PROCEEDINGS OF THE

Third International Conference on Autonomous Robots and Agents

Massey University, Palmerston North, New Zealand
12th - 14th December 2006

Edited by S.C. Mukhopadhyay and G. Sen Gupta

Massey University
CIC NUS, SINGAPORE
HRS Hoare Research Software Ltd
SINGAPORE POLYTECHNIC Opening Minds Shaping Lives
IEEE
IET
International Conference Fund
IEEE Instrumentation IM

ICARA

2006

International Conference on Autonomous Robots and Agents

Mukhopadhyay
Sen Gupta

ISBN-10 0-473-11566-2
ISBN-13 978-0-473-11566-1

Proceedings of the

Third International Conference on Autonomous Robots and Agents

Massey University
Palmerston North, New Zealand
12-14 December 2006

Edited by

S. C. Mukhopadhyay and G. Sen Gupta

Institute of Information Sciences and Technology
Massey University
Palmerston North, New Zealand

ISBN-10: 0-473-11566-2
ISBN-13: 978-0-473-11566-1

The papers in this book comprise the proceedings of the conference ICARA 2006. They reflect the authors' opinions and, in the interest of timely dissemination, are published as presented and without change. Their inclusion in these proceedings does not necessarily constitute endorsement by the editors or by Massey University.

ISBN-10: 0-473-11566-2
ISBN-13: 978-0-473-11566-1

Cover Design: Kim Gribbon

Cover Images:

ICARA 2006	Trax Robot IIST Massey University	Castle Point Wairarapa New Zealand	HanSaRam V KAIST Korea
Massey University Main Entrance	Gannet Colony Napier New Zealand	**Massey University**	Lady Knox Geyser Rotorua New Zealand
Tararua Wind Farm Manawatu New Zealand	Paul Dibble Sculpture Main Street Palmerston North	RoboErectus ARICC Singapore Polytechnic	**Palmerston North**
Kiwi National Icon New Zealand	RoboSapien CIC National University of Singapore	**New Zealand**	Maori Art Exhibit 2004 Te Papa Museum Wellington

Printed by: Massey University
Palmerston North, New Zealand

ORGANISING COMMITTEE

Honorary Chair

Kim Jong-Hwan, Korea

General Co-Chairs

Gourab Sen Gupta, NZ
Chris Messom, NZ

Programme Co-Chairs

Bruce MacDonald, NZ
Kim Mun Sang, Korea
Prahlad Vadakkepat, Singapore
Ray Jarvis, Australia
Serge Demidenko, NZ
Zhou Changjiu, Singapore

Technical Programme Chair

Subhas Mukhopadhyay, NZ

Proceedings Chair

Liyanage DeSilva, NZ

Publicity Co-Chairs

Dikai Liu, Australia
Jang-Myung Lee, Korea
Peter Kopacek, Austria

Advisory Committee

Bob Hodgson, NZ
Dave Chong, Singapore
Emil Petriu, Canada
Gurvinder Virk, UK
Janina Mazierska, NZ
Mel Siegel, USA
Richard Archer, NZ
Toshio Fukuda, Japan

Local Arrangement Co-chairs

Christine Allport, NZ
Ri Weal, NZ

Webmasters

Lisa Lightband, NZ
Tim O'Dea, NZ

Technical Programme Committee

Abder Koukam, France
Abdolhossein Sarrafzadeh, NZ
Babu B.V., India
C. Amarnath, India
Dale Carnegie, NZ
Djamila Ouelhadj, UK
Donald Bailey, NZ
Elmer P. Dadios, Philippines
Enrico Haemmerle, NZ
Gurvinder Virk, UK
Godfrey Onwubolu, Fiji
Gordon Wyeth, Australia
Hamid G. Hosseini, NZ
Huub Bakker, NZ
Huang Lou Lin, NZ
Iouliia Skliarova, Portugal
Jacky Baltes, Canada
James Peters, Canada
Joaquin Sitte, Australia
Karsten Berns, Germany
L.F. Wang, USA
Masoud Mohammadian, Australia
Mathew Palakal, USA
Mattias Wahde, Sweden
Mel Siegel, USA
Michael Cree, NZ
Nadir Ould-Khessal, Canada
Naoyuki Kubota, Japan
Napoleon Reyes, NZ
Nazeer Ahmed, Pakistan
N.N. Sharma, India
Osman Tokhi, UK
Peter Bentley, UK
Philippe Bidaud, France
P. Seshu, India
Peter Xu, NZ
Rini Akmeliawati, Malaysia
Roelof K. Brouwer, Canada
Rory Flemmer, NZ
Russel Stonier, Australia
Sung Bae Cho, Korea
Takehito Azuma, Japan
Tan Kay Chen, Singapore
Tan Kok Kiong, Singapore
Tapabrata Ray, Australia
Thomas Braunl, Australia
Valery Sklyarov, Portugal
Werner Friedrich, NZ
Yvan Baudoin, Belgium
Yoshikazu Mori, Japan

Welcome Message

It gives us immense pleasure to welcome you to the Third International Conference on Autonomous Robots and Agents, ICARA 2006.

Robotics and its applications in automation are advancing rapidly. While the last two decades of the 20th century belonged to communication, we believe the first three decades of the 21st century will be dominated by research and advancement in robotics. It is a very important technology that humankind needs to harness to enhance productivity. Robots are now roaming production floors, they are increasingly used in the service industry and they have found their way into our homes. Robotic pet dogs have become a reality and the day is not far off when humanoid robots will be common in many homes. Development of robots for edutainment, personal care and companionship is the next growth area. A conference like ICARA helps to bring together researchers from different fields such as artificial neural networks, neuro-fuzzy systems, genetic programming and evolutionary computation, apart from the traditional fields of image processing, motion control, embedded systems and communication. Research in robotics is also pushing the boundaries of miniaturisation and providing insight into alternate energy sources and new materials.

ICARA has a relatively brief history. It started off in 2000 as an International Symposium on Autonomous Robots and Agents (ISARA 2000), which was held in Singapore in conjunction with the Singapore Robotic Games (SRG). A total of 16 papers were presented. Since then, the conference has grown in leaps and bounds. The second time around, the name was changed to *ICARA – International Conference on Autonomous Robots and Agents* - and it was held in 2004 here at Massey University, Palmerston North, New Zealand. For ICARA 2004, 98 papers were received, of which 79 were accepted and presented. The phenomenal growth of the conference can be judged from the fact that for ICARA 2006, just over 160 papers were received. Of these, 132 papers will be presented at the conference in parallel oral sessions, a poster session and a special session on *Computational Intelligence for Communication Robots*. In addition, there will be three keynote addresses and three invited presentations on varied topics related to robotics. We foresee that ICARA will establish itself in the Australasian region as a premier conference in the rapidly advancing field of robotics.

Extended versions of selected papers from ICARA 2006 will be considered for publication in an issue titled “Autonomous Robots and Agents” which will be published by Springer-Verlag in the book series “Studies in Computational Intelligence” in June 2007.

Several people and organisations need to be thanked and acknowledged for their contribution to the success of the conference. We wish to thank all the authors and special session organisers for their contribution, our sponsors for the financial and technical support and all the members of the organising committee for their tireless efforts. Thanks go to all the paper reviewers for doing a thorough and timely job. Special thanks to Prof Emil Petriu, Prof Toshio Fukuda and A/Prof Kay Chen Tan for accepting our invitation to deliver the keynote addresses. We wish to thank Prof Mel Siegel, Dr Dave Chong and Prof Gurvinder Virk who will present invited papers.

We hope that you will have a very fruitful and stimulating time at the conference and a pleasant stay in New Zealand.

Gourab Sen Gupta	Subhas Mukhopadhyay	Chris Messom
ICARA 2006 General Co-Chair	ICARA 2006 Technical Programme Chair	ICARA 2006 General Co-Chair
g.sengupta@massey.ac.nz	s.c.mukhopadhyay@massey.ac.nz	c.h.messom@massey.ac.nz

Massey University (Turitea Campus)
Palmerston North
New Zealand

ORGANISERS

Institute of Information Sciences and Technology (IIST), Massey University, Palmerston North

SPONSORS

IEEE NZ North Section
IEEE NZ Central Sub-Section
IET New Zealand
Centre for Intelligent Control, National University of Singapore
Singapore Polytechnic
Hoare Research Software Ltd
International Science and Technology (ISAT) Conference Fund, Royal Society of New Zealand

TECHNICAL COOPERATION

FIRA (Federation of International Robot Soccer Association)
IEEE Instrumentation and Measurement Society, USA

Table of Contents

Session T3A: Autonomous Robots and Agents - I

Session T3B: Multi-Agent Systems and Networks

Session T3C: Humanoid Robots

Session T4A: Intelligent Control and Cooperative Robotics

Session T4B: Genetic Algorithm and Emotion Recognition

Session T4C: Robotics Path Planning

Session T5A: Mobile Robots

Session T5B: Unmanned Vehicle and Vehicle Tracking

Session T5C: Autonomous Robots and Agents – II

Session W1: Keynote Address

Session W2: Poster Session

Session W3: Invited Presentation

Session W4A: Vision Applications and FPGA in Robotics

Session W4B: Modelling and Simulation in Robotics

Session W4C: Smart Robots

Session H2A: Robot Soccer

Session H2B: Robot Hardware for Motion Control

Session H2C: Novel Strategies in Robotic Systems I

Session H3A: Special Session - Computational Intelligence for Communication Robots

Session H3B: Design Optimisation and Control Algorithms

Session H3C: Novel Strategies in Robotic Systems II

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Realtime Autonomous Navigation with a 3D Laser Range Sensor

Simon Thompson, Yusuke Kida, Atsushi Miyazaki and Satoshi Kagami
Digital Human Research Center
National Institute of Advanced Industrial Science and Technology (AIST)
Aomi, Tokyo, Japan
simon.thompson@aist.go.jp

Abstract

This paper presents ongoing development of the real time autonomous capabilities of a mobile robot platform equipped with a 3D laser range sensor. A Sick Laser Range finder is mounted on a platform with an actuator to allow a tilting motion. One 3D scan takes 1*s*, the data points rectified to an initial position to allow for continuous ego-motion during sensing. A two point randomised plane segmentation method is applied to the 3D data. Localisation is then performed using the entire 3D data set. Once localisation is performed the system plans a path through the environment using 3D data to detect a variety of obstacles such as low-lying, floating and overhead obstacles. Real time navigation performance is confirmed by an experiment through a complex environment.

Keywords: mobile robots, localisation, 3D sensing

1 Introduction

Mobile robot navigation is a crucial task for any useful application of mobile robot technology. Navigation is a complex behaviour encompassing such subcomponents as sensing, localisation, obstacle avoidance, path planning and path execution. To date mobile robot systems achieve navigation by making strict assumptions about the environment they operate in. The classic assumption being that the world is planar and that a 2D sensor is satisfactory for general operation. While this assumption can be true for a number of operation environments, in general 3D sensing is required for mobile robot operation in arbitrary environments.

Much work has been done in the related field of Simultaneous Mapping and Localisation (SLAM) to produce robust and reliable algorithms for 2D localisation systems using 2D sensors [1]. These algorithms are now being applied to 3D sensor data with much success, due primarily to the goal of 3D mapping and not autonomous navigation. Due to the difficulties inherent in 3D data capture and processing, 3D data is being captured online and in some cases mapping is done online too, but maps and data are held in representations which are unsuitable for the task of navigation. If autonomous navigation is performed it is generally done by extracting a 2D data scan from the 3D and performing traditional 2D navigation.

Stereo vision has been used as a 3D sensor for mobile robot navigation with mixed results [2] [3]. The high speed capture of images makes it good for dynamic environment, but low camera resolution, the dependence of stereo algorithms on specific image characteristics, and stereo baseline limitations make measurement accuracy a big problem.

Planar laser range finders have been used to generate 3D views of the environment by either moving a fixed sensor through the environment during ego-motion, and accumulating the range data [4], or by panning or tilting the sensor itself and making scans of the environment [5]. In the case of accumulation of data, the robot must move through unsensed regions to accumulated data. In the case of moving the scanner itself, current approaches have not achieved real-time navigation due to technical difficulties in data capture or in computational load in data processing.

This paper presents a mobile robot system capable of real time autonomous navigation used a 3D laser range sensor. 3D laser range data is captured in real time and is used to perform localisation and path planning with obstacle avoidance. The system can sense low lying obstacles as well as those not connected to the floor region. By assuming navigable regions lie on a common ground plane, the robot can use 3D sensor data efficiently to plot a 2D course through the environment. Initial steps toward environment modeling are taken with a plane segmentation algorithm applied to the 3D data also in real time with good results.

The rest of the paper is organised as follows: section 2 discusses relates work in 3D sensing for mobile robotics. Then section 3 describes the robot system used in the current work. The 3D laser sensor configuration is presented in 3.1. Section 3.2 then describes the plane segmentation algorithm and presents some examples of typical plane segmentation results. Section 3.3 then describes the localisation methods while section 3.4 describes how planning and obstacles avoidance is performed. Section 4 then presents some experimental results, illustrating the systems real time navigation performance in complex environments. Section 5 presents a discussion of the results, before section 6 makes some concluding statements and offers some thoughts on future work.

2 Related Work

Previous 3D systems have concentrated on making 3D maps of the environment. A common approach is to use 2D data for real time autonomous behaviours while capturing 3D data for later map building. Good examples of this type of approach are [4], a system which makes dense point cloud maps of underground mines and [6] who used an EM algorithm to learn 3D models of indoor environments made up of infinite planes. Another similar approach is that of [7], who describe an outdoor 3D mapping system using accumulated point clouds to perform plane segmentation and to abstract geometric representations of buildings. This work uses a reduced 3D Hough Transform search space to identify planes which fit 3D points in the point cloud. Planes are then merged and box shaped building structures identified.

Another approach is to abstract 3D point data to highly reduced surfaces and then to perform the autonomous localisation and mapping tasks. [8] proposed a system with a laser range finder which rotates about it's optical axis, capable of performing 3D scans within 1-8 seconds. In this approach planes were segmented from each 3D scan using an enhanced split and merge procedure which extracts maximally connected homogeneous surface regions. The extracted surfaces were then used to perform localisation and to build a 3D surface map, containing multiple sub-maps, of the environment. For each 3D scan, surface reduction requires 2-3 seconds, and it is unclear as to whether the robot was moving continuously during data capture. Similarly [5] report a 3D range sensing system which uses an EKF-SLAM approach to build a single map of infinite plane features. In that work the number of points sensed for each 3D scan is very high (361 × 601) and a single scan requires one minute.

The current work improves on previous systems by performing 3D laser range sensor based autonomous navigation in almost real time. Sensor data capture rates in the current system are considerably faster than in those previously reported, and fast localisation procedures and an efficient plane segmentation method ensure real time performance is maintained. The current system does not, however, create maps of the environment, using a given 2.5D map to perform localisation. The ability to perform plane segmentation provides a good basis for extensions to 3D mapping.

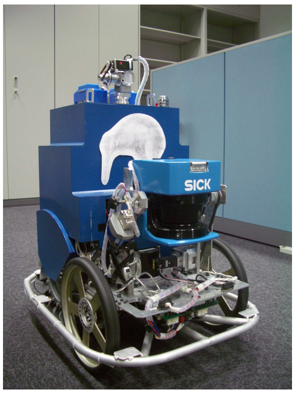


Figure 1: The mobile robot platform with a front mounted Sick laser range sensor which can be rotated about it's horizontal axis.

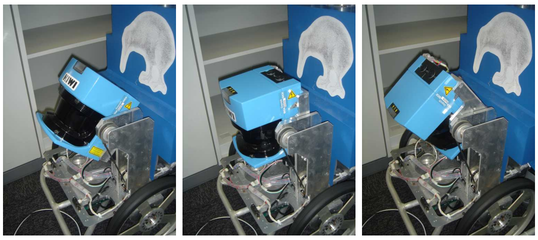


Figure 2: The laser range finder rotated to various positions during a sensor sweep: left 40 degrees, middle 0 degrees, and right -40 degrees

3 Mobile Robot Platform

The mobile robot platform is a car-like mobile robot with active front wheels and passive rear wheels. Figure 1 shows the mobile robot with a front mounted SICK laser range finder. It is linked to an off-board computer with wireless ethernet. The robot uses wheel encoders to generate odometry readings and has a laser range finder mounted in the front. The laser is mounted so it can rotate about its horizontal axis, tilting the scan line up and down. An autonomous navigation system has previously been developed allowing the robot to move to manually given goal positions in a known map. To do this the robot performs localisation, path planning and path execution. Laser range sensing and localisation previously was in 2D, with path planning and execution using smooth cubic and fourth order polynomial trajectories as proposed by [9] and [10].

Sensing, some sensory data preprocessing and control execution are carried out on-board the robot while most data processing, localisation and path planning are done off-board. Communications consists of transferring sensory data off-board and control

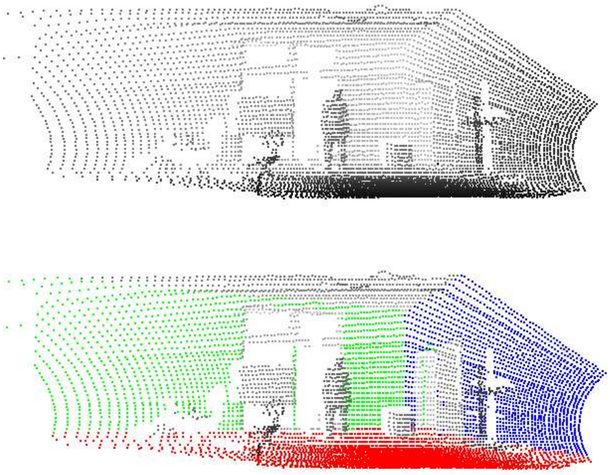


Figure 3: An example of a) laser range data from one 3D scan and b) the associated plane segmentation results.

parameters on-board. The off-board computer has dual Xeon 3.06GHz processors.

3.1 3D Range Sensing

The laser range finder is mounted on a tilt mechanism so it's axis of rotation passes through the focal point of the laser. The robot can thus rotate up and down about its horizontal axis scanning the environment. The Sick laser system is configured in high speed mode (500kbps) and is read through a high speed serial USB connection. Each single laser scan of 181 points (180 degrees) is performed and read in about 13ms.

By rotating the laser up and down in a sinusoidal motion the laser can sense the environment in 3D. In the current configuration the laser scans about 80 degrees in elevation, using a total of 80 individual scan lines, in approximately 1 second. Figure 2 shows the laser range finder being rotated throughout its range of motion.

During scanning, the real-time kernel module handling the high speed USB connection converts laser range distance to 3D point data. By monitoring odometry data during the 3D scanning process, individual data points can be adjusted for robot motion, and the full scan can be consistent from one view point even when the robot is moving.

3.2 Plane Segmentation

Plane segmentation is performed on the181 × 80 3D points generated by the 3D laser range scan. A two point algorithm is used which attempts to identify planes by randomly generating pairs of points that vote for a set of discretised possible planes (in pitch and yaw). Once probable plane orientations are identified the planes are further segmented into distance using a further voting method. The plane segmentation algorithm was developed by [11] for segmentation of stereo vision data, and is here applied to 3D laser range sensor data. The algorithm is similar to the 3D Hough Transform based plane segmentation described in [7] although here, the 2D sampling method improves the efficiency of the voting process.

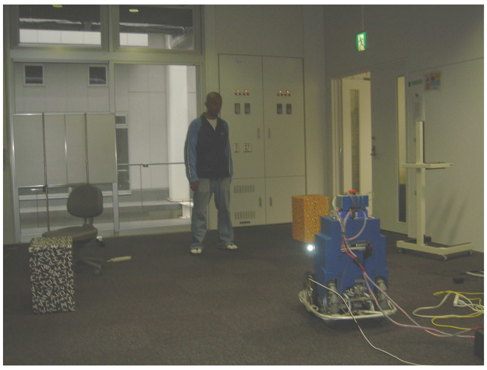


Figure 4: A photograph of the environment in which the sensor data from figure 3 was captured.

The plane discretisation resolution is 0.5 degrees in pitch and yaw angles. Distance is discretised at 5mm but a subsequent process of merging close planes means that that parallel planes with a difference in distance of more than 10cm can be detected. These parameters can be adjusted to suite the application with the trade-offs being computation time and the number of planes detected. In the current system plane segmentation was performed with the above resolution and the maximum number of planes was set to 10. Plane segmentation took about 120ms.

Results of the plane segmentation algorithm applied to laser data can be seen in figure 3. The robot has made a 3D scan in a complex environment including a human, boxes, other obstacles as well as walls and floor. A photograph of the environment is shown in figure 4. The plane segmentation process correctly identifies the three dominant planes in the environment with high accuracy, even correctly discounting a closed door from the surrounding wall plane.

3.3 Localisation

Localisation is a crucial component of current navigation systems due to the metric map representations in typical approaches. Although excellent localisation results have been published using 2D laser range sensors, it is easy to imagine situations where 2D scans could fail to uniquely identify an environment or to mismatch obscured or dynamic parts of the environment. Due to the increased information of 3D sensing it is desirable to use 3D data to resolve such situations and to further enhance the robustness of robot localisation systems. The framework behind the localisation system developed in this work is that of particle filter localisation

[12]. This is now a widely used method for mobile robot localisation which uses sampling of the robots position to approximate its location probability density function. A motion model is used to predict a sample's future position according to control or odometry information, and predicted positions are evaluated according to how their expected sensory view matches the current sensor view. A Gaussian approximation of the uncertainty distribution can be made by finding the covariance of position estimates within the particle set. This can then be used as a measure of localisation estimate certainty.

For 2D sensory data, the matching process can be done efficiently enough to allow a large enough number of particles (or samples) to accurately approximate the PDF and ensure continuous localisation. For 3D sensing, however, the computational cost of matching rises dramatically, especially the cost of generating expected views from each hypothesised robot location. This effectively requires the reduction of the number of particles or abandonment of real time constraints. This limitation is especially problematic in complex and repetitive environments (requiring high number of particles) where 3D sensing is most beneficial.

Localisation could be done using detected plane parameters, however, this would reduce the system to operating in only planar environments. Even in common environmnets such as corridors, often the most relevant localisation information lies in environmental features which are not part of the large planar surfaces which plane segementation algorithms are suited to detect. In the future it is planned to incorporate plane parameters in a multi-modal localisation process, but for the sake of generality in environment, it was decided to implement a raw data point based localisation method.

3.3.1 3D View Generation

The current system uses the hardware based graphics processing routines of OpenGL to reduce the time requirements of 3D matching. For hypothesised robot positions generated by the particle filter localisation system, quick ray tracing is done via calls to OpenGL to generate expected depth views from any location in the map. This is done by drawing the visual scene from the hypothesised view point in a non-visible image buffer and extracting the corresponding Z-buffer. The Z buffer gives the depth to expected obstacles for each pixel in the view image. This expected view can then be matched with the current 3D sensory data. The OpenGL routines are executed in specialised graphics hardware which gives a speedup of about 10 times over current software implementations. This effectively allows a 3D localisation system using hardware depth view generation to use 10 times more particles.

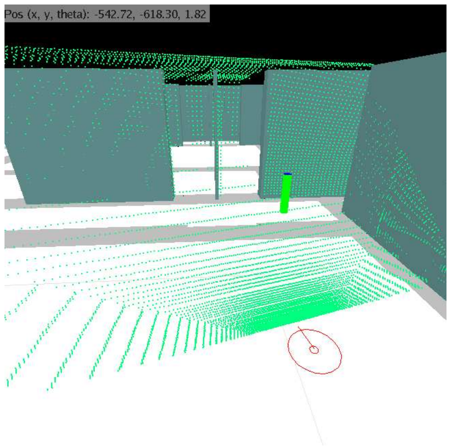


Figure 5: An expected laser view generated via OpenGL Z-buffer, drawn against the 2.5D map

3.3.2 3D Point Saliency

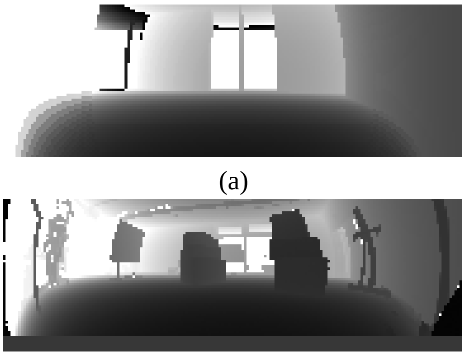
When dealing with a large amount of sensory data points which contains a large amount of redundant information, data points which provide crucial localisation information can be lost in the noise. It is desirable therefore to identify salient points in some manner to weight the matching process towards areas containing cues for localisation. It is clear that plane segmentation can provide cues to the saliency of indvidual points. One initial observation is that points observed on the ground plane do not give localisation information so can be discarded. Another observation is that points which lie on planes which are parallel to the main axis of localisation uncertainty provide less discriminatory information that those points which lie on planes orthogonal to the uncertainty distribution. Sampling data points from planes based on their orientation in relation to uncertainty distribution in the localisation estimate increases the discriminatory ability of the point set.

3.3.3 3D View Matching

Once an expected view is generated for a given localisation hypothesis it is matched against the current 3D laser view. Currently a simple Sum of Absolute Differences (SAD) between the salient points of the current and expected view is being used to determine the similarity. This has proved of limited discriminatory ability between similar views, but was sufficient for the experiments described later. Another feature of the SAD evaluation is that it is extremely fast.

Each evaluation of a localisation hypotheses (particle) takes approximately 11ms with the SAD calculation taking less than 1ms. Currently the system runs with 50 particles, a bare minimum required for successful localisation using the described methods.

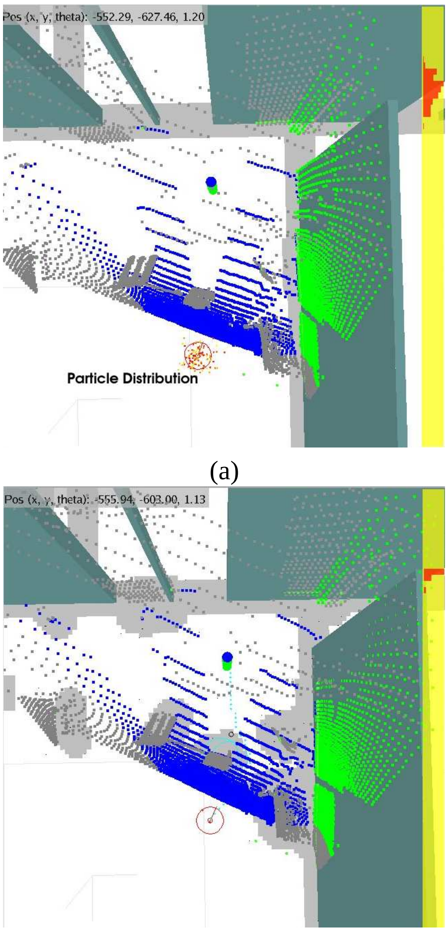
Figure 5 shows an example of a expected depth view generated by the OpenGL depth buffers. Here the 2.5D



(a)

(b)

Figure 6: Depth maps used for matching: (a) expected depth map, (b) current laser view.


Pos (x, y, theta): -552.29, -627.46, 1.20
Particle Distribution
Pos (x, y, theta): -555.94, -603.00, 1.13

(a)

(b)

Figure 7: Particle filter localisation: (a) particles initially distributed about robot position, (b) particles condense to an accurate location estimate. Orientation distribution shown in yellow sidebar.

map is shown along with the green points of the generated depth map. Figure 6 show the 2D depth images for an expected view and a current sensor view as they appear for matching.

Finally figure 7 shows the (a) initial and (b) localised results (after 5 frames) from a localisation sequence. The small red/yellow circles positioned about the robot show hypothesised robot (x,y) locations, while the yellow sidebar and red histogram show the spread of uncertainty in rotation. In the initial configuration the location and orientation is spread out while after location both have condensed around the correct location estimates.

3.4 Path Planning

Path planning is achieved using the method proposed by [10] and extending it to consider 3D data. That method uses a 2D A* path planner [13] to generate the optimally shortest path through a 2D grid map to a goal location. A local subgoal is created along the 2D path (at about 5m) and is used as the target for generating cubic and fourth order curvature polynomials. Cubic trajectories are easy to generate but do not consider obstacles. Extending cubics to fourth order polynomials allows for trajectories to explicitly consider the presence of obstacles while being generated. These local trajectories are continuous in curvature and allow the robot to make smooth motions through the environment.

Because there is an assumption of a single traversable ground plane, we extend the system to incorporate 3D data by reducing the 3D data to 2D, defining regions in a map to be either passable or not. The reduction to 2D is not a simple projection, as the process considers the physical dimensions of the robot and ignores overhead obstacles under which the robot could pass safely. In this work we consider any 3D depth point between 5 and 1200cm in height to be an obstacle which must be avoided.

4 Experiment

An experiment to test the real time performance of the 3D autonomous navigation system was conducted. The environment the experiment was conducted in is shown

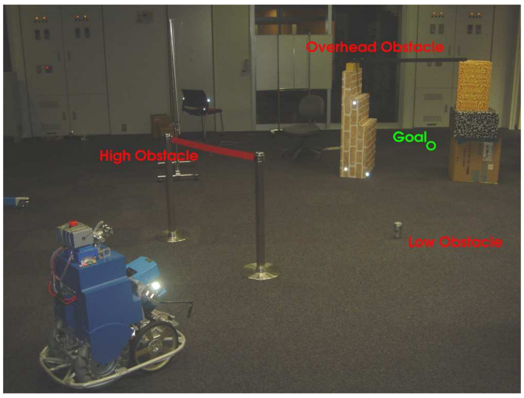


Figure 8: Setup of experiment to test real time performance of 3D autonomous navigation system: robot must reach goal navigating through low/high obstacles and under overhead obstacles.

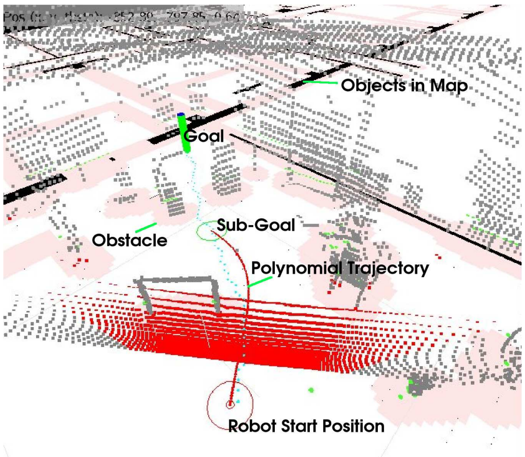
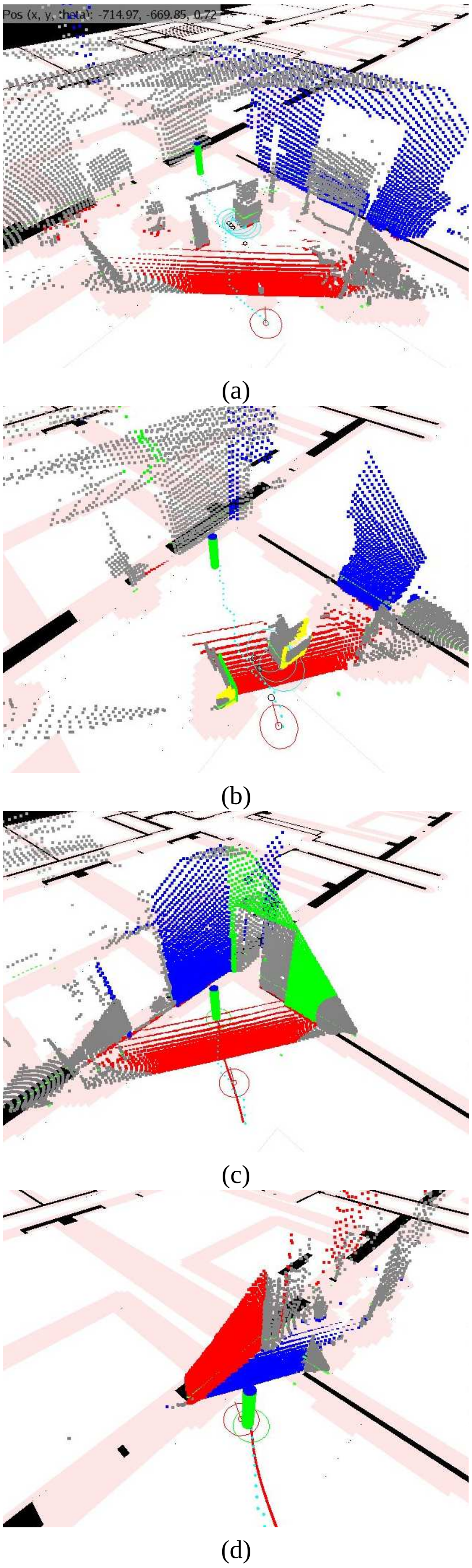


Figure 9: Robot view at start of navigation experiment.

in figure 8. The robot must reach a manually assigned goal location while navigating past low lying and floating "high" obstacles and passing under overhead obstacles. The robot continuously scans the environment using the 3D sensor while performing plane segmentation, 3D localisation, obstacle detection and path planning. In this experiment the robot moves at approximately $0.1m/s$. Seventy 3D scans were captured and processed during the experiment. The speed was set to test localisation ability, and allow for reactive planning when encountering dynamic objects. The robot can execute accurate trajectories with 2D sensing at up to $1.0m/s$ and can reconstruct consistent 3D scans up to a speed of about $0.5m/s$. In human inhabited environments, the robot can navigate autonomously with 3D sensing at about $0.3m/s$.

Figures 9 and 10 show screen-shots of the system while the experiment was being conducted. Figure 9 shows the initial configuration of the experiment with the robot planning a polynomial trajectory past the floating object to a sub-goal on the way to the goal. The robot successfully navigated past all obstacles to reach the goal position. The rest of the screen-shots show (figure 10) (a) the robot reaching the low lying obstacle (an aluminum can) and planning around it, (b) the robot approaching the overhead obstacle (a beam supported by boxes), and planning to pass below it, (c) passing below the beam, and finally (d) the robot reaching the goal location. In each image a 2.5D map used for localisation is drawn in 2D (black regions denoting occupied space, while pink regions show a clearance region around obstacles, through which the robot cannot pass. Plane segmentation results are shown by various colour labeling of the 3D data, while unclassified data are shown in gray scale.

Figure 11 shows a plot of the localisation results annotated with labels for start and goal locations and general location of obstacles. The localisation results track the general path followed by the robot, but are quite noisy



(a)

(b)

(c)

(d)

Figure 10: An autonomous navigation experiment using 3D laser range sensor data

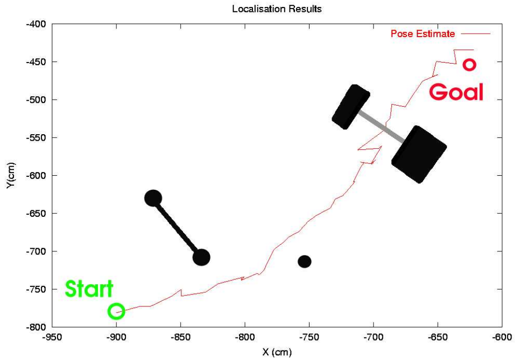


Figure 11: Localisation results for real time navigation experiment

in places, particular when the robot approaches large obstacles. This is to be expected as large obstacles obscure the view of wall segments which are contained in the map. No ground truth is available for the experiment, however visual inspection of the screen-shots in figure 10 show that the current sensor data is positioned correctly relative to the map. These results are indicative of localisation performance throughout the experiment.

During the experiment the autonomous navigation system on the off-board computer ran at a cycle rate of about 1*s*. This total time was comprised of the following (all times approximate): 300*ms* for sensing (retrieving 3D sensor data from on-board robot); 500*ms* for localisation; 30*ms* for path planning; and 120*ms* for plane segmentation.

5 Discussion

The system as implemented can perform real-time autonomous navigation using 3D sensor data from a laser range finder system, although the cycle time of 1s is approaching the outer limits of feasibility. The most time consuming aspects of the system as it stands are the 3D data capture itself (1*s*) and the 3D localisation (500*ms* for 50 particles, sensing and localisation operating in parallel).

Data capture using the 3D sensor is limited by the 2D scanning and data transfer rate of 13ms. Total time required for 3D scanning can be somewhat reduced by trading off vertical resolution for capture time. The sinusoidal motion pattern could be modified to ensure a more even spread of scan line capture across the vertical view field. Currently the highest velocity of sensor rotation occurs when viewing directly in front of the robot, while the sensor rotates slower and slower toward the upper and lower limits. Adjusting rotation velocities to concentrate scans toward more informative regions could improve robot perception, and possibly reduce the total number of 2D scans required for a full 3D scan. The performance of the system is sufficient for the experiment described above, assuming very low robot velocities. The robot could be speed up considerably if the environment was assummed to be entirely static.

The plane segmentation system works well and is relatively efficient. Currently plane segmentation is used to weight the saliency of the data points for localisation. More needs to be done with the results however, and the authours are planning to develop a 3D environment modeling system which would be used incrementally for mapping as well as for localisation. The plane parameters themselves could be used for localisation, although for arbitrary environments it would still be desirable to include localisation methods which operate on raw data points. The plane segmentation results presented here, however, illustrate the feasibility of incorporating surface reduction methods into a real time autonomous navigation system.

6 Conclusion

This paper has presented a real time 3D range sensor based autonomous navigation system. Real time 3D sensing allows mobile robot systems to navigate through arbitrary environments that may not conform to strict 2D planar assumptions. The current system senses, segments planes, localises, plans and executes paths using 3D data, all in real time. Real time performance was confirmed by performing an autonomous navigation experiment through a complex environment, moving continuously while capturing and processing 3D data.

As sensing time is the biggest limiting factor in acheiving real time navigation with a 3D sensor an investigation into the optimal sensor rotation pattern to make data capture faster would be beneficial. Perhaps the swinging sensor mounted higher and with a tilting range of only 45 degrees would be sufficent to perform autonomous localisation and mapping. This would improve the speed of the system considerably.

Besides optimising the existing system, it is planned to use the plane segmentation results to incrementally update a surface map of the environment. Reducing planes to polygons and using these to perform localisation would add to the autonomous nature of the system. The 3D localisation method would not have to fundamentally change as depth buffer generation for polygon based surface maps would be very fast. Incorporating full 3D localisation with a feature based method, or moving to a iterative scan matching approach would be beneficial for localisation accuracy however.

7 References

[1] S. Thrun, “Robotic mapping: a survey”, in G. Lakemeyer and B. Neberl (eds), Exploring Artificial Intelligence in the New Millennium, Morgan Kauffman, pp 1-36 (2002).

[2] D. Murray and J. Little, “Using real-time stereo vision for mobile robot navigation”, *Autonomous Robots*, 8(2), pp 161-171 (2000).

[3] S. Thompson and S. Kagami, “Revising stereo vision maps in particle filter based SLAM using localisation confidence and sample history”, *Proceedings IEEE International Conference on Autonomous Robots and Agents*, Palmerston North, pp 218-223 (2004).

[4] S. Thrun, D. Haehnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer and W. Whittaker, “A system for volumetric mapping of abandoned mines”, *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei (2003).

[5] J. Weingarten and R. Siegwart, “EKF-based 3D SLAM for structured environment reconstruction”, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, pp 3834-3839 (2005).

[6] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard and S. Thrun, “Using EM to learn 3D models of indoor environments with mobile robots”, *Proceedings 18th International Conference on Machine Learning*, Williams College (2001).

[7] D. Wolf, A. Howard and G. S. Sukhatme, “Towards geometric 3D mapping of outdoor environments using mobile robots”, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton (2005).

[8] P. Kohlhepp, P. Pozzo, M. Walther and R. Dillmann, “Sequential 3D-SLAM for mobile action planning”, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, pp 722-729 (2004).

[9] B. Nagy and A. Kelly, “Trajectory generation for car-like robots using cubic curvature polynomials”, *Proceedings International Conference on Field and Service Robots*, Helsinki (2001).

[10] S. Thompson and S. Kagami, “Smooth trajectory planning with obstacle avoidance for car-like mobile robots”, *Proceedings 23rd Conference of the Robotics Society of Japan*, Tokyo (2005).

[11] Y. Kida, S. Kagami, T. Nakata, M. Kouchi and M. Mizoguchi, “Human finding and body property estimation by using floor segmentation and 3D labeling”, *Proceedings International Conference on Systems, Man and Cybernetics*, The Hague, pp 2924-2929 (2004).

[12] S. Thrun, W. Burgard and D. Fox, “Robust Monte Carlo localisation for mobile robots”, *Artificial Intelligence*, 128, pp 99-141 (2001).

[13] J. Kuffner, “Efficient optimal search of Euclidean-Cost grids and lattices”, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai (2004).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Micro Aerial Vehicles: Towards Autonomous Flight

M. Meenakshi and M. Seetharama Bhat
Department of Aerospace Engineering
Indian Institute of Science, Bangalore, India
meena@aero.iisc.ernet.in, msbdcl@aero.iisc.ernet.in

Abstract

This paper presents the real time validation of fixed order robust H_2 controller designed for the lateral stabilisation of a micro air vehicle named Sarika2. A digital signal processor (DSP) based onboard computer named *flight instrumentation controller* (FIC) is designed to operate under automatic or manual mode. FIC gathers data from multitude of sensors and is capable of closed loop control to enable autonomous flight. Fixed order lateral H_2 controller with the features such as incorporation of level 1 flying qualities, gust alleviation and noise rejection is coded on to the FIC. Challenging real time hardware in loop simulation (HILS) is done with dSPACE1104 RTI/RTW. Responses obtained from the HILS are compared with those obtained from the offline simulation. Finally flight trials are conducted to demonstrate the satisfactory performance of the closed loop system. The generic design methodology developed is applicable to all classes of mini and micro air vehicles.

Keywords: micro air vehicle, robust controller, flight instrumentation controller, hardware in loop simulation, autonomous flight.

1 Introduction

Recently, the desire for low cost, portable, low-altitude aerial surveillance has driven the development and testing of smaller and smaller aircrafts, known as micro air vehicles (MAVs). MAVs are envisioned as rapidly deployable, autonomous aerial robots equipped with a camera system. The ducted or shrouded rotary-wing MAV would peer through windows, over roofs or above tree canopies to support missions like search-and-rescue [1] and target acquisition and thus suited for hover-and-stare. Equipped with small video cameras and transmitters, fixed wing MAVs have great potential for surveillance and monitoring tasks in areas either too remote or dangerous to send human scouts. It should be noted that fixed-wing miniature aerial vehicles [2] [3], [4] typically fly around 20 m/s in open skies and thus not suited for hover-and-stare.

MAVs are radio controlled remotely piloted systems. These vehicles are difficult to fly due to their unconventional design and unpredictable flight characteristics. Another potential source of instability for MAV is the relative magnitudes of wind gusts, which are much higher at the MAV scale. Hence, to alleviate the necessity of an expert pilot, some form of augmented control must exist. Thus, in [5], a single lateral stability augmentation system (SAS), which meets the flying qualities, disturbance rejection and noise attenuation specifications for the entire flight envelope (flight speeds ranging from 16m/s to 26m/s) of a MAV named Sarika2 is designed. The controller is designed at the central operating point of Sarika2 (20m/s) and is used for its entire flight envelope. The software in simulation (SiL) results showed excellent performance over its entire flight envelope [5]. However, SiL does not validate the implementation of a controller. In a sampled time digital computer with real life I/O there is no guarantee that the real time performance constraints can be met [6]. The application of HILS technology to the controller design process can reduce expensive field tests, improve product quality, facilitate examination of subsystem interactions, and resolve critical safety issues prior to in-vehicle testing [7]. This paper presents the development and real time validation of DSP based flight control system that gathers data from multitude of sensors and is capable of closed loop control to enable smooth flight of MAV. Simple augmented control is the initial step to enabling autonomous flight.

The overview of this paper is as follows. Sections 2 and 3 respectively give a brief description of the vehicle and the FIC. Real time Implementation of fixed order robust H_2 controller is given in Section 4. The concept of HILS, which uses the real time controller (FIC) and the simulated aircraft model is given in section 5. Section 5 also gives the HILS results and analysis. Finally conclusions are drawn in section 6.

2 Sarika2 - Description

The MAV named Sarika2 (figure 1) was designed at the Department of Aerospace Engineering, Indian Institute of Science, Bangalore. It incorporates a clipped delta wing with the controls through the inboard aileron and outboard elevator on the wing and a rudder on the vertical fin. Sarika2 carries a power plant, fuel, landing gear, radio receiver, batteries, servo controls for engine, rudder, and elevon and sensors. The fuselage carries the power plant in the front and vertical fin and rudder at the back. The fuselage of the constant area tapers over the rear part to reduce the drag. It is like a box covered on three sides while the top is open. The high set wing largely covers the top after assembly. The devices needed for the flight of the Sarika2 are all placed in the fuselage. The swept back wing made up of balsa and thermocole, with its elevon and their electro mechanical servo actuators form a unit which can be mounted on the fuselage and held in the position by the rubber bands as in conventional model aircraft. Table 1 summarises the pertinent geometric and mass properties of the vehicle.

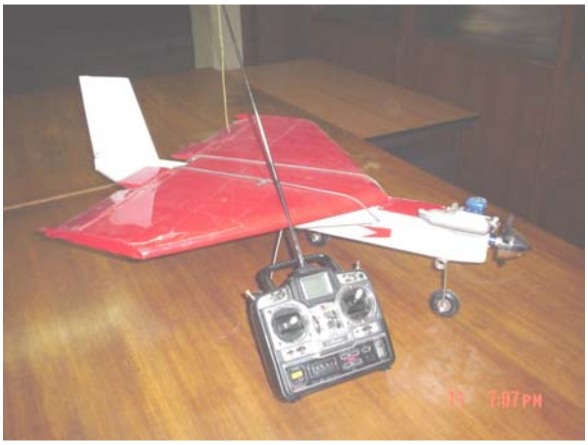


Figure 1: Micro air vehicle Sarika2.

Table 1: Geometric and mass properties of Sarika2.

Parameter	**Notation**	**Value**
Mass	m	1.75 kg
Wing span	b	0.6 m
Wing area	S	0.195 m^2
Moment of Inertia	I_{xx} I_{yy} I_{zz}	0.043 kgm^2 0.0686 kgm^2 0.07507 kgm^2

3 Onboard Flight Instrumentation Controller

The integration of a lightweight onboard computer, which is capable of performing closed loop control functions in real time, is a technically challenging task since off-the-shelf solution did not exist with the stringent constraint on size (limited space within Sarika2), weight and power. As a result, DSP based FIC (figure 2) (139.7mm×50.8mm, 55g) has been developed. The FIC is designed around two Texas Instruments DSPs, TMS320VC33 and TMS320LF2407. TMS320LF2407 DSP is used for data acquisition and motor control application while TMS320VC33 DSP is used for mathematical computations and data memory management. Communication between the two DSPs is through inbuilt synchronous serial channel operating at 5MBPS.

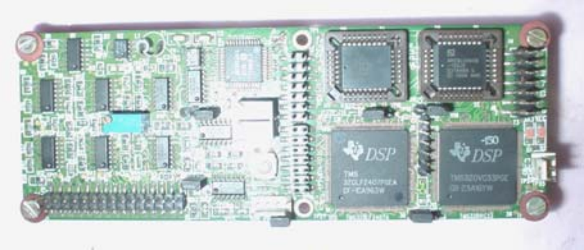


Figure2: Flight instrumentation controller.

TMS320LF2407 is a 16 bit fixed point processor operating at 30MHz frequency. LF2407 has on-chip peripherals like 16 channel 10 bit ADC, 2K flash memory, PWM I/O, UART and SPI. This DSP has 6 internal systems to capture PWM signals and 6 units to generate PWM waveforms.

TMS320VC33 is a 32-bit floating point DSP operating at 60 MHz frequency. VC33 has 1.1 MB on chip memory and serial synchronous channel. A memory expansion system is mounted on the board and support up to 256 K X 32 of SRAM (programme memory) and 4 MB non-volatile flash memory (FPROM) for flight data recording. A 1 MB X 16 FPROM is provided as boot memory for VC33.

An external 8 channel, 12 bit ADC, AD7891 having an output range of ±5 V is interfaced to TMS320LF2407 DSP for better resolution and accuracy of signal measurement. Five digital I/Os on TMS320LF2407 DSP and single I/O line on TMS320VC33 DSP are provided to handle external interrupts.

FIC supports a maximum of 19 sensor interfaces. A second order Butterworth low pass filter of 10 Hz cut-off frequency and 0.6 damping ratio is provided to all the analog input signals. Filtered outputs are pre-amplified as per the specifications of the sensors before connecting to ADC. Eleven of these conditioned signals have an output range of 0 - 3.3 V and are interfaced to inbuilt ADC of TMS320LF2407. Remaining eight channels have an output range of ±5V and are interfaced to external ADC, AD7891.

The FIC provides four primary functions: programme and external data storing in a memory, high-speed data acquisition (with 10Hz low pass filter and necessary signal conditioning), internal data logging, generation and capture of PWM signals. The data can be logged to a maximum of 4 MB (FPROM). Due to the complexity of the real time control of Sarika2, small errors in software coding may cause the serious damage to its airframe. To mitigate this problem, hardware digital switching logic is provided so that the pilot can switch-over from automatic to manual mode at will (so that command signals received from the ground bypasses the controller block, and directly get linked the actuator servo) and can safely land the aircraft. Manual mode of flight is also useful for the estimation of the aircraft flight parameters since some of these parameters get camouflaged on closing the feedback loop.

4 Real-time Implementation of Fixed Order Robust H_2 Controller

Floating point-programming technique of DSP is used to develop all onboard software for the FIC. Main algorithm comprises of four main steps i.e.

1. Initialisation routines, 2. delay routine, 3. sensor calibration routine, 4. interrupt enable routine, which enables the timer interrupt, TINP, the capture interrupt CAP, the external interrupt2, XINT2 and waits for the interrupt. The various interrupts are handled in the following fashion.

a. If timer interrupt,

i) TMS320LF2407 samples 8 channels of ADC; ii) implements control law, iii) transfers the sampled data and the controller output to TMS320VC33 via serial peripheral interface (SPI) and stores them in flash memory.

b. If capture interrupt,

TMS320LF2407 receives the pulse width of the command input sent from the transmitter and assigns this new value to the temporary variables used for controller implementation. Based on the new controlling signal, processor generates the new PWM signal.

c. If external interrupt2,

1. Processor disables all other interrupts as external interrupt is of top priority.
2. TMS320LF2407 sends a data request signal to TMS320VC33 through SPI
3. On receiving the data request signal from TMS320LF2407, TMS320VC33 starts sending the data from the flash memory to TMS320LF2407.
4. TMS320LF2407 receives the data sent by TMS320VC33 and simultaneously transmits to the Hyper Terminal through serial communication interface (SCI).

The efficacy of the software is demonstrated by implementing a third order robust H_2 controller (given in equation (1) [5]) designed to meet the level 1 flying qualities of the lateral dynamics of Sarika2. Level 1 flying quality requirements are:

1. Dutch roll damping ratio ≥ 0.5
2. Roll subsidence time constant <1 s
3. Spiral mode-minimum time to double the amplitude >12 s

Along with the level 1 flying qualities, the controller is also able to reject low frequency wind gust disturbance input and to attenuate high frequency sensor noise signal.

$$K(z) = \frac{1}{\Delta(z)}\begin{bmatrix} -0.68(z-1.07)(z-0.63) & -0.58(z-0.99)(z-0.61) & 0.17(z-1.66)(z-0.65) \\ -0.05(z+2.92)(z-0.94) & -0.02(z+5.24)(z-0.96) & 0.03(z+1.46)(z-1.04) \end{bmatrix}$$

where, $\Delta(z) = (z-0.72)\left(z^2-0.30z+0.29\right)$. (1)

The controller continuously gathers the feedback (measurement) signals (i.e. lateral acceleration (a_y), roll rate, (p) and yaw rate, (r)) from sensors along with the possible pilot command inputs from the radio transmitter. Simultaneously, controller processes the received data and generates the actuator commands through its PWM generation units allowing for assisted flight control. The system is designed to record the current state of the system (sensor, actuator and ground pilot command input) during the flight trials.

5 Hardware in Loop Simulation

In order to validate that FIC can indeed meet real time performance constraints; the HILS technique is employed. In HILS, either a simulated plant is wired to a real controller, or a real plant is wired to a simulated controller. In this work, a real controller is wired to the simulated aircraft as shown in figure 3.

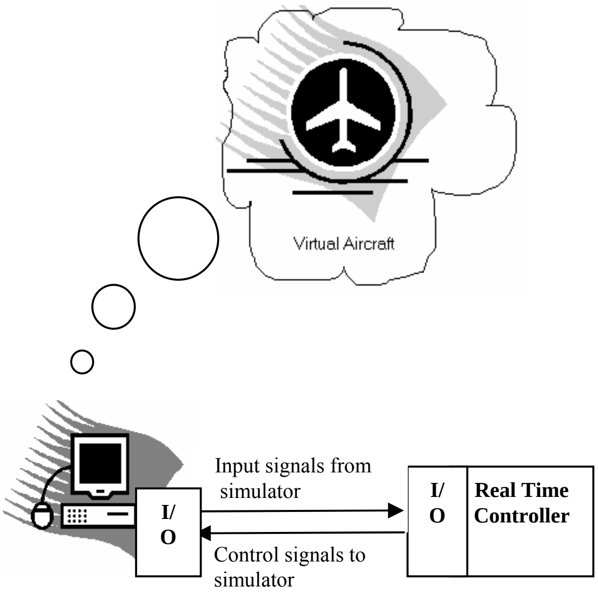


Figure 3: HIL simulated plant, real controller.

The feedback (measurement) signals (i.e., a_y, p and r) to the real time controller are determined by the lateral flight dynamic model, running on the desktop computer housing dSPACE1104 RTI/RTW. dSPACE1104 subsequently generates the equivalent analog signal in volts through its DAC unit. Sensitivities of the MEMS based sensors used for Sarika2 are: 1.11mv/deg/s (with an offset of 2.25volts), Accelerometers: 500mv/g (with a bias of 2.5volts). Hence, the sensor outputs are generated according to the relation:

$$V_{eq}\,(p \text{ or } r) = 0.063603 \times (p \text{ or } r) + V_{offset}\,(p \text{ or } r) \quad (2a)$$

$$V_{eq}\,(a_y) = 0.05 \times (a_y) + V_{offset}\,(a_y) \quad (2b)$$

FIC receives this equivalent sensor signal through its ADC unit. These signals are then converted in to equivalent milliseconds using software. The controller continuously gathers this converted sensor signals along with the possible pilot command inputs. Simultaneously, controller processes the received data and generates the actuator commands through its PWM generation units. The simulated flight dynamics model receives these outputs through its capture units allowing for assisted flight control.

5.1 Controller Validation

In order to validate the controller in real time, powerful open loop and closed loop tests using MATLAB/SIMULINK tools and RTI/RTW are employed. Under open loop test, controller receives the standard test inputs (e.g., step, pulse, ramp etc) in time-indexed mode using computer scheduler utilities.

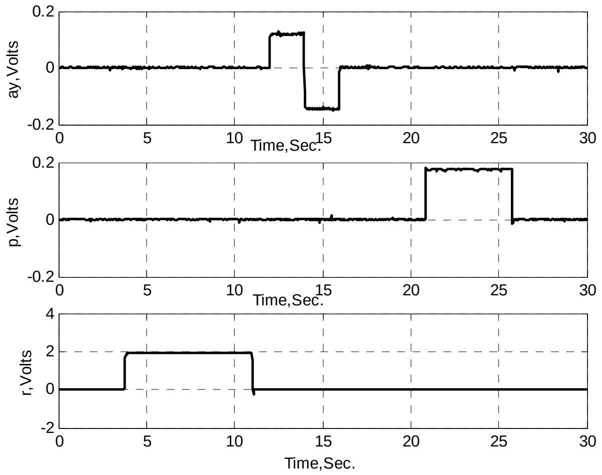


Figure 4: Scheduler input sequence to SAS.

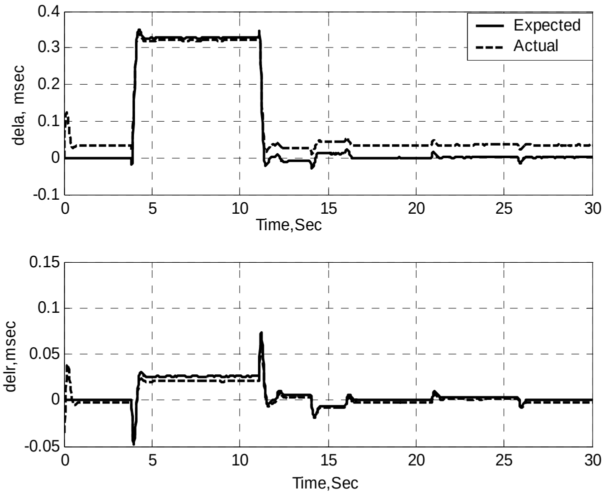


Figure 5: Output of controller to aileron and rudder channels

Figure 4 shows the inputs used for open loop test and figure 5 shows the corresponding output of the controller, recorded for a small interval of time.

The input pattern is a pulse of amplitude 2 volts and a duration of 8s (from t = 4s) on yaw rate channel followed by a doublet input of amplitude ± 0.1 volts and a duration of 2 s (from t = 12 s) on lateral acceleration a_y channel, and a pulse of amplitude 0.2 volts on roll rate channel (from t = 21to 26 s.). For the comparison, figure 5 also shows the expected output of the controller obtained using SIMULINK tools.

From figure 5, one can see the excellent match between the actual output obtained from FIC and the expected output from the SIMULINK tools with a small offset of 0.04ms (equivalent to 0.16 degrees aileron deflection) on aileron channel. This small difference is due to the limitations of fixed-point processor, TM320LF2407.

5.2 Closed Loop Test Results of Lateral SAS

FIC receives the equivalent feedback signals (2a) and (2b) through DAC/ADC interface of dSPACE1104/FIC. The closed loop responses are simulated at three different flight conditions, i.e., at 16 m/s, 20m/s and 26m/s. Figures 6a and 6b represent the real time command signal from radio frequency receiver, controller output and emulated sensor signals corresponding to a flight speed of 20 m/s.

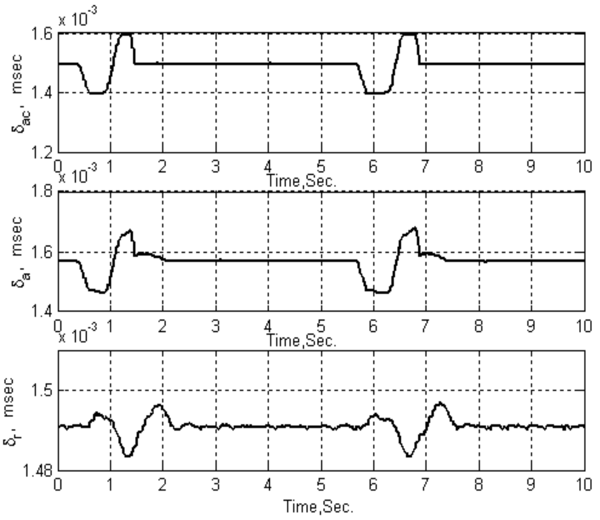


Figure 6a: Command input and controller outputs.

Figure 6a includes the command input applied on the aileron channel by moving the aileron joystick and the corresponding inputs to the aileron and rudder actuators from the controller. The command input is around - 0.1 to 0.1 (±3.84 degree aileron deflection) with respect to the neutral value of 1.5 ms. This input is equivalent of moving the joystick to one end, holding there for a while and then moving it to the other end and then back to the neutral position. In figure 6b the a_y response shows an excursion of a maximum of 0.15 volts to -0.15 volts. A 0.15 volts corresponds to 0.3g lateral acceleration. Similarly a -0.15 volts corresponds to -0.3g lateral acceleration, which is comparable with its offline simulation results.

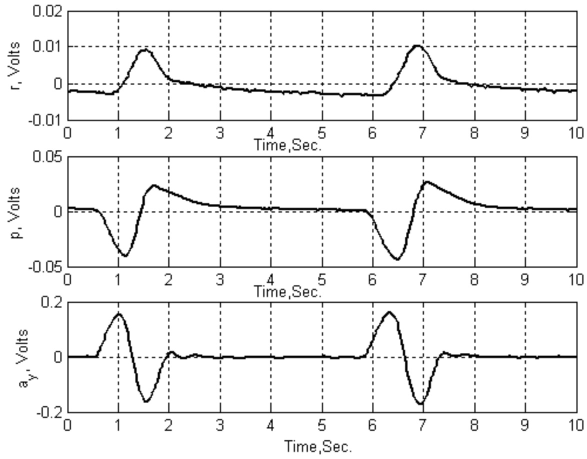


Figure 6b: Outputs of the emulated sensors.

Simultaneously, the real-time roll rate response ranges from -0.04 to 0.02 volts with respect to the neutral position. This corresponds to a roll rate of -36 deg/s with a -0.1 ms command input. Corresponding offline simulation shows a roll rate of -29.6 degrees/s. Hence, real time simulation responses match well with a small difference compared to that of offline simulation responses. This difference is due to the limitations of the fixed-point processor used for the implementation of the controller.

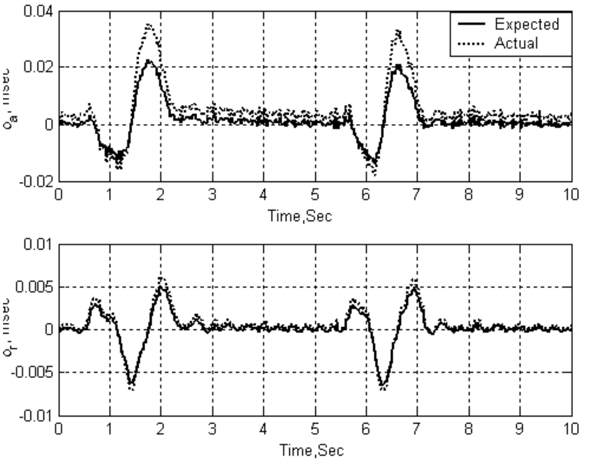


Figure 6c: Actual and expected outputs of the controller.

Figure 6c shows the comparison of actual and expected output of the controller at 20m/s. The actual outputs are from the FIC and the expected outputs are from the simulation using SIMULINK tools. From this comparison, a delay of 0.1 sec is noticed between the actual and expected output of the controller. This is due to the transportation delay unit introduced during simulation (in dSPACE 1104). However, this delay will not occur in reality. On the aileron channel, the actual output shows a small offset of +0.01 ms. in contrast to that of simulated output. On the other hand, the mismatch on rudder channel is almost zero. A 0.01 ms PWM signal on the aileron channel gives a disparity of 0.4 degree aileron deflection in real time. However, since the controller is robust, this mismatch is acceptable.

Similarly, the open loop and closed loop HILS is repeated at different flight speeds like, 16m/s and 26 m/s. The responses are found to be satisfactory.

5.3 Flight Test Results

Next, Sarika2 is flown (figure 7) in traditional manual or radio controlled mode without flight stabilisation for further data analysis.

Figure 7: Sarika2 in flight.

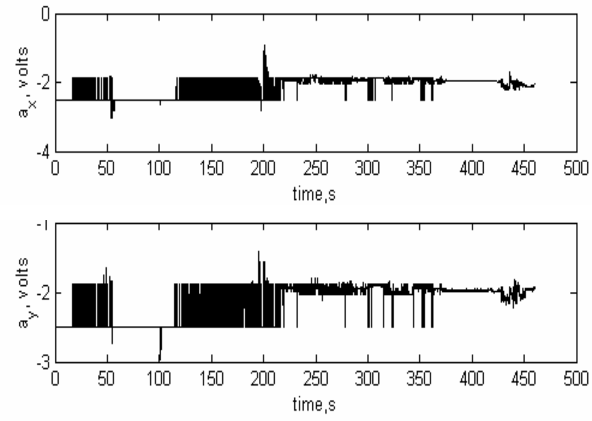


Figure 8: p, r responses from flight trial.

The sensors output (figures 8 and 9) recorded during taxiing run with flight stabilisation and flight would indicate quality of signals from low cost MEMS rate gyros and accelerometer, impact of engine vibration on sensors output, ground reaction on the vehicle during the taxiing and would give confidence on the plant model used for the controller design. Figures 10 and 11 show the pilot command inputs and the corresponding controller outputs given during the test trials. The recorded data is used to verify the satisfactory functioning of the controller. However, the controlled flight proved that the aircraft remains stable under closed loop with improved damping. Moreover, the pilot on ground is satisfied with the performance of the controller during the test trials, which proves that the controller works satisfactorily.

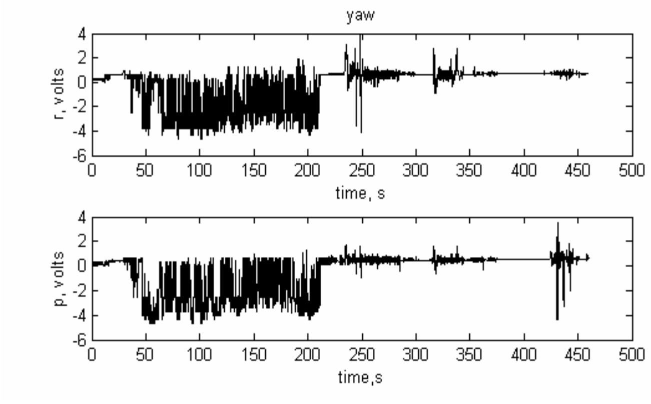


Figure 9: a_x, a_y, responses from flight trial.

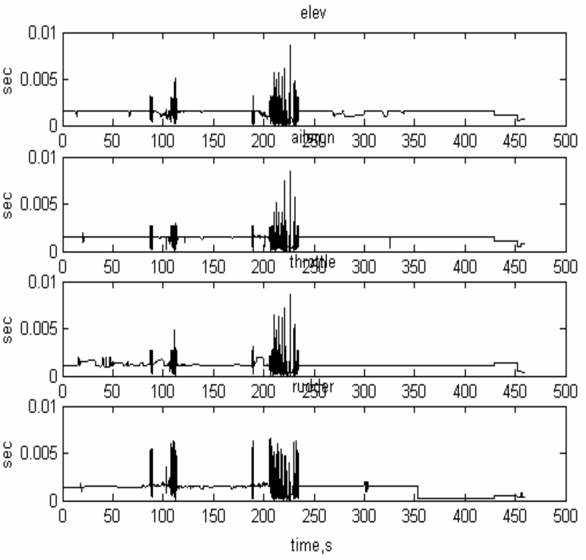


Figure 10: Aileron and rudder command inputs.

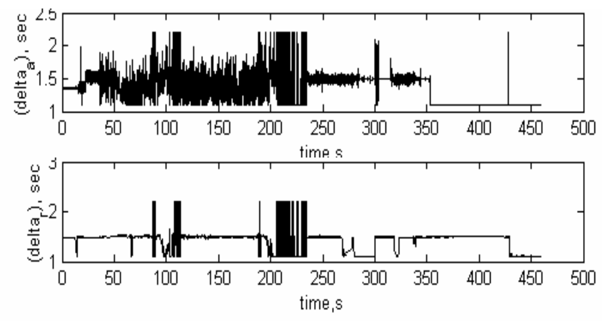


Figure 11: Controller outputs to aileron and rudder.

6 Conclusion

This paper explains the way to enable autonomous flight of the MAV through the inclusion of onboard flight instrumentation computer, capable of using rate gyros and accelerometers to assist in the control of the aerial vehicle. This paper also details the real time validation of robust fixed order lateral H_2 controller using HILS experimental setup. Flight trials are conducted to demonstrate the satisfactory performance of the designed controller.

7 References

[1] J. Blitch, "World Trade Center search-and-rescue robots", *Proceedings IEEE International Conference on Robotics and Automation*, Washington, D.C., pp 319-325 (2002).

[2] S.M. Ettinger, M.C. Nechyba, P.G. Ifju and M. Waszak, "Vision-guided flight stability and control for micro air vehicles", *Proceedings IEEE/RSJ International Conference on Robots and Systems*, Lausanne, pp 2134-2140 (2002).

[3] R. Fearing, K.H. Chiang, M.H. Dickinson, D.L. Pick, M. Sitti and J. Yan, "Wing transmission for a micromechanical flying insect", *Proceedings IEEE International Conference Robotics and Automation*, San Francisco, pp 1509-1516 (2000).

[4] J.M. Grasmeyer and M.T. Keennon, "Development of the Black Widow micro air vehicle", *Proceedings 39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, pp 1-9 (2001).

[5] M. Meenakshi and M. Seetharama Bhat, "Robust fixed order lateral H_2 controller for micro air vehicle", *Proceedings 9th International Conference on Control, Automation, Robotics (ICARCV2006)*, Singapore, accepted (2006).

[6] J.C. Stasko, R.L. Crandell, M.T. Dunn, M. Sureshbabu and W. Weber, "The versatile hardware in the loop laboratory: beyond the ad-hoc fixer", *Proceedings American Control Conference*, Philadelphia, pp 508-512 (1998).

[7] J.R. Wagner and J.F. Keane, "A strategy to verify chassis controller software: dynamics, hardware, and automation", *IEEE Transactions on Systems, Man and Cybernetics Part A*, 27(4), pp 480-493 (1997).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Automatic Book Identification and Retrieval using an Autonomous Robot

Aneesh N. Chand and Godfrey C. Onwubolu
Mechatronics Laboratory, School of Engineering and Physics
University of the South Pacific, Suva, Fiji Islands
chand_an@fit.ac.fj, onwubolu_g@usp.ac.fj

Abstract

In this research we develop a robot intended to eliminate the human component in the book retrieval process by autonomously navigating to and retrieving a specific book from a bookshelf. Notably a salient feature of this autonomous robot is the employment of an elementary book detection and identification technique in the formulation of a time-optimal and computationally basic method in contrast to vision-based techniques which has been used by researchers in similar work.

Keywords: automatic book retrieval, book detection and identification

1 Introduction

An emerging paradigm shift in robotic engineering developments is the development of *service* and *task-oriented* robots for providing assistance to humans in daily life [1]. As such there are now available a broad genre of service robots including but not limited to, nursing and security robots [2], escort robots [3] and hall polishing, cleaning and vacuuming robots. Such robots offer the merit of setting unprecedented levels of automation.

Commonly underlying to these robots is that they are developed for real-life applications that are usually labour-intensive and requires explicit human presence and effort; deployment of these robots then has the connotation of establishing a reduction in the need for human intervention in the task to be executed. This directly results in a reduction of human involvement and time and relegates humans to a safer but more potent role as planner. In addition, [4] cites numerous instances of autonomous robots being used in material handling and warehousing, in the assembly line and in the execution of common household chores.

In retrospect, we also designed a robot for a scenario that requires considerable human effort; the principal objective of this research is the design, formulation and synthesis of the hardware and software systems in the development of a Linux-based fully autonomous book retrieval robot. The current search for books is labour-intensive; this research therefore aimed to develop a service robot that would automate the process of book searching in a typical library system. Moreover emphasis was placed on the formulation of a computationally elementary but effective book identification method.

A picturesque analogy of the conceptual framework of the research is illustrated in figure 1. A user searching for a book using the online database first retrieves the book call number. The user then enters the call numbers through a laptop computer onboard the robot. The robot navigates to the appropriate shelf and begins searching for the target book. Upon searching for the target book, the robot then retrieves the book and retraces its path back. Customised software written in a high level language (C++) provides the intelligence and custom made hardware provides the functionality for this robotic vehicle.

2 Book Detection and Identification

Researchers working on similar robots have shared a high degree of commonality in the technology they have used for the identification and selection of books [5] [6] [7]; this has been the employment of vision based techniques. However, the research reported in this paper employs a much economical, simpler yet equally effective identification technique with much greater computational simplicity compared to image processing. Image processing in its most basic form incurs computationally exhaustive tasks including but not limited to image processing, line detection, magnification, histogram equalisation, binarisation, filtering, translation of pixels to text using Optical Character Recognition (OCR), contour detection and expansion. This can be considered to be a major demerit of image processing and hence it comes as no surprise that results state that it takes a maximum of one (1) second to localise a label from an image and an additional 5-10 seconds for the image processing and OCR phase in the identification and realisation of a single book [7]. With this magnitude of duration, the scanning and localisation of an entire bookshelf will no doubt incur a much higher processing time and render the robot inefficient, compared to the time it takes a human to identify a book. Conversely, a barcode scanner, the device used for scanning and identifying books in this research, provides input digital data ready for processing unconditionally and instantaneously. The key criterion in the selection of the barcode scanner was that the output is available

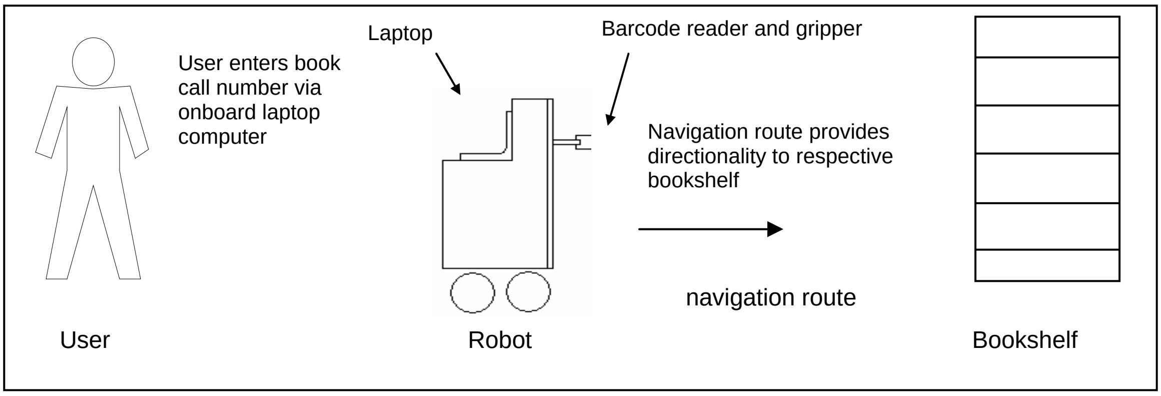

Laptop
Barcode reader and gripper
User enters book call number via onboard laptop computer
Navigation route provides directionality to respective bookshelf
navigation route
User
Robot
Bookshelf

Figure 1: Conceptual framework.

digitally. This feature makes it possible to implement comparisons in software through the use of elementary decision structures without any form of conditioning to the input data. Another imperative merit of a barcode scanner is that its utilisation implies machine and platform independency concerning software development and deployment. This allows its integration over any computer and operating system compared to cameras which require explicit installation of appropriate drivers. More details of using a barcode scanner for book identification are given in subsequent sections.

Any object has to have a unique identity in order for it to be recognised. Barcodes provide an inexpensive way of doing this. This is what makes the task of book identification and selection straightforward: books have barcodes placed on their spines with their specific information (book call number) available in it; more importantly this information is available digitally, which implies that once the robot reaches a specific shelf, it can easily start scanning the spines of the books and store the information in a buffer and constantly compare the scanned call number to the call number entered by the user. Once it makes a match, it indicates that the target book has been identified. The comparing of the scanned and entered call number is implemented in software and it is one of the most basic features of artificial intelligence.

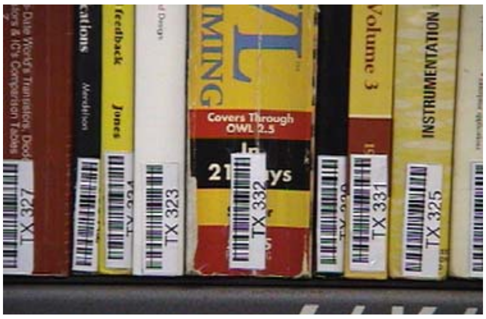


Figure 2: Barcodes provide digital signatures to books.

A typical barcode scanner forms the scanning device. The barcodes are positioned on the lower spines (figure 2) and the barcode scanner is mounted on the robot to scan the barcodes. Once the barcode scanner matches the book call number of a scanned barcode with the user-entered book call number, it identifies a match and the robot halts promptly. A gripper and book retrieval mechanism, positioned precisely over the barcode scanner, then start executing a systematic sequence of mechanical manoeuvres that pry the target book out of the book shelf using the upper edge of the book spine. It is imperative that the barcode scanner used has a fast response time and a large reading distance. The barcode scanner used in this research was the handheld PS2 port-interface type scanner with a reading distance of 10cm. Initial results and current operation of the barcode scanner looks viable as an identification device and more results are expected to be derived during formal testing which will validate the notion of using a barcode scanner as a book identification device. A scanner used in similar work [8] has achieved considerable success. The following sections outline the design details of the book retrieval robot.

3 Hardware Design

The robot deployed for the purpose of book retrieval consists of a mobile wheel-driven platform base encompassing a gripper and book retrieval manipulation system with four axes of movements. The size of the robot is chosen such that it is able to accommodate a laptop computer (the principal controller) while simultaneously minimising the weight, mechanical complexity and constructional cost. The maximum height of one and a half meters (1.5m) to which the gripper rises is attributed to the anticipated height of the bookshelves. This book retrieval robot is capable of retrieving a book of dimensions 25cm x 17cm x 8cm and of weight 1kg.

A differential drive steering mechanism is used to provide locomotion to the robot with the main chassis being built over this steering base. The base has passive castors for stability and a line tracing mechanism to follow the navigation route. The navigation route is a white line leading to and running along a bookshelf. A two finger gripper configuration is used to securely grip a book. . The barcode scanner, gripper and book retrieval mechanism are constructed precisely over each other, as the barcode scanner provides the reference point for the book extraction execution. To provide vertically directionality to the barcode scanner, gripper and the book retrieval mechanism, a lead screw and pulley based mechanism is used to provide vertical actuation so that the robot can scan and retrieve books from different heights; sensors indicate the different heights of the bookshelf rows. The robot encompasses mechanical components which are a hundred per cent custom made and built in-house. The robot is shown in figure 3.

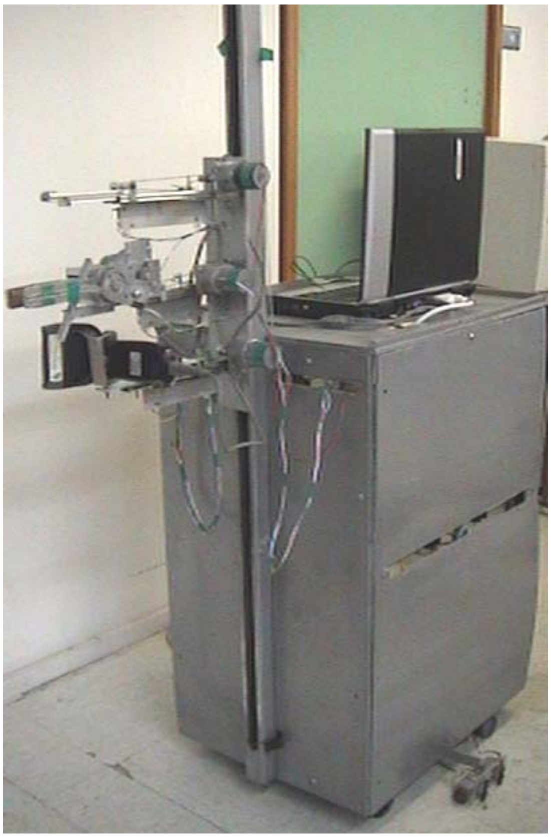
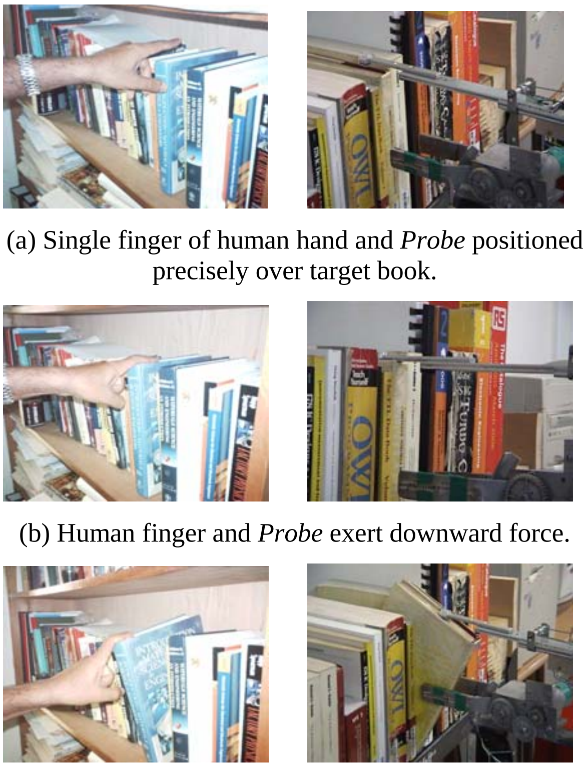


Figure 3: The book retrieval robot.

3.1 Book Retrieval Mechanism

The mechanism for systematically prying a book out of a shelf is constructed based on replication of the actions of the human hand. Figure 4 shows the various sequences of movements of the human hand in retrieving a book out of a shelf. Based on these movements, the book retrieval mechanism was conceived and hence the Probe (the symbolic name allocated to the book retrieving mechanism) constructed to replicate the human hand. This particular book retrieval mechanism was selected as it does not require the assumption that there are any spaces between the books. The Probe is a relatively simple mechanism made out of aluminium. It has inclined grooves fabricated, opposite to the direction of book retrieval motion, on the end that retrieves a book to create the necessary friction. A motor is used to actuate this back and forth horizontally on a linear slider with a nylon rope drive being used to transmit power from a motor shaft.



(a) Single finger of human hand and *Probe* positioned precisely over target book.

(b) Human finger and *Probe* exert downward force.

(c) Human finger and *Probe* pull back. Downward force and friction present have the effect of prying the book out of the shelf.

Figure 4: Comparison of actions of human hand and the *Probe* in retrieving a book.

Once the scanner has identified the target book and the robot has halted, the *Probe* extends horizontally into the bookshelf and is positioned precisely over the target book. The *Probe* then starts to move vertically down, applying a downward force on the book. When a certain magnitude of force is attained (the magnitude of this force is determined empirically), the *Probe* then retracts back horizontally. Since there is a sufficient downward force and friction also present, the book starts to tip on its end and into the gripper fingers. The full operation of the *Probe* is graphically illustrated in figure 4(a)-(c). The full sequence of mechanical actions in a book retrieval operation is given in section 5.

3.2 Electronics

The control electronics of the book retrieval robot encompasses a custom designed Enhanced Parallel Port (EPP) based, bidirectional, software selectable, input and output parallel port-interfacing card [9], eight (8) motors and eight corresponding motor controller circuits as actuators. In addition, various sensors are integrated for obstacles detection, navigation and force feedback for the gripper and the book retrieval mechanism.

The principal controller for the robot is a P4 1GHz laptop computer, onboard the robot. The secondary controller is a Microchip Technology PIC16F877 eight bit CMOS microcontroller with built in EPROM. This microcontroller is available as a 40-pin DIP package containing a central processor, EPROM, RAM, timer(s), and TTL / CMOS compatible or user defined input and output lines. This microcontroller is responsible for coordinating and receiving sensory information. Also, the *interrupt* feature of the microcontroller is heavily utilised.

3.3 Barcode Scanner

The notion of using a barcode scanner for book identification has already been presented in section 2; the justification for its selection is given in this section.

For ease of implementation and mass-market appeal, the device chosen for the identification of books had to be an optimum non-volatile, passive (requiring no external power) memory product, inexpensive and readily available. A key criterion of the device had to be that its output should be available digitally. This feature would make it possible to implement comparisons in software through the use of elementary decision structures. The best candidate thus identified was the barcode reader/scanner.

The selection of a suitable barcode scanner was then made based on a number of considerations, listed in table 1. In conformation to these specifications, the barcode scanner thus selected was the *ProTracer ANL810-K* with PS2 interface

Table 1: Barcode scanner selection criteria.

Selection Factor	**Present**
PS2 interface?	Yes
Multiple barcode decoding?	Yes
Free from hardware emulation for laptop operation?	Yes

Selection criteria 1 was imperative in the software programming of the barcode scanner, further explained in section 4. With several types of barcode encoding formats present, criteria 2 ascertains the non-restriction of the usability of the barcode scanner while criteria 3 ascertains the operation of barcode scanner with the onboard laptop computer. For the reason that the internal physical construction (electronics) of a barcode scanner varies from manufacturer to manufacturer, its operation is therefore treated in oblivion to the internal electronics and hence they are not presented here; only the data written by the scanner to the laptop is of specific importance. This is discussed in the next section.

4 Software

The software that provides the robot cognition and intelligence is written in the C++ programming language under the Linux OS (Red Hat distribution version 9). Linux is chosen as it provides direct access to hardware and fares better than other conventional operating systems such as Windows in real time hardware operations.

All *behaviour*s of the robot (mechanical actions) have been represented as functions and all *attributes* are represented as data variables in the software. Invocations of these functions in their required sequences attain the book retrieval manoeuvres. Logically related functions and data are then grouped together and have been implemented in classes as methods and attributes which imply Object Oriented (OO) methodology. The software development phase has utilised the OO methodology to attain abstraction and modularity.

There are in fact three layers of the software, representing a three tier system. The first tier is the human-machine interface in the form of a Graphical User Interface (GUI). The GUI has been developed using Borland's Kylix C++ IDE. The second layer is the main robot control software encompassing the robot intelligence, logic behaviours and mechanical functions. This tier is developed using the Linux g++ compiler. As stated previously, a PIC microcontroller has also been used as a secondary controller and this constitutes the third tier encompassing the utilisation of features such as interrupts and an analog-to-digital converter (ADC). The software organisation is demonstrated in figure 5.

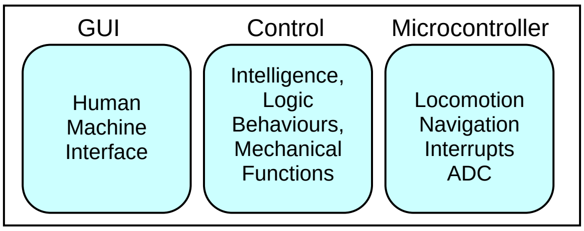

GUI
Control
Microcontroller
Human Machine Interface
Intelligence, Logic Behaviours, Mechanical Functions
Locomotion Navigation Interrupts ADC

Figure 5: Software nomenclature.

The software developed for the barcode scanner is fundamentally a half duplex, interrupt driven, software implementation of PS2 communications. Reception of a byte written at the PS2 port by the barcode scanner is automatically invoked whenever the barcode scanner scans in a barcode. The software codes for the data retrieval are part of standard C++ language/libraries and hence guarantees that the code used to retrieve data from the PS2 port will be platform-independent and thus compile under either Windows® or Linux platform while simultaneously having the merit of being machine-independent. Since it can be seen that the software retrieval of data scanned by the barcode scanner does not utilise any industry standard software such as MATLAB or

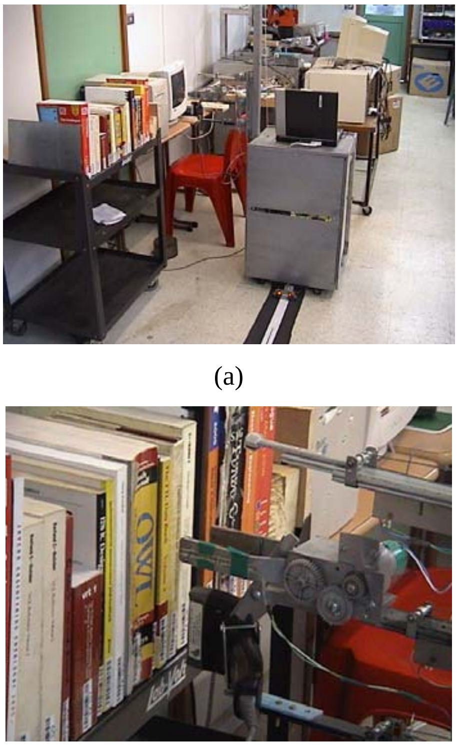
LabVIEW and uses elementary C++ programming, it does not constitute computation overhead as is the case with using computer vision for book detection and identification as done by [5], [6] and [7].

5 Theory of Operation

We now summarise the sequence of events in an automatic book retrieval process, synthesising the concepts that have been presented so far.

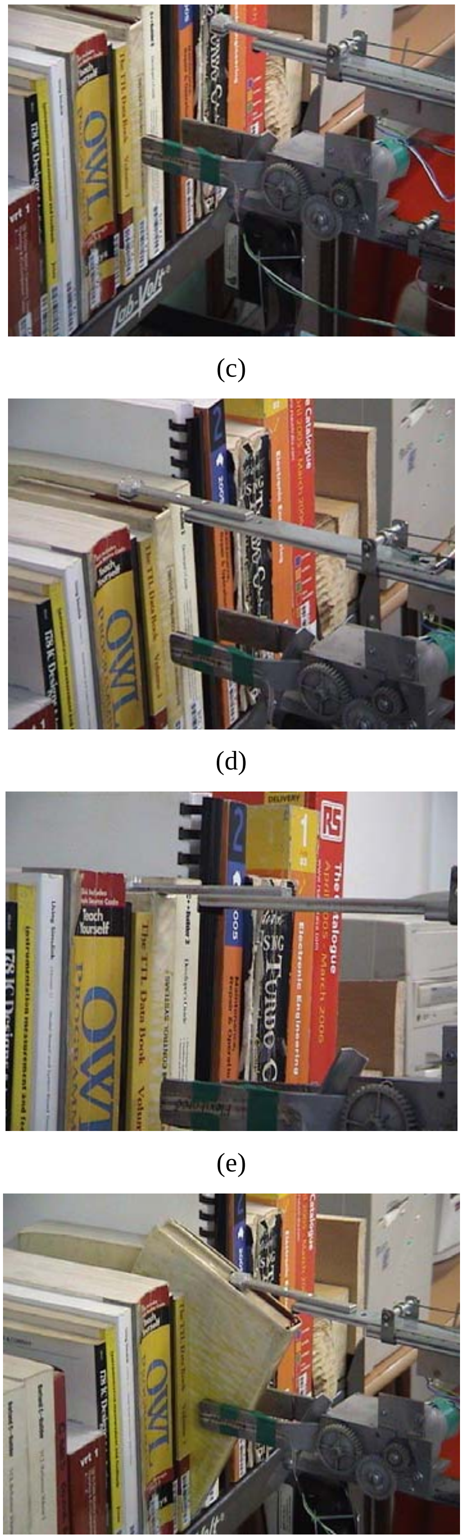
- The steering mechanism and chassis provides the locomotion for the robot. The guide paths provide the route for navigation (figure 6a).
- Once the barcode scanner has identified the target book, the robot halts precisely in front of it (figure 6b).
- Barcode scanner retracts back horizontally to make clearance (figure 6c).
- The *Probe* then enters the bookshelf exactly on top of the book (figure 6d).
- *Probe* then starts moving down to apply downward force (figure 6e).

When enough force has been attained, *Prob*e moves back horizontally to pry the book out of the shelf. The downward force and fiction present has the effect of tilting the book on it end. The gripper fingers then close, gripping the book (the gripper fingers are by default open spanning a distance greater than the maximum book width of 8cm) and then retracts back (figure 6f).



(a)

(b)



(c)

(d)

(e)

(f)

Figure 6: Sequence of events in a book retrieval operation.

6 Experimental Results

Several preliminary experiments were carried out for the validation and verification of the performance of the system. The major metrics of measurement were the optimum robot *speed* that suffices to accurately scan book barcodes, the *accuracy* of and the *time* required for the detection and identification of books. The accuracy and book seek-time analysis is important in comparing the results of other researchers and to determine the efficacy of the barcode scanner as the book identification device. From current experimental results, the robot is capable of correctly detecting and identifying books with 99% accuracy at a speed of 0.6m/s. It takes an average time of 0.45 seconds for the robot to detect, scan and realise the identity of a single book and is able to successfully retrieve books 53% of the time. These results were obtained in a controlled environment encompassing a number of required assumptions such as lighting conditions, presence of obstacles and dimensions of books. This seems reasonable as in a realistic environment there would be numerous dynamic factors and it would be quite complex to generalise the behaviour of these variables

7 Conclusion

The design, development and deployment of a prototype book search and retrieval robot has been presented and successfully attained. Customised cross-platform compatible software, written in C++ language, forms the intelligence of the robot while an onboard laptop computer, running on Linux platform of Red Hat Distribution Version 9.0, provides the processing and execution of intelligence of the robot. A Graphical User Interface (GUI), developed using Borland's Kylix® acts as the human machine interface allowing a user to submit a desired book call number. DC motors are used for locomotion. Infra red line tracer sensors are used for the navigation of the robot with lines embedded in the floors marking the routes and giving the robot a priori knowledge of the paths it has to take. Bar-coding technology is used to provide digital signatures to library books, encoding its call number into the barcode. The employment of an innovative and practical system, a typical and inexpensive barcode reader, provides the technology with which books are searched for. Current ongoing work includes improvising the performance of the robot and incorporation of features such as circumnavigation around obstacles, localisation and automatic self-recharging that will solve the problems of deployment in a realistic environment.

8 Acknowledgements

The writers acknowledge the Faculty of Science and Technology Research Committee of the University of the South Pacific for the provision of research funds.

9 References

[1] S. Yuta, “Autonomous mobile robotics research for daily life environment”, *Proceedings 4th IFAC Symposium on Intelligent Autonomous Vehicles (IAV2001),* Sapporo, pp 32-39 (2001).

[2] D.A. Carnegie, D.L. Loughnane and S.A. Hurd, “The design of a mobile autonomous robot for indoor security applications”, *Journal of Engineering Manufacture: Proceedings of the Institution of Mechanical Engineers, Part B,* 218 pp 533-543 (2004).

[3] A. Ohya, Y. Nagumo and Y. Gibo**,** “Intelligent escort robot moving together with human - methods for human position recognition”, *Proceedings Joint 1st International Conference on Soft Computing and Intelligent Systems and 3rd International Symposium on Advanced Intelligent Systems (SCIS & ISIS 2002),* Tsukuba 24B5-2 (2002*).*

[4] T. Fukuda and Y. Hasegawa, **“**Perspectives on intelligent control of robotics and mechatronic systems”, *Proceedings 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003),* Singapore (2003).

[5] T. Tomizawa, A. Ohya and S. Yuta, **“**Book Extraction for the Remote Book Browsing Robot”, *Journal of Robotics and Mechatronics,* 16(3), pp 264-270 (2004).

[6] T. Tomizawa, A. Ohya and S. Yuta, “Remote book browsing system using a mobile manipulator”, *Proceedings IEEE International Conference on Robotics and Automation,* Taipei, pp 256-261 (2003).

[7] M. Prats, R. Ramos-Garijo, P.J. Sanz and A.P. Del Pobil, “An autonomous assistant robot for book manipulation in a library”, *Intelligent Autonomous Systems*, 8, pp 1138-1145 (2004).

[8] J. Suthakorn, S. Lee, Y. Zhou, R. Thomas, S. Choudhury and G.S. Chirikjian, “A robotic library system for an off-site shelving facility”, *Proceedings IEEE International Conference on Robotics and Automation,* Washington DC, pp 3589-3594 (2002).

[9] G.C Onwubolu, S. Aborhey, R. Singh, H. Reddy, M. Prasad, S. Kumar and S. Singh, “Development of a PC-Based computer numerically controlled drilling machine”, *Journal of Engineering Manufacture: Proceedings of the Institution of Mechanical Engineers, Part B,* 218, pp 1509-1517 (2003).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Autonomous Agents with Need and Curiosity

Ramesh K. Rayudu
Institute of Information Sciences and Technology,
Massey University, Palmerston North, New Zealand
r.k.rayudu@massey.ac.nz

Abstract

This paper discusses our research towards development of a domain-independent autonomous agent that can learn and adapt to function in a domain. For an agent to work autonomously in a domain, it needs some motivation to get familiar with its environment. In this paper we proposed a model with two emotions that can motivate the agent. The model was developed and tested in a virtual world and their results are discussed. This paper concludes that our algorithm can perform its basic goals successfully without explicit domain based knowledge.

Keywords: autonomous agents, emotions, simulation, domain independent, modelling, agent architecture

1 Introduction

This paper discusses a model for developing an autonomous agent for complex environments. Since mid-nineties several autonomous agent theories have been proposed and developed, and a few have been deployed in environments such as satellites [1] and robots [2, 4]. However, most of the agents require explicit description of problem domain and is a time-consuming process. This paper focuses on developing a domain independent agent that uses emotions as a mechanism for organisation of agent behaviour related to a problem domain.

The central topic of this paper is an agent model with need and curiosity. Need and curiosity [6] are regarded as subsets of emotion. Emotion is that drives both self-regulation and self-governing properties in an autonomous agent [2, 5, 7].

Modelling emotion in agents and robots has been an active area of interest [3, 4]. Several architectures have been proposed and implemented but there is still a greater need to develop an architecture that is truly domain independent.

Emotion in an agent can be modelled by some adaptive mechanisms that can simulate motivations such that, using the motivations, it can learn about the problem domain [3]. There are several emotional states in a biological entity [9, 3] but, for this project, we chose only need and curiosity as main emotional factors to achieve our goals. The reason being, we believe that, the two emotions are primary to make an autonomous agent.

Based on these arguments, we have developed ANCA. ANCA (Agent with Need and Curiosity Architecture) is an autonomous agent architecture whose mental processes are guided by need and curiosity. These emotional states are adjusted to govern the motivational states of the agent. It is important to note here that the ‘need’ emotion can be application and goal-oriented.

This paper discusses the developmental details of ANCA. Section 2 introduces ANCA’s architecture and its details are discussed. The implementation of the architecture is discussed in section 3. The autonomous activity experiments and ANCA’s relational learning are detailed in section 4 and the results are discussed in section 5. Section 6 concludes and summarises our results.

2 A Model for Autonomous Agent

ANCA is an agent model for an autonomous agent with no domain specific knowledge. The general architecture of ANCA is shown in the figure 1 and its subsystems are discussed in this section.

Sensors monitor the world and receive the raw data related to the agent’s current state and its environment.

The process ‘sensation’ transforms the raw data received from the sensors into a format suitable for agent’s reasoning. It stores the raw data in terms of an n-tuple of state variables which contains data related to agent’s location and the objects of the world it discovers. Whenever, the agent discovers an object of its world, it creates an event and stores it in the memory of sensation.

The Notion system maintains a logical image of the agent’s perceived environment. The image can be read, analysed, stored and updated by the processes in Affection system. The image is stored in a form of beliefs.

The Affection system incorporates the emotional processes and organises their execution. The emotional states are activated by the assessment of their strengths and beliefs.

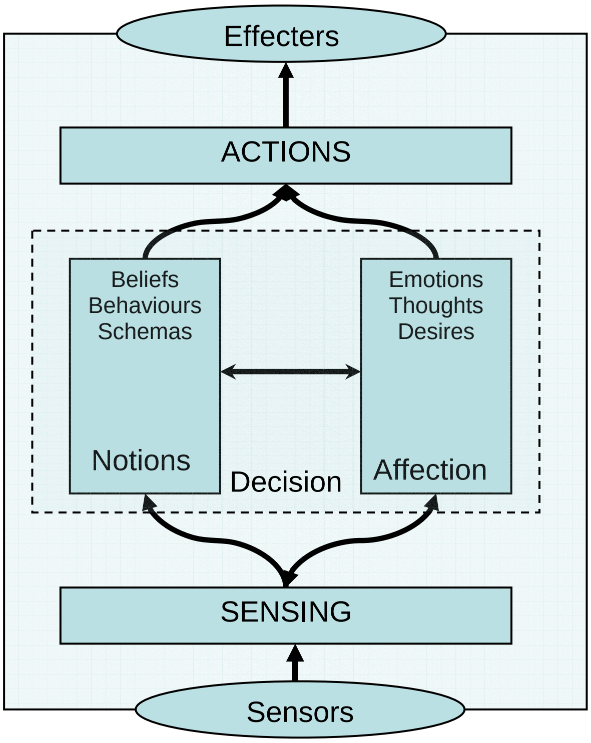


Figure 1: ANCA architecture.

The Action system comprises of the mechanisms to learn and execute the plans and behaviours of the agent. The learnt knowledge is stored in terms of schemas and behaviours and the plans are executed using the effecters.

The following section discusses ANCA's processes in detail.

2.1 Sensation

The Sensor information about the agent's world is stored in an n-tuple of variables

$$\left\langle v_1, v_2, v_3, ...v_i, ...v_n \right\rangle$$

where v_i is the data related to the agent and the world it encounters.

The variables are related to the agent location and information on the objects it discovers. All related information is stored in the list. Examples of this information include agent's absolute position; absolute location of the discovered object, object properties, relative position of the agent to the object, and pathway coordinates of the agent's movements.

The details of the object discovery and the pathway details leading to the object are aggregated and stored as events in the world.

2.2 Decision

As shown in figure 1, the decision section of ANCA comprises of Notion and Affection processes.

The logical image of agent's environment is stored in ANCA as a set of beliefs of Notion process. Beliefs are concepts developed and executed by 'thoughts' of Affection process that are unique and repeatable. Beliefs represent the learnt knowledge of the agent. They can be generated, updated, applied and destroyed by the processes of the Affection system.

A belief is characterised by

$$B_i = \left\langle v_i, C_i, SE_i \right\rangle$$

where v_i is data related to the agent and object, C_i is confidence value and SE_i is a set of emotions that are related to the belief. Beliefs are invoked by emotions when needed.

Emotions are part of the Affection process as shown in figure 1. Emotions such as need and curiosity are intrinsic in ANCA. However, there is also a need to incorporate 'application' related emotions in ANCA. For example, for an ambulance agent, one has to program the fact that the agent has to look for patients to transport and in shortest possible time[1].

Emotions in ANCA are a 4-tuple

$$E_i = \left\langle e_i, s_i, b_i, t_i \right\rangle$$

where e_i is the name of the emotion, s_i is the strength of the emotion, b_i is the Boolean that states the emotion state and t_i are set of related thoughts .

Emotions in ANCA can be termed as entities that specify a situation of the agent. It then affects the agent to perform an activity to reach a desired relation between the agent and the environment. Emotions in ANCA raise thoughts which, in turn, raise desires for execution.

Thoughts are the entities of the Affection system. They are activated by the emotions to fulfil a desire. Thoughts are related to performing tasks related to emotional requirements. For example, if the agent is *hungry*, that triggers the thought to *eat*. Thoughts are the required goals to perform the need. Thoughts in turn invoke a set of desires that the agent can choose to execute. For example, *find food* is a desire that can be invoked when the agent has to eat. Both thoughts and desires are a 3-tuple

$$T_i = \left\langle N_i, S_i, P_i \right\rangle$$

where N_i is the name of the desire, S_i is satisfaction value and P_i are the properties attached to the thought and desires.

Figure 2 shows the relationships between process mechanisms of decision and action processes. Emotions, thoughts and desires have a single direction

[1] ANCA is capable of learning the emotion the hard way but will be too late for the person being transported.

cause-and effect relationship and subsequently affect the actions of the agent. Beliefs, Behaviours and Schemas are ANCA's learning outcomes and are affected by both emotions and actions. If ANCA has a belief related to the goal to be solved, the thoughts and desires of the belief will be used in creating actions for execution.

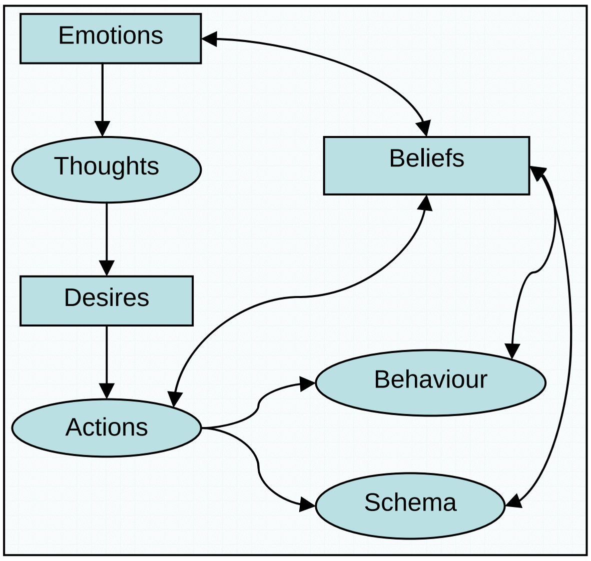


Figure 2: Process interactions in ANCA.

It is important to note here that the thoughts that become constructive and procedural in achieving a repetitive task are then turned into beliefs and stored. Thoughts also change the environment and send actions to the action system through desires.

2.3 Action System

The Action system reasons about the use of effecters in the progress towards its current goal. Effecters, the external mechanisms, achieve the goals of the actions. For example, the desire *look for food* might trigger an action to *move forward* or *move backward*.

The action process chooses and performs actions based on two mechanisms: Planned mode and Random mode. In planned mode, the actions are delivered through a fixed policy that has been planned and prepared for execution. Planned mode execution happens when the agent is pursuing a goal and it has a good understanding of either the environment or the goal. If the agent has partial or no understanding of the environment then the action process chooses random mode. In random mode, effecters are chosen at random and executed.

The relationship between random and planned modes is a learning algorithm that generates planned set of behaviours and schemas from the random mode exploration. Behaviours are a list of navigation steps the agent saves for future navigation in the world. Schemas are navigation steps the agent can ignore while learning new behaviours. For example, if the agent hits a wall and follows the wall to go around it, it creates a behaviour. If the agent turns right and hits another dead end, it creates a schema.

3 The Experimental Agent World

We demonstrate ANCA in a micro-world termed 'Hungry Agent'. The micro-world experiment demonstrates how the agent uses its emotions to survive in the world. The zoomed schematic of a 2000x2000 pixel model is depicted in figure 3. The environment has different kinds of food and objects that are movable and stationary. The environment also has rooms and access points for the agent to explore around. The main objective of the agent is to survive in the environment.

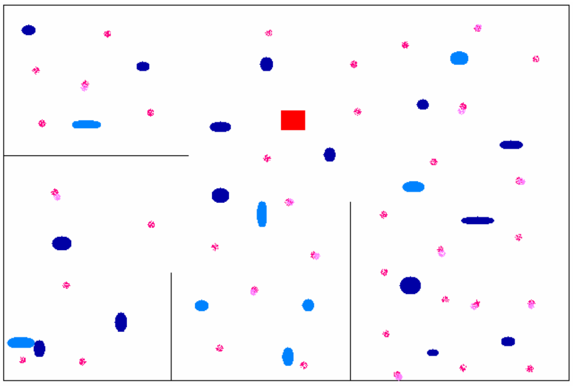


Figure 3: Zoomed part of agent world. The rectangle is the agent, small dots are food particles, and others are obstacles.

The agent is expected to roam around and get familiar with its world using the emotions of need and curiosity. The need is basically from agent's survivability such as finding and eating food, avoiding poisonous food and keeping a record of a food inventory. The curiosity is related to knowing more about the agent world, tracking and learning from its movements and providing support to agent's food inventory.

The agent sensors perceive the following

$$\left\langle xPos, yPos, ObjID, OxPos, OyPos, \left[Obj\Pr ops\right]\right\rangle$$

where *xPos* and *yPos* are agent's absolute co-ordinates, *ObjID, OxPos, OyPos* are object ID and its location coordinates, and *ObjProps* are properties of the discovered object.

The agent records all its movements and object characteristics until it encounters a new object. The cycle is complete when an object is discovered and, at that point, the learning algorithm saves the object details and analyses the agent movements to identify more behaviours, schemas and finally saves them as beliefs.

4 Results

To evaluate the performance of ANCA we have developed an agent with ANCA's learning and adaptive capabilities and another agent with no emotions (A2). The evaluation has been performed in terms of agent survival time, and behaviour utilisation.

In our first experiment we compared A2's survivability with ANCA. A2 doesn't incorporate any emotions and hence has no adaptability. The control related to actions in A2 has been programmed as procedures. For example, rules such as '*if the energy level is <40, then look for food and eat*' governs A2's actions.

We ran about 60 life cycles on ANCA and A2. The results are shown in figure 4.

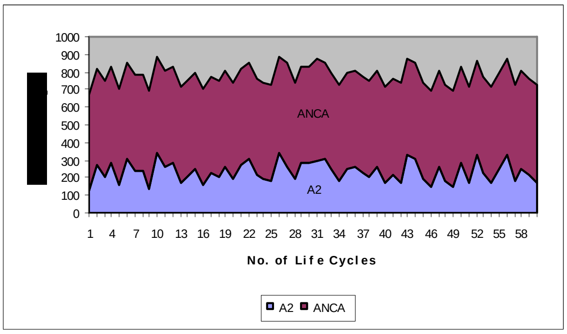


Figure 4: Survival of ANCA and A2.

From the chart, it is seen that ANCA outweighed A2 in survival. In fact, ANCA never died when we stopped the simulation at around 550000 steps. In all occasions A2 died after eating all the food that's around its vicinity and could not proceed further into newer territories. The average speed of travel for A2 has been the same as ANCA and even faster sometimes as the movement algorithm in A2 is a search algorithm with some delay steps.

The second test we performed was to analyse ANCA's learning and adaptive abilities. To do this, we have let ANCA and A2 to explore its world over certain number of steps and gave a goal to achieve. It is important to note here that the goal was same absolute location but the location of the agents has been different in each test. Figure 5 shows the average number of steps each agent took to achieve the goal.

It is clear from the graph in figure 5 that ANCA uses fewer steps to achieve the solution as it performs more actions in the simulation. The log files revealed that ANCA's use of saved behaviours and schemas have directly affected its future solution pathways. Since the goals are programmed emotions, they have high strength towards its achievement. Hence ANCA uses its behaviours and schemas to solve the problem.

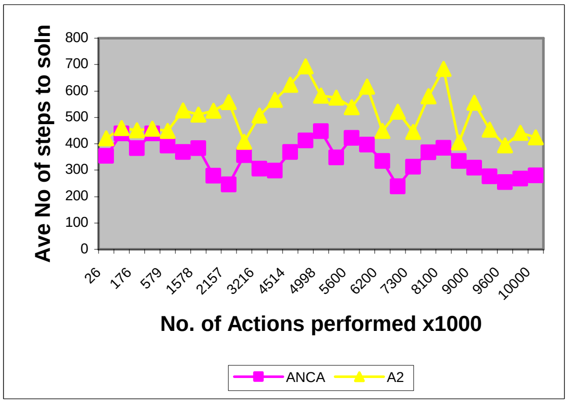


Figure 5: Adaptability of ANCA.

This experiment has shown that emotions are essential part of agency towards a more efficient and effective goal execution. We have also proven that that an agent can survive in an environment without any domain specific knowledge. However, the agent needs some domain based emotional and goal specifications.

5 Conclusions

A domain-independent autonomous agent architecture has been proposed and the different entities of the architecture detailed. ANCA uses need and curiosity emotions to develop motivation and uses thoughts and desires to learn and fulfil its responsibilities of its world. The proposed architecture has been implemented in a world where the agent primary goal is to survive. The experimental results have been compared to an autonomous agent (A2) with no emotions. Our experiments confirmed that ANCA performs better then A2.

Our future research will include testing ANCA in different micro-worlds, enhancing the sensing capabilities and improving ANCA's adaptability.

6 Acknowledgements

Thanks to Prof. Don Kulasiri and Dr. Sandhya Samarasinghe of Lincoln University, Canterbury, New Zealand, for their valuable suggestions in the early parts of this research.

7 References

[1] N. Muscettola, P. Nayak, B. Pell and B.C. Williams, "Remote agent to go boldly where no AI system has gone before", *Artificial Intelligence*, 103(1-2), pp 5-48 (1998).

[2] R.C. Arkin, M. Fujita, T. Takagi and R. Hasegawa, "An ethological and emotional basis for human-robot interaction", *Robotics and Autonomous Systems*, 42(3-4), pp 191-201 (2003).

[3] R. Nair and M. Tambe, "The role of emotions in multiagent teamwork", in J.M. Fellous and M.A.

Arbib (eds), *Who Needs Emotions?: The Brain Meets the Robot*. Oxford University Press, Oxford (2005).

[4] L. Steels, “When are robots intelligent autonomous agents?”, *Journal of Robotics and Autonomous Systems*, 15, pp 3-9 (1995).

[5] T.J. Norman, and D. Long, “Goal creation in motivated agents”, in *Intelligent Agents: Theories, Architectures and Languages*, Springer-Verlag, Heidelberg (1995).

[6] F. Kaplan and P-Y. Oudeyer. “Intelligent adaptive curiosity: a source of self development”, *Proceedings 4th International Workshop on Epigenetic Robotics*, pp 127-130 (2004).

[7] A. Sloman and M. Croucher, “ Why robots will have emotions”, *Proceedings 7th International Joint Conference on Artificial Intelligence*, Vancouver, pp 197-202 (1981).

[8] A. Sloman, R. Chrisley and M. Scheutz, “The architectural basis of affective states and processes”, in J.M. Fellous and M.A. Arbib (eds), *Who Needs Emotions?: The Brain Meets the Robot*. Oxford University Press, Oxford, pp 201-244. (2005).

[9] T. Turner and A. Ortony, “Basic emotions: can conflicting criteria converge?”, *Psychological Review*, 99(3), pp 566-571 (1992).

[10] M. Scheutz, “Agents with or without emotions?”, *Proceedings of FLAIRS 02*, AAAI Press, pp 89-94 (2002).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Multiagent System Model for Vehicle Platooning with Merge and Split Capabilities

Jean-Michel Contet, Franck Gechter, Pablo Gruer, Abder Koukam
Systems and Transportation Laboratory (SET),
University of Technology of Belfort-Montbéliard (UTBM), Belfort, France,
jean-michel.contet@utbm.fr, franck.gechter@utbm.fr, pablo.gruer@utbm.fr, abder.koukam@utbm.fr

Abstract

This paper presents a Multiagent based approach to the vehicle platooning problem. Our model is an alternative to centralised control solutions. In our approach, vehicles are autonomous entities in mutual interaction. The behaviour of each vehicle is determined from a physics inspired model that minimises interactions: each vehicle relates only with the preceding one in the platoon. This allows for the emergence of all required collective behaviours: steady platoon motion over arbitrary trajectories, vehicle merging and splitting. Furthermore, a set of safety properties which are crucial to the viability of the application, such as stability of inter-vehicular distance, are demonstrated thanks to the intrinsic properties of the physics inspired model. Other aspects such as trajectory matching and merge/split behaviour have been validated by both simulation and experimentation with mini-robots.

Keywords: platoon, reactive multiagent, longitudinal and lateral control, merge and split, physics inspired behaviour model

1 Introduction

In the last twenty years, many works have been done in research projects like PATH [1] to increase traffic safety and efficiency using automatic vehicles. These vehicles have a perception of the environment, and provide automated assistances (speed and distance control, obstacles detection, platooning, automatic car parking). Thanks to New Information and Communication Technologies (NICT) previous research can be put into practice in modern transportation. These public transportation systems provide new services such as car sharing or public open access to vehicle units. Collaborative driving is a research domain which aims to design automated vehicles that collaborate in order to navigate through traffic. Among this kind of systems, platoons are formed by a virtual train of semi-automated vehicles.

The main problem related to platoon systems consists in controlling the global platoon geometry: inter-vehicular distance and trajectory matching. Generally, the platoon control system splits up into longitudinal control and lateral control. Longitudinal control is one of the main aspects to be worked out. It consists in controlling braking and acceleration in order to stabilise the distance between the leader vehicle and the follower. This control takes as parameter the distance between the preceding and following vehicle, or the time which separates these vehicles, depending on the model used. Sheikholeslam and Desoer [2] proposed longitudinal control using linearisation methods. Ioannou and Xu [3] controlled the brakes and acceleration by fixed gain PID control (proportional, integral, derivative) with gain scheduling. Hendrick et al. [4] used a control mode based on a non linear method with PID. Lee and Kim [5] proposed a longitudinal control by fuzzy logic. In related work, the following of robot by fuzzy logic is proposed in [6]. Lateral control consists in aligning the vehicle direction in relation to the preceding vehicle. Daviet and Parent [7] proposed lateral control by a PID controller. This control consists in keeping close to zero the angle between the preceding and the following vehicle. In order to model vehicle platoon systems with longitudinal and lateral control, Gehrig and Stein [8] inspired on physical particles submissive forces. Yi and Chong [9] represented immaterial fixing with an impedance control model. Halle [10] used multiagent systems (MAS) in order to model immaterial vehicles fixing using constant values from [7].

In this work, we aim to introduce a multi-agent system that provides two functionalities. On one hand, longitudinal and lateral control and on the other hand, merge and split capabilities. Multiagent systems have become an attractive approach for problem-solving and have been used to a wide range of applications and simulations [11], such as localising and tracking targets [12]. Among the classical models, the reactive approach is one of the most interesting due to its robustness, adaptability and simplicity. In this article, MAS are composed by reactive agents. In our model, as an alternative to centralised control solutions, vehicles are autonomous entities in mutual interaction. The behaviour of each vehicle is determined from a physics inspired model that minimises interactions: each vehicle relates only with the preceding one in the platoon. This allows the emergence of all required collective behaviours: steady platoon motion over arbitrary trajectories, vehicle merging and splitting.

This paper is organised as follows. First part presents the main aspects of agent behavioural models, in an independent fashion, relatively to the intended control model. Second part is dedicated to the impedance-control model. Then, section four deals with merging and splitting. Finally, experimental results are presented and discussed.

2 The Multiagent Model

2.1 Roles and Interactions

In order to define the multi agent model, we first identify different roles. A role is a set of abstract behaviours, each characterised by a set of interactions. The vehicle platoon application includes two main roles: vehicle and platoon. Vehicle role specialises towards two sub-roles: head vehicle and follower vehicle. The head vehicle role interacts with the environment (road). The follower role interacts with another vehicle and with the environment. The nature of interactions can be diverse: explicit communication or simple mutual perception. In our case it depends on the physics-inspired model presented in a later session.

2.2 Agent Behaviour

We use the language of statecharts as the formalism for the specification of agent's behaviour. Statecharts are widely used in behavioural descriptions, because of their expressive power and well founded semantics. This has been the reason why we adopted the statecharts in the frame of our activity on the multiagent system's formal specification and verification [13].

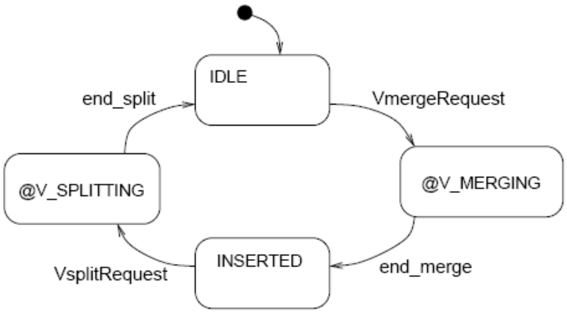


Figure 1: Vehicle agent behaviour.

Figures 1 and 2 describe respectively the *Vehicle* and *Platoon* role behaviours at a high level of abstraction. State names that begin by an ampersand character include a sub-statechart, which describes the behaviour of a sub-role. The follower vehicle (*Vehicle* for short) role behaviour is formed by a cycle of four sub-roles representing vehicle at rest, merging with the platoon, inserting in the platoon and splitting from it. The *Platoon* role behaviour represents three global situations: the platoon is steady, a vehicle is merging with the platoon and a vehicle is splitting from it.

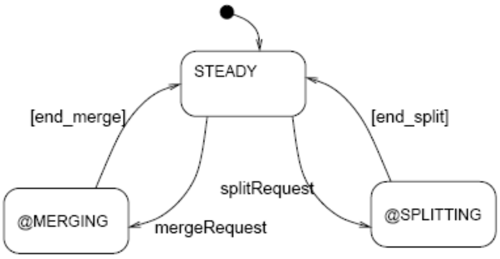


Figure 2: Platoon agent behaviour.

According to this hierarchy, Figures 3 and 4 describe the *MERGING* and *SPLITTING* sub-role behaviours of *Vehicle* and *Platoon* roles. The *MERGING* sub-role for the Vehicle (*Vehicle: MERGING*) waits for the detection of the last vehicle in the platoon, which will be his followed vehicle. Then, the merging process begins.

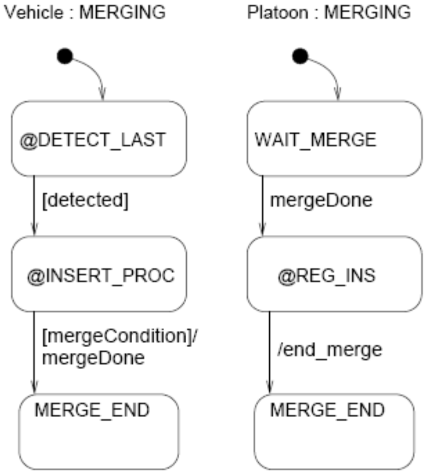


Figure 3: The MERGING sub-roles.

The merging process consists in first detecting the condition to insert, then launching the merging procedure. The mergeCondition is verified when the following distance is the required one to merge a new vehicle without danger. When the mergeCondition is satisfied, the *Platoon:MERGING* process is made aware and refreshes the values of a set of attributes, to take into account the merging at the platoon level.

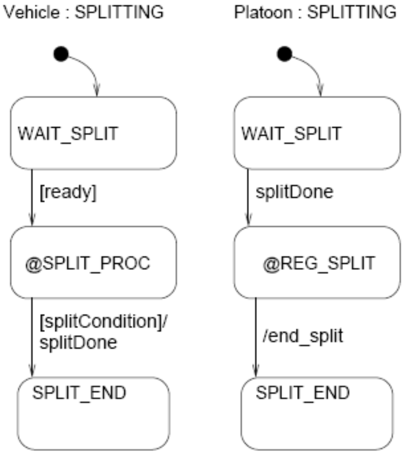


Figure 4: The SPLITTING sub-roles.

The SPLITTING sub-roles (figure 4) consists in first be ready to split, then launching the splitting procedure. The splitCondition is verified when the distance is convenient in order to split a vehicle out of the train. When the splitCondition is satisfied, the *Platoon:SPLITTING* process is made aware and refreshes the values of a set of attributes, to take into account the splitting at the platoon level.

The functions performed in the V_MERGING, V_SPLITTING and INSERTED states of figure 1 are based on algorithms defined from an interaction model. The following section presents the one that we adopt.

3 Interaction between Vehicle and Preceding Vehicle

3.1 Interaction Model

The connection between two vehicles is made by an impedance-control model like in [9]. The virtual link connection between each vehicle is a spring damper with stiffness *k*, damping *h* and non-stretched spring length l_0. These forces are composed to exerted force by the spring F_s, the shock absorber force F_a and the friction force of the surface F_f. The interaction with the environment accounts for friction parameters λ. Figure 5 shows a 3-vehicle part of the considering platoon. Each vehicle *i* is represented by its position $\vec{X_i} = [x_i, y_i]$. The mass of the vehicle is denoted as *m*. The distance between vehicles is:

$$d = \|\vec{X}_{n+1} - \vec{X}_n\|$$

Spring force $F_{s:}$

$$\vec{F_s} = -k(\|\vec{X}_{n+1} - \vec{X}_n\| - l_0)\vec{u_{n+1\ n}} \qquad (1)$$

Shock absorber force $F_{a:}$

$$\vec{F_a} = -h(\dot{\vec{X}}_{n+1} - \dot{\vec{X}}_n) \qquad (2)$$

Friction force of the surface $F_{f:}$

$$\vec{F_f} = -\lambda\dot{\vec{X}} \qquad (3)$$

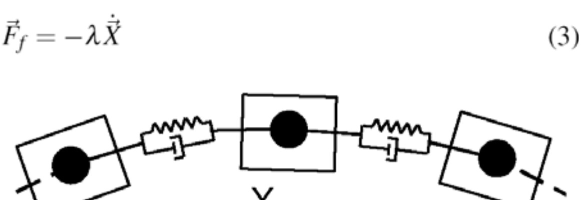


Figure 5: Simplification of forces applied to the vehicle.

From Newton's law of motion, in case of equilibrium of forces, relatively to the preceding vehicle:

$$m * \vec{\gamma} = \sum_{force} \vec{F_i} \qquad (4)$$

From this equation, acceleration γ can be computed. By discrete integration, we can then determine the speed and the vehicle orientation angle[1], which are taken as reference signals for vehicle motion.

3.2 Interest of the Model

The physics inspired model is used to specify a MAS. Each vehicle is represented by a reactive agent. The behaviour of a reactive vehicle agent is calculated from agent-environment and agent-agent interactions and perceptions. The physics inspired model presented in the previous section is similar to the one presented in [9]. The difference with previous work consists in the fact that we take into account only a local point of view. For each vehicle agent, perception is limited to an estimation of relative position of preceding-vehicle. Action is decided using only this perception. Since global stability is achieved from the local actions of each agent, it becomes important to establish a proof of the emergent global stability. We can verify that the new model is stable by following an approach based on an energy variation.

3.3 Proof of Stabilisation

The proof of vehicle stability is made by considering energy, which is composed of kinetic and potential energies. We take vehicle *n* as reference frame to express the energy knowing that the head vehicle index is 0. So $\vec{X}$ represents the difference between the distance separating vehicles *n-1* and *n* and the non-stretched spring length. Also $\dot{\vec{X}}$ expresses the speed difference between vehicles *n-1* and *n* (figure 5).

$$E = E_{pot} + E_{kin} = \frac{1}{2} * m * (\dot{\vec{X}}^t . \dot{\vec{X}}) + \frac{1}{2} * k * (\vec{X}^t . \vec{X}). \qquad (5)$$

however,

$$\frac{dE}{dt} = \frac{\partial E}{\partial \vec{X}} * \frac{\partial \vec{X}}{\partial t} + \frac{\partial E}{\partial \dot{\vec{X}}} * \frac{\partial \dot{\vec{X}}}{\partial t} \qquad (6)$$

so, since $\frac{\partial E}{\partial \vec{X}} = k * \vec{X}$ and $\frac{\partial E}{\partial \dot{\vec{X}}} = m * \dot{\vec{X}}$, we obtain

$$m * \ddot{\vec{X}} = -k * \vec{X} - h * \dot{\vec{X}} \qquad (8)$$

thus,

$$\frac{dE}{dt} = k * \vec{X}^t * \dot{\vec{X}} + \dot{\vec{X}} * (-k * \vec{X} - \xi * \dot{\vec{X}}) \qquad (9)$$

finally,

$$\frac{dE}{dt} = -h * \dot{\vec{X}}^t . \dot{\vec{X}} \qquad (10)$$

With h the coefficient of shock absorber, h is a positive number. So, if we take the limit,

$$\lim_{t \to \infty} E = 0 \qquad E > 0 \qquad (11)$$

Consequently,

[1] The choice of a command law takes into account the characteristics of a test vehicle used in our laboratory

$$\lim_{t\to\infty} \vec{X} = \vec{0} \qquad \lim_{t\to\infty} \dot{\vec{X}} = \vec{0} \qquad (12)$$

Thus, the derivative of energy is always negative (10) because energy will decrease until becoming null and energy E are the sum of two positive number (11). Consequently, ad infinitum, we will have the kinetic energy reaching zero like the potential energy. Thus relative speed between each couple of agent will become null (12) because of the kinetic energy. And the distance between vehicles will be constant since the potential energy is zero and X becomes null. In consequence agents will be followed at same speed and distance between agents will converge to the non-stretched spring length.

3.4 Parameters of the Model

This model uses five variables: mass *m*, coefficient of viscous friction λ, stiffness *k*, damping *h* and non-stretched spring length l_0. The mass of agent vehicle is set by the mass of real vehicle. However, other parameters must be set too. In order to find the values of parameters or a boundary, we used the law expressed in (4). We can deduce a differential equation of second degree.

$$m\ddot{\vec{X}}(t) = -h\dot{\vec{X}}(t) - \lambda\dot{\vec{X}}(t) - k\vec{X}(t) + kl_0 - h\dot{\vec{X}} \qquad (13)$$

This equation can be solved by putting it on a point form. And we can discuss from the level damping.

$$\Delta = \omega_0^2 * (\varepsilon^2 - 1) \qquad (14)$$

with

$$\varepsilon = \frac{h+\lambda}{m*2*\omega_0} \qquad \omega_0 = \sqrt{\frac{k}{m}} \qquad (15)$$

In order to have an absorbed system, pseudo periodical damping must be negative ($\Delta < 0$) thus ε must be under value of one ($\varepsilon < 1$). So we can deduce an over estimate of variables.

$$\frac{h+\lambda}{2+m*\sqrt{m*k}} < 1 \qquad (16)$$

The last parameter is non-stretched spring length l_0, its value is the safety distance. This distance depends of the safety stop. According to the schedule equation, we obtain:

$$l_0 = \frac{-V_0^2}{2*\gamma} \qquad (17)$$

4 Vehicle Merging and Splitting

The model presented in this article include merge and split capabilities. It has been implemented on the base of a MAS architecture. Naturally, vehicle agents include a set of attributes. Attribute values can be changed as a consequence of inter-agent communication, i.e. by a message from another agent vehicle. When a vehicle receives message, it forwards this message to the preceding or following vehicle and refreshes its own attributes value. Particularly, this mechanism is used to increment the mass weighting attribute value in order to increase the spring force during the merge or split phase.

4.1 Merging

The Merge phase is described by the statechart presented in figure 3. The merging phase agrees results in a new vehicle being added at the end of train. The merging vehicle is initially parked and waits for the train. When the merge vehicle detects the last platoon vehicle, it follows it by applying the impedance control model. If the merging vehicle is close enough to the last train vehicle, it sends a message to the preceding vehicle with new weighting and index.

4.2 Splitting

Split phase is described by the statechart of figure 4. Any vehicle can split from train. When a vehicle wants to split, it waits for car parking. When vehicle is close enough to the car parking, it sends new weighting and index to the following and preceding vehicle. Then, it splits from the vehicle train.

5 Experimentations

Experiments have been made with the same parameter values in order to draw a parallel between computer simulation and robot application.

Table 1: Parameter values.

m	λ	k	h	l_0
500 kg	200 Nm^2kg^{-2}	400 Nm^{-1}	10 kgs^{-1}	50 mm

These parameters have been chosen in order to respect (16) and (17).

5.1 Computer Simulation with Madkit Platform

The model described in the previous section has been implemented thanks to the multi-agent platform[2] proposed by Ferber and Guknecht [14]. Computer simulation is used to validate some model characteristics. The simulation runs with a platoon of 4 following vehicles. The first vehicle follows a preset trajectory: a square with rounded angles.

Regular trajectory error: the simulation of home stretch has shown that the error in following is below the millimetre. Figure 6 illustrates a case of rotational motion, showing the leader and follower trajectories. It shows an increase in trajectory error from any vehicle to its follower. This is due to the impedance control model, in which each vehicle anticipates the position of the following one. It can be completed in

[2] Madkit5, http://www.madkit.org

order to deal with this problem. This error can be scaled up towards real vehicle length: 1.2 m compared to the 2 m length of a vehicle.

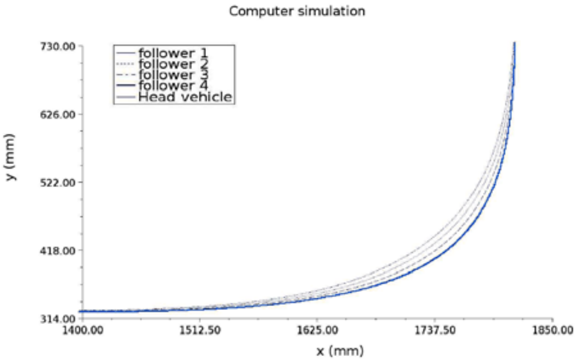


Figure 6: Computer simulation.

Obstacle avoidance: figure 7 illustrates a case of obstacle avoidance. The leading platoon vehicle avoids an obstacle on the road: we can see that all following vehicles also avoid this obstacle and preserve the platoon structure.

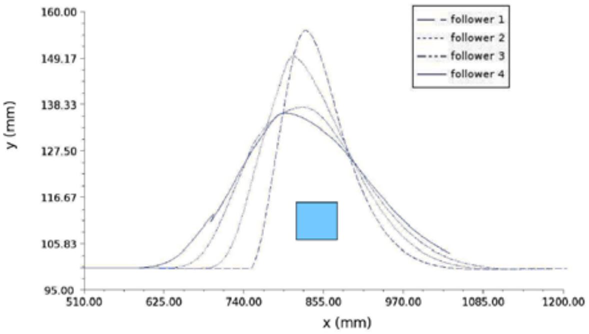


Figure 7: Obstacle avoidance.

Vehicle merging: the merging process was simulated in order to visualise the duration of transitory phase of distance stabilisation, as illustrated by figure 8.

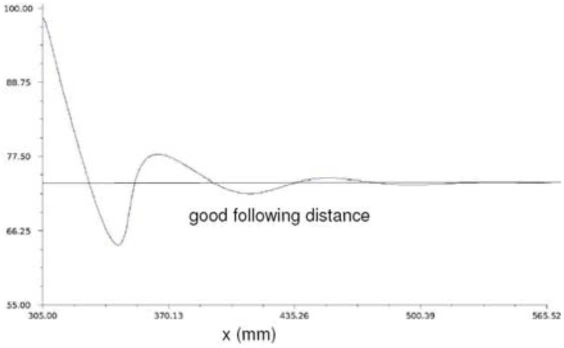


Figure 8: Safety length.

The graph shows the expected pseudo-periodical damping. We can see that the damping time is below 7 seconds and the maximal deviation from the safety length is 10 % of the non stretched spring length.

5.2 Real Experimentation with the Robot Soccer Platform

Experiments have been made to test the controller model under real-world conditions. These real experiments were done with a robot-soccer platform. As for the real vehicle, some small 2-wheel drive Mirosot6[3] soccer robots have been used (figure 9). These robots move on a playground, the size of which is 2.20 m by 1.80 m. Robots are controlled by a standard PC computer that sends data to each robot through a RF interface. The perception is performed by a CCD camera placed above the playground (cf. figure 7). A standard PC computer has been used to execute the SMA software.

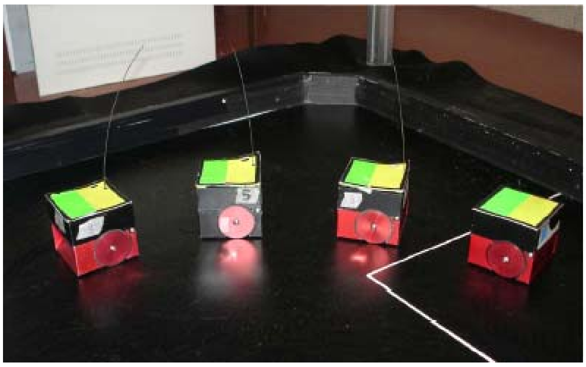


Figure 9: Robot soccer platform.

This experiment allows us to check the model flexibility and adaptability. For instance, we can over estimate the errors by half a vehicle length (i.e. 35 mm). A platoon system with two vehicles was simulated. As in the previous experiments, the first vehicle follows a preset trajectory. Figure 10 shows platoon motion at high speed (more than 50 km/h, if scaled up to real vehicle size) and the effect of centrifugal force on the following vehicle, in the curved sections of the trajectory.

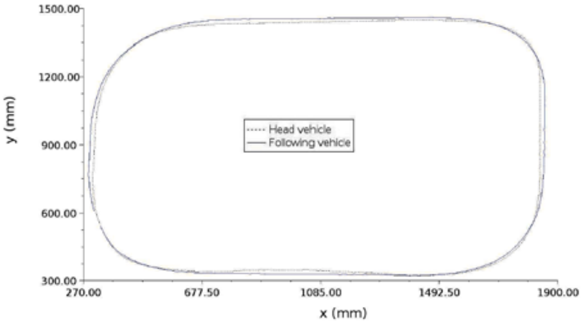


Figure 10: Real simulation.

A platoon system with two following vehicles was simulated. The errors can be scaled up from robot size to real vehicle size. The resulting value is 2.5 m compare to the 2 m length of the vehicle.

[3] http://www.merlinrobotics.co.uk/merlinrobotics/

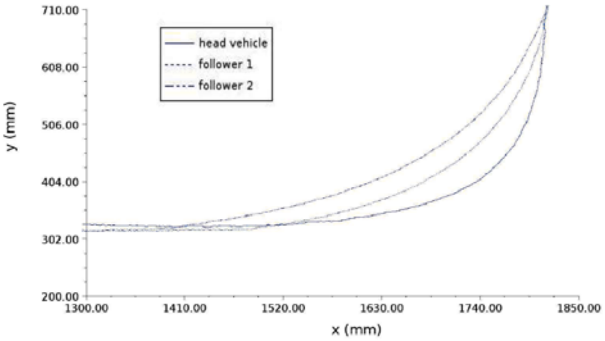


Figure 11: Real simulation.

6 Conclusion

The model proposed in this paper consists in a MAS-oriented approach to vehicle platoons, where vehicles agents interact using laws inspired by physics. This model is based on simple agents with neither cognitive abilities nor representations of the collective goal. One of the interesting points is the emergent structure obtained. The use of physics inspired forces enables an easier tuning of the behavioural parameters and the demonstration of stability. These experiments allowed us to demonstrate the essential characteristics of this kind of resolution method: *Flexibility/Adaptability* (clinging and avoidance capabilities) and *Reliability* (low error rate in curve and damping time). With the real simulation, we note that the system admits the speed constraints and curve radius of town roads. We are now working on the use of formal models in order to prove some application properties and to ensure a zero default embedded software design for real prototypes. Moreover, we are also advancing into further research on other interaction models.

7 References

[1] J.K. Hedrick, M. Tomizuka and P. Varaiya, “Control issues in automated highway systems”, *IEEE Control Systems Magazine*, 14(6), pp 21-32 (1994).

[2] S. Sheikholeslam and C.A. Desoer, “Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study”, *IEEE Transactions on Vehicular Technology*, 42(4), pp 546-554 (1993).

[3] P. Ioannou and Z. Xu, “Throttle and brake control systems for automatic vehicle following”, *IVHS Journal*, 1(4), p 345 (1994).

[4] J. Moskwa and J.K. Hedrick, “Nonlinear algorithms for automotive engine control”, *IEEE Control Systems Magazine*, 10(3), pp 88-93 (1990).

[5] H. Lee and M. Tomizuka, “Adaptive vehicle traction force control for intelligent vehicle highway systems (ivhss)”, *IEEE Transactions on Industrial Electronics*, 50(1), pp 37-47 (2003).

[6] M. Lee, M. Jung and J. Kim, “Evolutionary programming-based fuzzy logic path planner and follower for mobile robots”, *Proceedings IEEE Conference on Evolutionary Computation (ICEC)*, 1, pp 139-144 (2000).

[7] P. Daviet and M. Parent, “Longitudinal and lateral servoing of vehicles in a platoon”, *Proceedings IEEE Intelligent Vehicles Symposium*, Tokyo, pp 41-46 (1996).

[8] S.K. Gehrig and F.J. Stein, “Elastic bands to enhance vehicle following”, *Proceedings IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oakland, CA, pp 597-602 (2001).

[9] S. Yi and K. Chong, “Impedance control for a vehicle platoon system”, *Mechatronics*, 15(5), pp 627-38 (2005).

[10] S. Halle and B. Chaib-draa, “A collaborative driving system based on multiagent modelling and simulations”, *Transp. Res. C, Emerg. Technol.*, 13(4), pp 320-45 (2005).

[11] H.V.D. Parunak, “‘Go to the ant’: engineering principles from natural multi-agent systems”, *Annals of Operations Research*, 75, p 69 (1997).

[12] F. Gechter, V. Chevrier and F. Charpillet, “Localizing and tracking targets with a reactive multi-agent system”, *Proceedings 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, New York, 3, pp 1490-1491 (2004).

[13] P. Gruer, V. Hilaire, A. Koukam and P. Rovarini, “Heterogeneous formal specification based on object-z and statecharts: semantics and verification”, *Journal of Systems and Software*, 70(1-2), pp 95-105 (2004).

[14] O. Gutknecht and J. Ferber, “Madkit: a generic multi-agent platform”, *Proceedings International Conference on Autonomous Agents*, Barcelona, pp 78-79 (2000).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

The Effect of Potential Field Sharing in Multi-Agent Systems

J.L. Baxter[1], E.K. Burke[1], J.M. Garibaldi [1], M. Norman[2]
[1]Automated Scheduling, Optimisation and Planning Group,
School of Computer Science and Information Technology,
University of Nottingham, Nottingham, England.
[2]Merlin Systems Corp Ltd,
ITTC Tamar Science Park, Plymouth, England.
jlb@cs.nott.ac.uk, ekb@cs.nott.ac.uk, jmg@cs.nott.ac.uk, marknorman@merlinsystemscorp.co.uk

Abstract

This paper describes two implementations of a ‘potential field sharing’ multi-agent system which we term as ‘pessimistic’ and ‘optimistic’. Unlike other multi-agent systems in which coordination is designed explicitly, it is an emergent property of our system. The agents perform no reasoning and are purely reactive in nature. The motivation behind this work is to develop a conceptually simple system that can be used in a number of applications. Experiments were conducted in the simulated search and rescue problem, although it is believed that the system is applicable to other common problems such as formation control, coverage and hunting. Statistical analysis showed that systems of six to eight agents that shared potential fields outperformed the equivalent non-sharing system. We conclude that potential field sharing has a positive impact on agents involved in a search and rescue problem.

Keywords: potential fields, multi-robot systems, multi-agent collaboration systems, autonomous systems, search and rescue

1 Introduction

Potential fields [1] have been used in robot navigation [2] for a number of years, despite a number of well-known problems [3]. Examples of problems include oscillation near obstacles and in narrow passages. Modifications to the potential field algorithm have been proposed [4] to overcome these problems. Other approaches to potential fields include that of Reif et al. [5] in which an individual agent’s motion is a result of an artificial force imposed by other agents and components of the system, and that of Damas et al. [6] in which the potential fields are modified to enhance the relevance of obstacles in the direction of the agent’s motion. Howard et al. [7] divide their potential field into two components, a field due to obstacles and a field due to other agents. Pathak et al. [2] stabilise their agent within a surrounding circular area (‘bubble’) using two potential field controllers, the agent is centred within a bubble and then its orientation is corrected.

In our approach, an agent’s motion is a result of the force imposed by obstacles. In addition, a ‘local group’ of agents share information on common potential field regions so that an agent’s motion can be a result of obstacles not perceived by the individual agent. The concept of a local group is similar to that of dynamic robot networks [8]. However, instead of broadcasting trajectories and plans of agents, in our system the potential field information is broadcast. To the best of our knowledge, no previous research using potential fields has incorporated the concept of sharing potential field information amongst agents.

Unlike perception-reasoning-execution architectures [9] the system presented in this paper does not reason about its environment. We describe our system as a reactive system [10]. Motion is based purely on the potential field created by the sensor data in real-time. There is no concept of teamwork [11] or role selection [12] in that each agent performs actions as an individual. Agents are not aware that they are part of a collective; coordination becomes an emergent property of the system through the implicit sharing of potential fields by the agents.

In this paper, we use potential fields in the search and rescue problem. However, we believe our system may be applied to many other problems such as robotic soccer [6] [13], the coverage problem [7] and hunting [14].

2 Potential Field Implementation

Before a description of the ‘sharing’ agents is given, the ‘individual’ agent system will be summarised, as the sharing agents are based upon this individual agent system and they are compared against it during the experimentation in section 3.

The individual agent system, as the name suggests, is made up of a number of agents who attempt to solve a task individually without any communication or coordination with one another. The system is implemented using potential fields modelled on simple electromagnetic theory (as described in [15]).

The system is composed of positively charged particles, which are used to calculate the field by the inverse square law (Coulomb's Law) below (1):

$$F = \frac{q_1 q_2}{r^2} \qquad (1)$$

where q_1 and q_2 are the charges of two particles and r is the distance between them. The resultant force, F, either repels or attracts the particles to one another. Using (1), we calculate eight individual forces F_1-F_8; r is the range to an object (in metres) obtained from an ultra-sonic sensor reading, q_1 is the charge of the agent and q_2 is the charge of the object. For simplicity, all objects are represented by a positive unit charge. Rather than formally resolving into a single force, the agent's motor control is a simple action selection of either move forward or rotate. When moving forward the speed of the agent is relative to the force acting upon it (2), when rotating the angular speed is 0.5 r/s and the forward speed is 0.025 m/s. The agent calculates the minimum F (F_{min}) and rotates in the direction of F_{min}. If the direction of the agent matches the direction of F_{min} then the agent moves forwards.

$$S = 1 - \frac{F}{10} \qquad (2)$$

Using this conceptually simple algorithm the agent moves away from areas of positive charge (obstacles). As a target is indistinguishable to an ultra-sonic sensor a camera is used to distinguish between obstacles and targets using blob detection (the camera tracks for the colour blue, the target). However, rather than giving the target a negative unit charge (-1) the task is said to be complete once the target has been found (this is a simplification due to the nature of the task described in section 3). The orientation of the camera is fixed to the forward orientation of the agent.

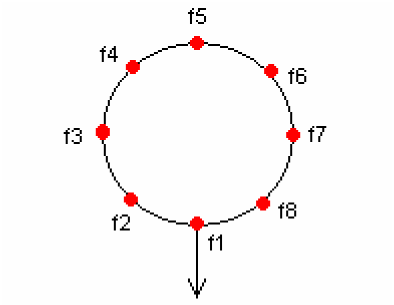


Figure 1: Model of agent. Dots represent position of ultra-sonic sensor. The straight-line arrow indicates the forward orientation/motion of the agent.

2.1 Potential Field Sharing

As noted above the sharing agents are based upon the individual agent system. In fact, they are identical apart from the implicit sharing of potential field information. Agents share their potential field information with other agents within a local group. Therefore, by knowing only the relative positions (based upon odometric readings and the initial location of all agents) of other agents in the system, agents can assign themselves to a local group. Simple geometrical calculations (the intersection of lines/circles), are used to see which agents (if any) are in the local group. The world is represented as a two dimensional plane. If any of the lines (representing the direction of ultra-sonic sensors) from any agent intersects a circle (representing the local radius) of another agent then the potential field information of the involved agents are shared. An example is given below in figure 2.

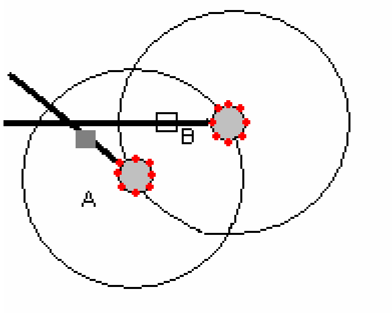


Figure 2: Two agents. The large circles represent the range of the agent's sensors and thus the grey square is an obstacle only observed by agent A. The white square represents what agent B would "see" if the pessimistic system was implemented.

We describe two types of sharing systems. They are referred to as 'pessimistic' and 'optimistic' in the context of sensor noise. Consider, for example, the situation as in figure 2 in which Agent A detects an obstacle that is not within Agent B's sensor range. As they both belong to the same local group (within 2m of each other), a shared potential field is constructed. Agent A would suggest a high charge, whereas Agent B would suggest a low charge. In the pessimistic system, the highest charge is selected and so Agent B is repulsed by an obstacle outside its own sensor range. In the optimistic system, the lowest charge is selected and thus Agent A is no longer repulsed by the obstacle initially detected by its sensors. The advantage of the pessimistic system is that agents are less vulnerable to false negatives (and so avoid obstacles that they have not detected due to sensor noise), but the disadvantage is that they are more susceptible to false positives (and so avoid obstacles they do not need to). The advantage of the optimistic system is that agents are less vulnerable to false positives but are more susceptible to false negatives.

3 Experiments

All experiments were conducted using the player/stage [16] simulator. Real world experimentation is one of our key objectives, and as such, the robots modelled in the simulator are those that we intend to use in experiments in the future. The robots simulated for the experiments were Miabot

Table 1: Mean completion time (seconds) for each multi-agent system in each world, for two to eight agents (to 1 d.p.).

	Number of Agents						
	2	3	4	5	6	7	8
Individual							
world 1	300.0	250.7	255.9	225.1	205.2	273.0	288.2
world 2	153.8	129.6	70.1	31.7	32.4	33.2	29.9
world 3	52.2	91.2	69.6	73.6	62.7	112.7	134.4
Pessimistic							
world 1	203.4	211.1	172.6	133.9	121.6	124.3	120.6
world 2	77.2	57.9	43.2	49.9	8.0	4.3	11.9
world 3	115.5	141.8	67.9	78.8	64.7	133.5	95.9
Optimistic							
world 1	262.5	198.8	150.2	109.4	98.0	110.2	116.0
world 2	132.1	42.6	31.7	26.3	11.4	4.3	9.5
world 3	115.5	128.5	70.5	105.7	66.7	106.6	110.0

Pros [17], with an ultra-sonic range finder and avr-cam modules. The ultra-sonic range finder is composed of eight ultra-sonic sensors that give a 360° field of view with a range of 3cm - 1m. The avr-cam has a field of view of 30° and can track up to eight blobs at the same time. The environment simulated has approximately the same dimensions (5m x 3m) as our real robot arena. As an initial test-bed, the problem of search and rescue was chosen. It is defined as follows; Agents have the task of navigating an unknown environment in order to find a target (once the target is found it is presumed rescued).

The environment consists of obstacles, agents, a deployment zone and a target. Obstacles were generated within the environment (world) randomly; the size of the obstacle, its position and its orientation all varied. The positions of the target and the deployment zone for the agents were also generated randomly. However, these positions were fixed throughout the experiments. The only difference between each experimental run is the noise within the simulation in the form of ‘bad’ data e.g. sonar passing through obstacles. The deployment zone is a position in the world that all the agents start in (evenly spaced 0.1m from one another in concentric circles starting from this initial position).

For these experiments, three worlds were generated. For each world, all three of the systems (individual, pessimistic and optimistic) attempted twenty runs that were repeated for groups of two to eight agents. The time recorded for the first agent to find the target was the metric recorded. Failure to complete the task within three hundred seconds resulted in a score of three hundred seconds. The means of the results are provided in table 1. Two statistical tests were chosen to analyse the data. The Kruskal-Wallis test was chosen, as it is useful in detecting a difference in the medians of distributions. The Friedman test was chosen to detect the existence of association between characteristics of a population.

3.1 Kruskal-Wallis Rank Sum Test

To investigate quantitatively which system performs better we used the Kruskal-Wallis rank sum test. This involves ranking all of the times provided from each system; the mean of the sum of ranks for each (see table 2) was taken and the significant differences noted. This was repeated for all three worlds and all seven sizes of groups. The null hypothesis (h_0) and alternative hypothesis (h_1) were as follows:

h_0: The medians of the k populations do not differ.
h_1: At least two of the medians differ.

As detailed in [18], the Kruskal-Wallis H test statistic was produced whilst correcting for ranking ties. If the corrected H value was greater than the selected P-value (0.1), h_0 could be rejected in favour of our h_1. If the differences between the means of ranks (table 4) did not satisfy (3) (where k = 3, N = 60 and z = 2.8 (to 1 d.p.)) and was negative this meant that the first group had a significantly smaller completion time than the second group. However, if it was positive, it can be concluded that the second group had a significantly smaller completion time. This is because a larger mean difference relates to a longer completion time.

$$\left|\overline{R}_i - \overline{R}_j\right| \leq z\sqrt{\frac{k(N+1)}{6}} \tag{3}$$

3.2 Friedman Rank Sum Test

To investigate what effect the number of agents had on performance, the Friedman rank sum test was conducted. This involves ranking all of the times

Table 2: Mean rank sum for each multi-agent system in each world, for two to eight agents (to 1 d.p.).

	Number of Agents						
	2	3	4	5	6	7	8
Individual							
world 1	38.0	35.6	41.0	38.8	41.1	44.8	46.3
world 2	22.5	42.1	35.2	32.2	45.7	50.5	47.0
world 3	12.0	19.5	27.3	23.1	27.0	28.5	32.2
Pessimistic							
world 1	22.5	29.2	27.8	27.0	26.9	24.4	24.1
world 2	39.1	26.5	29.2	32.5	22.1	20.0	24.3
world 3	39.9	37.9	30.1	32.1	32.1	30.5	27.9
Optimistic							
world 1	31.0	26.8	22.7	25.7	23.5	22.3	21.1
world 2	29.9	22.9	27.1	26.8	23.7	21.0	20.3
world 3	39.6	34.1	34.1	36.4	32.5	32.5	31.5

Table 3: Column sum of ranks for each multi-agent system in each world, for two to eight agents (to 1 d.p.).

	Number of Agents						
	2	3	4	5	6	7	8
Individual							
world 1	97.5	81.0	77.0	68.5	64.5	82.0	89.5
world 2	121.0	118.0	79.0	62.5	53.0	59.5	67.0
world 3	46.5	66.0	92.5	81.0	79.0	101.5	93.5
Pessimistic							
world 1	101.5	100.0	89.5	75.5	60.0	68.0	65.5
world 2	109.0	117.0	102.5	93.5	57.0	26.0	55.0
world 3	91.5	110.0	74.0	74.0	64.5	81.5	64.5
Optimistic							
world 1	120.0	101.0	81.5	72.5	58.5	64.0	62.5
world 2	120.0	114.5	102.0	85.5	59.0	31.0	48.0
world 3	85.5	89.5	76.5	82.5	65.0	82.5	78.5

provided from each system from all group sizes. The sum of ranks for each (see table 3) was taken and the significant differences noted. This was repeated for all three worlds. The null hypothesis (h_0) and alternative hypothesis (h_1) were set as follows:

h_0: The number of agents of the k populations has no effect.

h_1: The number of agents has an effect.

As detailed in [18], the Friedman Q test statistic was calculated. If the Q value was greater than the selected P-value (0.1) the h_0 could be rejected in favour of our h_1. If the difference between column sums (tables 5-7) did not satisfy (4) (where k = 20, n = 7 and z = 2.8 (to 1 d.p.)) and was negative this meant that the first group had a significantly larger completion time than the second group. If positive, it can be concluded that the first group had a significantly smaller completion time.

$$\left|R_i - R_j\right| \le z\sqrt{\frac{kn(n+1)}{6}} \tag{4}$$

4 Results

Table 4 shows the significant differences between the means of ranks for the, individual system (R1), pessimistic system (R2) and the optimistic system (R3). Rows with no significant differences have been omitted for brevity. As only values above, 11.8 (to 1 d.p.) are significant it can be seen that both the pessimistic and optimistic systems perform better than the individual system, in groups of six or more in worlds one and two. For groups of less than six the results are mixed. The individual system performs significantly better in world 3 for group sizes of up to five beyond that the differences become insignificant. It can also be observed that there was no significant difference between the pessimistic and optimistic systems. Table 5 shows significant differences between the pairs of column sums for the individual system. As only values above 38.6 (to 1 d.p.) are significant, it can be seen that in world 2 the individual system performs better with four or more agents. However, for world 3 the individual system performs better with two agents only. It can also be observed that there is no significant impact of group size in world 1. Table 6 shows the same data for the pessimistic system. It appears to perform better with six or more agents in all three worlds (which is in line with the results from table 4). Table 7 again shows the same data for the optimistic system, which performs better in worlds 1 and 2 with four or more agents. There were no significant differences in world 3.

Table 4: Significant differences between the individual (R1), pessimistic (R2) and optimistic (R3) systems (to 1 d.p.).

	Differences	
Two Agents	$\overline{R}_1 - \overline{R}_2$	$\overline{R}_1 - \overline{R}_3$
world 1	15.5	7.0
world 2	-16.6	-7.4
world 3	-27.9	-27.7
Three Agents		
world 2	15.7	19.2
world 3	-18.4	-14.7
Four Agents		
world 1	13.2	18.3
Five Agents		
world 1	11.7	13.1
world 3	-9.0	-13.3
Six Agents		
world 1	14.3	17.6
world 2	23.6	22.0
Seven Agents		
world 1	20.4	22.6
world 2	30.4	29.5
Eight Agents		
world 1	22.2	25.2
world 2	22.6	26.7

Table 5: Significant differences between the numbers of agents, for the individual system. (to 1 d.p.)

	Number of Agents				
	4	5	6	7	8
world 2					
2 agents	42.0	58.5	68.0	61.5	54.0
3 agents	39.0	55.5	65.0	58.5	51.0
world 3					
2 agents	-46.0	-34.5	-32.5	-55.0	-47.0

Table 6: Significant differences between the numbers of agents, for the pessimistic system. (to 1 d.p).

	Number of Agents		
	6	7	8
world 1			
2 agents	41.5	33.5	36.0
3 agents	40.0	32.0	34.5
world 2			
2 agents	52.0	83.0	54.0
3 agents	60.0	91.0	62.0
4 agents	45.5	76.5	47.5
5 agents	36.5	67.5	38.5
world 3			
3 agents	45.5	28.5	45.5

Table 7: Significant differences between the numbers of agents, for the optimistic system. (to 1 d.p.).

	Number of Agents				
	4	5	6	7	8
world 1					
2 agents	38.5	47.5	61.5	56.0	57.5
3 agents	19.5	28.5	42.5	37.0	38.5
world 2					
2 agents	18.0	34.5	61.0	89.0	72.0
3 agents	12.5	29.0	55.5	83.5	66.5
4 agents	N/A	16.5	43.0	71.0	54.0

5 Discussion

From the results in the previous section, it can be observed that both the pessimistic and optimistic systems perform better than the individual system (when in groups of 6 or more). It can also be observed that both the pessimistic and optimistic systems perform better in larger group sizes (6 or more and 4 or more respectively). These results make sense as the presence of more agents implies a higher probability that the agents can take advantage of potential field sharing, instead of reverting to the individual behaviour. The second observation is interesting as it shows that a system less inclined to avoid obstacles performs better with a smaller group size than a system that is inclined to avoid obstacles. This also makes sense as the smaller the group size, the less likely it is that more (previously unseen) obstacles will be discovered. The observation that there is no significant difference between the performance of the pessimistic and optimistic systems is also interesting as this implies that the ability to detect more obstacles has the same benefit as the ability to ignore more sonar noise. This finding was unexpected and requires further investigation.

A major limitation of the pessimistic and optimistic systems is their reliance upon accurate odometric readings. In the experiments carried out in this paper it was assumed that no errors occurred. In the real world errors occur frequently due to wheel slippage. This will have to be accounted for in real world experiments.

Future work includes adapting the multi-agent systems to include group member recognition in order to improve agent dispersal. We also intend to investigate the effect of the local group radius (both increasing and decreasing its size). Other possible future work includes, applying this work to a very large-scale robotic system (hundreds of robots) [5] and implementing the sharing agents in other common problems such as robotic soccer [6] [13], formation control [19], the coverage problem [7] and hunting [14]. It is hoped the work in this paper can form a basis for future work in the real world.

6 References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *Proc. 1985 IEEE Int. Conf. on Robotics and Automation*, St. Louis, Missouri, pp 500-505 (1985).

[2] K. Pathak and S.K. Agrawal, "An integrated path planning and control approach for nonholonomic unicycles using switched local potentials", *IEEE Transactions on Robotics*, 21(6), pp 1201-1208 (2005).

[3] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation", *Proc. IEEE Conf. Robotics and Automation*, pp 1398-1404 (1991).

[4] J. Ren, K.A. McIssac and R.V. Patel, "Modified Newton's method applied to potential field-based navigation for mobile robots", *IEEE Transactions on Robotics*, 22(2), pp 384-391 (2006).

[5] J. Reif and H. Wang, "Social potential fields: a distributed behavioral control for autonomous robots", in J.C. Latombe, R. Wilson K. Goldberg and D. Halperin (eds), *Algorithmic Foundations of Robotics*, A.K. Peters, Wellesley, MA, pp 431-459 (1995).

[6] B.D. Damas, P.U. Lima and L.M. Custdio, "A modified potential fields method for robot navigation applied to dribbling in robotic soccer", in R.R. Gal, A. Kaminka and P.U. Lima (eds), *RoboCup 2002: Robot Soccer World Cup VI*, Springer Verlag, Berlin, pp 65 - 77 (2003).

[7] A. Howard, M.J. Mataric and G.S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem", *Proceedings 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, Fukuoka, Japan, pp 229-308 (2002).

[8] C.M. Clark, S.M. Rock, and J.C. Latombe, "Motion planning for multiple mobile robot systems using dynamic networks", *Proc. IEEE Int. Conference on Robotics and Automation*, 3, pp 4222 -4227 (2003).

[9] X.W.T. Liu and J. Baltes, "An intuitive and flexible architecture for intelligent mobile robots", *Proc. 2nd International Conference on Autonomous Robots and Agents (ICARA)*, Palmerston North, New Zealand, pp 52-57 (2004).

[10] R. Arkin, "Reactive robotic systems", in M. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, MA, pp 793-796 (1995).

[11] P.R. Cohen, H.J. Levesque, I.A. Smith, "On team formation", in J. Hintikka and R. Tuomela (eds), *Contemporary Action Theory 2: Social Action*, Kluwer, Dordrecht, pp 87-114. (1997).

[12] H.L. Sng, G. Sen Gupta and C.H. Messom, "Strategy for collaboration in robot soccer", *Proc. 1st IEEE International Workshop on Electronic Design, Test and Applications (DELTA '02)*, Christchurch, New Zealand, pp 347-351 (2002).

[13] D.H. Kim, Y.J. Kim, K.C. Kim, J.H. Kim and P. Vadakkepat, "Vector field based path planning and petri-net based role selection mechanism with q-learning for soccer robots", *Int. J. Intelligent Automation and Soft Computing*, 6(1), pp 75- 87 (2000).

[14] Z. Cao, M. Tan, L. Li, N. Gu and S. Wang, "Cooperative hunting by distributed mobile robots based on local interaction", *IEEE Transactions on Robotics*, 22(2), 403-407 (2006).

[15] B.H. Chirgwin, C. Plumpton and C.W. Kilmister, *Elementary Electromagnetic Theory, Vol. 1*, Pergamon Press, New York (1971).

[16] B. Gerkey, R.T. Vaughan and A. Howard, "The player/stage project: tools for multi-robot and distributed sensor systems", *Proc. 11th International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, pp 317-323, (2003).

[17] Merlin Systems Corporation, "Merlin Robotics", http://www.merlinrobotics.co.uk, visited on 10/07/06.

[18] J. D. Gibbons, *Nonparametric Methods for Quantitative Analysis* (3rd edition), American Sciences Press, Columbus, OH (1997).

[19] S. Monterio and E. Bicho, "A dynamical systems approach to behavior-based formation control", *Proc. Intl. Conf. Robotics Automation*, Washington, DC, pp 2606-2611 (2002).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Line-Crawling Bots That Inspect Electric Power Transmission Line Equipment

James F. Peters[1], Maciej Borkowski[1], Christopher Henry[1], Daniel Lockery[1], David S. Gunderson[2], Sheela Ramanna[3]
[1]Department of Electrical and Computer Engineering, [2]Department of Mathematics
University of Manitoba, Winnipeg, Canada
[3]Department of Applied Computer Science
University of Winnipeg, Winnipeg, Canada
jfpeters@ee.umanitoba.ca, maciey@ee.umanitoba.ca, chenry@ee.umanitoba.ca,
dlockery@ee.umanitoba.ca, gunderso@cc.umanitoba.ca, s.ramanna@uwinnipeg.ca

Abstract

This paper presents an approach to solving the problem of commanding and target tracking for ALiCE II (Autonomous Line Crawling Equipment) bots that cooperate with each other to navigate along a wire stretched between towers and to inspect electric power transmission equipment. ALiCE II is a second generation version of a new family of autonomous line-crawling robotic devices produced to reduce the hazards and cost of performing manual human inspection of hydro equipment. Specifically, this paper gives the details for the current design of the ALiCE II telecommunication subsystem that makes it possible to use the internet to send navigation and vision system commands via email to the bot and receive bot responses such as its location and images captured by bot cameras. The contribution of this article is an approach to commanding and target tracking for remote, autonomous bots used for inspecting electric power transmission line equipment.

Keywords: autonomous, bot, commanding, reinforcement learning, target tracking, wireless

1 Introduction

This paper focuses on the solution to the problems of commanding and target tracking for ALiCE II, a line-crawling bot (LCB) that lives in a society of cooperating bots designed to inspect electric power transmission equipment. Work on this form of robotic device has been reported recently (see [1, 2, 3, 4, 5, 6]). The problem of commanding an LCB is solved in a fairly straightforward fashion by using a wireless communication system that makes it possible to upload commands to a bot and to receive information (e.g. position and images) via email. One of the goals of the LCB is to classify the health of hydro equipment. The problem of aiming the camera at the hydro equipment (also called *target tracking*) by an LCB is more difficult to solve. Both classical as well as reinforcement learning approaches to solving the target tracking problem are reported in this paper.

This paper is organised as follows. A communication system and approach to commanding an LCB are briefly presented in sections 2 and 3. A vision system and target tracking methods for the LCB are presented in section 4.

2 Wireless Communication System

This section briefly presents the main logistical components of the cellular wireless internet communication system (see figure 1). The communication link (denoted by a lightning icon) represents a Point-To-Point Protocol (PPP) connection to the Telus network[1]. The PPP connection protocol is used to establish a direct connection between two nodes (e.g. ALiCE II and a PPP server) [8].

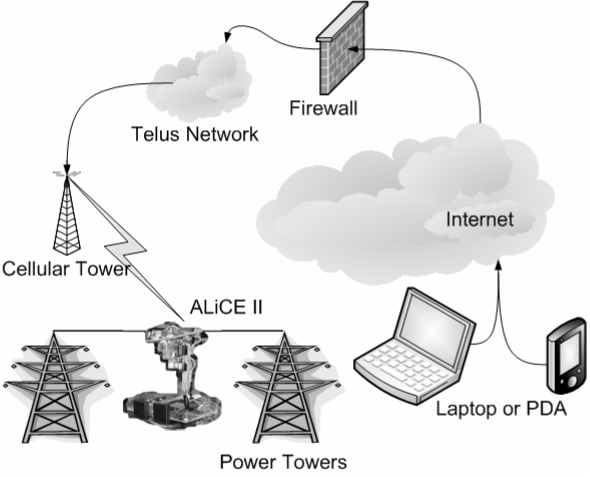


Figure 1: ALiCE II wireless network.

ALiCE II[2] connects to the internet in a manner similar to the traditional modem operating over the telephone lines of a Plain Old Telephone System (POTS). In both cases, a connection is made between the modem

[1] The term "network" described in this paper refers to a Wide Area Network that covers a large geographical area, involves large arrays of computers, and in part (or completely) consists of circuits provided by a common carrier [7].

[2] The main processing unit of ALiCE II is the TS-5500 compact full-featured PC compatible Single Board Computer based on the AMD 0x586 class processor [9].

and a server by establishing a PPP communication link. The only difference between these two cases is the medium in which the connection is made. Specifically, ALiCE II uses the Sierra AirCard 580 Wireless WAN Modem (see figure 2) to establish the connection to the PPP server over the Telus cellular network. PPP has three main components [10], a method for sending data over the connection between two nodes, a Link Control Protocol (LCP) for maintaining the data link, and a suite of Network Control Protocols (NCPs) used to establish and configure different network-layer protocols. Specifically, the NCP protocol used to establish the IP address is the PPP Internet Protocol Control Protocol (IPCP) which negotiates the IP address only after the data link has been established and tested [10].

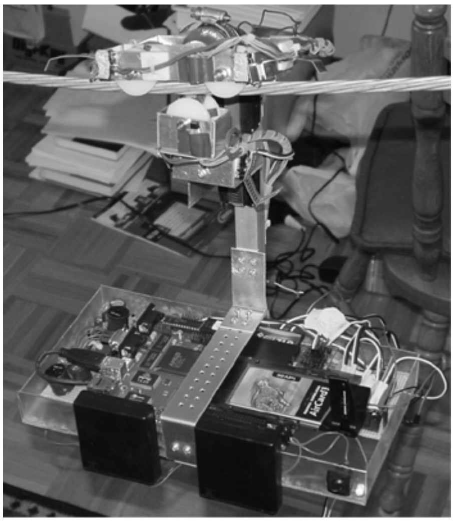


Figure 2: ALiCE II oblique view.

3 Client-Server Architecture

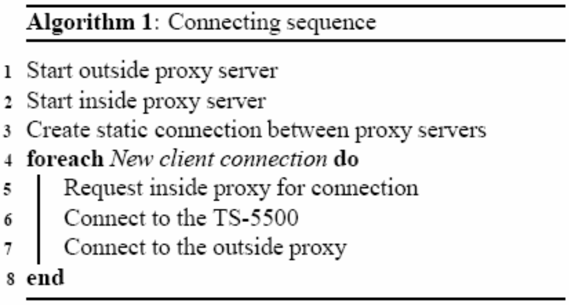
This section gives a brief overview of the client-server architecture used for commanding ALiCE II.

3.1 Connecting Sequence

This section contains a short description of the sequence of steps required to access the TS-5500 from the internet. The summary of this process is given in algorithm 1.

First, the two proxy servers are started (see figure 3). Then the inside server initiates the connection with the outside server. This connection (called a static connection) is used to forward requests to the inside server from outside the firewall. After the static connection is created, the servers are ready and are waiting for incoming connection requests.

Assume that a client wants to open a web page on the TS-5500. The client connects to the outside server using port 80 (see step 4 in algorithm 1). The outside server requests the connection from the inside server using the static connection. The inside server connects to the TS-5500 using port 80 and closes the connection loop with the client by creating the dynamic connection with the outside server.



Algorithm 1: Connecting sequence

```
1 Start outside proxy server
2 Start inside proxy server
3 Create static connection between proxy servers
4 foreach New client connection do
5 |   Request inside proxy for connection
6 |   Connect to the TS-5500
7 |   Connect to the outside proxy
8 end
```

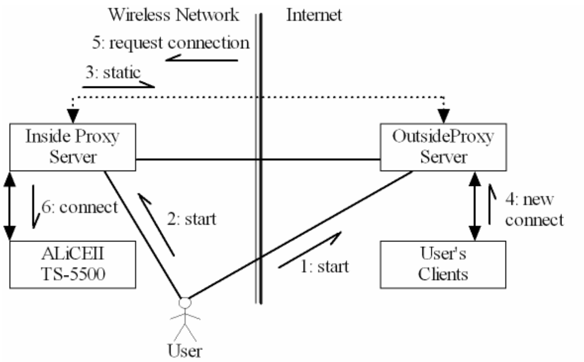


Figure 3: UML model for PPP.

3.2 Web Server

The TS-5500 board is equipped with the Apache [11] web server. It is capable of running PHP [12] scripts. A web based interface was implemented to allow the user to remotely control the Linux [13] operating system running on the TS-5500. By logging into the ALiCE II web page the user can run Linux commands that are available in a terminal window. The commands are executed and the accompanying results are shown on the web page. Finally, several of the most commonly used scripts were implemented with buttons to decrease the time needed to perform service tasks on the TS-5500.

3.3 Email Server

The TS-5500 has ability to receive and send e-mails. A Perl [14] script controls the operations of handling incoming e-mails, execution of task requests, and sending response e-mails containing task results. Similar to the web-based user interface, the user can run any Linux command on the TS-5500 board in the same manner as a terminal window. The command is executed and results are sent back to the user via email. Moreover, an e-mail can contain several commands, which will be executed sequentially. Furthermore, the email server has the additional advantage of being able to incorporate frequently used command sequences into the script processing the emails. For example, an image acquisition sequence has been implemented in the email server. If the Subject Field of an email contains only the words "get image," the TS-5500 captures an image and attaches it to an email with the destination being the originator of the command. See table 1 for a summary of commands recognised by the ALiCE II mailing script.

Table 1: Sample ALiCE II command.

Subject	Action
Linux cmd (e.g. uptime, ls, ps)	Execution of cmd
Get image	Capture image; Send image to originator of email

4 ALiCE II Vision System

This section gives a brief overview of the ALiCE II vision system. Only target-tracking by a monocular vision system using classical as well as reinforcement learning is considered (consideration of the ALiCE II binocular vision system made possible by coupled bots, each with its own camera system, is outside the scope of this paper). The ALiCE II TS-5500 is running the Linux [13] operating system on the Elan 0x586 class processor [9, 15]. The main constraint limiting the possible algorithms which can be used for tracking is the relative low computational power of the CPU, which is run by a 133 MHz clock. As a result, the TS-5500 has computational power comparable to a Pentium III processor with a clock ≈ 70 MHz.

The ALiCE II camera has two degrees of freedom, namely it is mounted on two servos. The movement of each servo can be controlled separately. Pan and tilt camera movements are possible with a combination of servos attached to a digital camera shown in figure 4 (note, the pan servo is obscured by the camera in this figure). A bot suspended from a sky wire stretched between electric power transmission towers usually sways from side-to-side as it moves along a wire. Hence, target tracking is necessary once a bot identifies a target (e.g. insulator or tower cross-brace) in order to move a camera so that it continues to focus on the object of interest. What follows is a comparison of several types of tracking systems using classical as well as reinforcement learning methods.

4.1 Classical Target Tracking

Figure 5 shows the overview of a tracking system using classical image processing methods. The arrows show the flow of the information between the components of the system.

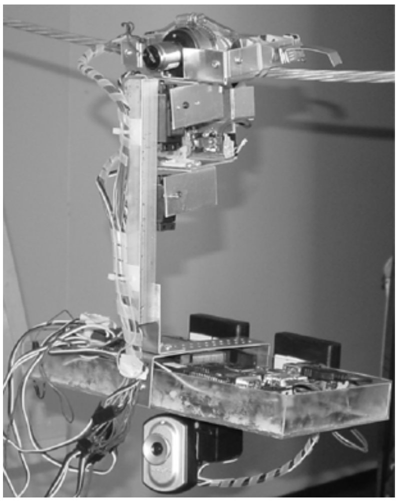


Figure 4: ALiCE II monocular vision system.

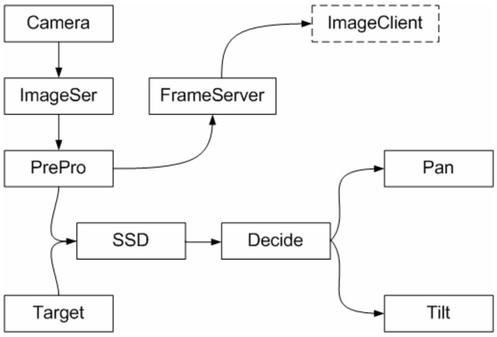


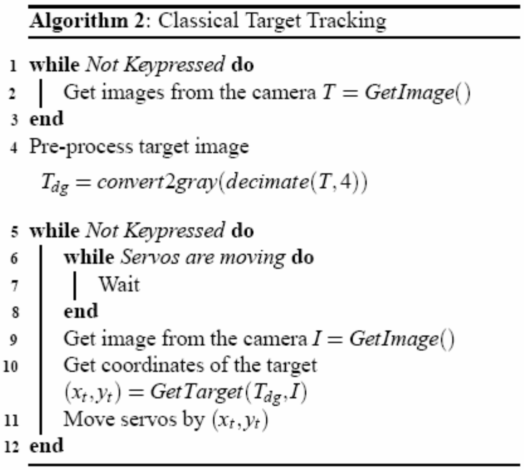
Figure 5: Target tracking system overview.

The *Camera* object in figure 5 represents a Creative WebCam NX Ultra [16], and the drivers. For this project the SPCA50X USB Camera Linux Driver has been used [17]. The *ImageSer(ver)* object in figure 5 is based on the *spcaview* programme included in the spca driver. The purpose of this server is to acquire images from the camera and pass them to the application. The *PrePro(cessing)* object prepares the image for the actual tracking. Due to the low processing power of the TS-5500, the size of the image is reduced (both in the 2D and colour space). The smallest image size available from the camera is 160 × 120 pixels. The most convenient format is RGB (one byte for each colour: Red, Blue and Green). Therefore, the total size of such an image is 160 × 120 × 3 = 57,600 bytes. The quickest method for spatial dimension reduction is decimation. This means that in each dimension (horizontal and vertical) every n^{th} pixel is selected. For this project, the number of pixels is reduced by 4 in each dimension to 40 × 30 pixels. In the colour domain, the RGB colours are transformed into grey levels using

I = (RED + GREEN +BLUE)/3.

This equation is very convenient, since it can be implemented using only integer operations.

The resulting image has the size of 44 × 30 × 1 = 1320 bytes, and is over 40 times smaller than the original image. This image, called the *source image*, is used to find the position of the target. Despite its small size, the source image contains sufficient information about a scene. Algorithm 2 gives the classical target tracking method underlying figure 5, where *GetImage( )* (not shown) takes a pre-processed image *I*, decimates *I* by a factor of 4 to obtain I_d, converts I_d to greyscale to obtain I_{dg}, uses *SSD( )* (not shown) to compute the sum of squared differences $SSD(T_{dg},I_{dg})$ between a target T_{dg} and I_{dg}, finds the minimum $(x_t,y_t) = argmin_{(x,y)}\ ssd(x,y)$, and moves the servos by (x_t,y_t).



```
Algorithm 2: Classical Target Tracking
while Not Keypressed do
    Get images from the camera $T = GetImage()$
end
Pre-process target image
    $T_{dg} = convert2gray(decimate(T,4))$
while Not Keypressed do
    while Servos are moving do
        Wait
    end
    Get image from the camera $I = GetImage()$
    Get coordinates of the target
    $(x_t,y_t) = GetTarget(T_{dg},I)$
    Move servos by $(x_t,y_t)$
end
```

4.2 Reinforcement Learning

To experiment with control of camera movements using reinforcement learning (RL), it is helpful to partition an image into sectors representing states, which can then be correlated with horizontal *h* and vertical *v* movements of the camera. For example, the image I_{dg} is partitioned into the 9 states shown in figure 6(a), where the numbers in each section of figure 6(b) respectively denote incremental horizontal (pan) and vertical (tilt) movements of the camera. Due to the low computational power of the TS-5500 the number of possible actions for any given state was restricted to 12. The twelve actions are step increments ranging from 0 to 11. A selected action is then multiplied by the numbers of a partition in figure 6(b) to determine servo direction. Each step increment to the servos provides a rotation of 0.9 degrees with an accuracy of ±0.25%. Once the actions have been established, it is important to devise a means of rating (i.e. "rewarding") the camera movements.

S_6	S_7	S_8
S_3	S_4	S_5
S_0	S_1	S_2

(a) states

1 -1	0 -1	-1 -1
1 0	0 0	-1 0
1 1	0 1	-1 1

(b) horizontal and vertical movements

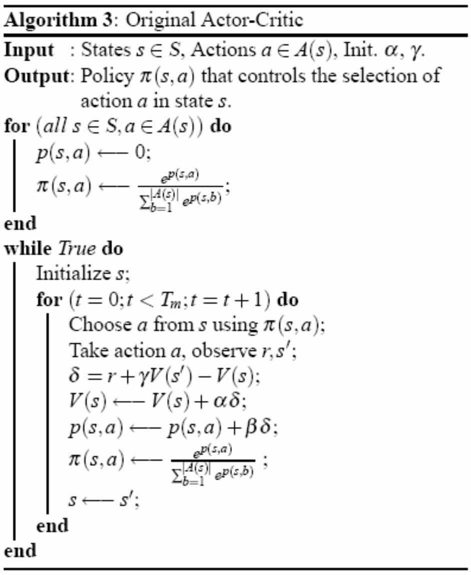
Figure 6: Partition of image.

In this work, rewards are a function of the Euclidean distance measured from the coordinates of the target to the centre of I_{dg}. The reward function is calculated using (1)

$$REWARD = 1 - \frac{DISTANCE}{MAXDISTANCE}. \quad (1)$$

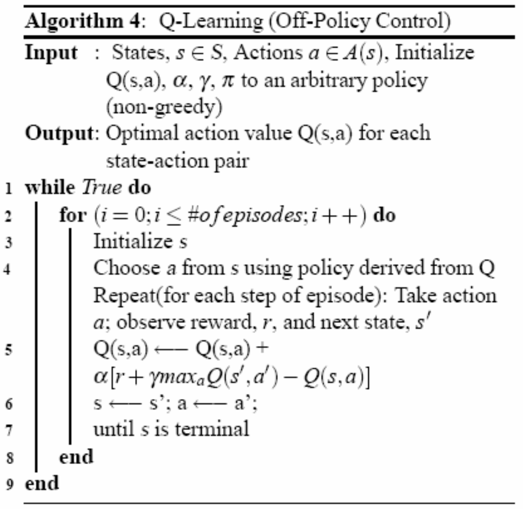
The result of using (1) is a normalised reward function which equals 1 (goal) when the target is located at the origin, and 0 (worst case) when the target is at *MAXDISTANCE* (i.e. at the outside corners of I_{dg}.).

Given the above framework for states, actions and rewards, it is then possible to experiment with RL-based target tracking. We briefly report the results obtained using the Actor-Critic (AC) and Q-Learning methods (see [18]). AC is an on-policy RL algorithm that involves a policy *π(s,a)* called an *actor* and an estimated value function called a *critic* [18]. In algorithm 3, the critic is responsible for critiquing a policy and providing an error measure for guiding the actor towards an optimal policy to achieve a particular goal.



```
Algorithm 3: Original Actor-Critic
Input  : States $s \in S$, Actions $a \in A(s)$, Init. $\alpha$, $\gamma$.
Output: Policy $\pi(s,a)$ that controls the selection of
        action $a$ in state $s$.
for (all $s \in S, a \in A(s)$) do
    $p(s,a) \longleftarrow 0$;
    $\pi(s,a) \longleftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$;
end
while True do
    Initialize $s$;
    for ($t = 0; t < T_m; t = t+1$) do
        Choose $a$ from $s$ using $\pi(s,a)$;
        Take action $a$, observe $r, s'$;
        $\delta = r + \gamma V(s') - V(s)$;
        $V(s) \longleftarrow V(s) + \alpha\delta$;
        $p(s,a) \longleftarrow p(s,a) + \beta\delta$;
        $\pi(s,a) \longleftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$;
        $s \longleftarrow s'$;
    end
end
```

Details about the form of AC in algorithm 3 are given in [19], which also introduces a rough set approach to the AC inspired by [20]. Q-Learning (QL) is an off-policy RL algorithm (see algorithm 4) that learns the state-action value, *Q*, regardless of the policy being followed [18]. The advantage is that a policy is capable of exploring all possible actions to gain experience while a greedy or maximising reward policy dictates the next action value revision. The details concerning Q-learning can be found in its original form in [21] as well as in [18].



```
Algorithm 4: Q-Learning (Off-Policy Control)
Input  : States, $s \in S$, Actions $a \in A(s)$, Initialize
         Q(s,a), $\alpha$, $\gamma$, $\pi$ to an arbitrary policy
         (non-greedy)
Output: Optimal action value Q(s,a) for each
         state-action pair
while True do
    for $(i = 0; i \leq \#ofepisodes; i++)$ do
        Initialize s
        Choose a from s using policy derived from Q
        Repeat(for each step of episode): Take action
        $a$; observe reward, $r$, and next state, $s'$
        Q(s,a) ⟵ Q(s,a) +
        $\alpha[r + \gamma max_a Q(s',a') - Q(s,a)]$
        s ⟵ s'; a ⟵ a';
        until s is terminal
    end
end
```

4.3 Results

Algorithms 2, 3, and 4 have been compared using the setup shown in figure 7, which is a replica of the hardware used in ALiCE II. Note, the distance of the camera from the target is fixed. Two display devices were used, one flat panelled monitor (upper right in figure 7) displays an image of a typical tower with insulators that moves around on the screen to simulate movements of ALiCE II being buffeted by wind while it is inspecting an insulator group. The second is a laptop (lower left in figure 7) that acts as observation equipment and displays the images captured by the ALiCE II camera (the area inside the white box in the display indicates a target that has been identified within the captured image). The camera in figure 7 is connected to a TS-5500 (underneath the white base shown in the centre of figure 7).

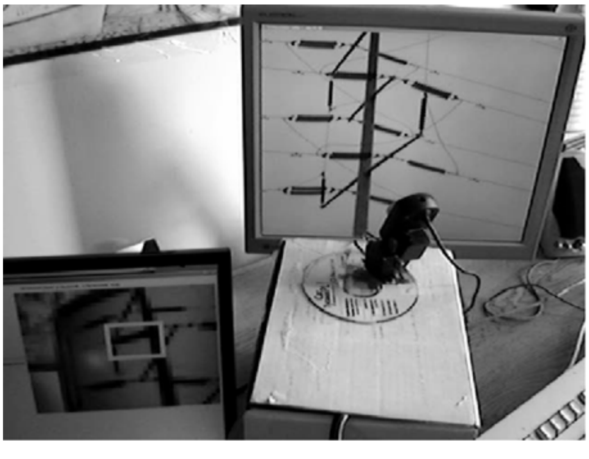


Figure 7: ALiCE II experimental setup.

The metric of an RMS error with regards to pixel distance away from the goal was used to rate the three algorithms. Figure 8 shows the RMS for all three methods. The lighting conditions were all very similar for the above experiments and steps were taken to ensure as much consistency as possible. The length of each experiment was 5 minutes and as can be seen in the results, the number of samples did vary somewhat.

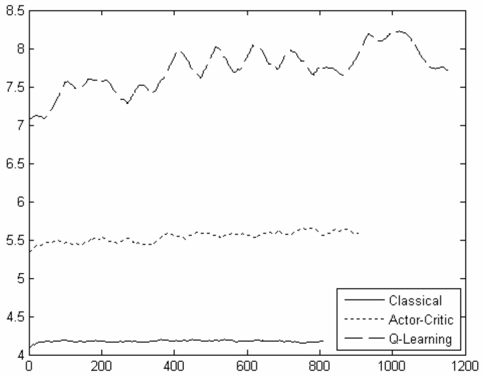


Figure 8: RMS metric for all three methods.

There is a notable benefit with the classical tracking method over the learning methods in a noise free environment. This is because the classical target tracking method is a deterministic algorithm and does not perform any learning. Consequently, this algorithm defines its state space separately than the reinforcement learning algorithms. In effect, every possible target coordinate can be considered a state and the only action available in each state is to move the servo's the calculated distance (using (1)) to the target. In a sense, it selects the right action to take in every state. As a result, this algorithm is important because it demonstrates the desired behaviour of the reinforcement learning algorithms and provides a baseline for comparison when plotting the results.

Next, the Q-learning method distinctly produced the worst results. This is due largely to the algorithm losing the predetermined target. As mentioned above, the algorithms were evaluated on the distance of the target from the origin. However, a distance of zero is produced when no target is present. Consequently, the Q-learning results were adjusted any time the target distance was zero for an extended period of time. This can be easily corrected in the software by assigning the maximum distance when the target is not present. Additionally, it should be noted that the Q-learning algorithm was tested separately before being introduced to our environment. Both learning methods implement the same Object Oriented interface and were tested using simple examples from [18] to verify correct behaviour.

Lastly, we ran extra experiments with the actor-critic method, including a 15 minute test and a 2 hour test. The average RMS error for the 15 minute test was slightly reduced from the original 5 minute test. Most likely this is due to convergence of the learning actions in selecting the optimal choice in each state. The 2 hour results showed possible convergence of the AC and classical methods. However, more work on the reported algorithms is necessary to complete the evaluation of the target tracking methods.

5 Conclusion

This paper reports work on target tracking and commanding for a line-crawling robot called ALiCE II. Commands are sent to ALiCE II via email messages. A steady stream of images for equipment that ALiCE II finds "interesting" (i.e. images showing an object with a recognised problem) can be sent to a base station for evaluation. Preliminary results of experiments with conventional as well as reinforcement learning-based target tracking methods are also reported in this article. These results suggest that the classical target method is more suitable for camera movement control. Future work will include the use of RL methods to control pan and tilt operations to achieve the best view of a target object.

6 Acknowledgements

We wish to thank Wes Mueller and Manitoba Hydro for their help during this project. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grants 185986, 194376 and grant T247 from Manitoba Hydro.

7 References

[1] J.F. Peters, T.C. Ahn and M. Borkowski, "Obstacle classification by a line-crawling robot: a rough neurocomputing approach", *Proceedings 3rd International Conference on Rough Sets and Current Trends in Computing (RSCTC)*, Malvern, USA, 2475, pp 549-601 (2002).

[2] J.F. Peters, T.C. Ahn, M. Borkowski, V. Degtyaryov and S. Ramanna, "Line-crawling robot navigation: a rough neurocomputing approach", in C. Zhou, D. Maravall, D. Ruan (eds), *Autonomous Robotic Systems: Soft Computing and Hard Computing Methodologies and Applications*, Physica-Verlag, NY, pp 141-163 (2003).

[3] J.F. Peters, "Rough ethology: towards a biologically-inspired study of collective behaviour in intelligent systems with approximation spaces", *Transactions on Rough Sets III, LNCS*, 3400, pp 153-174 (2005).

[4] J.F. Peters and S. Ramanna, "Hierarchical behavioural model of a swarmbot", *Symposium on Methods of Artificial Intelligence (AIMETH04)*, Gilwice, Poland, pp 105-106 (2004).

[5] J.F. Peters, C. Henry, D. Lockery, M. Borkowski and S. Ramanna, "Approximation space-based rewards for swarms that learn", in T. Burczynski, W. Cholewa, W. Moczulski (eds), *Recent Developments in Artificial Intelligence*, Silesian University of Technology, Poland, pp 139-144 (2005).

[6] F.Y. Zhou, J.D. Wang, Y.B. Li, J. Wang and H.R. Xiao, "Control of an inspection robot for 110KV power transmission lines based on expert system design methods", *Proceedings of 2005 IEEE Conference on Control Applications (CCA05)*, Toronto, Canada, pp 1563-1568 (2005).

[7] W. Stallings, *Data and Computer Communications* (6th edition), Prentice Hall, Upper Saddle River, NJ (2000).

[8] W. Simpson, "RFC 1661: The Point-to-Point Protocol (PPP)", http://www.ietf.org/rfc/rfc1661.txt, visited on 04/04/2006.

[9] Technologic Systems, "TS-5500 User's Manual", http://www.embeddedarm.com, visited on 10/07/2005.

[10] G. McGregor, "RFC 1332 The PPP Internet Protocol Control Protocol (IPCP)", http://www.ietf.org/rfc/rfc1332.txt, visited on 04/04/2006.

[11] The Apache Software Foundation, "Apache Web Server", http://www.apache.org, visited on 17/07/2006.

[12] PHP, "PHP: General-Purpose Scripting Language", http://www.php.net/, visited on 17/07/2006.

[13] Linux Online, "Linux operating system", http://www.linux.org/, visited on 15/07/2006.

[14] The Perl Dictionary, "Perl programming language", http://www.perl.org/, visited on 14/07/2006.

[15] Technologic Systems, "Elan SC520 Microcontroller User's Manual", http://www.embeddedarm.com/, visited on 04/04/2005.

[16] Creative Worldwide, "Creative WebCam NX Ultra", http://www.creative.com/, visited on 15/07/2006.

[17] Sourceforge.net, "SPCA50X USB Camera Linux Driver", http://sourceforge.net/, visited on 10/05/2005.

[18] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA (1998).

[19] J.F. Peters, C. Henry and S. Ramanna, "Reinforcement learning with approximation spaces", *Fundamenta Informaticae*, 71(2-3), pp 323-349 (2006).

[20] Z. Pawlak, *Classification of Objects by Means of Attributes*, Report 429, Institute for Computer Science, Polish Academy of Sciences (1981).

[21] C.J.C.H. Watkins, *Learning from Delayed Rewards*, PhD Thesis, King's College, University of Cambridge, UK (1989).

A Comprehensive Approach for Building an Integrated and Reusable Agent based Applications

Nora Houari and Behrouz H. Far
Department of Electrical and Computer Engineering,
University of Calgary, Calgary, Alberta, Canada
nhouari@ucalgary.ca, far@ucalgary.ca

Abstract

System development methodologies play a significant role in guiding system developers and their adoption improves the process of system development. In the last few years an increasing number of methodologies for building multiagent systems (MAS) have emerged. These differ in the agent perspectives they capture, the development process phases they consider and the target deployment environment, which presents a challenge when trying to use them in a project. This paper presents a comprehensive approach that combines both methods engineering principles and annotations of design decisions adopted during system development. Making these design decisions explicit helps the methodology user know how system decomposition was carried out, what are the crosscutting and interrelated concerns that affect the system's reuse, maintenance, adaptation and integration. We used the proposed methodology to develop a multi-agent decision support system geared towards assisting software engineering organisation teams in the development of the project lifecycle, to achieve project performance (e.g. schedule, effort, quality, etc).

Keywords: multiagent system, methodology, decision support system.

1 Introduction

The last two decades have seen the development of complex and distributed software engineering systems. A key element in this advance is the development of increasingly powerful and natural high-level abstractions. Software agent concept is widely viewed as an advance in this direction. Like human agents, a software agent can carry out a task and has its own characteristics, such as autonomy, social ability, reactivity, proactiveness, cooperative, learnable and adaptable [1]. An agent is thus considered a natural abstraction of the real world, with its own goals, and interacts with other agents to achieve mutual benefits. Agents are deployed within a specific environment, they perceive it, act upon it and affect it. Thus, the agenthood paradigm captures different abstraction facets, such as its internal structure; inter-agent and social structure; and agent intra-environment structure. As higher levels of abstractions are required to systematically support the process for the ever-growing complexity of software, this further complicates the already difficult problem of reusing software concepts.

Kryuchten [3] pointed out that "for a software reuse technique to be effective, it must reduce the cognitive distance between the initial concept of a system and its final executable implementation". Decomposing a system into context-independent agent components is not trivial. Generally some concerns are spread over different components, creating crosscutting and interconnected dependencies. Applying the separation of concerns principle requires adopting some design decisions that deal with identifying these concerns and explicitly representing them as first class standpoints, since they influence the system integrity, maintenance, reuse and adaptation.

Recently a number of agent-oriented development methodologies have been proposed [4, 5, 6, 13]. These vary in (a) the agent perspectives they acquire (internal, inter-agent, intra-environment); (b) the process development phases captured (requirements, analysis, design, etc); (c) supporting tools and (d) target implementation platform. It has been realised that no universal accepted approach or framework for studying or evaluating these methodologies, nor a methodology that fits all domains, can be found. Despite attempts to classify and compare these methodologies and to unify some, developers must still go through all the methodologies to select one or more that can be reused for their system with further adaptation. This is no easy task as not all assumptions and design decisions made in developing the methodology are explicitly represented or documented.

To overcome some of the issues with existing MAS methodologies, particularly when considering a multiagent application that must be integrated in a specific environment, be maintainable, and target reuse concepts, we propose a comprehensive methodology that adopts the perspective of MAS characteristics, that is, fundamentally a composition of both descriptive and prescriptive knowledge. *Descriptive* knowledge embodies the design methods engineering and techniques used. Most existing methodologies deal with this category of knowledge. *Prescriptive* knowledge on the other hand represents design decisions made along the methodology, which usually relate to the crosscutting and interrelated aspects of the application domain. This knowledge should be annotated as first class system standpoints, as it affects system integrity, adaptation, maintenance and reuse. We use the DIBRA approach [7] for designing the individual agents. DIBRA integrates (a) agent internal structure (desire, belief, and intention, DBI), (b) inter-agent, inter-environment, and agent autonomy structure in the agent's rapport element, and (c) the agent's adaptation of its beliefs and plans. The process and techniques we used for the system development applies engineering method endowed with annotation of design decisions, in particular those related to intertwined system concerns.

We use this approach to build a decision support system to assist software development teams during the project lifecycle to track roles, tasks and activities to achieve project performance (schedule, effort, quality, etc).

2 Related Works and Motivation

Several methodologies for building agent based systems have been proposed [13], each created with distinct purposes. For instance some focus on *agent characteristics* including internal structure, such as the BDI model [2], social structure [8, 9]; others concentrate on particular *phases* of the development lifecycle (requirements analysis, design, etc.); still others use specific *supporting tools* [10, 11] and others target particular *platforms* [12]. These methodologies are applicable in the appropriate contexts, but their adaptation and reuse are limited outside these contexts as the key underpinning design decisions, particularly the crosscutting and inter-dependent concerns, are not well described and represented for application, reuse and integrity.

In [13] the authors evaluate ten existing agent methodologies with regards to process, techniques, model related criteria and supportive features, then use one open meta model to unify them. Despite efforts to offer a clear comparison between methodologies, this work has some limitations: the comparison was based on documentation available for these methodologies and was not applied to different projects to identify their performance; some evaluation criteria were subjective, especially the usability and understandability of the methodologies, process steps, techniques and models.

The work of [14] studied three methodologies and their related concepts and tried to find a means to unify them under a meta-model. This was a first step towards interoperability between few methodologies, but there was no evaluation or validation of the unified model in terms of its flexibility for reuse with different methodologies and for users unfamiliar with agent oriented methodologies.

Authors of [15] proposed an approach for multi agent system development targeting reuse and maintenance. They used generative programming and aspect-oriented software development. Although this takes care of crosscutting concerns it lacks explicit annotation and representation of key design decisions addressed during design to help with reuse and adaptation of the methodology.

It is accepted that the creation of a universally applicable methodology is an unrealisable goal; what is practical is to create a methodology which simultaneously satisfies system requirements and fulfils software practitioners' needs. This paper argues that such flexibility may best be obtained by (1) exploring concerns that are uniquely or most appropriately addressed by the agenthood paradigm; (2) using methods engineering that provide descriptive knowledge in terms of process and techniques; (3) endowing the development process with design decisions representations and annotations that specify the prescriptive knowledge used when dealing specifically with decomposition and crosscutting interconnecting concerns of the system. These three fundamentals formalise the descriptive and prescriptive knowledge used in the MAS development methodology, which leverages the system integrity, maintenance, reuse and adaptation.

3 An Application Case Study of the Approach

We applied our new approach in software engineering project lifecycle performance prediction, particularly software delivery time. Software development systems are usually multi team and multi stage development; they may belong to one organisation or be shared among partners; due to the wide distribution of data, information and knowledge, it is impossible to effectively locate, share or learn from these projects, raising an urgent need for data integration and orchestration. Solving such complex problems, without time consuming and often unavailable human intervention, can only be accomplished by invoking autonomous agents. Software agents were introduced to project team members to assist them with time-consuming activities and automate repetitive user tasks. The agents are software assistants representing different roles of the teams' developers and coordinate their activities. Each agent specialises in small and distinct subsets of the overall objective of the system. Centralised control is not necessary; agents cooperate through their perception of the environment and adaptation ability. Figure 1 shows the system architecture. To carry out a project, different data come from different teams; monitoring the project delivery rate for example requires combination of means from the various participants. Agent support mechanisms are geared towards effectively providing proactive monitoring and prediction throughout the project development process. As well as keeping track of current events (e.g. reaching certain milestones on time or with delay), agents can learn from experience to provide users with advance warning that a problem is likely to occur. Multi-agent architecture offers several advantages. Agents solve a complex task by dividing it among them in a distributed manner. The system structure is presented in figure 2, and was deployed using JADE [16] as an implementation platform system. The case study is described in detail in [17].

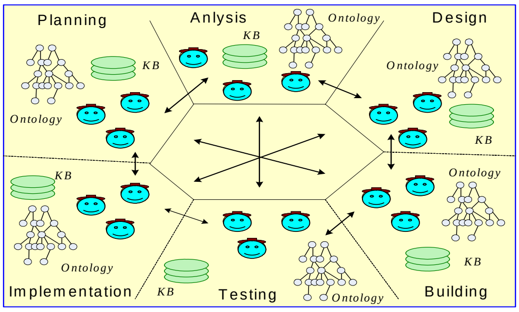


Figure 1: Application case study system architecture.

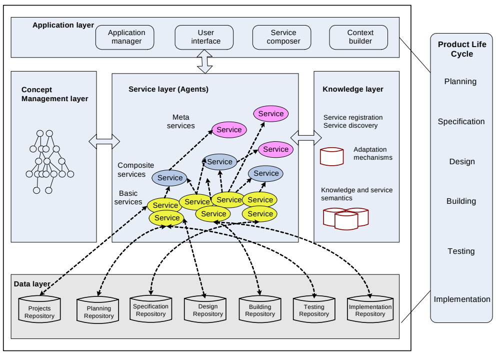


Figure 2: System structure.

4 A Comprehensive MAS Development Approach

In section 2 we examined some of the work done in multi-agent-oriented methodologies and discussed the way they handle concerns related to the development of multiagent systems. We then assessed the need to endow agent-development methodology with explicit representation and annotation of design decisions, particularly those dealing with system decomposition and how crosscutting interrelated concerns are addressed. In this section we present our new approach that addresses these issues.

The process of software engineering development involves the use of *descriptive* knowledge and *prescriptive* knowledge. Descriptive knowledge deals with the methods and rules applied, that is, knowledge of "*how*", especially how best, to perform some tasks. This type of knowledge is addressed by most existing methodologies. Prescriptive knowledge on the other hand represents *"know how"* and the strategy used, and answers the question why a component is this way, for example. This type of knowledge is difficult to extract and represent because it is personalised and context-specific. Here we adopt representation and annotation of this knowledge through the four main stages of system development: (a) system requirements analysis and priorities; (b) system decomposition; (c) system functionalities and properties; and (d) system integrity with its environment.

Our approach embraces the IEEE definition of a methodology and enhances it with the annotation and depiction of first class design decisions for the crosscutting and interwoven agent concerns

In reviewing agent-oriented methodologies, we noticed that the terms methodology, method, techniques, process, activity, etc. are used indiscriminately [18]. For consistency, here we follow the IEEE definitions for such terms; thus (1) a *methodology* is described as "a comprehensive, integrated series of techniques or methods creating a general systems theory of how a class of thought intensive work ought to be performed"; (2) a *method* is a set of "ordered processes or procedures used in the engineering of a product of performing a service"; and (3) a *technique* is "a technical and managerial procedure used to achieve a given objective" [19].

According to the IEEE definition, methods and techniques are related. In fact methods and techniques are used to carry out tasks inside the different processes which comprise a methodology. A process is a "function that must be performed in the software life cycle". A process is composed of *activities*: an activity is "a constituent task of a process". A *task* is the smallest unit of work subject to management accountability; related tasks are usually grouped to form activities [19]. Figure 3 summarises the relations between these definitions. Our approach adds the first class representation of key design decisions that address the crosscutting and interwoven concerns of system development.

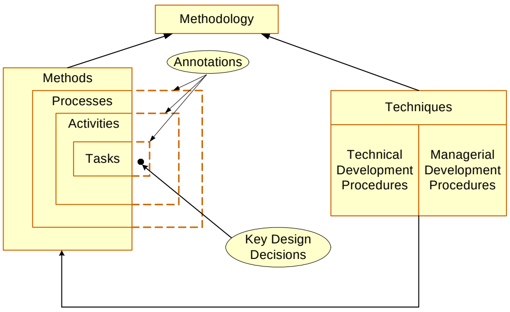


Figure 3: Illustrative view of a reusable methodology.

Our agent types are based on DIBRA [7] where internal structure, external structure, and integration of agent within its environment are taken into account. The following section reviews a summarised version of the methodology process and activities.

5 Methodology Activities and Process

In our multiagent development approach we used the entire software development process and followed a top down structured design method, taking into account (a) system priorities and constraints, (b) ensuring the system would achieve its objectives and requirements, (c) and endorsing system integration with the environment. We used scenarios as a foundation to illustrate system properties and processed an iterative and incremental development approach. The approach activities and development process are briefly summarised as follows.

5.1 Approach Activities

Given our concerns of integrating both methods engineering and annotation of key design decisions, especially those dealing with interconnected and dependent concerns, we carry out the following activities (see figure 4):

1. **Goals decomposition:** from system requirements, we develop a set of tasks scenarios illustrating the system properties of the system. These scenarios represent tasks relevant to different roles of the system's users; from these scenarios a list of system goals (desires) is extracted.
2. **Plans structuring:** once the system goals are identified, we perform the schedule of tasks so that those on which other tasks depend are completed early; here goals are arranged into executable states towards their achievement. Questions in this activity include how to organise data structures, how to schedule tasks to be assigned to agents, how to exploit data locality, whether to communicate in small or large messages, etc.
3. **Capabilities-setting:** to fulfil each goal, a set of assertions (beliefs) with underlying proportional semantics is required. These constitute the ontology and knowledge base that the designated agents need to be endowed with. During this activity identification of agent types, resources and services as well as the deployment environment are featured. Here also the resources available to agents are captured (i.e. databases, objects and the integration of the system with the environment, SQL queries, plug-in module, etc.).
4. **Roles-assignment:** in our approach roles are underpinned by the organisation structure, since the agents will assist their users within the organisation. A set of soft roles is considered as a position within an organisation. Specifying a role consists of grouping set of goals into specific objectives. Revealing role interaction is important at this point because it affirms the system functionality. Role interactions are subject to constraints and temporal regulation norms that regulate system behaviour [8]. A high degree of interaction may mean that the functionality of a particular designated agent performing a role is poorly isolated. Role interaction determines the extent to which the agent assigned role will support separation of concerns. Here we make explicit representation of key design decisions adopted for role assignment as they are concerned with system decomposition, especially the representation of crosscutting and interwoven concerns. This annotation helps with system integrity, reuse, adaptation and maintenance.
5. **Orchestration:** denotes the activity in which the agents are assigned roles. In our methodology individual agents are assigned singular roles and collaborate if needed to support a distinctive organisational human role. Agents achieve their goals by performing responsibilities of the role assigned and manipulating objects. The primary performance goal of orchestration is to balance the workload among agents and to reduce inter-agent communication. For an agent to execute its goals it needs a reasoning process, interaction, collaboration and preservation of its autonomy. Here the DIBRA [7] structure of agents is endorsed (figure 6). Orchestration is more dependent on the implementation platform and the environment where the system is to be integrated.
6. **Mapping:** the collaborative agents that result from the above steps constitute a full-fledged multi-agent application. The application manager takes care of mapping the agents and associated resources to the implementation platform.

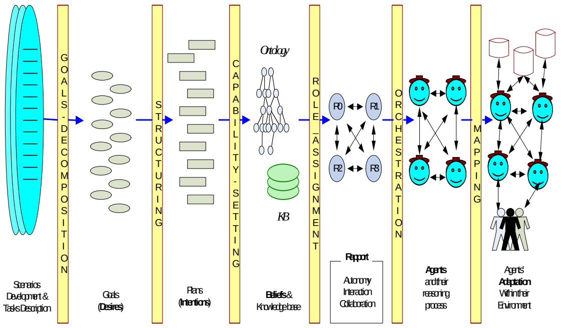


Figure 4: Methodology activities.

5.2 The Development Process

In building our approach we first identify the desires from the system requirements (scenarios) these are the goals of the system, then capture the intentions these are the plans, followed by the required knowledge which constitutes the beliefs both explicit and implicit in the form of potential deductions based on logical reasoning; to communicate with the environment agent needs a dialogue (rapport) established from communication language, protocol, norms, and contracts. As an agent, it needs to adapt to its environment. The building blocks of our agent modelling are shown in figure 5.

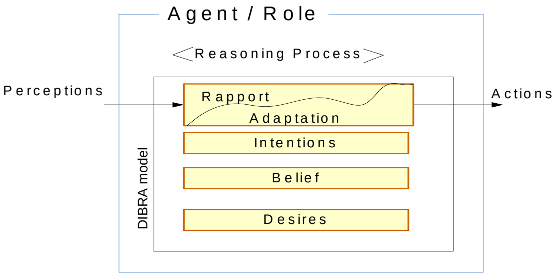


Figure 5: Individual agent architecture.

Our approach follows the iterative and incremental software modelling phases. These are requirements, analysis, design, building, testing and implementation. It has seven main steps: system requirements, system architecture identification (components and connectors), desires (goals) identification, finding intentions (plans), designating the beliefs, specifying the rapport (interaction, collaboration, and autonomy) with the environment, and finally building the adaptation (figure 6).

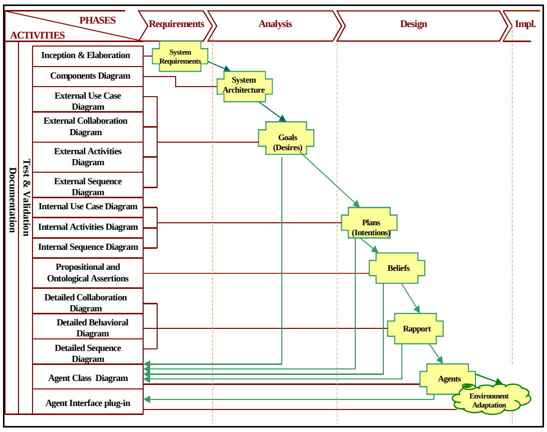

PHASES
ACTIVITIES
Requirements
Analysis
Design
Impl.
Documentation
Test & Validation
Inception & Elaboration
Components Diagram
External Use Case Diagram
External Collaboration Diagram
External Activities Diagram
External Sequence Diagram
Internal Use Case Diagram
Internal Activities Diagram
Internal Sequence Diagram
Propositional and Ontological Assertions
Detailed Collaboration Diagram
Detailed Behavioral Diagram
Detailed Sequence Diagram
Agent Class Diagram
Agent Interface plug-in
System Requirements
System Architecture
Goals (Desires)
Plans (Intentions)
Beliefs
Rapport
Agents
Environment Adaptation

Figure 6: Methodology process.

Because of space constraints, here we underline only the requirements, analysis, and design phases; we used UML for the diagrams and propositional semantics for beliefs representation. These phases are briefly summarised in the following steps:

Step 1: System requirements and assessment: this initial step describes the system requirements. It is the conceptualisation of the required system from the customer's point of view, and it denotes the services and functionalities that the system has to provide. It states the goals expected from agents.

Step 2: System architecture: in this step we identify the system components and connectors. This step is a high level description of the system that helps make the system more understandable, intellectually manageable, guide development implementation and evolution of the system for future modification.

Step 3: Capturing the desires (goals): the desires of the system are specified from the requirements, they are abstract artefacts. We use UML use case, collaboration, activity and sequence diagrams to represent the system services, more precisely from the external point of view. We identify what services the system should provide who are the actors, what they do and for what purpose without dealing with the internal details.

Step 4: Extracting the intentions (plans): from the external point of view, we create internal diagrams that show how the internal entities of the system interact, these help identify intentions for the previously specified goals and reveal more details that are not obvious to extract from the external point of view.

Step 5: Identifying the beliefs (knowledge): beliefs are the required knowledge that the plans (intentions) use to fulfil the goals. These are stored in the form of a set of assertions, which comprise standards propositional operators (conjunctions, disjunctions, negations and application) a set of quantifiers (as in predicate calculus) modal and temporal operators, and other devices for quantifying assertions with a level of uncertainty, as well as ontological assertion.

Step 6: Specifying the rapport: (autonomy, interaction and collaboration): here the different types of agents are generated. Agents are the refined roles with the specification of agent reasoning process deducted. As agents need to collaborate to fulfil the requirements, they need to interact with one another and with the external environment. We use message passing as a unique form for method invocation. Agents distinguish different type of messages as speech acts and use a collaboration protocol. This protocol is a set of public rules that dictate the conduct of an agent with other agents to achieve a desired final outcome in sharing the knowledge and performing actions that satisfy the desired goals to fulfil some utility functions. Messages are in a defined XML (eXtensible Markup Language) format and transported using the Simple Object Access Protocol (SOAP). After identifying basic parts of system entities -agents, desires, intentions, beliefs, and rapport- we gave names to agents.

Step 7: Agent adaptation: adaptation refers to changes in the agent knowledge or behaviour, and it includes belief and plan adaptation. It is closely related to the implementation platform and the integrated environment. Here several issues such as reusability integrity are considered, also the refinement of agents' interaction, object, services resources, and permission and constraints.

At the end of the design phase we create the agent class diagram. It contains the agent name, desires, intentions, beliefs, rapport, and adaptation capabilities. At this point the system is ready to implement. It is quite straightforward, because we have almost the whole ready to code agents' capabilities and behaviours. It is possible to go back to previous artefacts of the earlier steps and make some adjustments for consistency with the interaction and the adaptation of the agents.

6 Conclusion and Ongoing Work

Given the large existing number of agent-oriented methodologies, where each was created with specific purpose, targeting some perspectives of agent characteristics and capturing designated software development phases; system developers are faced with the dilemma of adopting these methodologies to the project they have at hand. The approach presented in this work uses method engineering supported with annotations and representations of key design decisions in particularly those dealing with system decomposition, crosscutting and interconnected concerns that affect the system integrity, reuse, and evolution. These are elements of the prescriptive knowledge used during the development. The approach captures the whole lifecycle of the software development and explicitly outlines the activities and the process of the methodology. We used the methodology to develop a decision support system to assist project development teams in achieving project performance (i.e. delivery time, effort, and quality).

We are currently developing a tool that identify MAS methodology candidates from library of methodologies, this tool guides the system developers in adopting the methodology to the project they have at hand, this ongoing work is done with collaboration of the major known methodologies' authors as it requires representation and annotation of first class design decisions, such as those dealing with system decomposition and crosscutting interrelated concerns of agent characteristics. The tool will be used for different types of projects to evaluate the performance of the methodology when both the descriptive and prescriptive knowledge are captured.

7 References

[1] F. Zambonelli, N. Jennings and M. Wooldridge, "Organizational abstractions in the analysis and design of multi-agent systems, *Proc. 1st Intl. Workshop on Agent-Oriented Software Engineering at ICSE* (2000).

[2] D. Kinny, M. Georgeff and A. Rao, "A methodology and modelling technique for systems of BDI agents", *Proc. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, LNAI*, 61 (1996).

[3] C. Krueger, "Software reuse", *ACM Computing Surveys*, 24(2), pp 131-183 (1992).

[4] M. Amors, L. Fuentes and A. Vallecillo, "Bridging the gap between agent-oriented design and implementation using MDA", *AOSE 2004, LNCS 3382*, pp 93-108 (2005).

[5] M. Wooldridge, N. Jennings and D. Kinny, "The Gaia methodology for agent-oriented analysis and design", *Journal of Autonomous Agents and Multi-Agent Systems*, 3, pp 285-312 (2000).

[6] T. Stockheim, D. Frey and P. Woelk, "Integrated multi-agent-based supply chain management", *Proc. 1st International Workshop on Agent-based Computing for Enterprise Collaboration*, IEEE Computer Society Press (2005)

[7] N. Houari and B.H. Far, "Integrated approach for developing autonomous and interactive software agents," *Proc. 10th IEEE Intl. Conf. on Emerging Technologies and Factory Automation (ETFA'05)*, Catania, Italy (2005).

[8] V. Dignum, *A Model for Organizational Interaction Based on Agents, Founded in Logic*, PhD Thesis, University of Utrecht (2003).

[9] J. Filipe, "Information fields in organization modelling using an EDA multi-agent architecture", *Proc. AAAI Spring Symposium on Agent-Mediated Knowledge Management (AMKM'03)*, Stanford, USA (2003)

[10] M. Pavlos, E. Petraki and N. Spanoudakis, "Engineering JADE agents with the Gaia methodology", *Agent Technology Workshops*, LNAI 2592, Springer Verlag, Berlin, pp 77-91 (2003).

[11] AgentBuilder, "An integrated toolkit for constructing intelligent software agents", http://www.agentbuilder.com/

[12] G. Laleci, Y. Kabak, A. Dogac, I. Cingil, S. Kirbas, A. Yildiz, S. Sinir, O. Ozdikis and O. Ozturk, "A platform for agent behaviour design and multi agent orchestration, *Proc. AOSE '04, LNCS 3382*, pp 205-220 (2005).

[13] B. Henderson and P. Giorgini, *Agent-Oriented Methodologies*, Idea Group (2005).

[14] J. Sudeikat, L. Braubach, A. Pokahr and W. Lamersdorf, "Evaluation of agent oriented software methodologies: examination of the gap between modelling and platform", *Proc. AOSE '04, LNCS 3382*, pp126-141 (2005).

[15] U. Kulesza, A. Garcia, C. Lucena and P. Alencar, "A generative approach for multi-agent system development", *Proc. SELMAS 2004, LNCS 3390*, pp 52-69 (2005).

[16] Java Agent Development Framework (JADE), "Home Page", http://jade.tilab.com/

[17] N. Houari and B.H. Far, *A Model for Software Engineering Project Decision Making Support Based on Collaborative Agents*, Technical Report, University of Calgary Press (2006).

[18] R De Hoog, "Methodology for building knowledge based systems: achievements and prospects", in J. Liebowitz (ed.), *The Handbook of Applied Expert Systems*, CRC Press, Boca Raton, FL (1998).

[19] *IEEE Standard Glossary of Software Engineering Terminology,* IEEE Std 610.121990, IEEE Computer Society, New York (1990).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Networked Remote Operation of Humanoid via Motion Interpretation and Image Recognition

Junichi Kato[1], Narihito Takahashi[1], Yuuki Ueda[1], Yukiko Sugihara[1] and Yasuo Matsuyama[2]
[1]Graduate School of Computer Science / [2]Department of Computer Science,
Waseda University, Tokyo, Japan
canichi@wiz.cs.waseda.ac.jp, nari@wiz.cs.waseda.ac.jp, y-ueda-1020@wiz.cs.waseda.ac.jp,
yukiko@wiz.cs.waseda.ac.jp, yasuo@wiz.cs.waseda.ac.jp

Abstract

New remote operation methods for biped humanoids are presented and examined. The system we examine includes motion interpretation and image recognition. The flexibility difference between the human and humanoid movements is absorbed by the motion interpretation. Through the network, the human operator can share the same view as the humanoid sees. The system has supervised and autonomous modes. In the supervised mode, human motions are input to generate humanoid actions. In the autonomous mode, the image recognition and the 3D stereo vision are utilised. When the humanoid visually identifies cue objects in its environment, the operation is switched to the autonomous mode where the humanoid can act independently of the human operator. The motion interpretation, the image recognition and the humanoid action are examined for the supervised and autonomous modes. Experiments show that the designed system works in both modes effectively.

Keywords: biped humanoid, motion recognition, remote operation, autonomous mode, stereo vision

1 Introduction

Humanoids have become rapidly sophisticated these days in various directions [1]. Supporting theories and technologies include computational intelligence, processor power and networks besides the growing knowledge of the human body and cognition.

The humanoid appearing in this paper is biped (two-legged) and two-handed. Hereafter, it will be simply called the humanoid. There is an exclusive class of tasks which can be performed by the humanoid. Such a class of jobs can be expressed well by “the surrogate to human actions”. The humanoid can behave like a human with allowable clumsiness. Its expected tasks include the following:

1. Dangerous jobs can be taken over by the humanoid.
2. Senior people can be well treated by the humanoid with its affection, especially in cases where their relatives live in distant locations.

These examples share the importance that the total system needs to satisfy below.

a. The humanoid operations need to be compatible with human sense and perception.
b. The operation can be done remotely through the network.
c. The operation needs to be completed within bearable delay for human and/or machine clients.

Therefore, we develop a humanoid control system which directly uses human motions as *soft commands.*

Thus, the main characteristics of the presented system are as follows.

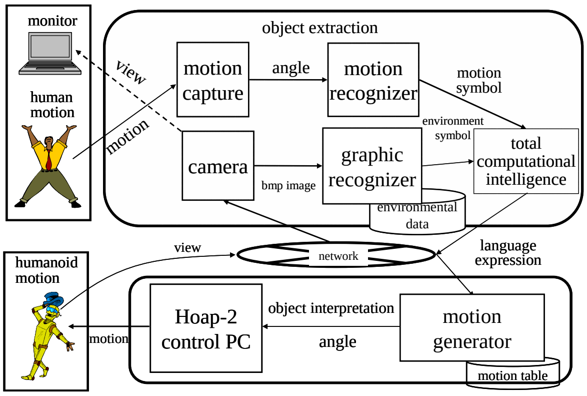


Figure 1: Remote operation of humanoid with motion interpretation.

1. Human motions are captured to create the commands. But, this is not based on a mere master-slave system. A mechanism of the motion recognition for absorbing the man and machine’s individuality [2] is included. This is an essential device in order to absorb the difference between human and humanoid movements.
2. The operator, i.e., the human user, can watch the scenery through the humanoid’s eyes. This is an important factor since the remote operation through the network is addressed in this study.
3. The humanoid has both supervised and autonomous modes. That is, the humanoid can switch itself from the supervised mode to the autonomous mode if it detects a cue in the environment. This is another property that the presented system is beyond a mere master-slave style.

Figure 1 illustrates the total paths including subsystems to realise the above characteristics: the motion generation and capturing, the motion interpretation, the image recognition, the command generation, and the action generation by the humanoid. We note here that the changeable environment of the humanoid is visually fed back to the remote operator through the network (the dashed line).

2 System Organisation

The total system has the role to recognise or estimate what is the human operator's intention. For this purpose, the system makes use of motion data and image data to be integrated.

2.1 Motion Capturing

The purpose of motion capturing is to supply enough information so that the target humanoid can be operated precisely. So far, we use a magnetic system by Ascension Technology. Note that the motion capture puts out only a set of numerical numbers. Therefore, the transformation to the BVH format (Bio Vision Hierarchical data) is necessary.

Sensors give a good amount of information on the positions, angles and rotation matrices. This information is usually too much for the humanoid. Therefore, the extraction and recognition of the core information is important. Figure 2 illustrates a view where a motion generator (a human operator) wears a set of sensors. His posture is for the system calibration. The animation is used to check to see if the human posture is correctly captured and extracted. On the method of the animation and related robotics, readers are requested to refer to [3].

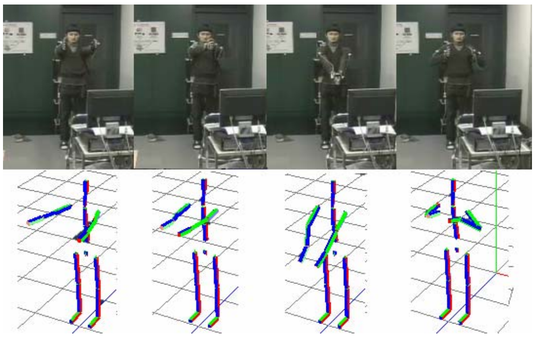


Figure 2: Motion capturing and animation.

2.2 Motion Recognition

The humanoid control via the human motion is not quite straightforward. Each angle of human joints does not have specific meaning. But the total human motion with an intention has significance. The moving capability of the humanoid is considerably different from that of a human's. The ordering by the flexibility is

[human] >= [animation] >> [humanoid].

Therefore, the data of the human motion needs to be further reduced effectively towards the recognition at the box of "motion recogniser" in Figure 1.

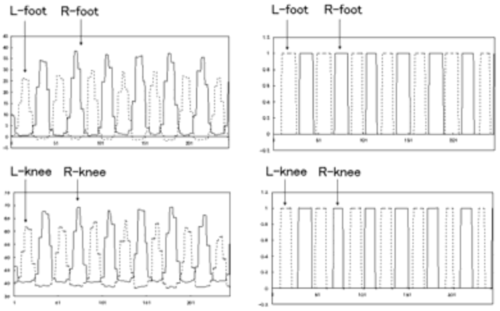
Due to the difference in the motion granularity, we apply a tight convention to the recognition.

[Recognition] A human motion is *successfully recognised* if it is reproduced by a humanoid within the *acceptable flexibility difference*.

For instance, if a human operator walks and so does a humanoid, the recognition is successful. The flexibility is absorbed if both motions agree in a language level. The language may be an abstract one. Thus, the number of recognisable motions depends on 1. the motion capturer, 2. the motion interpreter and 3. the humanoid's flexibility. As will be found in section 2.5, there are 11 elemental motions for the lower half of the body (see figures 8 and 9). Their combinations are unlimited. On the upper half of the body, there are no such discrete elements since the motion is nearly free from a fall (figures 7 and 10).

2.2.1 Rule-Based Motion Recognition

The simplest and computationally fastest recognition method for the operator's motion is the rule-based motion extraction. Figure 3a (upper) shows periodical movements of the left and right legs. Figure 3a (lower) shows periodical movements of the left and right knees.



(a) Angle. (b) Driving signal.

Figure 3: Motion measurement and labelling.

These four time series have precise phase shifts for "walking". Therefore, this movement is recognised to generate a humanoid command of "walking." Figure 3b is a set of time-dependent commands which specify the driving of {L-foot, R-foot, L-knee, R-knee} with an appropriate timing for the humanoid without fall. Except for such a set of meaningful patterns, irregular motions are regarded as "noisy movements". If the operating condition is robust so that the operator generates clear motions like figure 3, such a rule-based command-issuing works well. But, the operator's body size is often different. Also, wobbles may occur during the operation.

2.2.2 Probabilistic Motion Recognition

After the consideration in section 2.2.1, one realises that probabilistic or stochastic mechanisms are necessary so that noisy movements are absorbed properly. Such candidates are the Bayesian network and the hidden Markov model (HMM).

1. The Bayesian network matches our system to express behavioural actions. Thus, the granularity of expressible actions is medium. The computational demand is also medium.
2. HMM can express diverse granularities of actions. There is an example of this application by this paper's group [3]. The main problem is the demand on the computational resources.

Since the target of this paper includes the humanoid's *real-time* supervised walking and its autonomous mode, the Bayesian network is judged to be more suitable than the HMM this time. This is due to the HMM's excessive computational load to the PC processor. Thus, the content of the box "motion recogniser" in figure 1 is a Bayesian network.

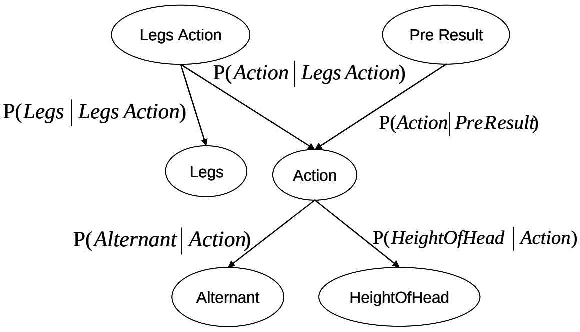


Figure 4: Bayesian network diagram.

The Bayesian network learned from 15,000 frames of data generated by three subjects is illustrated in figure 4. Each node stands for judged actions. Arcs have transition probabilities. The node "Leg Action" specifies "Left Leg Up" and "Right Leg Up". The node "Legs" has the states {Left High, Right High, Even}. The node "Pre Result" stands for the recognised result of the one-frame before.

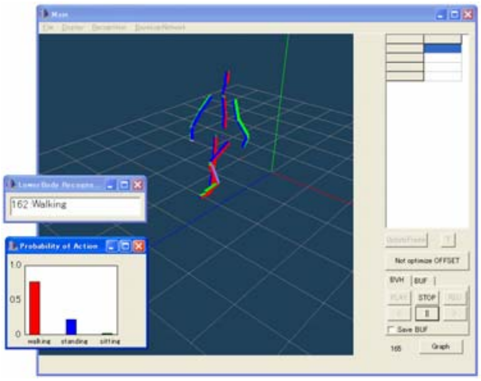

162 Walking

Figure 5: Real time animation and recognised results.

Figure 5 illustrates the operator's motion and the recognised result described by sentences and bars (two popup windows on the left side). Thus, the motion recognition before the action is successful.

2.3 Image Recognition

The humanoid in this paper (HOAP-2 by Fujitsu Automation) has two CCD cameras for eyes. But, there is no automatic focusing. Therefore, the distance measurement using the epipolar geometry is necessary [4]. The object position *(x, y, z)* is estimated as follows.

Let (u, v, f) and (u', v', f) be two points on the epipolar line. Here, f is the focus. Let B be the distance between the left and right eyes. Then, the disparity is

$$d=u-u' \qquad (1)$$

and the coordinates are

$$x=uB/d, \quad y=vB/d, \quad z=fB/d. \qquad (2)$$

Thus, the distance measurement on the target by the image recognition comprises the following steps.

a. Measure the images by the left and right eyes.

b. On each image, the labelling, or the region merge, is computed. The method is the pixel merge by considering four surrounding pixels {Northwest, North, Northeast, West}.

c. Find corresponding object positions for the left and right eyes using the labelled images.

d. Using the epipolar geometry of (1) and (2), the object's position (x, y, z) is computed.

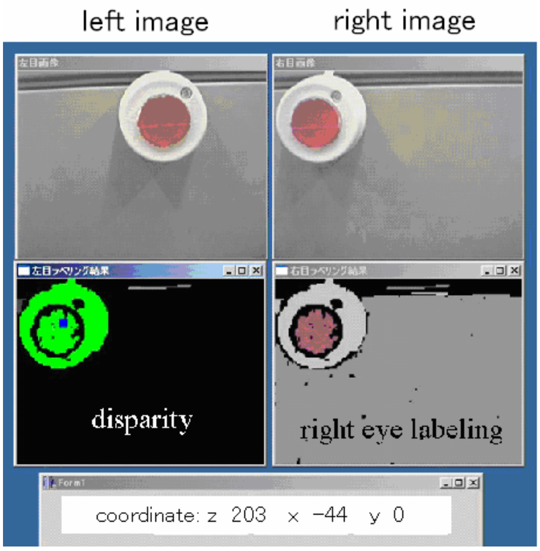

coordinate: z 203 x -44 y 0

Figure 6: Object recognition.

The result of this computation, or the image recognition, is illustrated in figure 6 which contains a set of images of a push button. In this figure, the following images and the estimated position are shown.

1. The captured image by the left eye (top left).
2. The captured image by the right eye (top right).
3. The disparity *d* superimposed on the right eye image (middle left as a distance image).
4. The labelling on the right eye image (middle right as a distance image).
5. The target's *z* position in *mm* whose zero is set to the right eye (bottom left: *z=203*).
6. The target's *x* position whose zero is set to the right eye (bottom centre: *x=-44*).
7. The target's *y* position whose zero is set to the right eye (bottom right: *y=0*).

The measurement of *(x, y, z)* is an important step for the generation of the autonomous mode for the humanoid. This will be described in section 3.

2.4 Motion Generation I: Arms and Hands

For the generation of the remote actions by the humanoid, meaningful upper body motions need to be realised within acceptable delay. Among all movements, those of arms, hands and elbows are especially important.

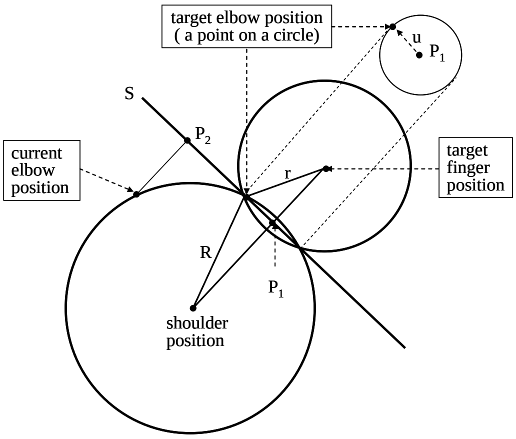


Figure 7: Elbow positioning.

Each arm has four degrees of freedom:

a. elbow angle,
b. forward and backward shoulder movements,
c. left and right shoulder movements,
d. shoulder torsion.

The hand movement is generated by specifying the finger position *(x, y, z)*, and then by setting the angles for (a) ~ (d). This can be performed by the computation of the inverse kinematics. But, one degree of freedom, 4-3=1, remains to be filled since the specification of *(x, y, z)* gives only three degrees of constraints. Therefore, we set one more constraint: the movement of the elbow is kept to a minimum linearly so that the quick finger positioning is realised.

Figure 7 illustrates the computing method for the elbow positioning. Here, *R* is the length of the humanoid's upper arm, and *r* is the length of the forearm including the hand. The target elbow position is computed as follows.

1. The coordinates of {current shoulder position, current elbow position, target finger position} are given.
2. By drawing spheres with the radii *R* and *r*, the surface *S* is found.
3. The point P_1 which is the perpendicular foot on the surface *S* is computed.
4. The perpendicular foot P_2 on the surface *S* from the current elbow position is computed.
5. The radius *u* of the intersection circle of two spheres is computed.
6. The target elbow position *P* is computed by:

$$P = (P_1 - P_2)\,(1 - u/|P_1 - P_2|\,) + P_2 \qquad (3)$$

7. Since the target elbow position and the target finger position are given, all angles can be computed by using *arcsin* and *arccos*.

2.5 Motion Generation II: Lower Body

The human operator, or the motion generator, issues action commands or motions to the humanoid at unconstrained timing. Even for such a sequence of commands, the humanoid has to mimic the operator *without falling down*. Therefore, the control of the humanoid's lower body needs to be executed by considering the following:

a. Avoiding falling down is the ultimate purpose.
b. Each motion by the operator needs to be supported.
c. Any possible sequence generated by the human operator needs to be realised.
d. The sequence of the motions is compatible with the humanoid's autonomous motion.

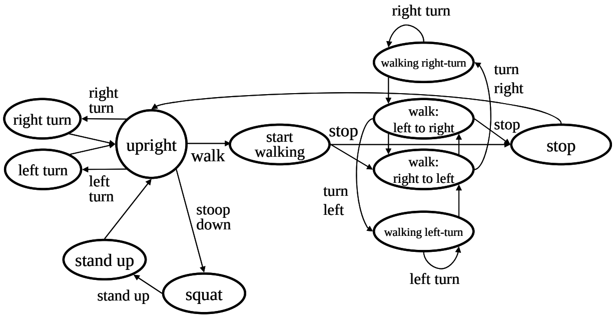


Figure 8: Lower body motion generation.

Figure 8 illustrates the designed state diagram which realises continuous and stable actions of the humanoid. It is designed according to the following demands.

1. Each transition needs to be the direction from "start motion" to "terminating motion".
2. Each transition is provided with pre-computed "non-falling movement".
3. While receiving the operator's new motion command, the humanoid remains at the current state, i.e., continues the current motion for a little while.
4. Then, a state transition occurs to the new state, i.e. the humanoid starts a new action.

We note here that the combination of items 3 and 4 is a key to avoiding the falling down.

3 Autonomous Mode

3.1 Autonomous Tasks

"Autonomous mode" stands for the action condition which allows the humanoid to act free from the human operator's motion. But, this does not mean that the humanoid moves at random. Once the humanoid finds a target object, it acts to complete a set of necessary tasks to complete designed jobs. The target object can be either physical or software-based. In this paper, the object is physical and can be identified through the cameras, i.e., the humanoid's eyes.

3.2 Design of Autonomous Tasks

The level of the autonomous tasks can be versatile depending on the equipped parts' sophistications. As was explained, the following items are equipped in this paper's humanoid:

a. Two legs to walk,
b. two arms and hands to specify direction,
c. two sets of fingers to grasp,
d. two CCD cameras as eyes, without automatic focusing.

For the design of the autonomous mode for the humanoid, the minimum, yet demanding, requirement is "never fall" which is related to item (a). Items (b) and (c) need to be used while satisfying "do not hurt itself or others". Item (d) does not create any forbidden tasks. By considering these properties, we set the following series of tasks as a change to the autonomous mode.

1. The humanoid walks (including straight, left and right turns) according to the human operation (see figures 2 and 8). While on this course, the humanoid stops for a short while and sends back the captured images to the operator through the network. Until the end of this phase, the humanoid is in the supervised mode.
2. The humanoid finds a specified physical object or a cue by its eyes. Then, the humanoid goes into the autonomous mode.
3. By using the 3D stereo matching of figure 6, the object's position measurement is performed.
4. By using the computed data, the humanoid tries to handle the physical object. Here, "never fall down" and "do not break anything" need to be obeyed.

In the experiment, the detection of a push button and handling it will be described as an example of this class of tasks.

4 Total Computational Intelligence: Joint Execution of Supervised and Autonomous Modes

4.1 Supervised Operation with Motion Recognition and Image Recognition

As was explained in section 1, the supervised mode by the human operator does not mean a mere master-slave operation. Even in the supervised mode, the motion recognition is utilised (see figures 1 and 5).

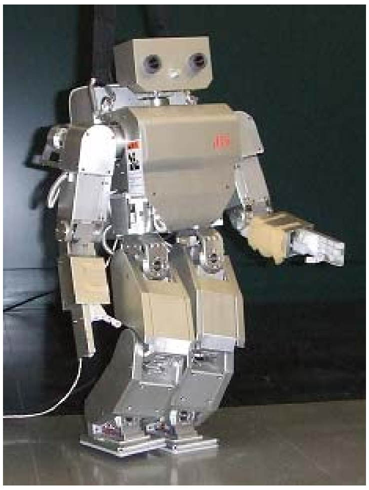


Figure 9: Supervised mode: walking.

After the calibration illustrated in figure 2, the supervised mode starts. Figure 9 is a scene in which a humanoid walks while mimicking the operator's gestures. The differences of freedom degrees, angle ranges, weights and dimensions are successfully absorbed by the motion recognition and the command generator. We note that the network exists between the human operator and the humanoid (see figure 1).

4.2 Autonomous Mode

The autonomous mode starts when the humanoid finds a push button within its view. Then, the humanoid autonomously adjusts its position by walking to get closer to the push button. The humanoid visually measures the button's 3D position (see figure 6). If the humanoid recognises that the button is within its reach, it tries to push the button. If the pushing is successful, the environment is changed, e.g., the light turns on. We note here that this is a fairly complex task for the humanoid for the following reasons.

a. The 3D stereo image recognition per se is a complex task for a humanoid.

b. The control of the finger position is also a difficult task because of its ill-conditioned nature and physical problem, e.g. (3).

c. Just touching the button causes no effect. The button needs to be pushed adequately.

d. Pushing the button too much causes the humanoid to fall down.

Therefore, the movement of the finger to press the button is not a single stream. The movement has intermediate steps for the adjustment. Then, the autonomous action "pushing the button" is completed.

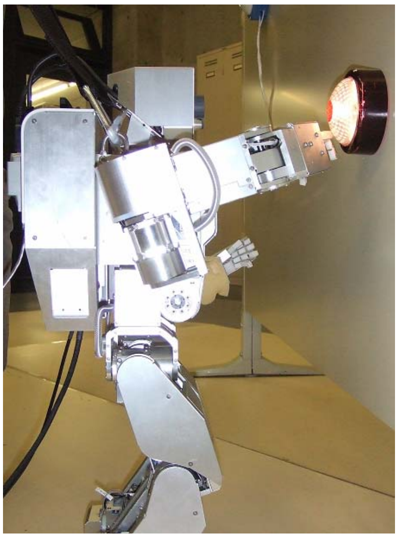


Figure 10: Autonomous mode: pushing a button without falling down.

Figure 10 illustrates the scene where the humanoid completes the pushing of the button. The rate of the overall success without re-adjustment is 60 % at the date of this paper's preparation. As was anticipated, this rate is higher than the same job in the supervised mode alone.

5 Concluding Remarks

In this paper, the human motion was directly utilised as a command generator for the biped and two-handed humanoid. The system is not a mere master-slave style. The process includes the phase of the gesture recognition whose abstraction absorbs the operator and humanoid's individuality. The human operator and the humanoid share the same view through the humanoid's eyes and the network communications. By such properties, the human operator was able to control the humanoid successfully.

This paper's humanoid is equipped with another important ability. It is the existence of the autonomous mode. Once the humanoid finds a cue object which is related to a task, a series of autonomous actions start.

This paper's humanoid control system includes the motion capturing, its recognition, the network, and the ability to act in the autonomous mode. This advancement will lead to the following future steps for developing further sophistications.

1. A hierarchical combination of multi-grained recognisers needs to be developed.

2. By utilising the network, the unification of the human operator, the network and the humanoid needs to be enhanced. This includes the cooperation of multiple humanoids. A version without remote operations exists [5]. Thus, the utilisation of the networking and the autonomous mode for multiple humanoids is an important next step.

3. Greater adaptation to real world environments needs to be developed. Offices, streets and hospitals would be welcome recipients.

6 Acknowledgements

The last author is grateful to his former students, Mr. Kazuya Tanikawa of IBM Japan and Mr. Ryusuke Okuhara of Canon, for their early contributions to this paper.

7 References

[1] R.R. Murphy, *Introduction to AI Robots*, MIT Press, Cambridge, MA (2000).

[2] Y. Matsuyama, S. Yoshinaga and H, Okuda, "Action transmission system and its method", *Patent Open*, 2006-146435, Japan (2006).

[3] Y. Matsuyama, Y. Yoshinaga, S. Okuda, K. Fukumoto, S. Nagatsuma, K. Tanikawa, H. Hakui, R. Okuhara and N. Katsumata "Towards the unification of human movement, animation and humanoid in the network", *Lecture Notes in Computer Science*, 331, pp 1135-1141 (2004).

[4] B. Klaus and P. Horn, *Robot Vision*, MIT Press, Cambridge, MA (1986).

[5] Y. Matsuyama, T. Shiga, T. Chikagawa, N. Takahashi and Y. Ueda, "Network communication strategies for cooperative physical agents", *Proceedings Asia-Pacific Symposium on Information and Telecommunication Technologies*, Yangon, Myanmar, pp 148-153 (2005).

Eye Fatigue Reducing System Mounted on a Remote-controlled Humanoid Robot

Keisuke Tsurumi, Nozomi Shirakura, Junichi Takeno,
Department of Science and Technology, Science and Robot Laboratory,
Meiji University, Kanagawa, Japan
ce66729@isc.meiji.ac.jp, ce55710@isc.meiji.ac.jp, takeno@cs.meiji.ac.jp

Abstract

We develop and research remote-controlled humanoid robots that assist human activities or take the place of humans in working environments by enabling humans to communicate with robots. Acquisition of visual information that is similar to information acquired directly by humans through a robot from a remote site with a low eye fatigue is an important interface between robots and humans. We will describe in this paper the construction of a stereoscopic remote vision system that reduces eye fatigue by reducing parallax on visualised objects as much as possible in a stereoscopic image as seen by an operator. To verify the usefulness of our proposed vision system, we conducted an experiment and obtained results indicating that the eye fatigue on the operator can be reduced. In a situation in which the robot is working near the system, the system is most effective at reducing the eye fatigue to improve work efficiency.

Keywords: humanoid robot, stereoscopic vision, eye fatigue

1 Introduction

In an inferior or dangerous environment for humans, robots are particularly necessary. We think that dangerous work for humans such as removing materials from mines and working in a nuclear reactor plant should be done by robots in the place of humans. Also since the required labour for humans is reduced to simpler tasks, humans and robots would need to effectively communicate with each other.

In recent years, the development of robots has been remarkable and many autonomous robots such as bipedal walking robots with very high operating accuracy have been developed. For effective communication and support in the same type of working environment, humanoid robots are appropriate. The reason is that the environment is made for human use and all tools (such as pencils and screwdrivers), vehicles (such as airplanes and automobiles), and buildings have appropriate shapes for humans to use. [1] To communicate and work with humans, robots must think and act for themselves. However, to give an autonomous robot artificial power to enable it to adapt to various conditions so that it can communicate and work with humans, more advanced research is required and it will take a long time. Also research on the capability of robots to work in all directions at a remote site based on the intentions of a human operator is important. [2] At this time, accurate transmission of the environment information around the robot to the remote operator is the most important. For the human operator, it is ideal to create a virtual reality space in which the operator senses intuitively through all five senses as if he or she were at the site.

2 Realisation of Stereoscopic Vision

In human eyes, images on the right and left retinas are slightly different. This difference makes a human sense depth and therefore realises stereoscopic vision. To create a stereoscopic vision system on a robot, two CCD cameras are mounted on the head of the robot and a vertical two-segment division frame sequence system and the liquid crystal shutter glasses are adopted (figure 1).

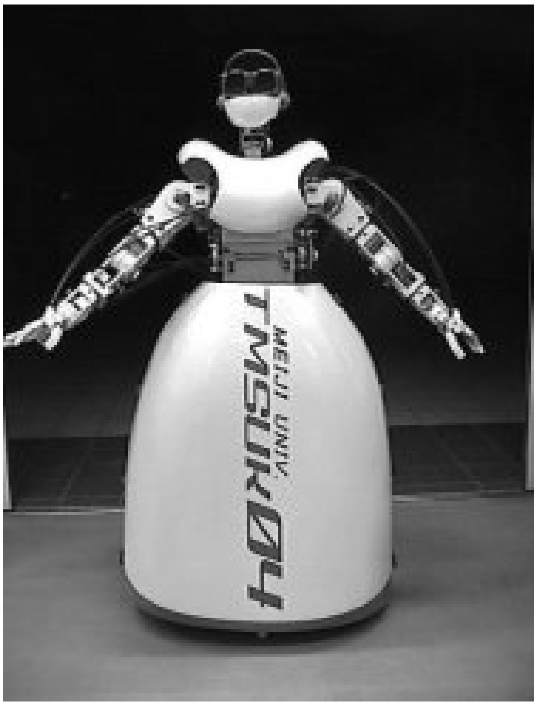


Figure 1: Humanoid robot (TMSUKIV-Visor).

The vertical two-segment division frame sequence system transmits the right and left camera images by compressing them vertically for the upper half and the lower half of the screen and combining them. Infrared signals are used to display the right and left camera images alternately at high speed on the display so that only the left camera image is seen by the left eye and only the right camera image is seen by the right eye. The shutters of the liquid crystal shutter glasses are closed alternately by linking them with the right and left camera images to achieve stereoscopic vision. Also, the vertical two-segment division frame sequence system is easy to control over the Internet and therefore is appropriate for stereoscopic vision at a remote site. [3]

In addition, we have constructed a vision system that is intended to reduce the eye fatigue on operators who view stereoscopic images acquired with this system.

3 Vision System

We think that reducing the eye fatigue on operators who view the stereoscopic images allows safer operation of the robots. If two cameras are used for stereoscopic vision, parallax always occurs to an object in the images produced by the two cameras. Human stereoscopic vision uses the parallax of the right and left eyes. When humans view a stereoscopic image on the display, larger parallax causes larger eye fatigue. Consequently, we have constructed a system that can reduce eye fatigue on the operators by eliminating the parallax on the object that they wish to observe closely.

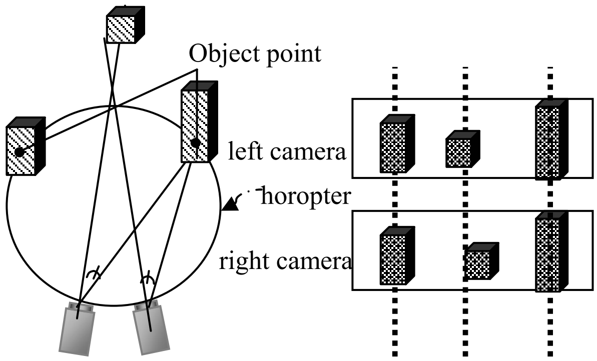
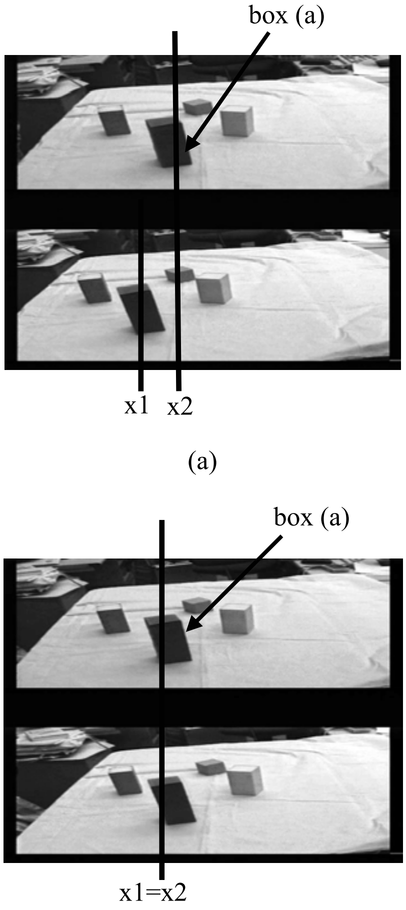


Figure 2: Object point.

The vision system that we have constructed can eliminate the parallax by adjusting automatically the angle of convergence of the cameras to set the object point to the object to be observed closely. The operator can acquire the images without parallax on the right and left camera images for the object to be observed closely by selecting the object in the stereoscopic image using a mouse. If two cameras are installed so that their optical axes intersect, some points that do not generate parallax on the object in the right and left camera images exist except the intersection point of the optical axes. These are called object points (figure 2). The object points are points on the circle called the horopter that pass the right and left cameras and their optical axes (cross points) [4]. In the next section, we explain the method of calculating the image matching and the motor control value to eliminate parallax on the object to be observed closely and the method of acquiring images without parallax.

3.1 Calculation of Image Matching and Motor Control Value

Figure 3 shows a display of the vertical two-segment division frame sequential system in which the images of the right and left cameras are compressed vertically into half size and displayed on the upper half and the lower half respectively. The stereoscopic images are created by enlarging the upper and lower images of figure 3 two times and displaying them alternately at high speed.



(a)

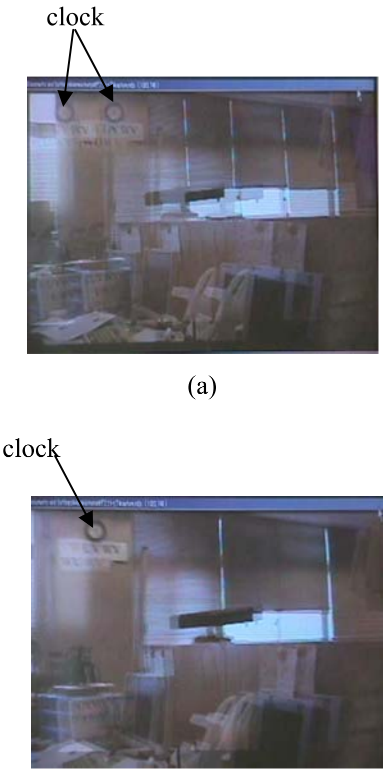
(b)

Figure 3: Image matching in vertical two-segment division images.

Suppose that an operator wants to observe closely a blue box in a stereoscopic image. Since the operator is viewing a stereoscopic image, the image of figure 3 is not actually displayed. If the operator selects the box (a) on the stereoscopic image display with a mouse, “x1” on the box (a) in the upper image is selected on the computer. (figure 3a) Based on the centre of the selected position around the 30-pixel-square image, a difference calculation is performed from the left end of the lower half image to find the position (on the x2 line) that generates the minimum difference of image. The motor control value is calculated by calculating the camera direction from |x1-x2|. This value is used to control the camera motor and shift the lower half image to the right (figure 3b). The upper and lower images of figure 3b are enlarged two times and displayed alternately at high speed to acquire the stereoscopic image without parallax for the object to be observed (box a).

3.2 Acquisition of Image without Parallax in Stereoscopic Image

Figure 4 shows a stereoscopic image that the operator views when operating the robot. To create this image, infrared signals are used and right and left camera images are displayed alternately at high speed so that only the left camera image is seen by the left eye and only the right camera image is seen by the right eye. The operator can view this stereoscopic vision by wearing the liquid crystal shutter glasses as described above.



(a)

(b)

Figure 4: Acquisition of image without parallax on object being viewed in stereoscopic image.

Suppose that the operator wants to observe closely the clock in the image of figure 4a. In this image, the clock is seen as a double image and therefore parallax occurs between the right and left camera images. Consequently, while observing the clock in this image, eye fatigue is generated and the operator's eyes will tire easily. Use of the vision system that we have constructed eliminates parallax between the right and left camera images and reduces eye fatigue.

The flow of this vision system is as follows.

1. The operator wears the liquid crystal shutter glasses and views the stereoscopic image with stereoscopic vision.

2. The operator selects an object (clock) to be observed closely in the stereoscopic image with a mouse.

3. Image matching is performed between the upper and lower images of the vertical two-segment division frames for the selected object and the motor control value is calculated.

4. The camera motors are controlled with the motor control value that has been calculated in 3 and a stereoscopic image without parallax for the object to be observed can be acquired (figure 4b).

4 System Configuration

This robot system consists of two main components: the robot body and the operation cockpit for operating the robot. The robot and the operation cockpit (figure 6) each contain a computer for the vision system. The operation cockpit contains a display that shows images from a remote site in stereoscopic vision. (figure 5).

First, the computer in the robot acquires the image from the stereoscopic vision cameras (figure 5a). The computer passes the acquired image through a field sequencer and a frame sequencer and transmits it to the computer in the operation cockpit over a WLAN using the vertical two-segment division frame sequential system (figure 5b). The computer in the operation cockpit receives the transmitted image and passes it through an infrared ray generator to display it on the stereoscopic display (figure 5c). By wearing the liquid crystal shutter glasses that synchronise with the infrared ray generator to view this image, the operator can acquire a stereoscopic image. The operator can operate the robot while viewing this image and can select an object to be observed closely in the stereoscopic image (figure 5d). When the object to be observed is selected on the display, the computer in the operation cockpit performs image matching and calculates the motor control value. The computer transmits the motor control value to the computer in the robot (figure 5e). The computer in the robot uses this motor control value to acquire an image without parallax for the selected object to be observed (figure 5f).

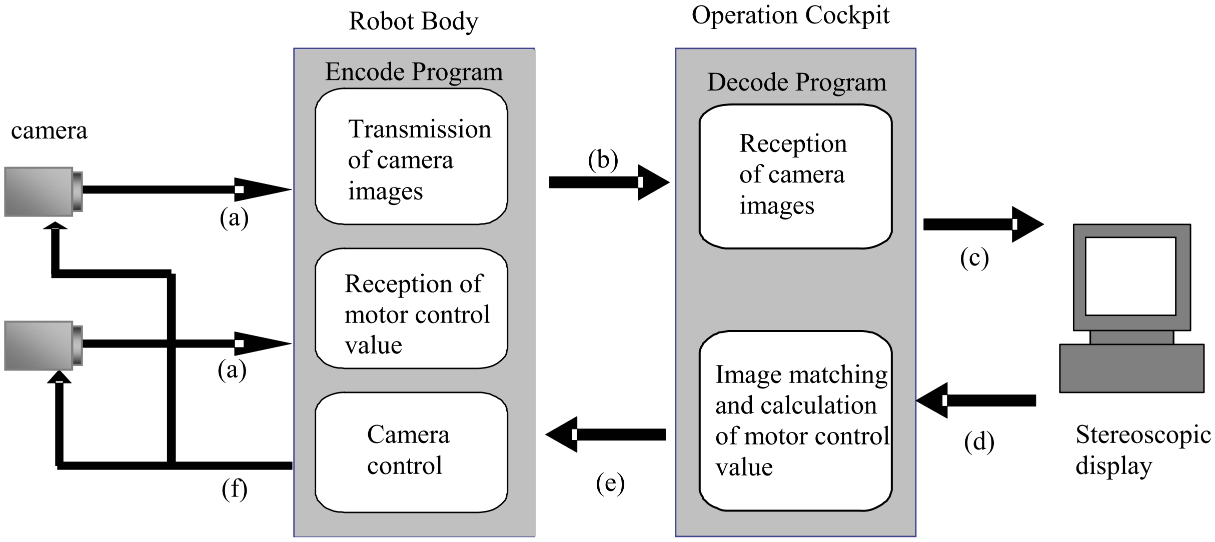

Robot Body
Operation Cockpit
Encode Program
Decode Program
camera
Transmission of camera images
Reception of camera images
Reception of motor control value
Image matching and calculation of motor control value
Camera control
Stereoscopic display
(a)
(a)
(b)
(c)
(d)
(e)
(f)

Figure 5: Robot system configuration.

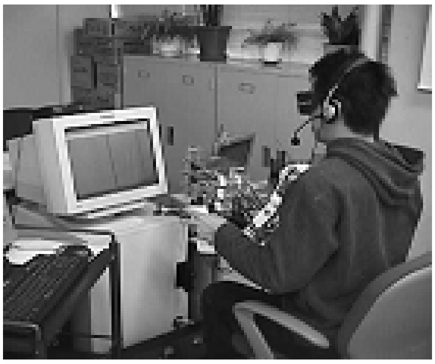


Figure 6: Operation cockpit.

5 Experiment

In this experiment, we prepared random characters on paper in different situations, and selected subjects to view the characters through the stereoscopic image.

To verify the effectiveness of the vision system we constructed, we conducted an experiment focusing on two points: the degree of character recognition and visibility.

5.1 Purpose and Outline of the Experiment

Since a stereoscopic image allows an object to be seen projected or retracted and therefore easily gives a sense of distance, it is very useful for working with a robot. However, when viewing a stereoscopic image the eyes get tired and it takes a long time to acquire stereoscopic vision. When looking at an object, the operator need not concentrate and therefore does not get so tired. However, when looking at characters, the situation is different. Since there are similar characters such as "V", "U" and "W", the operator must concentrate to recognise them properly and therefore the operator's eyes tire easily. Consequently, we verified the effectiveness of our vision system by displaying similar characters at random. We selected 20 subjects through a random process

5.2 Environment and Method of the Experiment

We selected three characters "U", "V", "W" that are similar and are not easily recognised. We prepared 10 characters on paper by combining these three characters at random such as "VVWWUVWVUV" and placed it at four locations: 0.3 m, 1 m, 3 m and 5 m from the cameras. Figure 7a has parallax and figure 7b has no parallax. In addition, we tested it in three next situations in each distance.

For each of the following three situations:

1. The distance of convergence was infinite (cameras were placed in parallel.)
2. The distance of convergence was 1 m.
3. The vision system was used.

We asked the subjects two questions:

1. Can you recognise the characters properly?
2. Are the characters easy to see?

For a (1) question, the subjects read it aloud to see character string such as "VVWWUVWVWV". We count the number of characters which we read aloud precisely.

For a (2) question, the subjects choose one of the following three characters for their evaluation.

Evaluation of the question on visibility:

○: Easy to see (+1)

□: Not easy to see but somehow readable (-1)

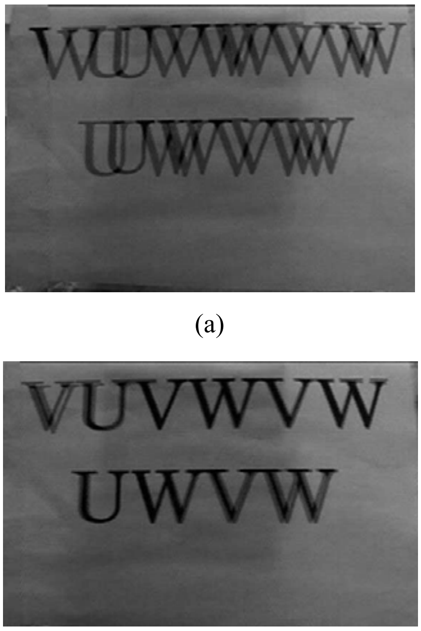
×: Cannot read at all (-1)

Number of subjects: 20

Character font sizes:

When the character sizes are the same, characters placed at a further distance are naturally more difficult to see. We adjusted the sizes while viewing the image so that all the characters were easy to see on a similar level. The character font sizes that were actually used are shown below.

0.3 m: 36 points
1 m: 72 points
3 m: 150 points
5 m: 350 points



(a)

(b)

Figure 7: Vanity of characters.

5.3 Discussion of Results

We compiled table 1 and figure 8 about the number of characters which were able to recognise precisely. Figure 8 is the graph of table 1. Each value took average of 20 subjects. Likewise figure 9 is the graph of table 2. Evaluation value subtracts the number of “□”or “×” from the number of “○”.The graph in figure 8 shows that more characters could be recognised correctly when the conditions were in this order: use of vision system, distance of convergence at 1 m, and infinite distance of convergence. Also when the distance of convergence was 1 m, the number of correct answers was less at 0.3 m compared with the use of the vision system but was not very different at other positions. However, the evaluation of visibility shown in the graph in figure 9 indicates a large difference between the condition with no parallax for characters at a distance of 1 m and use of the vision system, except at the position of 1 m. (Since no parallax exists at 1 m in both cases, it is to be expected that no difference would be found.) The infinite distance of convergence facilitates reading of characters to some extent but gives considerable eye fatigue. Use of the vision system gives visibility and correct recognition. Although the difference is small, the visibility was the highest at 1 m and decreased gradually at farther distances.

Table 1: Number of correct answers.

	Distance to object			
Distance of convergence	0.3m	1m	3m	5m
(1) Infinite	8.4	7.9	8.2	8.6
(2) 1m	8.2	9.5	9.4	9.5
(3) Vision system	9.5	9.5	9.5	9.6

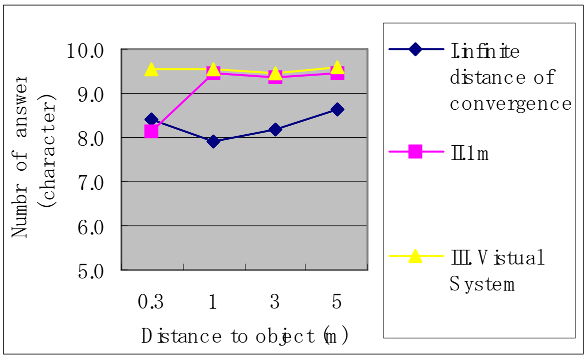


Figure 8: Number of correct answers.

Table 2: Evaluation of visibility.

	Distance to object			
Distance of convergence	0.3m	1m	3m	5m
(1) Infinite	-14	-12	0	8
(2) 1m	-16	20	6	10
(3) Vision system	14	20	18	16

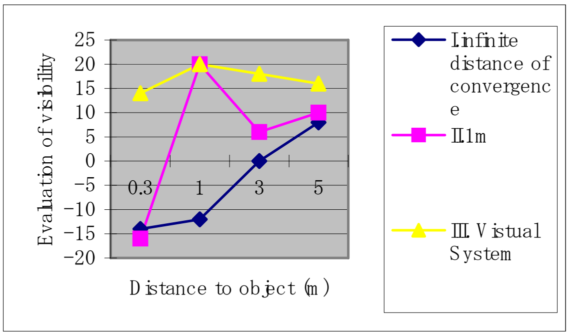


Figure 9: Evaluation of visibility.

Although this result may be caused simply by the difference in character sizes, we also think that since the distance between the subjects and the display was 1 m, the 1 m distance between the cameras and the object being viewed was more natural. Also, the result indicates clearly that characters placed at a farther distance are easier to see with parallax than characters at a closer distance without parallax. This fact is proved by the graph in figure 9 which shows that characters at the position of 5 m were easier to see than characters at the position of 0.3m when the distance of convergence was set to 1 m.

From the results of this experiment, we can note the significance of the following three points:

1. If the distance of convergence is set to infinite, since the parallax is basically larger at a shorter distance from the cameras, both the recognition of characters and their visibility deteriorated. At a distance of approximately 0.3 m, however, since the parallax is too large, each character was seen as two completely separated characters. Consequently, the recognition accuracy was high but visual fatigue was also high.

2. Objects at distances farther than the distance of convergence were easier to see than those at closer distances.

3. There is a possibility that optimal visibility is obtained at a distance between the cameras and the object being viewed that approximates the distance between the subjects and the display.

6 Conclusion

This paper presented the construction of a vision system that is intended to reduce the eye fatigue on an operator when acquiring visual information through a robot from a remote site that is similar to information acquired directly by humans. We have explained our realisation of a stereoscopic vision system to create visual information similar to information acquired directly by humans, the so-called three-dimensional space, and have explained our method for reducing the eye fatigue on the operator when viewing stereoscopic images. Also, we have described an experiment using 20 subjects for a remote-controlled robot on which this vision system is mounted and we have verified the effectiveness of this vision system. If our vision system is used to eliminate parallax, a great advantage is obtained particularly in visibility. Consequently, eye fatigue can be largely reduced. Also concerning the recognition of characters, the rate of correct answers was the highest using our vision system although the difference was approximately one character at the most. Since the misreading of even one character could lead to a great difference in meaning and possible misunderstanding, the advantage of using our vision system is also seen in our results on character recognition. We summarise the advantages of using our vision system to operate a robot as follows:

1. Since the eye fatigue on the operator's can be reduced by eliminating parallax for easy-to-see stereoscopic vision, the operator can operate the robot in a safer manner.

2. Since large parallax occurs for objects at farther distances, this system is the most effective when working near the robot.

In the future, we will create a system that can select an object using line of sight instead of a mouse to eliminate parallax on the object being viewed. We intend to achieve cursor operation through eyeball movement and mouse click operation through the blinking of the eyes. This would enable the operator to continue using both hands at the operation cockpit and therefore operate the robot more smoothly.

7 Reference

[1] K. Akaihi, T. Isozumi, M. Hirata, S. Ohta and M. Ishizaki, "Development of the humanoid robot HRP-2", *Kawata-Gihoh*, 23, pp20-25(2004).

[2] S. Tachi, "Real-time remote robotics: toward networked telexistence, *IEEE Computer Graphics and Application*, 18(6), pp 6-9 (1998).

[3] H. Hoshino, K. Suzuki, T. Nishiyama and K. Sawada, "Development of tele-existence visual display system for humanoid robot control," *Transactions of the Virtual Reality Society of Japan*, 7(1), pp 39-47 (2002).

[4] M. Ohkura, Y. Yanagida, T. Maeda and S. Tachi, "Measurement of horopter and alleys in auditory space", *Proceedings 2002 International Conference on Auditory Display*, Kyoto, Japan, pp 333-341 (2002).

[5] Y. Miyazawa and J. Takeno, "Humanoid robotics", *Proceedings 21st Annual Conference of the Robotics Society of Japan*, p 115 (2003).

[6] M. Morita, J. Takeno, "New visual system of remote control type humanoid robot" *Proceedings 8th World Multiconference on Systemics, Cybernetics and Informatics, The International Institute of Informatics and Systemics (IIIS)*, Florida, 3, pp 71-76 (2004).

[7] N. Shirakura, M. Morita and J. Takeno, "Development of a human interface for remote-controlled robots using an eye-tracking system", *International Conference on Mechatronics and Automation*, Niagara Falls, Canada, pp 351-356 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Contribution to the Development of Contact Interaction-Based Humanoid Robot Navigation System: Application of an Optical Three-Axis Tactile Sensor

Hanafiah Yussof[1], Masahiro Ohka[1], Hiroaki Kobayashi[2], Jumpei Takata[2],
Mitsuhiro Yamano[3] and Yasuo Nasu[3]
[1]Graduate School of Information Science, Nagoya University, Nagoya, Japan
[2]Graduate School of Engineering, Nagoya University, Nagoya, Japan
[3]Faculty of Engineering, Yamagata University, Yonezawa, Japan
hanafiah@nuem.nagoya-u.ac.jp

Abstract

Autonomous navigation in walking robots requires that three main tasks be solved: self-localisation, obstacle avoidance, and object handling. This report presents a development and application of an optical three-axis tactile sensor mounted on a robotic finger to perform object handling in a humanoid robot navigation system. Previously in this research, we proposed a basic humanoid robot navigation system called the “groping locomotion method” for a 21-DOF humanoid robot, which is capable of defining self-localisation and obstacle avoidance. Recently, with the aim to determining physical properties and events through contact during object handling, we have been developing a novel optical three-axis tactile sensor capable of acquiring normal and shearing force. The tactile sensor system is combined with 3-DOF robot finger system where the tactile sensor in mounted on the fingertip. Experiments were conducted using soft, hard, and spherical objects to evaluate the sensor’s performance. Experimental results reveal that the proposed optical three-axis tactile sensor system is capable of recognising contact events and has the potential for application to humanoid robot hands for object handling purposes.

Keywords: humanoid robot navigation, contact interaction, optical three-axis tactile sensor, robotic finger

1 Introduction

A humanoid robot is a robot with an overall appearance based on that of the human body [1]. Humanoid robots are created to imitate some of the physical and mental tasks that humans undergo daily. The goal is that one day it will be able to both understand human intelligence and reason and act like humans. If humanoids are able to do so, they could eventually coexist and work alongside humans and could act as proxies for humans to do dangerous or dirty work that would not be done by humans if there is a choice, hence providing humans with more safety, freedom and time. Eventually, the “working coexistence” of humans and robots sharing common workspaces will impose on robots with their mechanical-control structure at least two classes of tasks: motion in a specific environment with obstacles, and manipulating various objects from the human environment [2]. As far as this working coexistence is concerned, a suitable navigation system combining design, sensing elements, path planning and control embedded in a single integrated system is necessary to guide humanoid robots’ activities so that they can work and share workspaces with humans.

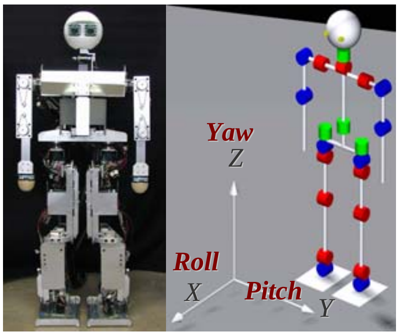


Figure 1: *Bonten-Maru II* and configuration of DOFs.

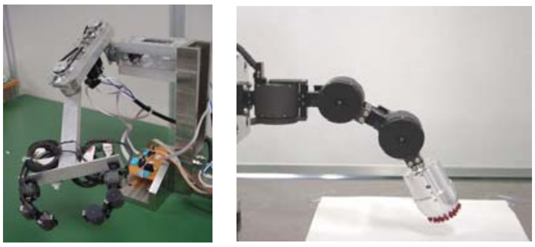


Figure 2: Humanoid robot arm with fingers and the finger mounted with optical three-axis tactile sensor.

To date, research on humanoid robots has arrived at a point where the construction and stabilisation of this type of robot seems to be no longer the key issue. At this stage, it is novel practical applications such as autonomous navigation [3], telerobotics and development of intelligent tactile sensor devices [4] that are being studied and attracting great interest. The main purpose of this project is to develop an autonomous contact interaction-based navigation system for bipedal humanoid robots. The system comprises three important elements: self-localisation, obstacle avoidance, and object handling. We have previously proposed a basic humanoid robot navigation system called “groping locomotion method” capable of defining self-localisation and obstacle avoidance. This system is based on contact interaction with the aim of creating suitable algorithms for humanoid robots to effectively operate in real environments. In this report, we focus on development of the object handling. Here we present the development of an optical three-axis tactile sensor capable of acquiring normal and shearing force, with the aim of installing it on a real humanoid robot arm. In addition, we explain the tactile sensor system’s structure and principle. Furthermore, this report includes an experimental evaluation of the performance of an integrated system comprising the tactile sensor and robotic finger to recognise and grip hard object, soft objects and also a spherical object.

In this project, we used the previously developed humanoid robot named *Bonten-Maru II*, shown in figure 1. To define the tactile sensors’ performance, as well to ensure safety of the humanoid robot and tactile sensor, we conduct an experiment and evaluation of the optical three-axis tactile sensor using a newly developed prototype multi-fingered 3-DOF humanoid robot arm. The arm was designed to replicate *Bonten-Maru II*’s arm. However, the arm’s control system has been refined to comply with the multi-finger control system and the tactile sensor system. Figure 2 shows the newly developed humanoid robot arm with fingers and the finger mounted with the optical three-axis tactile sensor.

2 Motivation

It is inevitable that the application of humanoid robots in the same workspace as humans will result in direct physical-contact interaction. This will require the robot to have additional sensory abilities. Besides sensor systems that help the robot to structure their environment, like cameras, radar sensors, etc., a system on the robot’s surface is needed that enables to detect physical contact with its environment. A tactile sensor system is essential as a sensory device to support the robot control system [4-6]. This tactile sensor is capable of sensing normal force, shearing force, and slippage, thus offering exciting possibilities for application in the field of robotics for determining object shape, texture, hardness, etc. We believe that the demand for tactile sensing devices will grow in parallel with rapid progress in robotics research and development.

2.1 Humanoid Robot *Bonten-Maru II*

We have previously developed a 21-DOF (degrees-of-freedom), 1.25-m tall, 32.5-kg anthropomorphic prototype humanoid robot called *Bonten-Maru II*. The robot has a total of 21 DOFs: six for each leg, three for each arm, one for the waist, and two for the head. The high number of DOFs and configuration of joints that closely resemble those of humans provide *Bonten-Maru II* with the possibility of realising complex trajectories to attain human-like motion. Each joint is driven by a DC servomotor with a rotary encoder and a harmonic drive-reduction system, and is controlled by a PC with the Linux OS. *Bonten-Maru II*’s structure design and control system are used in experiments and evaluation of this research.

2.2 Relevancy of Contact Interaction

As mentioned above, the application of humanoid robots in the same workspace as humans will inevitably result in contact interaction. Our survey of journals and technical papers resulted in finding only a very small number of works reporting the application of a contact interaction method to navigate humanoid robots in real environments. Most reports on navigation of walking robots are related to perception-guided navigation [3], dealing particularly with visual-based navigation that has been a relevant topic for decades. In visual-based navigation, which is classified as non-contact interaction, despite the rapid growth in visual sensor technology and image processing, identification accuracy problems due to approximate data obtained by the visual sensor and interruption of environment factors such as darkness, smoke, dust, etc. seem to reduce the robots performances in real environments. Meanwhile, contact interaction offers better options for humanoid robots to accurately gauge the structure of their environment, making it easier for them to perform tasks that required human-like handling. For example, they may perform object manipulation in assembly and disassembly processes, rehabilitation, etc. Therefore, contact interaction is a relevant topic in research on humanoids. Indeed, contact interaction is a fundamental feature of any physical manipulation system and the philosophy to establish a working coexistence between human and robot.

2.3 Sensors and Tactile Sensing

A sensor is a device that measures some attribute of the environment. Being one of the three primitives of robotics, besides planning and control, sensing plays an important role in robotic paradigms. These paradigms are supposed to reflect the five main sensing modalities in humans; sight, sound, smell, taste and touch. Tactile sensing is the process of determining physical properties and events through contact with objects in the world. While a great deal of research has been applied to the development of sensors especially visual and auditory sensors, comparatively little progress has been made with regards to sensors translating the sense of touch [6]. Consequently, a tactile sensor offers exciting possibilities for use in mechatronics devices and measuring instruments in many fields of science and engineering. Robotics is one of the application areas that have generated the most interest.

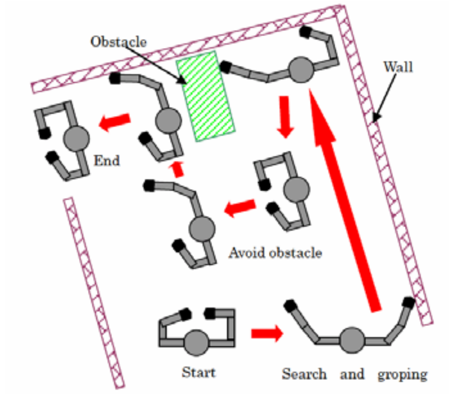


Figure 3: Path planning for humanoid navigation.

3 Contact Interaction-Based Humanoid Robot Navigation

Autonomous navigation of walking robots requires that three main tasks be solved: self-localisation, obstacle avoidance, and object handling [3]. In this research project, we have previously proposed a basic humanoid robot navigation system called "groping locomotion" [7-9] on the anthropomorphic 21-DOF humanoid robot *Bonten-Maru II*.

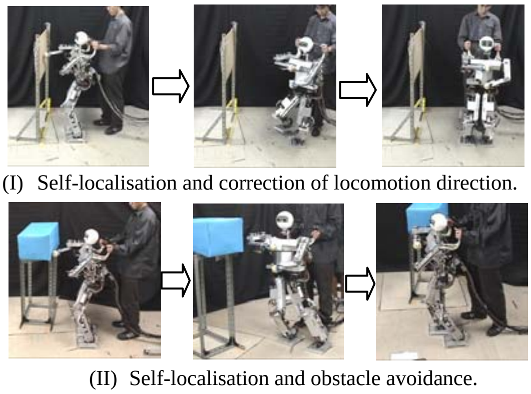
3.1 Groping Locomotion Method

The groping locomotion method consists of self-localisation for the humanoid robot to recognise its surroundings by touching and groping a wall's surface to obtain the wall's orientation. The robot then responds to its surroundings by correcting its position and orientation. During groping locomotion, however, the existence of obstacles in the correction area creates the possibility of collisions. Hence, we proposed an autonomous method of obstacle avoidance that operates in conjunction with groping locomotion. Six-axis force sensors were attached to both of the humanoid's arms as end-effectors that directly touch objects and provide force data which are subsequently converted to position data by the robot's control system to recognise its surroundings.

Figure 3 displays path planning in the groping locomotion. The research methodology applies a multi-tasking algorithm in the robot's control system, consisting of four important processes: searching for a wall, groping a wall's surface, correction of the robot's position and orientation, and obstacle avoidance. The algorithm is based on formulations for solving kinematics problems, interpolation of the end-effector, and force-position control. Using the proposed algorithm, we managed to demonstrate autonomous locomotion of the humanoid robot in experiments, as shown in figure 4.

3.2 3-DOF Robot Finger

In future, the newly developed 3-DOF fingers with the mounted optical three-axis tactile sensor will replace the six-axis force sensor on the humanoid arm for object handling purposes. The humanoid's finger comprises of three micro-actuators (YR-KA01-A000, Yasukawa) and connects to a PC via a motor driver and motor control board. The PC is installed with the Windows OS, the image analysis software Cosmos32 and a Visual C++ compiler. Figure 5 displays the integrated control system structure of the 3-DOF robotic finger and optical three-axis tactile sensor used in the experiment presented in this report.



(I) Self-localisation and correction of locomotion direction.

(II) Self-localisation and obstacle avoidance.

Figure 4: Photographs of robot's locomotion.

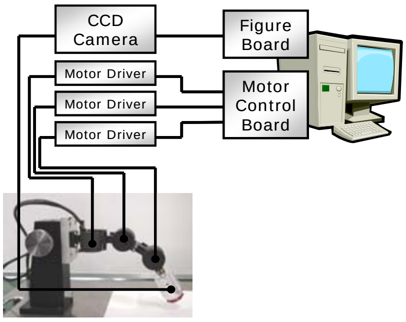


Figure 5: Layout structure of the humanoid's finger with the optical three-axis tactile sensor.

4 Optical Three-Axis Tactile Sensor

In this research, with the aim of establishing object handling ability in a humanoid robot navigation system, we have developed a novel optical three-axis tactile sensor capable of acquiring normal and shearing force. This sensor is designed to be mounted on a humanoid robot's fingers [10].

4.1 Tactile Sensor Structure

The structure of the optical three-axis tactile sensor used here consists of an acrylic hemispherical dome, an array of silicon rubber sensing elements, a light source, an optical fibre-scope, and a CCD camera. Figure 6 shows a diagram of the tactile sensor's structure. The silicone rubber sensing element comprises one columnar feeler and eight conical feelers, as shown in figure 7. The eight conical feelers remain in contact with the acrylic surface while the tip of the columnar feeler touches an object. The sensing elements are arranged on the hemispherical acrylic dome in a concentric configuration with 41 sub-regions as shown in figure 8, which also displays the structure of the fingertip with the sensing elements.

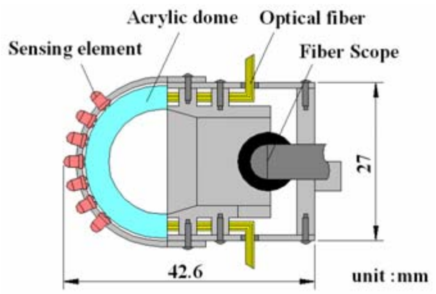


Figure 6: Structure of hemispherical optical three-axis tactile sensor.

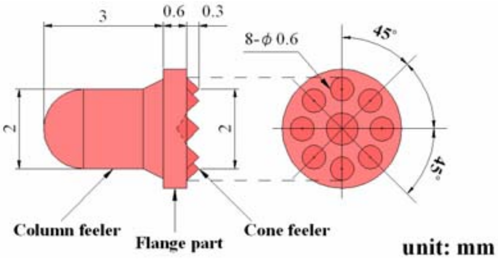


Figure 7: Sensing element.

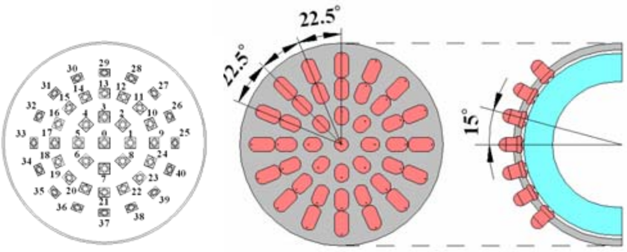


Figure 8: Arrangement of sensing elements on fingertip.

4.2 Sensing Principle

The optical three-axis tactile sensor is based on the principle of an optical waveguide-type tactile sensor. Figure 9 shows the sensing principle of the optical three-axis tactile sensor system. The light emitted from the light source is directed towards the edge of the hemispherical acrylic dome through optical fibres. When an object contacts the columnar feelers, resulting in contact pressure, the feelers collapse. At the points where the conical feelers collapse, light is diffusely reflected out of the reverse surface of the acrylic surface because the rubber has a higher reflective index. The contact phenomena consisting of bright spots caused by the feelers' collapse are observed as image data, as shown in figure 10, which are retrieved by the optical fibre-scope connected to the CCD camera and are transmitted to the computer.

In this situation, the normal force of F_x, F_y and F_z values are calculated using the integrated grey-scale value G, while shearing force is based on the horizontal centroid displacement. The displacement of the grey-scale distribution u is defined in equation (1), where i and j are orthogonal base vectors of the X and Y axes of a Cartesian coordinate, respectively. Each force component is defined as shown in equation (2).

$$\boldsymbol{u} = u_x \boldsymbol{i} + u_y \boldsymbol{j} \tag{1}$$

$$F_x = f(u_x),\ \ F_y = f(u_y),\ \ F_z = g(G) \tag{2}$$

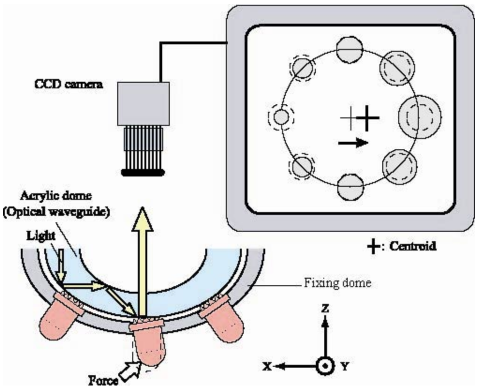


Figure 9: Principle of optical three-axis tactile sensor system.

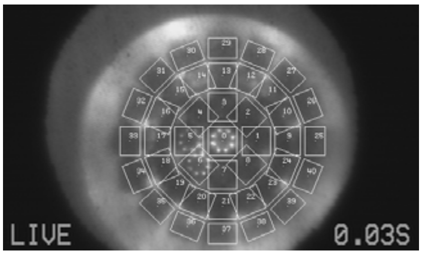


Figure 10: CCD camera-captured image of contact phenomenon in the optical three-axis tactile sensor.

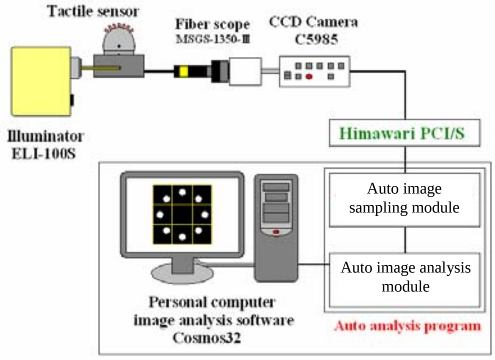


Figure 11: Layout of optical three-axis tactile sensor system structure.

Figure 11 shows the layout of the tactile sensor system. Image data acquired by the CCD camera are divided into 41 sub-regions as indicated in figure 8. The dividing procedure, digital filtering, integrated grey-scale value and centroid displacement are controlled on the PC using auto analysis programme applying the image analysis software Cosmos32.

5 Experiments and Results

In this section, we conduct experiments to evaluate the performance of the optical three-axis tactile sensor system using the 3-DOF robot finger. Unfortunately, the parameters of the human hand and fingers that are involved in sensing the hardness and/or softness of an object have not been fully researched. However, in this experiment we specified the force parameter values to perform force-position control. The minimum force parameter is specified at *0.3~0.5 N* (threshold 1), with the maximum side being *1.4~1.5 N* (threshold 2). Meanwhile *2 N* is fixed as the critical limit. To simplify the analysis, we only consider sensing element number *0* (refer figure 8).

5.1 Experiment I: Recognise and Grip Hard and Soft Objects

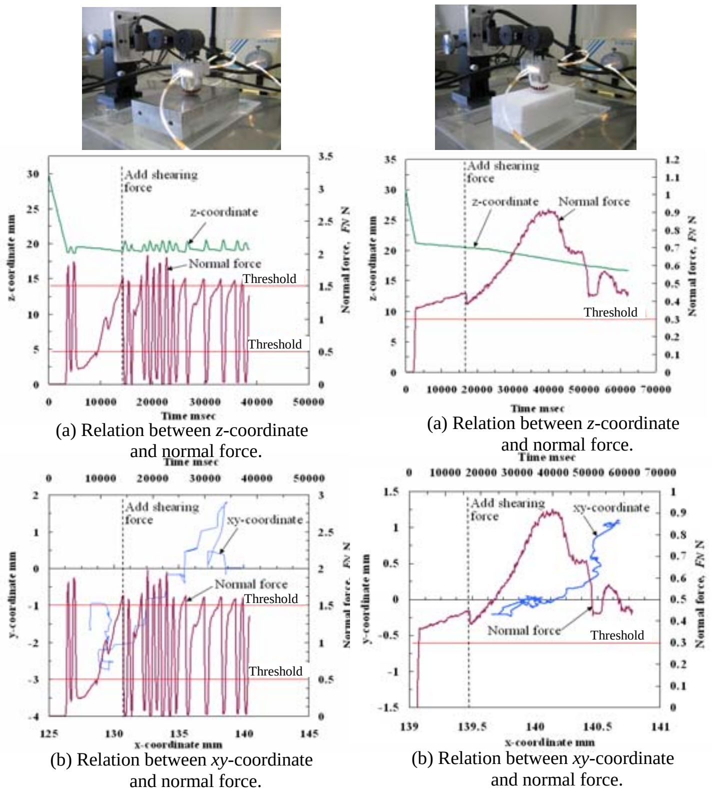
The ability to sense hardness and/or softness will be particularly important in future applications of the robotic hand. Therefore, we have conducted this set of experiment in order to recognise and grip hard and soft object. An aluminium block represents a hard object, and styrofoam represents a soft object. Figure 12 contains graphs that define the relation of the finger's trajectory at the force applied to the sensing element, normal force, and fingertip position data for each experiment. Here, when the tactile sensing element touches the object, normal and shearing forces are detected simultaneously and feedback to the finger's control system. The finger responds to the applied shearing force by generating trajectory at the *XY*-axis to define a stable gripping position. Meanwhile, the finger movement at the *Z*-axis is to adjust the applied normal force to grip the object.

Regarding the aluminium block, some disturbance to the normal force was observed at an early stage as shown in the graph. This phenomenon is the result of the control system adjusting the touching force and the finger is searching for a suitable gripping position. In this situation, due to the hard surface of aluminium block, the force detected by the sensor element is high and exceeds threshold 2. However, the finger feedback system controls the finger movement in order to search for a stable gripping force and position. Finally, the grip force was maintained within the specified force parameter values (between thresholds 1 and 2). Meanwhile, in the case of styrofoam, because of the surface's softness, the sensor element tended to gnaw into it resulting in a small detected shearing force. Therefore, the movement of finger at the *XY*-axes for styrofoam is very small as shown in figure 12(b). The sensor element is mainly detecting normal force, where finger movement at the *Z*-axis can be observed in figure 12(a). However, as the graph indicates, the tactile sensor managed to recognise touch with the soft object where normal force is detected. Finally, the finger movement stopped when the stable grip force was defined.

Furthermore, when the stability of the grip force and the position were satisfactory, the finger remained at that position to hold the object. At this point, we purposely added shearing force from the *X*-axis direction to both objects to define the proposed system's response against slippage. According to the graphs, the finger grip force increased drastically, at which point the tactile sensor detected shearing force. The finger feedback control system responded by adjusting the finger position and the applied force again to define a stable gripping position and force. This phenomenon can be observed in the figure 12.

5.2 Experiment II: Recognise and Grip a Spherical Object

In real-time applications, robots might have to comply



(a) Relation between *z*-coordinate and normal force.

(b) Relation between *xy*-coordinate and normal force.

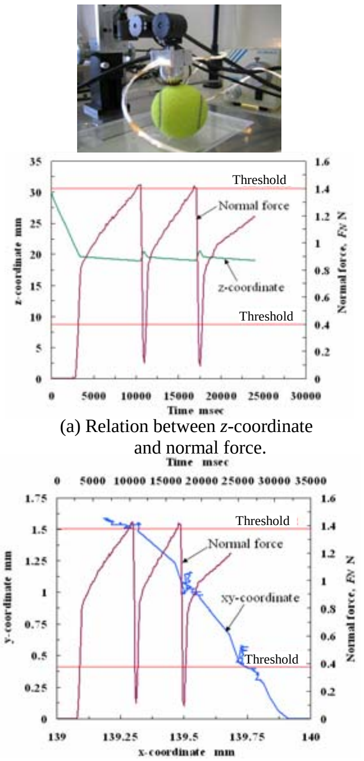
(i) Aluminium block.

(a) Relation between *z*-coordinate and normal force.

(b) Relation between *xy*-coordinate and normal force.

(ii) Styrofoam.

Figure 12: Grip movement of the robot finger on an aluminium block and styrofoam.



(a) Relation between *z*-coordinate and normal force.

(b) Relation between *xy*-coordinate and normal force.

Figure 13: Grip movement of the robot finger on tennis ball.

with objects of various shapes. It is therefore necessary for them to recognise the shape of an object and generate a suitable trajectory to grip it. In this experiment, we used a tennis ball to evaluate the performance of the proposed tactile sensor and finger system at recognising a spherical object.

In the experiment, the initial touch point was shifted slightly from the peak point of the tennis ball so that the movement characteristic of the robot finger against the sphere's shape surface could be evaluated. Referring to figure 13, when the touch force increased, the rotation moment acted on the tennis ball, causing the ball to try to rotate. The shearing force increased simultaneously and when the applied normal force exceeded the threshold 2, the ball started to slip out. Hence, the finger corrected its position to move nearer to the object's surface and adjust the gripping force. This process was repeated, finally resulting in a change to the finger orientation in order to comply with the spherical surface shape stopping when the finger reached a stable gripping position.

6 Conclusion: Potential Application of Optical Three-Axis Tactile Sensor to Humanoid Robot Hand

Application of tactile sensor in humanoid robot is still under development. Although current robot hands are equipped with force sensors to detect contact force, they do not make use of tactile sensors capable of detecting an object's hardness and/or softness, nor can they recognise the shape that they grip. For a robot hand to grip an object without causing damage to it, or otherwise damaging the sensor itself, it is important to employ sensors that can adjust the gripping power.

The optical three-axis tactile sensor system presented in this research is capable of clearly acquiring normal force and shearing force. Furthermore, it also can detect force at the *XYZ*-axes of a Cartesian coordinate. These characteristics have given the proposed tactile sensor the possibility of recognising the hardness and /or softness of an object. Experiments conducted in section *5.1* revealed the ability to recognise and grip hard and soft objects. Next, according to experimental results in section *5.2*, the proposed system also capable of recognising spherical shape and then adjusts the finger's gripping position and force it to hold the object.

Our overall conclusion from both experimental results revealed that the optical three-axis tactile sensor system, combined with the newly developed 3-DOF robot finger, can be readily applied to a robot's hand, and that the robot will be capable of controlling its gripping power by using the feedback algorithm of this embedded system.

Future work will involve further development of the contact-based humanoid robot navigation system project, applying the integrated system comprising the optical three-axis tactile sensor and robot finger. It is anticipated that using this novel tactile sensor system technology will bring forward the evolution of human and humanoid robots working together in real life.

7 Acknowledgements

A part of this research was supported by fiscal 2006 grants from the Japan Ministry of Education, Culture, Sports, Science and Technology (Grant-in-Aid for Scientific Research in Exploratory Research, No. 18656079).

8 References

[1] K. Hirai, M. Hirose, Y. Haikawa and T. Takenaka, "The development of Honda humanoid robot", *Proceedings of ICRA'98*, Leuven, Belgium, pp 1321-1326 (1998).

[2] M. Vukobratovic, B. Borovac and K. Babkovic, "Contribution to the study of anthropomorphism of humanoid robots", *Journal of Humanoids Robotics*, 2(3), pp 361-387 (2005).

[3] A. Clerentin, "Self localization: a new uncertainty propagation architecture", *Journal of Robotics and Autonomous Systems*, 51(2-3), pp 151-166 (2005).

[4] S. Omata, Y. Murayama and C.E. Constantinou, "Real time robotic tactile sensor system for determination of the physical properties of biomaterials", *Journal of Sensors and Actuators A*, 112(2-3), pp 278-285 (2004).

[5] O. Kerpa, K. Weiss and H. Worn, "Development of a flexible tactile sensor system for a humanoid robot", *Proceedings IROS2003*, Las Vegas (2003).

[6] M.H. Lee and H.R. Nicholls, "Tactile sensing for mechatronics: a state of the art survey", *Journal Mechatronics*, 9(1), pp 1-31 (1999).

[7] Y. Hanafiah, M. Yamano, Y. Nasu and M. Ohka, "Obstacle avoidance in groping locomotion of a humanoid robot", *Journal of Advanced Robotic Systems*, 2(3), pp 251-258 (2005).

[8] Y. Hanafiah, M. Yamano, Y. Nasu and M. Ohka, "Trajectory generation in groping locomotion of a 21-DOF humanoid robot", *Proceedings ICMT2005*, Kuala Lumpur, Malaysia, (2005).

[9] Y. Hanafiah, M. Yamano, Y. Nasu and M. Ohka, "Analysis of correction of humanoid robot locomotion direction in groping locomotion method", *Proceedings ATCI2005 / CISAR200*5, Kuala Lumpur, Malaysia, (2005).

[10] M. Ohka, H. Kobayashi and Y. Mitsuya, "Sensing precision of an optical three-axis tactile sensor for a robotic finger", *Proceedings RO-MAN 2006*, Hatfield, UK, pp 220-225 (2006).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

DAODAN:
An Affordable Research Platform for Humanoid Robotics

Jacky Baltes and John Anderson
Department of Computer Science,
University of Manitoba, Winnipeg, Canada
jacky@cs.umanitoba.ca, andersj@cs.umanitoba.ca

Abstract

This paper describes our latest humanoid robot DAODAN. The research goal for DAODAN is the development of an affordable research platform for humanoid robotics research with capabilities similar to humans. DAODAN is our own design that uses the Megarobotics AI-1001 motors as actuators. Onboard processing is provided by a Nokia 3230 mobile phone. This mobile phone has a built-in fixed camera as well as bluetooth wireless connectivity. An Analog Devices 3 axis accelerometer is added to the serial bus that is used to control the actuators and provides balancing information for the robot.

Keywords: humanoid robotics

1 Introduction

Humanoid robots have always inspired the imagination of robotics researchers as well as the general public. Up until 2000, the design and construction of a humanoid robot was very expensive and limited to a few well funded research labs and companies (e.g. Honda Asimov, Fujitsu HOAP). Starting in about 2001 advances in material sciences, motors, batteries, sensors, and the continuing increase in processing power available to embedded systems developers has led to a new generation of affordable small humanoid robots (some examples include: Pino [1], Manus I [2], Tao-Pie-Pie [3], Roboerectus [4], and Hansa Ram [5]).

These robots cost in the range from $1000.00 to $20,000 USD. Many hobbyists have built their own humanoid robots, especially in Asia.

The creation of these humanoid robots also coincided with an increased interest in several high profile research oriented international robotics competitions (e.g. RoboCup [6] and FIRA [7]). The researchers chose robotic soccer as a challenge problem for the academic fields of artificial intelligence and robotics. Robotic soccer requires a large amount of intelligence at various levels of abstraction (e.g. offensive vs defensive strategy, role assignment, path planning, localisation, computer vision, motion control). Robotic soccer is a dynamic real-time environment with multiple agents and active opponents that try to prevent the robot from achieving its goal. These competitions allowed researchers to compare their results to others in a real-world environment. It also meant that robustness, flexibility, and adaptability became more important since these robots had to perform for extended periods of time in variable conditions. This is in contrast to researchers that could previously fine tune their system to the specific conditions in their laboratory. The inaugural humanoid robotics competition at RoboCup and at FIRA were held in 2002.

Furthermore, in 2002 hobbyists formed several popular robotics events which had less of a research emphasis. The most popular example are the televised Robot Wars (Battle Bots) events where remote controlled wheeled robots with weapons try to destroy each other. The huge interest in humanoid robotics in Asia has led to the creation of humanoid robot fighting competitions in Japan (Japan One) and Korea (K-One). Several of these humanoid robots are now commercially available at a competitive price. Even though the robustness and cost make these remote controlled fighting robots attractive for humanoid robotics researchers, the robots are not immediately suitable as autonomous robot research platforms.

The main disadvantages are: (a) these fighting robots do not have sufficient processing power for on-board vision, and (b) they have few if any sensors (e.g. accelerometers, gyroscopes, force sensors) to support dynamic balancing and feedback control of the walking gait.

This paper describes our work in converting the Kondo KHR-1 humanoid robot kit from a remote controlled fighting robot into a fully autonomous soccer playing robot Abarenbou. To keep the cost down, our robots uses a commonly available personal digital assistant (PDA) or mobile phones with a built in camera as processing platform for vision and higher level reasoning.

The following section describes the hardware of DAODAN and our modifications to the hardware. Section 3 describes the methodology we used for developing new motions (e.g. walk, turn, and kick). The vision processing part of our system is described in section 4. Section 5 describes our pragmatic Al method for localising the robot in the playing field and mapping the environment around the robot. The agent architecture is described in section 6. The paper concludes with section 7, which also gives directions for future work.

2 DAODAN Hardware Description

Figure 1 shown DAODAN our latest and most sophisticated humanoid robot. It has 17 degrees of freedom. DAODAN uses the A1001 AI motors from the Korean company Megarobotics. These motors are provide about 10 kg/cm torque but have several features that make them attractive for complex robots such as humanoids.

Firstly, the motors use a TTL level serial protocol which means that they can be daisy chained. This greatly simplifies the cabling of the robot. Secondly, the AI motors provide position and torque feedback. This makes it easier to develop stable walking gaits. We are also currently investigating methods where this feedback is used to actively control the angle of the upper torso to reduce the torque on the ankle joints. The torque on the ankle joints is a limiting factor in the development of walking gaits for humanoid robots, since all but the most expensive RC servos do not have enough torque to support a robot in the single support phase. In spite of these advantages, the A1001 AI motors are reasonably priced at approximately $50.00 CAN. This is in stark contrast to the $150 CAN or more that other teams spend on the robotis servos or high quality digital servos from Hitec and Futaba.

DAODAN uses Li-Polymer batteries because of their good power density and maximum current.

DAODAN is being controlled by a Nokia 3230 mobile phone, which has a powerful arm processor. It includes 6 MB of RAM which is sufficient for most robotics applications. The mobile phone has a built-in bluetooth connection and a small display which are useful in debugging.

DAODAN also includes an Analog Devices 3-axis accelerometer. We combined a OP amp to amplify the signal and connect it to an AVR Butterfly microcontroller. The accelerometer can be used as a tilt sensor by measuring the acceleration due to gravity along the 3 axis. The AVR Butterfly includes an ATMega 169 microcontroller, which implements a serial protocol that is compatible to the protocol used by the AI Motor servo motors.

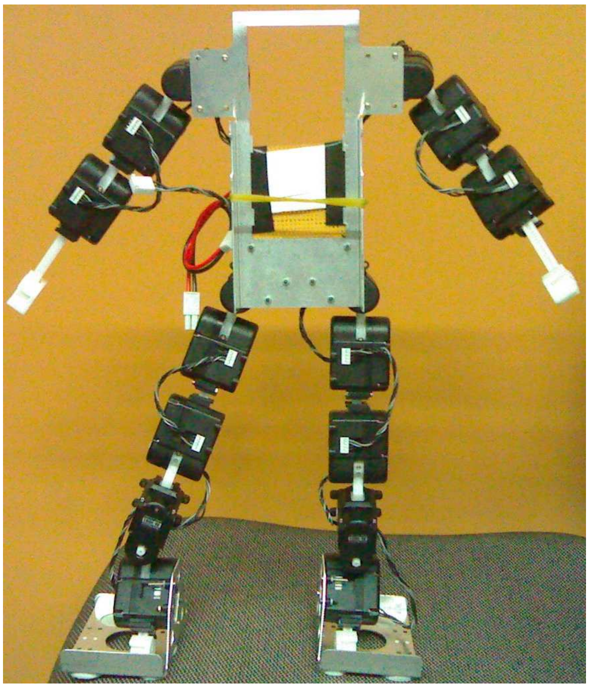


Figure 1: DAODAN.

3 Motion Development

Figure 2 shows the interface of the motion control software that we developed to control DAODAN. The interface allows one to move the robot into a specific position and save this position. The interface also allows one to set the trim (i.e. offset) for all joints as well as the home position.

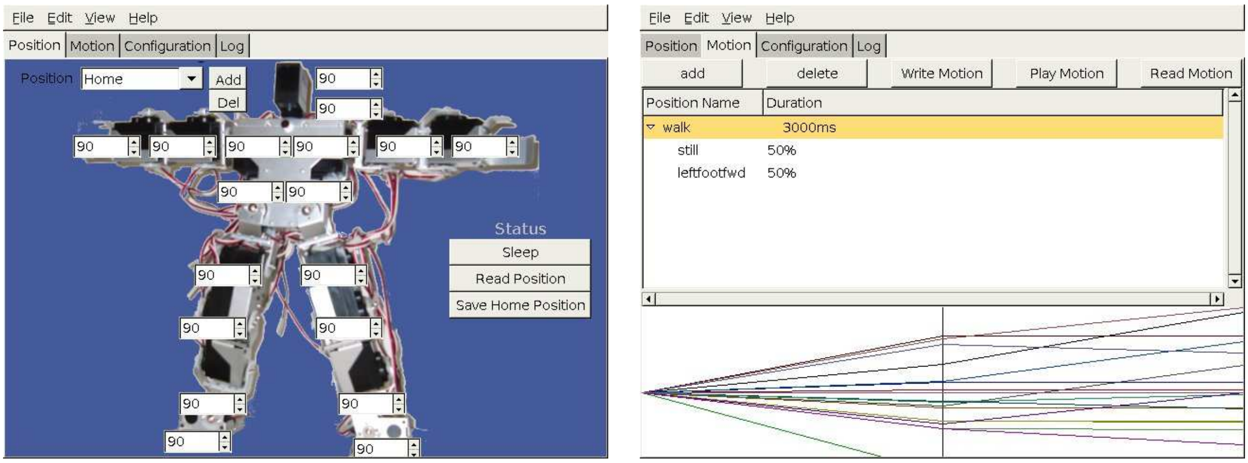
A separate window tab is used to combine positions into motions. Each motion has a cycle time associated with it and each part of a motion has a arrival time associated with it. Thus, the interface allows the user to easily adjust the speed of a whole motion or individual parts of the motion. The trajectory of all joints is shown in the bottom window.

Eight movements were programmed onto the robot: start walking, take step with right foot, take step with left foot, stop from left walk, stop from right walk, sideways step left, sideways step right, and kick with right foot.

These movements are then available to be played back as required by any of our programs running on the PDA.

4 Vision Processing

Our robots use a CMOS camera as the main sensor. The camera is used to approach objects in the field of view of the robot as well as localisation and mapping.



Position Name	Duration
walk	3000ms
still	50%
leftfootfwd	50%

Figure 2: Interface of our motion development system. The top window shows the development of a position, the bottom window shows the combination of these positions into motions.

To be robust enough to deal with the complex environment of robotic soccer, the vision processing makes little use of colours, but uses a very fast approximate region segmentation algorithm. First, the algorithms scans the image and extracts scan line segments (i.e. segments of similar colour) of approximately the right size. This step is similar to standard region segmentation algorithms.

However, we noticed that implementing a full union-find algorithm was too slow since it took about 2 secs. per image. Since most objects of interest in the environment are relatively small, we use a flood fill pixel merge algorithm, to find the associated region for a scanline. Note that the flood fill algorithms keeps track of which pixels have previously been visited and thus will visit each pixel at most once. The returned region is then checked for size (i.e. number of connected pixels), size of the bounding box, aspect ratio, and compactness. Only in the final step does the algorithm test whether the average colour of the region matches the object colour. If any of these tests fail, the object is rejected. Using only average colours of regions results in robust recognition of the ball and the goals and takes on average approximately 200ms.

An approximation of the relative position of objects is possible by determining the pan and tilt angles, and then calculating the distance to the centre of the image. It is assumed that these objects are on the ground plane. The relative position of an object at the centre of the image will have the closest approximation, so the camera is centred on important objects such as the ball before a decision is made as to what action to take next.

Goals are also detected as objects. Each goal is a distinct colour according to RoboCup rules. If both goal colours are found in one image, the regions of each goal colour are merged with other regions of the same goal colour. The goal colour that is present in the largest merged region is considered to be the goal currently being viewed.

To help the feature based localisation method described in the following section, we use a complex camera calibration based on the Tsai camera calibration algorithm [8]. This calibration is only done once for each robot. Given this calibration information, we are able to map points in the image accurately to their real world coordinates. This is essential since it allows us to determine the distance and orientation of the ball to a feature point (ball, goal post, line)

Before localisation can occur, features must be extracted from the image. The relevant features for localisation on the soccer field are lines, goals, and the centre circle. We use the lines and the goals to achieve localisation.

Every 5th column, the system scans from the bottom of the image towards the top. If there is a transition from a green pixel to a white pixel, the pixel p is remembered in a list. The scan continues upward, so there may be more than one transition pixel in a column.

Next, lines are found by running a gradient guided Hough transform [9]. For each point p_i, a set of adjacent points is determined. Triplets are formed from these by including one point to the left of the point p_i, and one point to the right of p_i. There are several triplets that can be formed this way out of the neighborhood of adjacent points. Each triplet votes for an unbounded line in the image. This vote is fuzzified by voting for a small range of slopes through the point p_i.

The peaks in the Hough accumulator space determine the equations of possible lines. For each peak in the accumulator space, we search along the pixels determined by the line equation to find start and end points of the lines. This results in a set of line segments.

The line segments are ordered based on their size. The longest line segment is assumed to represent the edge of the playing field. Given the distance and gradient of the line segment, the position and direction of the robot can be computed.

5 Localisation and Mapping

Knowing the position of the ball is important. Its relative position from the robot is easily determined from an image. However, without knowing the world position of the ball, the robot would often kick the ball out of bound or even into its own goal. Actions can not be taken toward kicking the ball until its world position is known.

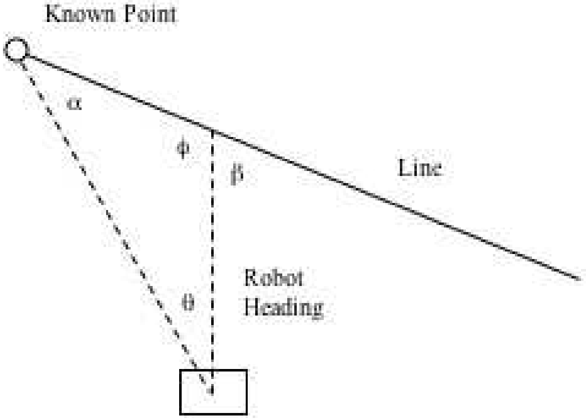


Figure 3: Localisation using a known point, its relative position and its orientation relative to a line

Absolute localisation of the robot will give absolute localisation of the ball. Absolute localisation can be done as long as a point is viewed with a known world coordinate, and knowing the robot's world bearing from it as shown in figure 3. One instance of this is when a goal post is seen. Once this is accomplished, dead reckoning can be used with some accuracy for a short time afterward.

An example of the quality of our localisation routine is shown in figure 4. The image as seen from the robot is shown on the left, and the calculated position is shown in the image on the right.

6 Agent Architecture

This section will give a brief introduction to the behaviour tree based agent architecture. The discussion will focus on the state transitions. More complex features of the agent architecture (e.g. behaviour trees and state references) are outside of the scope of this paper.

```
<State id="Scan For Target" >
<Enter>
%%v(angle) = 0;
if ( previousState == %%State("Target Right Forward") )
    {
      %%v(newAngle) = 20; /* Turn 20 degrees first */
      %%v(angleAdjust) = +10;
    }
 else
    {
      %%v(newAngle) = - 20; /* Turn 20 degrees first */
      %%v(angleAdjust) = -10;
    }
</Enter>
<Process>
  if ( ( %%v(newAngle) >= -190 ) &&
       ( %%v(newAngle) <= 190 ) )
    {
      if  (%%v(angle) != %%v(newAngle) )
        {
           turn( (%%v(angleAdjust) * TEN_DEGREE) / 10 );
           %%v(angle) = %%v(angle) + %%v(angleAdjust);
        }
      else
        {
           %%v(newAngle) = - %%v(newAngle) - 40;
           %%v(angleAdjust) = - %%v(angleAdjust);
        }
    }
  else
    {
      %%Transition("Random Walk");
    }
</Process>
</State>
```

Table 1: An XML schema for a behaviour that scans for a target by turning right/left with increasing sweeps

Designing an architecture that is flexible, versatile, and intuitive enough for an intelligent mobile robot is a difficult problem. This is especially true in the case of autonomous robots with limited processing capabilities.

The architecture uses a behaviour tree to balance the need for deliberate planning and reactive behaviours. The behaviours themselves are implemented as finite state machines. As the complexity of the task increases, the implementation of the state machine becomes more error prone.

We therefore developed a meta language to describe the behaviors in XML. The specification of the behaviors includes preconditions (enter functions) and post conditions (exit functions) of the behaviours. An slightly simplified example of a simple behaviour that scans for a target with increasing sweeps is shown in table 1.

The XML schemas include additional markup to refer to states by name (`%%State("Random Walk")` access variables (`%%v`) and to trigger transitions to other states (`%%Transition`).

Behaviours are organised into behaviour trees. Higher level behaviours can override or enable other lower level behaviours. For example, a "Perception" behaviour may disable the scan for target behaviour and enable the state "Target In Front" if it recognises the target.

One of the design goals of the meta language was to be highly efficient. Therefore, instead of adding a XML parser and interpreter to the agent, the meta language is parsed and interpreted offline and converted into highly efficient C code. This code is then compiled and executed on the PDA. For example, the example above shows that the programmer uses state names (e.g.

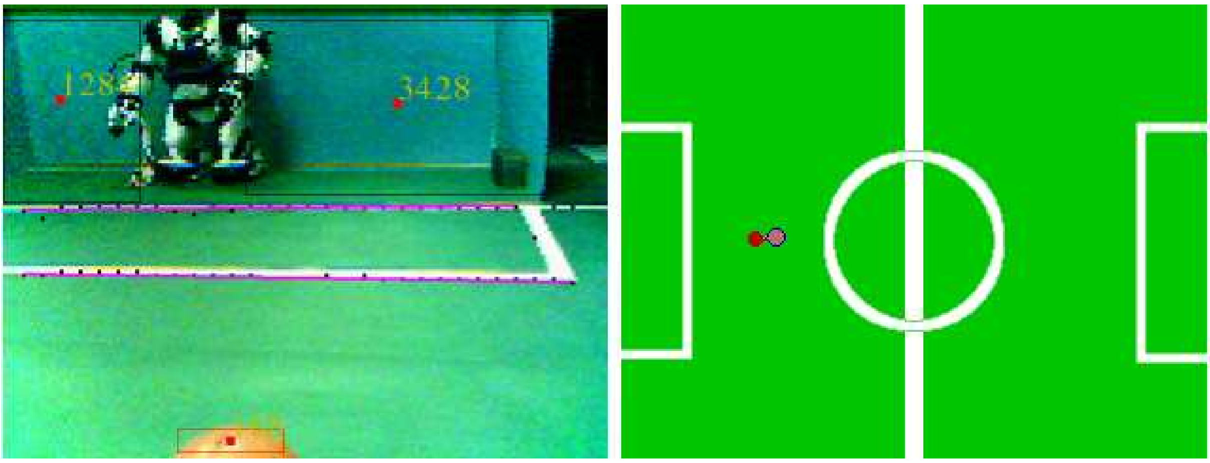


Figure 4: Image from the robot with highlighted line segments and the calculated position

"Random Walk," and "Scan For Target"). However, the states names are converted to integers in the C code.

An additional advantage of using a formalised state representation language with markup in the code is that it is possible to generate other representations. For example, our state compiler automatically generates a state transition graph. Figure 5 shows the state transition graph for a simple approach task. The robot first approaches a target and then walks away from it.

7 Conclusion

This paper describes the hardware and software of our latest humanoid robot DAODAN. Primarily intended as a platform to compete in the FIRA HuroSot and the RoboCup humanoid robotic soccer competitions, but it is very suitable for many other areas of humanoid robotic research.

We use a vision-based approach to determine the behaviour of the robot. The robot uses lines on the playing field to localise itself on the playing field and to map objects into the robot's environment.

The development of the complex architecture necessary for an intelligent soccer player is simplified through the use of an XML based meta language for behaviour trees. This meta language makes the pre-conditions, process stack, and state transitions explicit. The XML representation is converted into C code, which can be compiled into efficient code and thus does not introduce computational overhead through its use.

We are currently investigating methods for adding additional sensors (e.g. force feedback) that provide feedback about the balance of the robot. Even though active balancing is not necessary for a robot playing soccer on an even surface, it is at the heart of humanoid robotics research and is necessary to build robots that are able to move over uneven surfaces. For example, the FIRA HuroSot competition includes an uneven stepping field in the lift and carry competition.

8 References

[1] F. Yamasaki, T. Matsui, T. Miyashita and H. Kitano, "Pino the humanoid: a basic architecture", in Peter Stone, Tucker Balch, and Gerhard Kraetszchmar (eds), *RoboCup-2000: Robot Soccer World Cup IV,* Springer Verlag, Berlin, pp 269–278 (2001).

[2] R. Zhang, P. Vadakkepat, C. Chew and J. Janardhanan, "Mechanical design and control system configuration of a humanoid robot", *Proc. of 2nd Int. Conf. on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*, Singapore (2003).

[3] J. Baltes and P. Lam, "Design of walking gaits for tao-pie-pie, a small humanoid robot", *Advanced Robotics*, 18(7) pp 713–716 (2004).

[4] C. Zhou and P. Yue, "Robo-erectus: a low-cost autonomous humanoid soccer robot", *Advanced Robotics*, 18(7) pp 717–720 (2004).

[5] J.H. Kim, D.H. Kim, Y.J. Kim, K.H. Park, J.H. Park, C.K. Moon, J.H. Ryu, K.T. Seow and K.C. Koh, "Humanoid robot hansaram: recent progress and developments", *JACIII*, 8(1) pp 45–55 (2004).

[6] RoboCup Federation. *Robocup Humanoid League Rules,* "RoboCup Federation", http://www.robocup.org/regulations/humanoid/-rule_humanoid.htm (2001).

[7] J. Baltes and T. Bräunl, *HuroSot Laws of the Game,* The Federation of International Robot-soccer As-

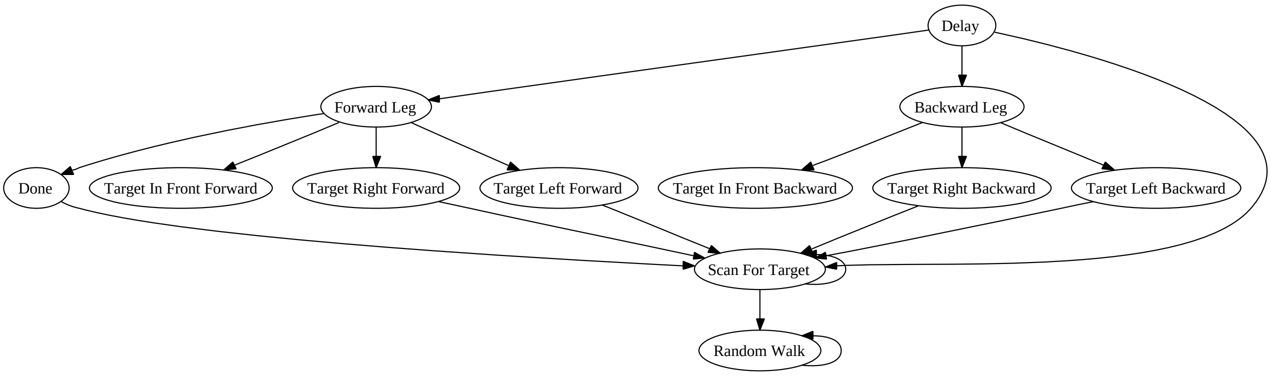


Figure 5: Automatically generated state transition graph for a simple approach and avoid task. Solid lines are state transitions.

sociation (FIRA), http://www.fira.net/hurosot (2006).

[8] R.Y. Tsai, “An efficient and accurate camera calibration technique for 3d machine vision”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp 364–374, Miami Beach, FL (1986).

[9] K.Y. Hough, Method and Means for Recognizing Complex Patterns, US Patent 3069654 (1962).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Foot Planning based on Global Motion Control Method and Fuzzy Logic System

H. Wongsuwarn[1] and D. Laowattana[2]
[1]Mechanical Engineering Department, [2]Institute of Field Robotics,
KMUTT, Bangkok, Thailand.
hataitep@fibo,kmutt.ac.th, djitt@fibo.kmutt.ac.th

Abstract

In this paper, we investigated two criteria that will be utilised to generate a balance biped trajectory. The first criterion is the moment criterion (ZMP) to maintain biped balance while the second criterion is the angular momentum criterion to generate human-like movement. In theoretical idea based on human motion, Biomechanical investigations have determined that for normal, level-ground human walking, spin angular momentum, or the body's angular momentum about the CM, remains small throughout the entire walking cycle, including both single and double support phases. In single support, we attempted to calculate a trajectory of swing foot that biped robots walk dynamically balance and look like human motion. In double support phase, the biped robot was stable walking. Normally, ZMP locate at middle both foot supports. We used that idea to apply at single support phase. A new ZMP position called "Virtual ZMP" based on global motion control. We calculated the virtual ZMP position at next step time by using fuzzy logic system. After that, we calculated the position of swing foot that are function of virtual ZMP on *x* and *y* axis. For position of swing foot on *z*-axis, we applied Actual ZMP position for this proposed by using fuzzy logic system. Finally, we calculated position of swing leg at next step time when we are predefined trajectory of whole body centre of mass, such as moved by joystick. The effectiveness of the proposed method is demonstrated by a simulation of walking on flat surface.

Keywords: angular momentum, control, fuzzy logic, global motion control, humanoid, walking planning

1 Introduction

Many researchers consider bipeds to be the ultimate robot systems, in which the environment is principally designed to meet the ergonomic demands of the human body. Stairs, chairs, offices, kitchens and sport stadiums are all designed with human-being compatibility. Alternatively, if they are to be used in environments where humans regularly work, biped robots will need to be smaller, lighter or more useful if they are to replace them. Wheeled robots, which presently dominate the mobile robot population, do not readily interact with human-shape space. Many aspect of humanoid walking have been solved [1-4].

In spite of experimental verification of various humanoid robots, it is important to further improve their capabilities in order that contemporary humanoid robots must be more autonomous intelligent and adaptable to environment and humans. The intelligent soft paradigms are demonstrated to be very powerful tools when applied in an appropriate manner for humanoid robots. The various pure and hybrid paradigms may be very efficient for realisation of online trajectory planning on line biped gait synthesis and control of static and dynamic balance during biped walking and running [5].

In order to compute humanoid movement that satisfies the desired ZMP trajectory, an iterative search algorithm was also introduced. Nagasaki et al. [6] used a gradient search algorithm to produce a stable humanoid-robot movement by adjusting an initial unbalance solution into a balance humanoid movement. An iterative approach that incorporates the concept of movement snapshots was adopted with additional consideration of angular momentum at CM by Naksuk et al. [7]. Huang et al. [8] developed an algorithm to produce humanoid-robot movement with maximum ZMP stability margin. This is accomplished by iteratively adjusting the relative location of the hip with respect to the location of the supporting foot. An analytic solution, which is more suitable for real-time trajectory generation, was proposed in [9] and [10]. The solution is based on solving differential equation with boundary value condition.

Time-local strategies for postural balance have been previously proposed. Studies relied on placing the centre of mass (CM) ground projection as close as possible to the innermost point of the support base to ensure stability [11, 12]. As known for centuries, stability is satisfied for the static case when the zero moment point, or ZMP, is positioned just beneath the CM [13,14]. Along similar lines, the ZMP distance from the support base boundary was employed as a stability margin [15]. Finally, recent studies explored the horizontal component (orthogonal to gravity) of angular momentum about the CM and the associated moment as indicators of postural stability [16,17]. However, by themselves none of these proposed stability metrics guarantee stability [18]. Moreover, not all of the proposed metrics are supported by biomechanical observations. During a walking gait cycle, the CM crosses the support base only during a short segment of the double support phase. Still further, the ZMP spans nearly the entire foot length during the single support phase. Finally, the angular momentum is highly regulated in walking but is less regulated for activities like a twirling hula-hoop motion or balancing on a tight rope [17-19].

In section 2 we propose a novel angular momentum-based, optimisation strategy expressed in terms of global quantities, or quantities representing whole body translational and rotational dynamics. Moreover, we introduce the global motion control (GMC) framework suitable for the control of high level integrated quantities. Though the proposed optimisation strategy and ZMP location may be expressed only in terms of global quantities, we extend our control framework from global state to joint space to effectively control balance as well as additional performance objectives. In section 3, we apply global motion control (GMC) and ZMP criteria to generate foot trajectory of our biped robot. And we apply fuzzy logic system to generate position of swing foot on *x*, *y* and *z* axis.

2 Walking Planning based on Global Motion Control (GMC) and ZMP Criteria

We apply global motion control (GMC) [19] and ZMP criteria [5] to generate foot trajectory of our humanoid robot. In theoretical idea based on human motion, biomechanical investigations have determined that for normal, level-ground human walking, spin angular momentum, or the body's angular momentum about the CM, remains small throughout the entire walking cycle, including both single and double support phases. To maintain balance in biped robot locomotion, supporting foot/feet must withstand moment acting on the biped robot, whenever the moment acting on the biped is more than the moment withstandable by the contact of the supporting foot/feet, the biped robot starts to fall down. In this research, we investigate two criteria that will be utilised to generate a new approach foot step planning of swing foot during dynamic walking based on global motion control (GMC) and ZMP criteria. The first criterion is the moment criterion (ZMP) to maintain biped balance while the second criterion is the angular momentum criterion to generate human-like movement.

For this foot planning method, we need to move the trajectory of whole body centre of mass ($\vec{r}_{CM,des}$). And, the robot automatically generates a trajectory o both stance leg and swing leg during dynamic walking (figure 1).

In his book *Legged Robots that Balance*, Raibert (1986) [2] speculated that a control system that keeps angular momentum constant during stance could achieve higher efficiency and better performance, motivated by biomechanical measurements showing the relatively small size of CM moments during human walking.

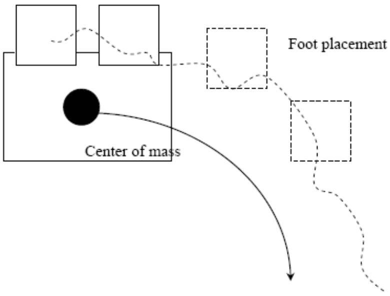


Figure 1: Show whole body centre of mass trajectory and foot trajectory.

Popovic, Gu and Herr (2002) [16] suggested that humanoid control systems should explicitly minimise global spin angular momentum during steady state forward walking $(\vec{L}_{des}(\vec{r}_{CM})=0)$ and a particular angular excursions. Consider a simple PD control law relating whole body angular excursion ($\vec{\theta}$), spin angular momentum ($\vec{L}(\vec{r}_{CM})$), CM position ($\vec{r}_{CM}$), CM velocity ($\dot{\vec{r}}_{CM}$) and CM target acceleration ($\ddot{\vec{r}}_{CM,tar}$), with desired whole body net moment about the CM ($\vec{\tau}_{des}(\vec{r}_{CM})$), and net CM force ($\vec{F}_{des}$), or

$$\vec{\tau}_{des}(\vec{r}_{CM}) = \overleftrightarrow{l}_1 \Delta\vec{L}(\vec{r}_{CM}) + \overleftrightarrow{l}_2 \Delta\vec{\theta}\,, \tag{1a}$$

$$\vec{F}_{des} = M\ddot{\vec{r}}_{CM,tar} - \overleftrightarrow{k}_1 \Delta\dot{\vec{r}}_{CM} - \overleftrightarrow{k}_2 \Delta\vec{r}_{CM}\,. \tag{1b}$$

In (1a) $\Delta\vec{\theta} = \vec{\theta}_{tar} - \vec{\theta}\,, \vec{\theta}_{tar}$ is the target body angular excursion; $\Delta\vec{L}(\vec{r}_{CM}) = \vec{L}_{tar}(\vec{r}_{CM}) - \vec{L}(\vec{r}_{CM})$ $, \vec{L}_{tar}(\vec{r}_{CM})\,.$ is the target spin angular momentum; and positive definite 3 by 1 matrices $\overleftrightarrow{l}_1$ and $\overleftrightarrow{l}_2$ are rotational stiffness and damping coefficients respectively. Analogously, in (1b) $\Delta\vec{r}_{CM} = \vec{r}_{CM,tar} - \vec{r}_{CM}\,, \vec{r}_{CM,tar}$ is the target CM position; *tar*. $\Delta\dot{\vec{r}}_{CM} = \dot{\vec{r}}_{CM,tar} - \dot{\vec{r}}_{CM}\,,$ $\dot{\vec{r}}_{CM,tar}$ is the target CM velocity; and positive definite 3 by 1 matrices $\overleftrightarrow{k}_1$ and $\overleftrightarrow{k}_2$ are stiffness and damping coefficients respectively. For practical purposes, instead of whole body angular excursions $\vec{\theta}$, which are not directly measurable quantities, one may consider using whole body principal angles ($\vec{\theta}_{eff}$)defined by the relative orientations of the principal axes of the whole body inertia tensor with respect to the non-rotating lab frame axes. Thus, $\Delta\vec{\theta} = \vec{\theta}_{tar} - \vec{\theta}$ is replaced by $\Delta\vec{\theta}_{eff} = \vec{\theta}_{eff,tar} - \vec{\theta}_{eff}$. For a humanoid robot in steady state walking, one may anticipate that the desired whole body angular excursion and the spin angular momentum would both be set to zero ($\vec{L}_{tar}(\vec{r}_{CM}) = 0, \vec{\theta}_{tar} = 0$) and the rotational stiffness and damping coefficients would then be adjusted to achieve a desired system response. Also, equation (1a) and (1b) replaced by

$$\vec{\tau}_{des}(\vec{r}_{CM}) = -\overleftrightarrow{l}_1 \vec{L}(\vec{r}_{CM,actual}) - \overleftrightarrow{l}_2 \vec{\theta}_{eff,actual}\,, \tag{2}$$

$$\begin{aligned}\vec{F}_{des} &= M\ddot{\vec{r}}_{CM,tar} - \overleftrightarrow{k}_1(\dot{\vec{r}}_{t\arg et} - \dot{\vec{r}}_{CM,actual}) \\ &\quad - \overleftrightarrow{k}_2(\vec{r}_{CM,t\arg et} - \vec{r}_{CM,actual})\end{aligned} \; . \tag{3}$$

The novelty of equation (2, 3) is that it employs the rotational analogue of the CM position and that it unifies all global quantities into one simple proportional derivative (PD) control law. We name this relationship the global motion control (GMC) PD law.

Vukobratovic and Borovac defined the ZMP as the "point of resulting reaction forces at the contact surface between the extremity and the ground". The ZMP, $\vec{r}_{ZMP}$ therefore may be defined as the point on the ground surface about which the horizontal component of the moment of ground reaction force is zero [5], or

$$\vec{\tau}_{G.R.}(\vec{r}_{ZMP})\,|_{horizontal} = 0 \tag{4}$$

Equation (4) means that the resulting moment of force exerted from the ground on the body about the ZMP is always vertical, or parallel to $\vec{g}$. Finally, the ZMP as a function of the CM position, net CM force ($\vec{F} = M\vec{a}_{CM}$), and net moment about the CM can be expressed as

$$\begin{aligned} x_{ZMP} &= x_{CM} - \left(\frac{F_x}{F_z + Mg}\right)\cdot Z_{CM} - \left(\frac{\tau_y(\vec{r}_{CM})}{F_z + Mg}\right) \\ y_{ZMP} &= y_{CM} - \left(\frac{F_y}{F_y + Mg}\right)\cdot Z_{CM} - \left(\frac{\tau_x(\vec{r}_{CM})}{F_z + Mg}\right) \\ z_{ZMP} &= 0 \end{aligned} \tag{5}$$

where $\vec{F} = M \cdot \vec{a}_{CM}$ and $g = 9.81m/s^2$

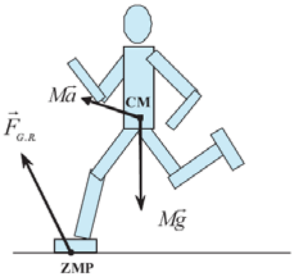


Figure 2: Zero moment point. The ZMP is where the ground reaction force acts, whereas the CM point is where inertia and the force of gravity originate [18].

As emphasised in figure 2, the most important notion of the ZMP quantity, applicable for both single and multileg ground support phases, is that it resolves the ground reaction force distribution to a single point. However, one needs to be careful to use this point in an appropriate manner. Most notably, both the vertical component of moment and the CM work performed by the ground reaction force cannot be computed solely on the bases of the ZMP trajectory and the resulting ground reaction force vector. Thus, we can define the virtual ZMP point based on the global motion control (GMC) PD law ((2) and (3)) and ZMP criteria (5). Physical meaning of virtual ZMP point is the biped robot that walks dynamic stable and human-like motion. From equation (2), (3) and (5), we can be expressed as

$$\begin{aligned} x_{ZMP,Virtual} &= x_{CM,actual}, -\left(\frac{F_{x,des}}{F_{z,des} + Mg}\right)\cdot Z_{CM.actual} \\ &\quad - \left(\frac{\tau_{y,des}(\vec{r}_{CM})}{F_{z,des} + Mg}\right), \\ y_{ZMP,virtual} &= y_{CM,actual} - \left(\frac{F_{y,des}}{F_{y,des} + Mg}\right)\cdot Z_{CM,actual} \\ &\quad - \left(\frac{\tau_{x,des}(\vec{r}_{CM})}{F_{z,actual} + Mg}\right), \\ z_{ZMP,actual} &= 0 \end{aligned} \tag{6}$$

where $g = 9.81m/s^2$.

Equation (6) is the virtual ZMP position at next step time. After that, we calculate the position of swing foot that are function of virtual ZMP. In double support phase, the biped robot is stable walking. Normally, ZMP locate at both foot supports. We use that idea to apply at single support phase (figure 3).

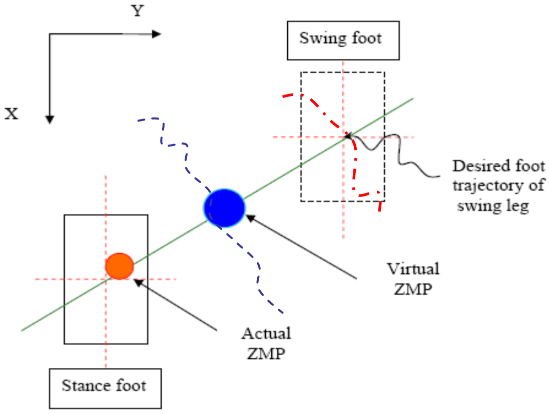


Figure 3: Theoretical idea of swing foot trajectory on *x* and *y* axis.

And from figure 4 we calculate an actual ZMP. And we can calculate a foot position of swing leg as function of actual ZMP.

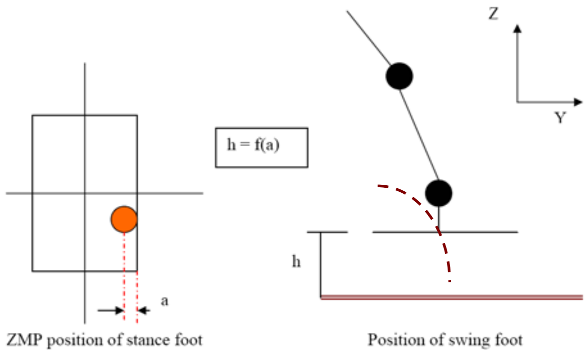


Figure 4: Theoretical idea of swing foot trajectory on *z* axis.

Finally, we can calculate position of swing leg at next step time when we are predefined trajectory of whole body centre of mass (see figure 1), such as moved by joystick.

3 Simulating Result

From section 2, we investigated a novel foot planning based on global motion control and ZMP criteria. Our humanoid robot can be automatically move a position of swing leg at next step time when we are predefined trajectory of whole body centre of mass.

For the next step motion, the knowledge of state positions, velocities, accelerations of swing foot, ZMP of stance leg, positions, velocities, accelerations of actual whole body centre of mass (WCOM), angular momentum of WCOM and effective angle of WCOM at time t_{n-1} defines the expectation of state at time t_n such as position of swing leg. Although, we known a position of virtual ZMP. The position of swing leg on *x*, *y* and *z* axis are difficult to define by equation form. As one of methodologies applied for biped gait synthesis and control of biped and humanoid robot, some researcher used fuzzy logic. Thus, we apply fuzzy logic as computational intelligence technique to investigate a position of swing leg at next step time.

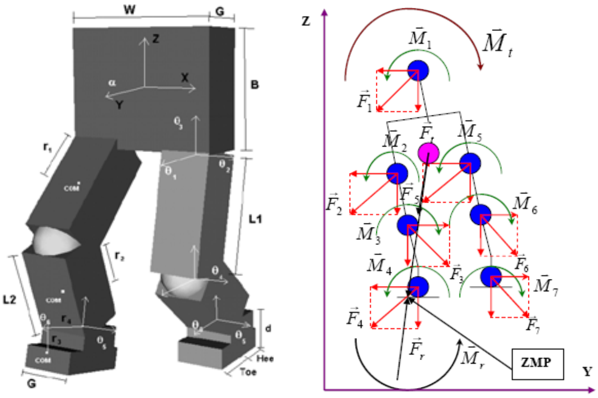


Figure 5: 7-segments legged posture shown in which the ZMP is located the CM (pink).

In this paper, the simulating robot system has seven body components: one hip, two thighs, two calves, and two feet. Body parts have certain masses and they all affect dynamics of the walking pattern. There is a total of twelve degree of freedom (DOF). Knee joints are able to only turn about *x* axis. The joint between the hip and thigh has three DOF. The ankle joint turns about *x* axis and *y* axis.

The humanoid robot used in the simulation is 1.45 m. tall and weight 50 kg. Detailed dimension and masses of its link are shown in Table 1.

Table 1: Mass and dimensions of the biped robot links.

Segment	Length	Mass	Inertial Moment
	(m)	(kg.)	(kgm^2)
Body	0.03	16.11	0.381
Thing	0.038	8.34	0.105
Calve	0.035	7.2	0.08
Foot	0.018	2.22	0.01

The desired position of swing leg is greatly influenced by the position of virtual ZMP, effective angle of WCOM, desired position of WCOM and actual ZMP of stance leg. Three membership functions are used for position of virtual ZMP, effective angle of WCOM, desired position of WCOM and actual ZMP of stance leg as shown in figure 6. And also, three membership function used for position of swing leg in each axis as shown in figure 7. Defuzzification based upon the centre of gravity is sued. Based upon this knowledge about human motion, the rule base of the fuzzy logic is shown in figure 8.

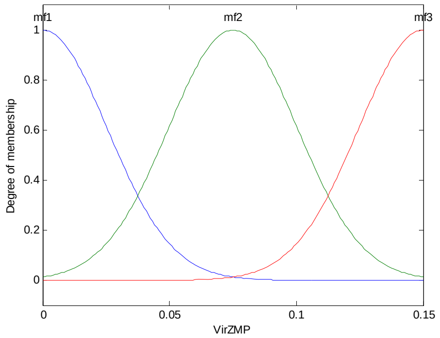


Figure 6: Fuzzy membership function for ZMP$_{\text{Virtual}}$

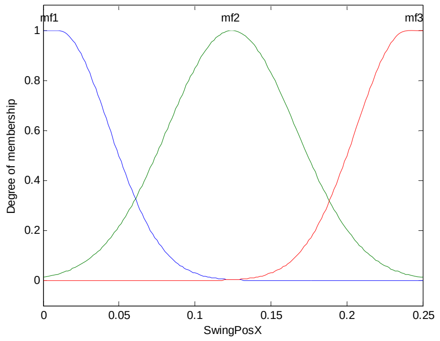


Figure 7: Fuzzy membership function for position of swing foot on *x*-axis

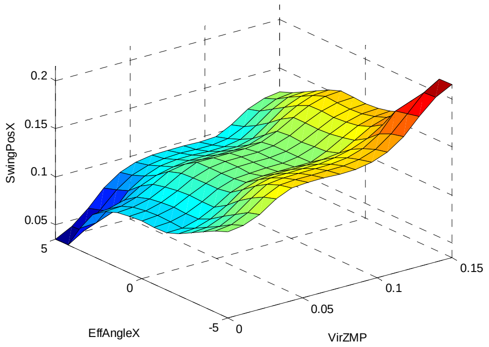


Figure 8: Rule base about virtual ZMP and effective angle acting to swing foot position on *x*-axis.

In the simulation, the initial condition of the robot was a single support phase. The interval step time was 10 ms and the ground projection of the WCOM was inside the support polygon of stance foot. Simple reference trajectory of desired WCOM was selected. The desired WCOM was moving on *x*-axis of the robot at 0.2 m/s^2. Thus, WCOM of the robot was going to outside the support polygon. Swing leg of the robot was moved automatically and touched on the ground. Figure 8 was a position of swing leg on *x*-axis and figure 9 was a position of swing leg on *z*-axis with initial position on 0.05 m.

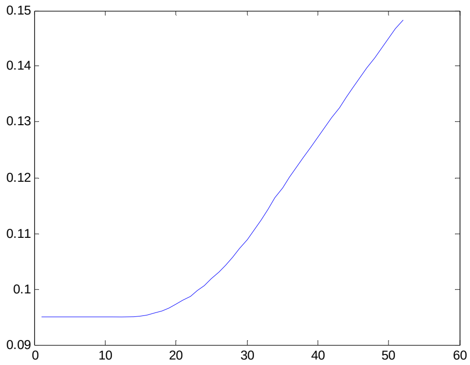


Figure 8: Position of swing leg on *x*-axis.

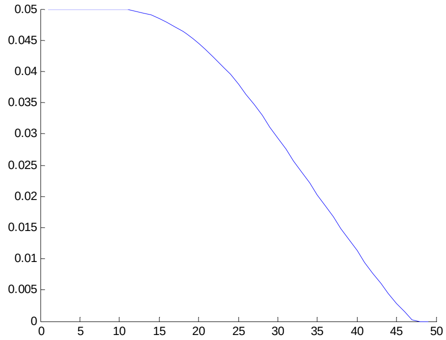


Figure 9: Position of swing leg on *z* axis.

Figure 10 shows the system's recovery from an initial displacement in lateral (positive x) direction and frontal (position y) direction. From the model's perspective, the left most edge of the foot support polygon is at 0.05 m. As is shown in figure 11, the ZMP within the foot support polygon, while the laterally displaced the actual WCOM begins outside the stance foot at 0.38 sec. Although, effective angle on *x* axis and *z* axis were close too naturally human, because of mass distribution of the robot was not same human body. But, they were decreases to zero that are good result.

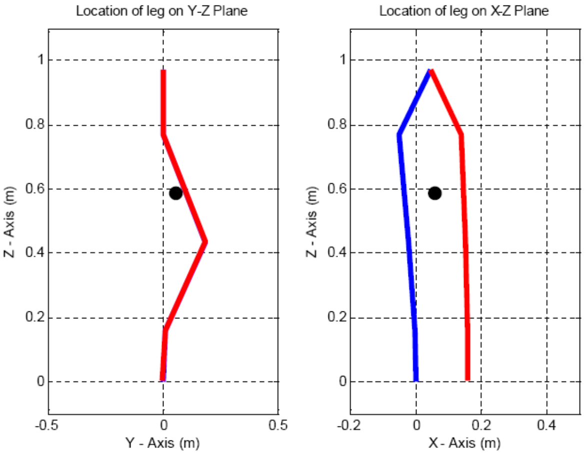


Figure 10: Final gait of a biped robot when touching on ground.

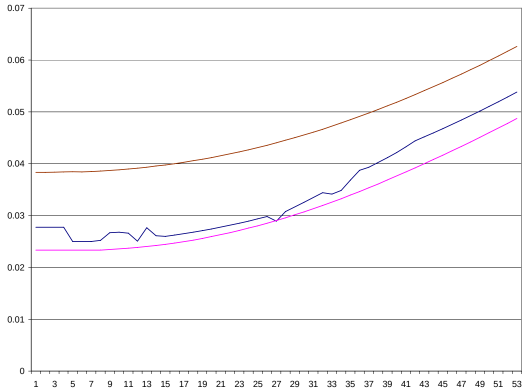
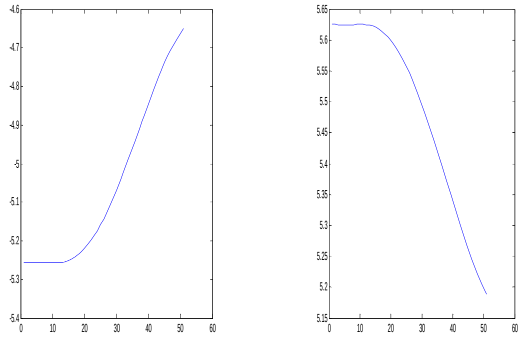


Figure 11: Actual ZMP (blue), WCOM (brown), Virtual ZMP (pink) on *x*-axis vs. time (10^{-2}sec).



x-axis

z-axis

Figure 12: Effective angle of the robot.

4 Summary and Future Work

In this paper, we present a novel foot planning for humanoid robot based on global motion controller. Moreover, we propose a time-local strategy where joint accelerations are varied to achieve whole-body dynamic postural balance by using fuzzy logic. This strategy optimises global body parameters such as spin angular momentum and body principal angles with respect to their equilibrium global state configuration as defined by the effective gravity vector.

In future investigations this theoretical model will be utilised and followed by several detailed studies on specific activities such as dynamic walking. These future investigations will simultaneously address human biomechanics and humanoid control. It is our hope that this work will lead to further investigation into online foot planning techniques that address postural balance of real legged systems, resulting in an even wider range of locomotors performance

5 Acknowledgements

This work was supported in part by the Thailand Research Fund (TRF) under Grant of TRF Golden Jubilee PhD fellowship PHD0022641. And this paper could not be finished if there were no generous supervision from Dr. Marko B. Popovic and Assoc. Prof. Hugh Herr at Biomechatronic Group, Media Lab, MIT for their investigation in global motion control and robust humanoid control.

6 References

[1] D.J. Todd, *Walking Machines: An Introduction to Legged Robots*, Chapman and Hall, NY (1985).

[2] M.H. Raibert, *Legged Robots That Balance*, MIT Press, Cambridge, MA (1986).

[3] S. Kajita and K. Tani, “Experimental study of biped dynamic walking”, *Proceedings IEEE Intl. Conf. on Robotics and Automation*, pp 13-19 (1995).

[4] H. Kazuo, H. Masato, H. Yuji and T. Toru, “The development of Honda humanoid robot”, *Proceedings IEEE Intl. Conf. on Robotics and Automation*, pp 1321-1326 (1998).

[5] M. Vokobratovic and B. Borovac, “Zero moment point - thirty-five years of its life”, *Intl. Journal of Humanoid Robotics*, 1(1), pp 157-173 (2004).

[6] K. Nagasaka, H. Inoue and M. Inaba, “Dynamic walking pattern generation for a humanoid robot based on optimal gradient method”, *Proceedings IEEE SMC Conference*, pp 17-22 (1999).

[7] N. Naksuk, Y. Mei and C.S.G. Lee, “Humanoid trajectory generation: an iterative approach based on movement and angular momentum”, *Proceedings IEEE-RAS/RSJ Intl. Conf. on Humanoid Robots*, pp 125-130 (2004).

[8] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie, “Planning walking patterns for a biped robot”, *IEEE Trans. on Robotics and Automation*, 17(3), pp 280-289 (2001).

[9] R. Kurazume, T. Hasegawa and Yoneda, “The sway compensation trajectory for a biped robot”, *Proceedings IEEE Intl. Conf. on Robotics and Automation*, pp 925-931 (2003).

[10] K. Harada, S. Kajita, K. Kaneko and H. Hirukawa, “An analytical method on real-time gait planning for a humanoid robot”, *Proceedings IEEE-RAS/RSJ Intl. Conf. on Humanoid Robots*, pp 118-124 (2004).

[11] O. Khatib, L. Sentis, J.H. Park and J. Warren, “Whole body dynamic behavior and control of human-like robots,” *International Journal of Humanoid Robotics*, 1, pp 29-43 (2004).

[12] L. Sentis and O. Khatib, “Task-oriented control of humanoid robots through prioritization”, *Proceedings IEEE RAS/RSJ Intl. Conf. on Humanoid Robots*, Santa Monica (2004).

[13] G. A. Borelli, *De Motu Animalium* (trans. P. Maquet), Springer-Verlag, Berlin (1989 [1680]).

[14] M. Vukobratovic and D. Juricic, “Contributions to the synthesis of biped gait”, *IEEE Transactions on Biomedical Engineering*, 16, pp 1-6 (1969).

[15] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Aria, N. Koyachi and K. Tanie, “Planning walking patterns for a biped robot”, *IEEE Transactions on Robotics and Automation*, 17(3), pp 280-289 (2001).

[16] M.B. Popovic, W. Gu and H. Herr, “Conservation of angular momentum in human movement”, *MIT AI Laboratory Research Abstracts*, pp 231-232 (2002).

[17] M.B. Popovic, A. Hofmann and H. Herr, “Angular momentum regulation during human walking: biomechanics and control”, *Proceedings IEEE Intl. Conf. on Robotics and Automation*, New Orleans, pp 2405-2411 (2004).

[18] M.B. Popovic, A. Goswami and H. Herr, “Ground reference points in legged locomotion: definitions, biological trajectories and control implications”, *International Journal of Robotics Research*, 24(12), pp 1013-1032 (2005).

[19] M.B. Popovic and H. Herr, “Global motion control and support base planning”, *Proceedings Intl. Conf. on Intelligent Robots and Systems*, Edmonton, Canada, pp 103-108 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Intelligence of Micro Probe Robot

Takashi Kubota and Tetsuo Yoshimitsu
Institute of Space and Astronautical Science,
Japan Aerospace Exploration Agency, Japan
kubota@nnl.isas.jaxa.jp

Abstract

In recent years, asteroid or comet exploration missions have received remarkable attention in the world. In small body explorations, especially, in-situ surface exploration by micro probe robot is one of effective and fruitful means and will make strong contributions towards scientific studies. JAXA/ISAS is promoting MUSES-C mission, which is the world's first sample and return attempt to/from the near earth asteroid. Before sample collection, the MUSES-C spacecraft deployed the tiny rover in November 2005. Unfortunately the tiny rover could not arrive at the asteroid surface, but it was confirmed that the functions of the robot were very good. This paper presents the design and system configuration of the developed rover in detail. This paper also describes the intelligence of the robot and flight results.

Keywords: micro exploration robot, autonomous behaviour, intelligence, asteroid exploration.

1 Introduction

Small planetary bodies such as asteroids, comets and meteorites have received worldwide attentions in recent years. These studies have been motivated by a desire to shed light on the origin and evolution of the solar system. Hence, exploration missions for small bodies have been carried out continuously since the late 1990s. To date, the missions of NEAR [1], Deep Space 1 [2], and Deep Impact [3] have been successfully performed, while Stardust [4], MUSES-C [5] and Rosetta [6] are currently in operation. These missions have mainly provided remote sensing in the vicinity of the small body, at a distance which cannot be attained from Earth. In-situ observations of small bodies are scientifically very important because their sizes are too small to have experienced high internal pressures and temperatures, which means they should preserve the early chemistry of the solar system. For future missions such as MUSES-C, in-situ surface observation by robots will make strong contributions towards those studies.

Most of the small planetary bodies are too small for direct observation from Earth. To obtain knowledge about them, spacecraft must be launched to travel to the vicinity of the target planetary body. When comet Halley came close to the sun in 1986, six spacecraft operated by Japan, Europe, the former Soviet Union and the USA flew by and remotely observed the comet. In 2000, more than ten years had passed since the worldwide enthusiasm about the comet Halley, and the spacecraft NEAR was put into orbit about the main-belt asteroid 433 Eros and the longstanding goal of remote sensing came true. The NEAR spacecraft also carried out a hard landing demonstration onto the asteroid surface after successfully making precise remote-sensing observations lasting for one year. The landing itself was successful, but only brought about very limited in-situ surface observations. Hereafter, in-situ observations of the small body surface by a lander, in-situ observation of multiple places on the surface by a rover, and sample return to bring fragments of the small body back to Earth for more precise investigation are expected to be carried out for greater scientific achievement. In such deep space missions, ground based operation is very limited due to communication delays and low bit-rate communication. Therefore, autonomy is required for deep space exploration using rovers. On the other hand, because the gravity and surface terrain are not known in advance, robotics and artificial intelligence technology must be used for rovers to explore a minor body.

The Institute of Space and Astronautical Science (ISAS) of Japan launched the MUSES-C spacecraft toward Asteroid 1998SF36 in May 2003. After the launch, the spacecraft was renamed "HAYABUSA" which means falcon. The MUSES-C project is demonstrating four key technologies required for future sample return missions from extra-terrestrial bodies as shown in figure 1. Those technologies are: 1) solar electrical propulsion with ion thrusters in interplanetary space as a primary propulsive means, 2) autonomous optical guidance and navigation, 3) automated sampling mechanism, and 4) direct hyperbolic reentry of the recovery capsule into Earth's atmosphere and to the ground. The target body of the MUSES-C mission is a near-earth asteroid which is named "Itokawa". The launch date was May 9th in 2003 and arrival at the asteroid was September 12th in 2005. Having left the asteroid in December 2005, the spacecraft returns back to Earth in June 2010. The mission duration from launch to Earth return is about 7 years. HAYABUSA spacecraft stayed for approximately three months around the asteroid and both mapping and sampling operations were carried out during that short period. For the MUSES-C mission, an asteroid surface exploration rover was developed, called MINERVA for MIcro/Nano Experimental Robot Vehicle for Asteroid. MINERVA is the first asteroid exploration rover to ever be developed and deployed. It is one of the technical challenges for the MUSES-C mission and its major objectives are as follows: establish a mobile system in the micro gravity environment of a small planetary body, demonstrate the autonomous exploration capability the rover is equipped with, and perform the first-ever planned scientific observations on an asteroid surface. On November 12th in 2005, the spacecraft deployed MINERVA at higher velocity, and MINERVA could not arrive at the asteroid surface. However the spacecraft could communicate with MINERVA and it was confirmed that the health of MINERVA was very good.

This paper presents the challenges of mobilising a rover on a minor body surface and discusses the necessary intelligence such an exploration rover must have. Section 2 discusses possible mobility systems for the micro-gravity environment on a small body surface, and presents a hopping mechanism that was developed in MUSES-C mission. Section 3 describes the intelligent functions of the micro probe robot. Section 4 presents the asteroid exploration rover MINERVA, which was designed to explore an asteroid surface for the MUSES-C mission. Conclusions are stated in section 5.

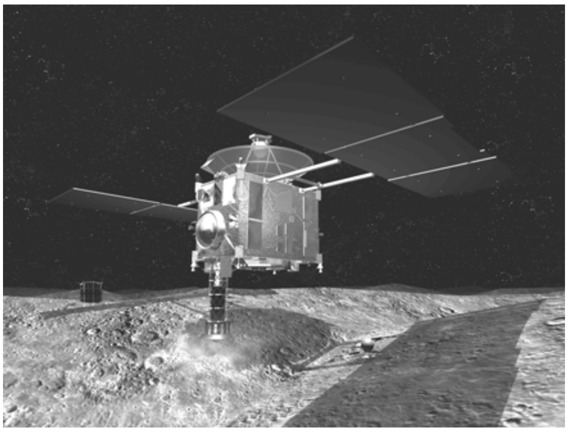


Figure 1: MUSES-C mission.

2 Mobility System

2.1 Mobility under Microgravity

The gravity on small asteroids such as Itokawa is on the order of 10 μG, which is extremely weak compared with that on Earth (1G equals one Earth gravity). Naturally, the escape velocity from the surface is very small as well, on the order of 20 centimetres per second. The largest asteroid yet discovered, Ceres, has a diameter of more than 900 km, however the majority of asteroids have diameters less than several kilometres in size, and surface gravity lower than one milli-G. Rovers that are intended to move over a small body surface need a mobility system suited for this milli- or micro-gravitational environment.

The main principle to be considered is that the traction force which drives the robot horizontally is very small in a micro gravity environment. Traction force is obtained by the friction between the robot and the surface. Assuming that Coulomb's friction law holds true in a microgravity environment, the traction force f is given by

$$f = \mu N \tag{1}$$

where μ denotes the friction coefficient and N is the contact force. In the case of a traditional wheeled mechanism, frequently used by planetary rovers on the Moon or on Mars, the contact force is opposite to the gravity, which results in the following equations:

$$f = \mu N = \mu mg \tag{2}$$

In (2), m is the mass of the rover and g is the gravitational acceleration on the surface. Equation (2) indicates that the available friction for wheeled mobility is very small in the microgravity environment (approximately 10^{-4} m/s^2). Thus, even if the friction coefficient is assumed to be large, friction between a robot and the surface becomes very low. Additionally, if the traction is larger than the maximum friction, the rover slips. Thus the available traction must be extremely small, which makes the horizontal speed extremely slow. Actually, NASA proposed a new robot [7] with wheeled mechanism as a payload of the MUSES-C mission, but it could have only driven only with the speed of 1.5 mm/s. To achieve distances of tens of metres, it would have used a mobility mechanism to hop.

Moreover, the surface of small bodies is a natural terrain with uneven surfaces. When the robot begins to move, any small disturbance may push the robot away from the surface. So the mobility system such as traditional wheeled mechanisms is not optimal under a micro gravity environment, where continuous robot-surface contact is not guaranteed. This reason motivates the search for another mobility system that specialises in hopping. Once a robot hops with some horizontal speed, it can move with no contact on the surface.

A hopping mechanism causes the robot to push the surface. This increases the contact force between the surface and the body artificially with the pushing force, which makes the friction larger and can provide mobility at a higher horizontal speed that cannot be attained by a wheeled mechanism. Once it has hopped into free space with some lateral speed, the robot moves in a ballistic orbit under the influence of the weak gravity. Lateral speed is also obtained by the friction acting during the short period after the actuator starts and before separation from the surface. In the hopping process, the friction force exerted by the hopping mechanism is expressed as:

$$f = \mu \; (mg + F) \qquad (3)$$

where F denotes the artificial pushing force which makes the robot hop. In the case $mg << F$, the friction force is approximated as:

$$f \sim \mu \; F \qquad (4)$$

Thus, a robot can use a friction force independent on the value of the gravity acceleration. In order to press its mass to the ground, some mechanism is required. By increasing the pushing force, the hopping mechanism is able to attain higher speeds which cannot be attained in principle by a wheeled mechanism. The maximum allowed speed is the escape velocity from the surface. If a larger speed than the escape velocity is achieved, the robot may never return to the surface again.

2.2 Hopping Mobility

There are several ways [7-10] to make a robot hop. Figure 2 shows a number of proposed hopping robots. The hopping mechanisms used in these craft are categorised into two groups. One is to use repulsed force by striking, sticking, or kicking the surface. The other is to use reaction force by rotating an internal torquer. The newly developed and novel hopping mechanism shown in figure 3, and used by MINERVA, is one where the robot incorporates a torquer inside of its structure. All previous designs have had a moving part outside of the rover body. However, MINERVA has no apparent moving parts outside, which has a big advantage in reliability for long-term outer-space use. After hopping into free space, the robot moves ballistically and comes back to the surface again, so long as its launch speed is less than the escape velocity from the surface. After a few bounces on the surface, it will settle and be ready to hop again. This action can be repeated, allowing the robot to explore the surface of the small planetary body. This proposed mechanism has several significant advantages:

1. The actuator is not outside the rover. Since the actuator is sealed inside the body, no consideration is required for dust, which may exist on the small body surface.

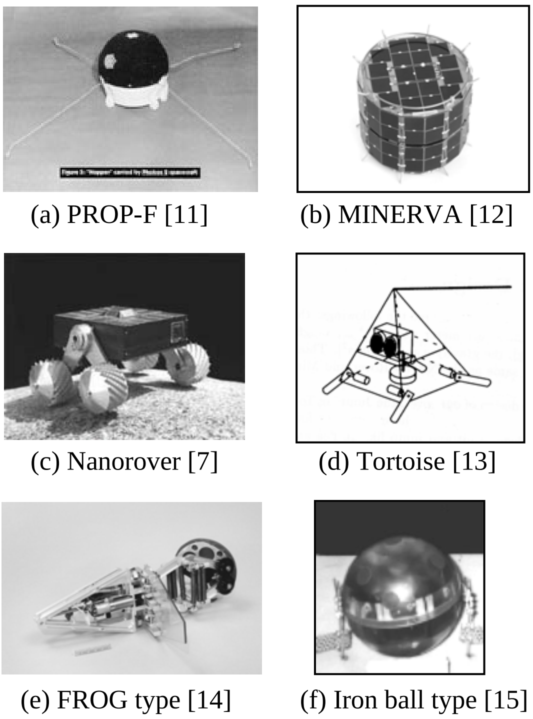
2. The torquer can also be used for attitude control during the ballistic orbit.

3. The contact force between the surface and the body is increased with help of the artificial pushing force made by the torquer, which makes the friction larger and provides for mobility at larger horizontal speeds.

4. DC motors can be used as a torquer, which are easily controlled. The imposed torque must be adjustable in order to control the hopping speed. A DC motor driven by PWM is used to provide torque adjustability.

5. For the micro-gravity environment of the small body surface, the required torque can be realised by small DC motors, which makes the robot light-weight.

Figure 4 shows the experimental results by drop tower test and the effectiveness of the hopping mechanism was confirmed.



(a) PROP-F [11]

(b) MINERVA [12]

(c) Nanorover [7]

(d) Tortoise [13]

(e) FROG type [14]

(f) Iron ball type [15]

Figure 2: Hopping robots.

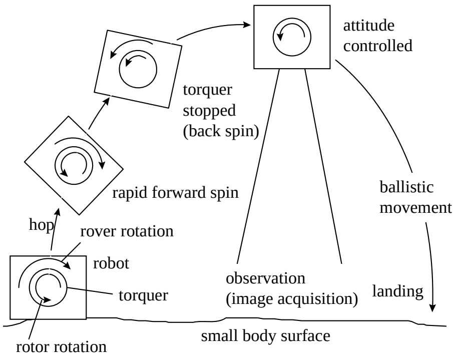


Figure 3: Hopping mechanism.

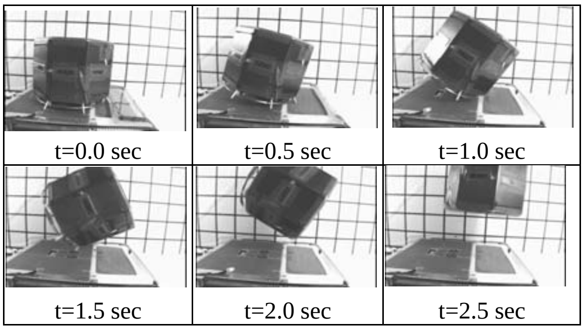


Figure 4: Micro gravity test results.

3 Intelligence of Micro Robot

The control system configuration of the micro probe robot is shown in figure 5. The system consists of CPU system, power supply system, thermal system, control system, and onboard system.

3.1 CPU System

A commercial CPU is used onboard. Based on radiation tests, the CPU is guaranteed to work in the high radiation environment of outer space. However, compared with the latest CPU commercially available, its processing speed is drastically slow with only 10 MIPS when the clock is driven at 10 MHz. The CPU system has 256 MB of ROM which preserves the program and 2 MB of RAM as a working area. Additionally it has a Flash ROM of 2 MB in which acquired data are stored. The surface of the asteroid may be a very harsh environment in terms of its temperature, which changes beyond the optimal operating range of the onboard devices. The rover has an automatic capability to wake up and shutdown the CPU depending on the surrounding temperature, so that operation is suspended during periods of extreme hot or cold.

3.2 Power Supply System

The micro probe robot is powered by solar energy, supplemented by secondary batteries, which enable a few minutes of activity with no help from the sun when the solar arrays are blocked by shadows. The solar cells are attached on each face of the rover. The power generated is dependent on the attitude of the rover, with a maximum power of 2.2 W expected when the rover is on an asteroid surface at a distance of 1 AU from the sun. Extra power is stored in the electric double layer capacitors. The onboard CPU works with the instantaneous power available from the solar arrays, but that is not sufficient for motor rotation, image capture and communication. For such operations, additional power support from the charged capacitors is necessary. Unfortunately the capacitors gradually degrade when temperatures go higher than 130 deg C.

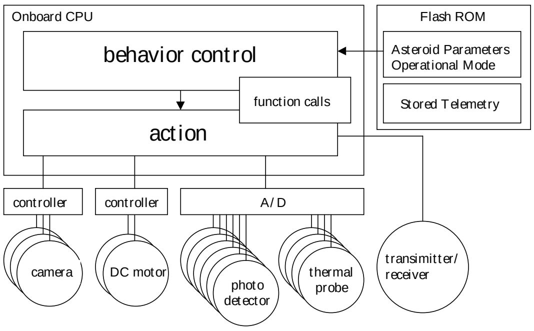


Figure 5: Hierarchy of onboard control system.

After a couple of days on the asteroid surface, the capacitors can no longer be used. However even if the capacitors are fully degraded, as long as the solar energy is supplied, the computer and communication system works in a low consumption mode, but the hop performance is degraded.

3.3 Thermal System

The operational temperature range of the onboard devices is from -50 deg C to +80 deg C. Beyond this range, the devices do not work well. The rover is equipped with an automatic capability to wake up and shutdown the onboard CPU depending on the surrounding temperature. When the temperature becomes more than +80 deg C or less than -50 deg C, the power supply to the CPU is suspended. The capacitors are not degraded if it is very cold, but will gradually degrade when the temperature goes higher than +130 deg C. After a couple of days on the asteroid surface, the capacitors can no longer be used because they will have experienced a couple of overheated daytimes. Even so, as long as the solar energy is supplied, the CPU works, but the hop performance is degraded and the rover may not move.

The onboard software also monitors the internal temperature. If this comes toward the threshold of working temperature, the actions which consume much power are suspended and the data in RAM are transferred to the Flash ROM in order not to be lost by the sudden shutdown of the CPU system. The onboard devices are covered by MLI heat insulation material. The heat capacity of the rover is very small as the temperature of the onboard devices is easily synchronised with the surrounding asteroid temperature. During the night, the asteroid temperature will fall below -100 deg C, but the temperature on the onboard devices will not drop as quickly because the heat transfer is blocked. This makes for a quick wake up of the rover at sunrise because the device temperature is not so low. Toward noon, the temperature may be rise higher than +80 deg C and the rover will take a nap for a while.

3.4 Control System

The hierarchy of the onboard control system is shown in figure 5. The lowest levels close to the devices are conducted by independent controllers. The next level is the action layer controlling the actuators, sensors and communication devices. For example, “take the picture from camera (A)”, “rotate the motor with duty pulse of X” are the commands on this level. The discrete commands to MINERVA from Earth can directly designate the commands on this layer when the CPU is working in the tele-operation mode.

The highest level is the behaviour control layer, which calls the action level commands to compose autonomous actions. Such behaviours as autonomous exploration by hopping and observation, navigation and guidance, thermal control, autonomous system, etc. are implemented in MINERVA. MINERVA has some operational modes, which differ from the program of this layer. When it is booted up, a certain Flash ROM area is first referred to in order to extract the operational mode and select the program. The switch of this operational mode is basically conducted by human command. The surface of the asteroid may be very cruel in temperature, which changes beyond the range of the onboard devices of MINERVA working well. The developed rover has the automatic capability to wake and shut down the CPU and is sensitive to the surrounding temperature so as to not work during hot or cold periods.

3.5 Onboard System

The appearance of MINERVA and the sensor allocations are shown in figure 6. MINERVA has three CCD cameras mounted on the turntable. The camera sight direction can be controlled by rotating the turntable. There are camera windows at the centre of the side faces of the rover, through which the cameras view the outside. All the cameras are commercially available and are sensitive at visible wavelengths. They have also passed radiation level tests and have been slightly tuned considering the hottest operational temperatures expected. The focal length cannot be adjusted onboard, thus the two cameras are set to look very closely and another one is to look far away.

MINERVA has six photo detectors (PDs) used as sun sensors. A single sun sensor senses the intensity of the incoming light, not the solar direction. Thus by using two outputs from the directly illuminated sensors, the solar direction can be detected. The allocation of the sun sensors is offset 90 deg from each other. At least three detectors will sense the light simultaneously and, by comparing the outputs of these detectors, the solar direction is computed. There are also pins sticking out from the body to protect the solar panels. Four of the pins are also used as thermal detectors, by which the temperature of the surface is measured.

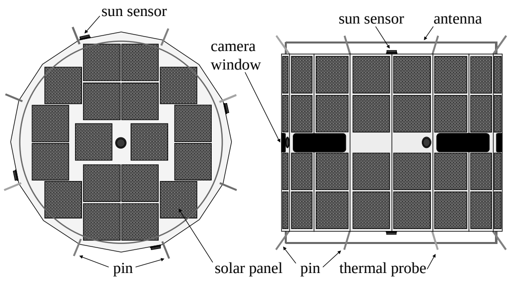


Figure 6: Allocation of onboard sensors.

4 Asteroid Exploration Robot

ISAS has developed an asteroid rover MINERVA [16] which is an experimental payload of the HAYABUSA spacecraft.

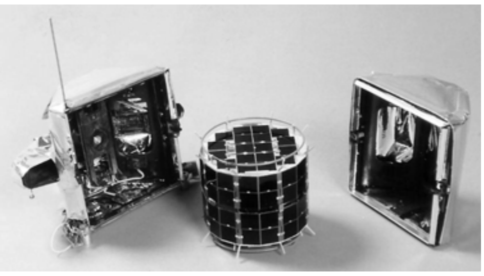


Figure 7: Flight model of MINERVA.

Figure 7 shows the flight model of MINERVA. MINERVA was expected to explore the asteroid surface after being deployed from the HAYABUSA spacecraft. Its shape is a hexadecagon-cylinder with a diameter of 12 cm and a height of 10 cm, its weight is expected to be 591g. Table 1 shows the specification of MINERVA. It is powered by solar energy, supplemented by condensers which enable a few minutes' worth of power when the sunlight is blocked by shadows. During cruise, MINERVA is also supplied with power by wired line from the OME (Onboard Mounted Equipment).

Table 1: Specifications of MINERVA.

Body size	Hexadecagonal cylinder ϕ: 120mm, height: 100mm
Weight	591g
Mobility system	Hopping ability 9cm/s (max)
Power supply	Solar cells (2.2W at 1AU) Condenser (4.6V, 20F)
Onboard CPU	32bit CPU(10MHz)
Memory	ROM:512kB, RAM:2MB Flash ROM:2MB
Temperature range	Min: -50 deg C Max: +80 deg C
Communication	9600 bps (max range: 20km)
Payloads	CCD camera × 3 Sun sensor × 6 Thermometers × 6
Power consumption	2.6 W for actuators (max.) 1.8 W for communication 8 W for camera 0.8 W for on-board computer

OME holds MINERVA during cruise. When MINERVA-deployment is triggered, the cover of OME is thrown away to allow MINERVA to be pushed towards the surface. After deployment, OME plays the role of a communication relay between MINERVA and the HAYABUSA spacecraft data recorder. Telemetry from MINERVA is received by OME via the antenna mounted on the asteroid-oriented face of the spacecraft. Commands from Earth are stored in the OME, and when communication becomes possible with MINERVA, the stored commands are transmitted. The communication bandwidth is 9,600 bps within a distance of 20 km.

On November 12th in 2005, HAYABUSA deployed MINERVA at higher velocity, and MINERVA could not arrive at the asteroid surface. MINERVA became an artificial planet and rotates around the Sun with the asteroid Itokawa. However the spacecraft could communicate with MINERVA and it was confirmed that the health of MINERVA was very good. Figure 8 shows the solar panel of HAYABUSA taken by MINERVA with autonomous functions.

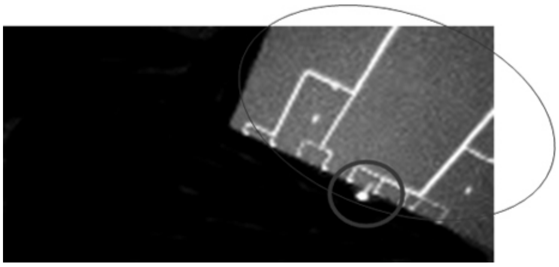


Figure 8: Image taken by MINERVA.

5 Conclusion

This paper has presented the design of an exploration robot which can move over the surface of small bodies with very weak gravity. First this paper discussed issues related to mobility in a micro-gravity environment. A new mobility system with hopping function was introduced. This paper also described the world first asteroid exploration rover "MINERVA", on board the HAYABUSA spacecraft. This paper presented the intelligence of the micro probe robot.

6 References

[1] R. Farquhar, "NEAR SO FAR", *Proceedings IAA Intl. Conf. on Low-Cost Planetary Missions*, IAA-L-0201 (2000).

[2] M.D. Rayman, P. Varghese, D.H. Lehman and L.L. Livesay, "Results from the Deep Space 1 Technology Validation Mission", *Acta Astronautica Journal*, 47, pp 475-487 (2000).

[3] M.C. Chiu, J. Veverka and E.L. Reynolds, "The Contour mission: status of implementation", *Proceedings IAA Intl. Conf. on Low-Cost Planetary Missions*, IAA-L-0205 (2000).

[4] K.L. Atkins, B.D. Martin, J. Vellinga and R. Price, "STARDUST: implementing a new manage-to-budget paradigm, *Proceedings IAA Intl. Conf. on Low-Cost Planetary Missions*, IAA-L-0202 (2000).

[5] J. Kawaguchi, K. Uesugi and A. Fujiwara, "The MUSES-C mission for the sample return: its technology development status and readiness", *Proceedings IAA Intl. Conf. on Low-Cost Planetary Missions*, IAA-L-0306 (2000).

[6] K. Wittmann, B. Feuerbacher, S. Ulamec, H. Rosenbauer, J.P. Bibring and D. Moura, "Rosetta lander in situ characterization of a comet Nucleus", *Acta Astronautica Journal*, 45, pp 389-395 (1999).

[7] B. Wilcox, "Nanorover technology and the MUSES-C mission, *Proceedings 4th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, W-6-2 (1997).

[8] T. Yoshimitsu, I. Nakatani and T. Kubota, "New mobility system for small planetary body exploration", *Proceedings IEEE International Conference on Robotics and Automation*, pp 1404-1409 (1999).

[9] P. Fiorini, S. Hayati, M. Heverly and J. Gensler, "A hopping robot for planetary exploration", *IEEE Aerospace Conference* (1999).

[10] S. Schell, A. Tretten, J. Burdick, S.B. Fuller and P. Fiorini, "Hopper on wheels: evolving the hopping robot concept", *IEEE Field and Service Robots* (2001).

[11]"Tpr_Rover", http://www.planetary.org/mars/.

[12] T. Yoshimitsu, T. Kubota, I. Nakatani, T. Adachi and H. Saito, "Hopping rover MINERVA for asteroid exploration", *Proceedings 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pp 83-88 (1999).

[13] E. Hale, N. Schara, J. Burdick and P. Fiorini, "A minimally actuated hopping rover for exploration of celestial bodies, *Proceedings IEEE Intl. Conf. on Robotics and Automation*, pp420-427 (2000).

[14] E. Hale, N. Schara, J. Burdick and P. Fiorini, "A minimally actuated hopping rover for exploration of celestial bodies, *Proceedings IEEE Intl. Conf. on Robotics and Automation*, pp420-427 (2000).

[15] K. Yoshida, "The jumping tortoise: a robot design for locomotion on micro gravity surface", *Proceedings 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pp 705-707 (1999).

[16] Y. Nakamura, S. Shimoda and S. Shoji, "Mobility of a microgravity rover using internal electro-magnetic levitation", *Proceedings 2000 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp 1639-1645 (2000).

[17] T. Yoshimitsu, T. Kubota and I. Nakatani, "Nano-rover MINERVA for deep space exploration, *Journal of Machine Intelligence and Robotics Control*, 3(3), pp 113-119 (2001).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Development of a U-learning System Featuring User-Adaptive Teaching Strategy

Yoshitaka Fujiwara[1], Hideki Yoshida[1], Yasunari Maeda[1], Tomoki Suzuki[2], Shin-ichiro Okada[3]
[1]Department of Computer Sciences
Kitami Institute of Technology, Kitami, Japan
[2]Tokyo Branch Engineering Department, Open-sourcing Division
SEISHOW Co Ltd, Tokyo, Japan
[3]Department of Computer and Information Sciences, College of Engineering,
Ibaraki University, Hitachi, Japan
fujiwara@cs.kitami-it.ac.jp, hy@cs.kitami-it.ac.jp, maedar@cs.kitami-it.ac.jp

Abstract

A ubiquitous learning system (u-learning system) allowing user access from a cellphone as well as a personal computer has been developed. The central part of the u-learning system is an on-demand learning assistance function called KUSEL, which is characterised by a user-adaptive teaching strategy, a user profile calculated by a causal network and learning material controlled by the execution plan. KUSEL is implemented by server side Java technology and Web technology. Its learning materials are made available using a Web browser on a personal computer or a special browser on a cellphone. An evaluation of the u-learning system was carried out by a local industry-government-university cooperation group. The main functions of the KUSEL user interface, including the distinctive feature of a user-adaptive teaching strategy, were assessed and it was found that these functions of KUSEL had the support of the large majority of PC and cellphone users.

Keywords: ubiquitous learning, user-adaptive teaching, causal network, Java, Web

1 Introduction

For most people using e-learning systems, the timely provision of adequate content is a strong requirement. In order to meet this requirement, adaptive educational systems have been studied [1], [2], [3].

At the same time, the authors have proposed a Java-based production system with a self-adaptive function and proposed its application to a learning assistance system [4], in which a flexible teaching strategy on the basis of the “user profile” [5] is used in order to determine the appropriate sequence of the learning materials.

In addition, with the development of mobile and wireless technologies, learning assistance systems which support self-study in a mobile environment (in town or in a vehicle such as a train) as well as in a conventional fixed environment (in an office or at home) are actively being studied. This type of e-learning system is frequently called a ubiquitous learning system (u-learning system) [6], [7], [8], [9].

Against the background described above, the authors have developed a u-learning system characterised by a user-adaptive teaching strategy and suitability for use in a mobile environment. This u-learning system is the core element of the local district activation plan of Kitami Institute of Technology and aims to contribute to the development of technical skills of local engineers as well as college students. The central part of the u-learning system is an on-demand learning assistance function called KUSEL, together with a user-adaptive teaching strategy and a data streaming function.

KUSEL provides a variety of self-study materials for technical and craft skills such as crane-operation and arc-welding, as well as certification exams for information technology engineers. These materials are available over the Internet and may be viewed using a special browser on a cellphone as well as a Web browser on a PC.

This paper first provides an overview of the u-learning system. Then the basic idea of a discriminative user-adaptive teaching strategy and the mechanism for implementing it is explained. Finally the results of an evaluation of the u-learning system, carried out by the local industry-university-government cooperation group, are described.

2 Overview of the U-learning System

The system configuration of the u-learning system is shown in figure 1. The central part of the system is an on-demand learning assistance function called KUSEL. KUSEL is designed based on Web technology and server side Java technology, and its content is available via Internet/intranet using a Web browser on a PC or a special browser on a cellphone.

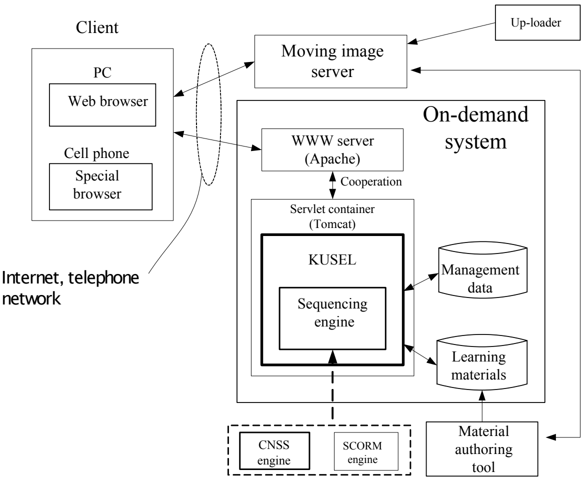


Figure 1: System configuration of the u-learning system.

The sequencing engine shown in figure 1 determines the way of providing the learning materials to the user. Currently two types of sequencing engine are available. One is constructed using the concept of "user-adaptive teaching strategy" mentioned above and the other is organised on the basis of the SCORM1.2 specification. The former engine, called CNSS, is the one we are proposing in this paper. The CNSS engine has two kinds of function: the first is for the selection of learning materials according to the execution plan and the second is for user profile calculation by means of a causal network (CN). The behaviour of the CNSS engine is described in section 3.2.

The material authoring tool in figure 1 generates learning materials, CN data and execution plans. These different types of information are grouped into appropriate "study courses" by the authoring tool. The study course means a unit of the self-study in this paper. For instance, "study course j" includes all the materials within chapter j through chapter j + n (n>=0), together with the CN data relating these materials, and execution plans available for the materials.

The learning materials are stored in KUSEL in HTML format for delivery to PC terminals and XHTML for cellphones. Each HTML/XHTML file can include text, still images and moving images. The moving image server in figure 1 is equipment for storing and distributing various types of moving image. The up-loader in figure 1 is a device for registering live or recorded moving images to the moving image server. In order to include a moving image into a HTML/XHTML file, the material authoring tool acquires the location information of the targeted moving image from the moving image server and embeds it into the HTML/XHTML file in the form of the appropriate URL. When a user takes a certain action to refer to the moving image in the HTML/XHTML file during his/her period of self-study, the corresponding URL is sent to the moving image server to download the desired image to his/her terminal display. The concept of the moving image reference from a cellphone is shown in figure 2.

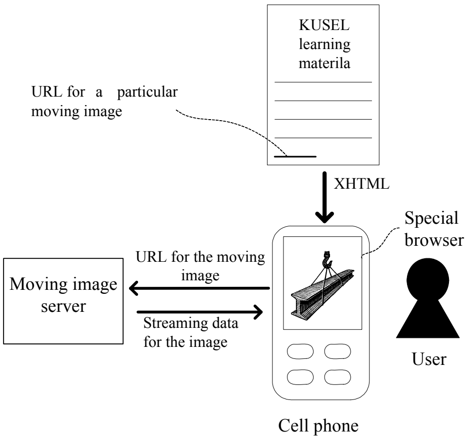


Figure 2: Concept of the moving image reference from a cellphone.

3 User-adaptive Teaching Strategy of KUSEL

This section describes how the idea of the user profile calculation by a CN [4], [5] is implemented in KUSEL. First, the user profile calculation in KUSEL is summarised in section 3.1. Then the user-adaptive teaching operation based on both user profile and execution plan is described in section 3.2.

3.1 Calculation of User Profile by a Causal Network

The learning materials of each study course are arranged in a tree structure and are provided to the user in a hierarchical manner: chapter-explanation → section-explanation → exercise problems. The graph structure of the CN used for the "user profile" calculation is generally expressed in a DAG (directed acyclic graph). However in KUSEL, the complete DAG is approximated by a tree. Therefore, the structure of the CN becomes identical with that of the learning materials of the study course. Each node in the CN has two states, "understood" and "not-understood." The state "understood" (or "not-understood") indicates that the corresponding node in the learning materials (chapter-explanation or section-explanation or exercise problem) is understood (or not-understood) by the user. In addition, the probability of the state "understood" of each node is defined as the "degree of understanding" of the node.

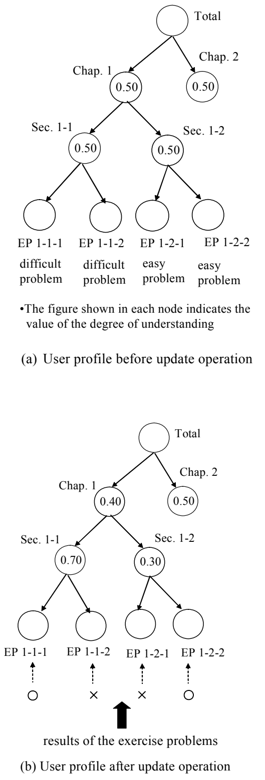
Then the set of values of the "degree of understandings" of all the nodes is interpreted as expressing the characteristics of the learner's understanding of the targeted study course. This characteristic is defined as the "user profile".

In addition, the user profile defined above is similar to the “knowledge vector” concept [1].

The degree of understanding of each node is calculated in the following way.

(1) CN initialisation: A set of conditional probabilities is given to every parent-child node pair of the CN. For instance, probabilities such as P (Ni-j = ○ | Ni = ○) and P (Ni-j = ○ | Ni = ×) are given between parent node Ni and its child node Ni-j. In the above expressions, symbol ○ and symbol × denote the state “understood” and the state “not-understood,” respectively. In addition, such probability as P (Ni-j = × | Ni = ○) is obtained by the formula (1 − P (Ni-j = ○ | Ni = ○)). Then the probabilities of the states of all the nodes are calculated by the specific probability propagation algorithm under the condition of “no instantiated nodes” [10]. Thus the initial “user profile” is obtained as the set of “degree of understandings” (DU) of all the nodes.

(2) User profile update: All the exercise problems belonging to all the sections of a certain chapter are executed by the user and their results (correct answer or incorrect answer) are stored. Then these results are reflected into the CN in the form of the “instantiated values” of the corresponding leaf nodes. Then the “degree of understanding” of the remaining nodes is updated by the probability propagation algorithm, resulting in an updated “user profile.”



(a) User profile before update operation

(b) User profile after update operation

Figure 3: Manner of updating the degree of understanding of nodes.

The manner of updating the degree of understanding of each node is illustrated in figure 3. In figure 3, the effect of the instantiation of the leaf node to its parent node is dependent on the value of the conditional probability given to the parent-child pair. Therefore, the updated values of the degree of understanding of sec. 1-1 and sec. 1-2 are different.

3.2 User-Adaptive Teaching Operation by the CNSS Engine

An execution plan used in KUSEL determines what learning materials should be presented to the user next, according to the values of the corresponding part of the user profile. The selection of the flow of materials based on the execution plan is shown in figure 4.

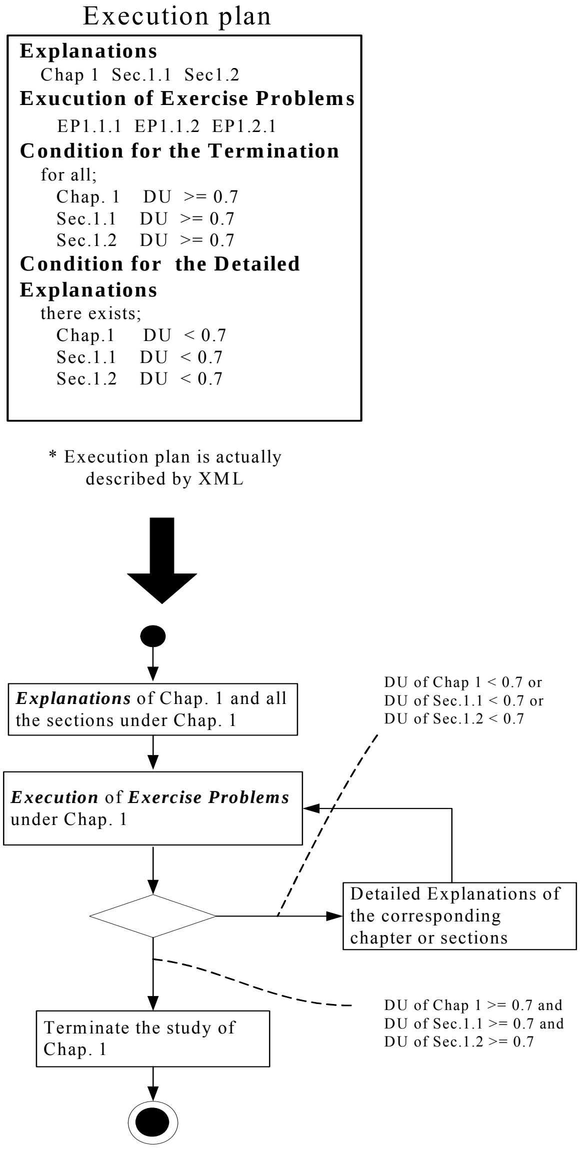

Execution plan
Explanations
Chap 1 Sec.1.1 Sec1.2
Exucution of Exercise Problems
EP1.1.1 EP1.1.2 EP1.2.1
Condition for the Termination
for all;
Chap. 1 DU >= 0.7
Sec.1.1 DU >= 0.7
Sec.1.2 DU >= 0.7
Condition for the Detailed Explanations
there exists;
Chap.1 DU < 0.7
Sec.1.1 DU < 0.7
Sec.1.2 DU < 0.7
* Execution plan is actually described by XML
Explanations of Chap. 1 and all the sections under Chap. 1
Execution of Exercise Problems under Chap. 1
DU of Chap 1 < 0.7 or
DU of Sec.1.1 < 0.7 or
DU of Sec.1.2 < 0.7
Detailed Explanations of the corresponding chapter or sections
DU of Chap 1 >= 0.7 and
DU of Sec.1.1 >= 0.7 and
DU of Sec.1.2 >= 0.7
Terminate the study of Chap. 1

Figure 4: Flow of learning materials selection based on the execution plan

Flexible material selection is achieved by use of the “degree of understanding” of the related CN nodes as branching conditions in the plan. The actual user-adaptive teaching operation based on the execution plan is realised in the CNSS engine. The CNSS engine is implemented using a Java-based production system [4]. That is, an execution plan is put into WM (Working Memory) and its interpretation logic is put into an inference unit in the form of a set of IF-THEN rules. The organisation of the CNSS engine is illustrated in figure 5.

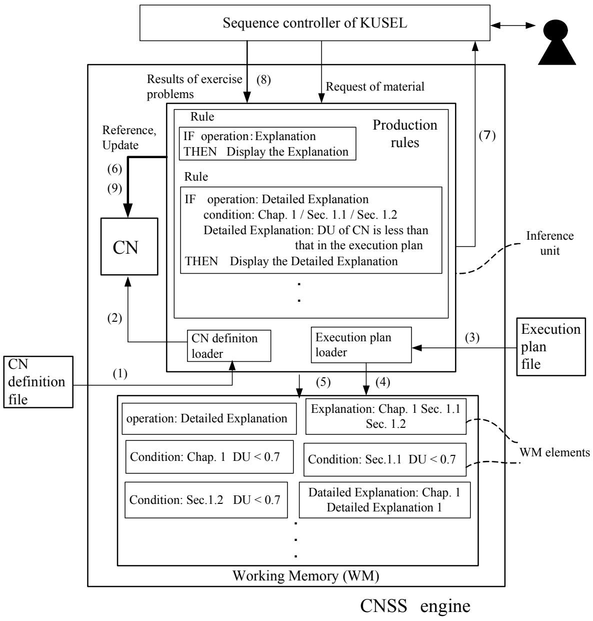

Sequence controller of KUSEL
Results of exercise problems
(8)
Request of material
Reference, Update
(6)
(9)
Rule
IF operation: Explanation
THEN Display the Explanation
Production rules
(7)
Rule
IF operation: Detailed Explanation
condition: Chap. 1 / Sec. 1.1 / Sec. 1.2
Detailed Explanation: DU of CN is less than that in the execution plan
THEN Display the Detailed Explanation
CN
Inference unit
(2)
CN definiton loader
Execution plan loader
(3)
Execution plan file
CN definition file
(1)
(5)
(4)
operation: Detailed Explanation
Explanation: Chap. 1 Sec. 1.1 Sec. 1.2
Condition: Chap. 1 DU < 0.7
Condition: Sec.1.1 DU < 0.7
WM elements
Condition: Sec.1.2 DU < 0.7
Datailed Explanation: Chap. 1 Detailed Explanation 1
Working Memory (WM)
CNSS engine

Figure 5: Organisation of the CNSS engine.

The behaviour of the CNSS engine is summarised below.

(1) Initialisation: First, the CNSS engine reads in a CN definition file and generates a corresponding CN structure ((1) and (2) in figure 5.). Then the engine reads in a certain execution plan specified by the target user and reconstructs it as a set of WM elements ((3) and (4) in figure 5.). The logic of interpreting the plan is pre-loaded in the CNSS engine in the form of IF-THEN rules. The WM contents in figure 5 show a part of the execution plan in figure 4.

(2) User-adaptive teaching operation: When the CNSS engine receives a request from the Sequence controller of KUSEL (this is not shown explicitly in figure 1) as to what materials are to be shown next, it determines the most appropriate material based on the execution plan and the current user profile, sending the “name information” of the chosen material to the sequence controller ((5), (6) and (7) in figure 5). The sequence controller then retrieves the appropriate HTML/XHTML file, corresponding to the name information, from the learning materials database (shown in figure 1), and displays it on the user screen. This cooperative work by the CNSS engine and the sequence controller is repeated until all the exercise problems of the sections within the chapter have been completed. Then CNSS engine receives the results of the exercise problems from the sequence controller, sending them to the CN which is within it. Next the CN updates the value of the degree of understanding of each node using the probability propagation algorithm ((8) and (9) in figure 5).

The manner of the CN update has already been described in figure 3. Finally, the CNSS engine determines what materials should be shown next, based on the execution plan and the updated user profile. Then the name information of the selected materials is returned to the sequence controller ((5), (6) and (7) in figure 5).

4 Evaluation of the U-learning System

The functions of the KUSEL user interface were evaluated by members from the local industry-government-university cooperation group. The evaluation was performed separately for two different types of client environment, PCs and cellphone terminals. The assessment was carried out in the form of a questionnaire. In the questionnaire entries were specified for different categories: the understandability of the window design, the ease of operation, and the usefulness of the user-adaptive teaching strategy. The assessors were required to choose one of the five possible answers from grade 1 (very bad) to grade 5 (very good) for each entry. The numbers of assessors for PC and cellphone terminals were 26 and 16, respectively.

From the results of the questionnaire survey it was clear that the main functions of the KUSEL user interface had the support of the majority of PC and cellphone users. In other words, the average rate of positive answers showing grade 3 or higher for each category was: (1) about 80% for both PC and cellphone users for the category of the understandability of the window design, (2) about 64% for PC users and 77% for cellphone users for the category of ease of operation, (3) about 90% for PC users and 75% for cellphone users for the category of usefulness of the user-adaptive teaching strategy.

The percentage distribution of the result for the user-adaptive teaching strategy is shown in table 1.

Table 1: Percentage distribution of the 5-grade evaluation of the user-adaptive teaching.

	Grade 5	**Grade 4**	**Grade 3**	**Grade 2**	**Grade 1**
PC	5	33	52	10	0
Cellphone	6	19	50	25	0

5 Conclusion

A ubiquitous learning assistance system that allows user access from a cellphone as well as from a personal computer has been proposed. The central part of the u-learning system is an on-demand learning-assistance function called KUSEL, which has the following features: (1) control of learning materials based on the execution plan, and (2) a user-adaptive teaching strategy based on a user profile.

The functions of the KUSEL user interface have been evaluated by the local industry-university-government cooperation group. The assessment was conducted in the form of questionnaire for the categories of the understandability of the window design, the ease of operation and the usefulness of the user-adaptive teaching strategy.

As a result, it was clear that the main functions of the KUSEL user interface received the support of the majority of both PC and cellphone users.

Hereafter, the improvement of the u-learning system is scheduled to increase the convenience of the user interface on the basis of the comments and suggestions, which are given through the questionnaire survey.

6 Acknowledgements

The development of the u-learning system was partially supported by the Ministry of Education, Culture, Sports, Science and Technology in Japan.

7 References

[1] N. Henz and W. Nejdl, “Adaptation in open corpus hypermedia”, *International Journal of Artificial Intelligence in Education,* 12(4), pp 325-350 (2001).

[2] G. Weber and P. Brusilovsky, “An adaptive versatile system for web-based instruction”, *International Journal of Artificial Intelligence in Education,* 12(4), pp 351-384 (2001).

[3] G. Weber and P. Brusilovsky, “Adaptive and intelligent web-based educational systems”, *International Journal of Artificial Intelligence in Education*, 13, pp 156-169 (2003).

[4] Y. Fujiwara, S. Okada, T. Suzuki, Y. Ohnishi and H. Yoshida, “Self-adaptive Java production system and its application to a learning assistance system”, *IEICE Trans. Inf. and Syst.,* E87-D(9), pp 2186-2194 (2004).

[5] Y. Fujiwara, S. Okada, N. Takakura and B. Matsuzawa, “Distributed knowledge-base execution system with user adaptability for distance learning system”, *IPSJ Journal,* 42(6), pp 1401-1411 (2001).

[6] W. Beuschel, “Ubiquitous e-learning: are we there yet?”, *Proceedings 3rd IEEE International Conference on Advanced Learning Technologies*, Athens, Greece, pp 414-415 (2003).

[7] H. Ogata and Y. Yano, “Supporting knowledge awareness for a ubiquitous CSCL”, *Proceedings E-Learn 2003*, Phoenix, AZ, pp 2362-2369 (2003).

[8] K. Sakamura and N. Koshizuka, “Ubiquitous computing technologies for ubiquitous learning”, *Proceedings IEEE International Workshop on Wireless and Mobile Technologies in Education*, Tokushima, Japan, pp 11-20 (2005).

[9] G. Zhan and Q. Jin, “Research on collaborative service solutions in a ubiquitous learning environment”, *Proceedings 6th International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dalian, China, pp 804-806 (2005).

[10] R. Neapolitan, *Probabilistic Reasoning in Expert Systems: Theory and Algorithms,* John Wiley & Sons, New York (1990).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Active Vibration Control of Flexible-Link Manipulators using Piezoelectric Sensors and Actuators

Krishnarao D. Dhuri and P. Seshu
Department of Mechanical Engineering
Indian Institute of Technology, Bombay, India
kddhuri@iitb.ac.in

Abstract

Use of piezoelectric patches for active vibration control of flexible manipulators is shown. Active vibration control has been simulated on single-link and five-link manipulator using PD controller. The manipulator is modelled using finite element approach. While derivative control is an effective strategy for vibration reduction, its use in conjunction with control of rotary actuators (which give rigid body motion), will give the best performance.

Keywords: flexible manipulator, piezoelectric actuator, active vibration control (AVC)

1 Introduction

Research on modelling and control of flexible manipulators has attracted considerable attention due to increasing emphasis on cost-effectiveness of such structures and high manoeuvre speeds. Achieving a high manoeuvre speed entails light weight structures raising issues of flexibility. Various attempts have been made to minimise the positioning error taking into account these flexibility effects.

In robotics, positioning of end effector has mainly been achieved by controlling the motors. But in recent years owing the developments in material sciences, smart materials have been used for vibration control to improve the end-effector positioning accuracy. Piezoelectric actuators have been successfully used for actively controlling the vibrations of stationary and rotating structures ever since Bailey and Hubbard's [1] initial effort of active vibration control of a cantilever beam. Baz and Ro [2] have used active constrained layer damping (ACLD) to control the vibrations of a rotating beam. Choi and Han [3] experimentally demonstrated the active vibration control of rotating beams. Besides the above-mentioned works, there are numerous research works on the vibration control of flexible stationary and rotating structures using piezoelectric actuators. Researchers have also tried piezoelectric materials for improving the positioning accuracy of the manipulators [4-7]. Kermani et al. [4] theoretically and experimentally demonstrated effective use of the PZT patches for trajectory tracking of a single link manipulator whose base is having linear motion degrees of freedom. Shin and Choi [6] have used collocated piezoelectric sensor and actuator for improving positional accuracy of two-link flexible manipulator.

Finite element (FE) analysis of linkages is well investigated. Midha et al. [8] have used instantaneous structural approach. But this approach doesn't give good results at higher speeds of manipulator manoeuvring. Turcic and Midha [9] have shown that Coriolis and centrifugal force cause significant changes in dynamics at high-speed operations. The above two approaches work well when the joint stiffness is relatively high or in case of mechanisms with a flywheel. But when the joint stiffness reduces, then dependency of rigid body motion and elastic degrees of freedom becomes significant. Nagarajan and Turcic [10] have studied this dependency. Subrahmanyan and Seshu [11] have shown that the elastic vibrations of an inertia invariant, parallelogram flexible five bar mechanism can be modelled using linear time invariant differential equations.

This paper demonstrates improvement in positioning accuracy of end effector, mounted on free end of multi-link flexible manipulator using piezoelectric actuators. The results are presented for single-link and parallelogram five-bar manipulators. Finite element model of multi-link manipulator is developed applying instantaneous structural approach. Proportional and derivative (PD) control law is used for implementing active vibration control of flexible manipulator using piezoelectric actuators.

2 Mathematical Modelling

A multi-link manipulator with piezoelectric sensor and actuator is shown in figure 1. In this study, piezo-film (PVDF) and piezoceramic (PZT) are used as sensor and actuator respectively for implementing active vibration control of manipulator. The dynamical equations of flexible manipulator can be derived using Lagrange's equation,

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = Q_i \qquad (i = 1, 2, \ldots) \qquad (1)$$

where q_i and Q_i are generalised coordinates and generalised forces acting in the directions of the generalised coordinates. T and U are the kinetic energy and potential of the system respectively. Superscripts dots, primes denote the differentiation with respect to time (t) and spatial coordinate respectively and its number shows the order of differentiation. A typical link is regarded as a beam element having uniform cross section along its length.

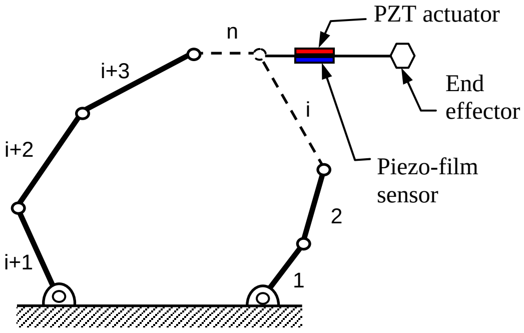


Figure 1: A multi-link manipulator with Piezo sensor/actuator.

Total kinetic energy of n link-manipulator, each link having length L_i is given by:

$$T = \sum_{i=1}^{n} \int_{0}^{L_i} \frac{1}{2} \rho \mathrm{V}^2 dx \tag{2}$$

where dx is infinitesimal part of the link with mass density ρ, having velocity, $\mathrm{V} = [\dot{u} \quad \dot{v}]$ with u and v being axial and transverse displacements of links respectively. Total potential energy of the elastic linkage is given by,

$$U = \sum_{i=1}^{n} \left(\int_{0}^{L_i} \frac{1}{2} EIv''^2 dx + \int_{0}^{L_i} \frac{1}{2} EAu'^2 dx + \int_{0}^{L_i} \frac{1}{2} \sigma^0 v'^2 dx \right) \tag{3}$$

where E, I, A and σ^0 are elasticity modulus, area moment of inertia, cross sectional area and axial stresses developed due centrifugal forces in high speed manoeuvring of manipulators.

2.1 Finite Element Modelling

Flexible manipulator is modelled by connecting together a series of beam elements in such a manner that the linkage can undergo translational, rotational and elastic deformations. In this section, the relations between different displacements are found out in Cartesian coordinate systems and various characteristic system matrices are derived. Figure 2 shows a typical 2-noded beam element [12] in its rigid and elastically deformed configurations. Three Cartesian coordinate systems are used; inertial frame $R(OXY)$, a moving frame $R_1(I\, \bar{x}\, \bar{y})$, whose x-axis is parallel to the beam element axis and its origin is at the left end of the beam element. Third Cartesian coordinate system, $R_2(Oxy)$ is parallel to second one and its origin is coinciding with that of the inertial frame, which is used for simplifying the mathematical manipulations. Consider u_1 through u_6 are the elastic deformations of the beam element at nodal points as shown in figure. Following relations can be expressed in $R(OXY)$ from figure 2,

$$\begin{aligned} X_{A'} &= X_A + u_1 \cos\theta - u_2 \sin\theta \\ Y_{A'} &= Y_A + u_1 \sin\theta + u_2 \cos\theta \\ \theta_{A'} &= \theta + u_3 \end{aligned} \tag{4}$$

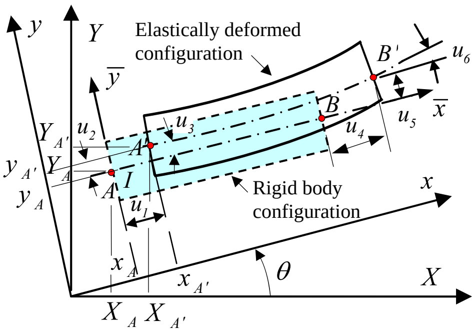


Figure 2: Elastic beam element.

After differentiating (4) twice wrt time and simplifying, we get the nodal accelerations in $R_1(O\,x\,y)$ as given in (5). Equation (5) can be rewritten as in (6), where $\ddot{u}_a$, $\ddot{u}_r$, $\ddot{u}$, $\ddot{u}_n$, $\ddot{u}_c$ and $\ddot{u}_t$ are absolute, rigid body, elastic, normal, Coriolis and tangential accelerations respectively [8].

$$\begin{Bmatrix} \ddot{x}_{A'} \\ \ddot{y}_{A'} \\ \ddot{\theta}_{A'} \\ \ddot{x}_{B'} \\ \ddot{y}_{B'} \\ \ddot{\theta}_{B'} \end{Bmatrix} = \begin{Bmatrix} \ddot{x}_A + \ddot{u}_1 - u_1\dot{\theta}^2 - 2\dot{u}_2\dot{\theta} - u_2\ddot{\theta} \\ \ddot{y}_A + \ddot{u}_2 - u_2\dot{\theta}^2 - 2\dot{u}_1\dot{\theta} - u_1\ddot{\theta} \\ \ddot{\theta}_A + \ddot{u}_3 + 0 + 0 + 0 \\ \ddot{x}_B + \ddot{u}_4 - u_4\dot{\theta}^2 - 2\dot{u}_5\dot{\theta} - u_5\ddot{\theta} \\ \ddot{y}_B + \ddot{u}_5 - u_5\dot{\theta}^2 - 2\dot{u}_4\dot{\theta} - u_4\ddot{\theta} \\ \ddot{\theta}_B + \ddot{u}_6 + 0 + 0 + 0 \end{Bmatrix} \tag{5}$$

$$\{\ddot{u}_a\} = \{\ddot{u}_r\} + \{\ddot{u}\} + \{\ddot{u}_n\} + \{\ddot{u}_c\} + \{\ddot{u}_t\} \tag{6}$$

The inertia forces caused due to these accelerations, produce the vibrations in manipulator arms while manoeuvring from one position to another.

2.1.1 Stiffness Matrix

The total stiffness consists of standard beam stiffness due to substrate beam and piezo and the geometric stiffness matrix pertaining to the effects of rotation. The standard stiffness matrix [12], $[k_E]$ of the element is modified for present study. For an element representing the substrate beam with piezoelectric patches, flexural rigidity is given by $EI = E_b I_b + E_p I_p + E_f I_f$ and for other elements, $EI = E_b I_b$; where E and I are modulus of elasticity and area moment of inertia of each layer as referred to the mid-plane of combined element. Subscripts p, f and b refer to PZT actuator, PVDF sensor and substrate beam respectively. Similarly, for an element representing substrate beam with piezo patches, EA in the stiffness matrix is calculated for equivalent section; where A is the cross-section area of the layer. As the stiffness of PVDF is very less compared to substrate beam and PZT, its effect is neglected.

2.1.1.1 *Geometric Stiffness Matrix due to Rotation*

The additional strain energy due to presence of axial stress σ_x^0 caused by rotation is given by

$$[k_G] = \int_0^{l_e} [G]^T \left[\sigma_x^0\right][G] A dx \tag{7}$$

where ***G*** is defined by, $\left\{\frac{\partial v}{\partial x}\right\} = [G]\{q_e\}$ (8)

where $\{q_e\}$ consists of nodal degrees of freedom. The stresses incurred in the manipulator links due to rotation are approximated from a quasi-static force analysis, wherein the displacements and internal forces are calculated due to inertial and external forces on the system [9].

2.1.2 Mass and Damping Matrices

The standard consistent mass matrix [12] is modified for the present study considering the effect of piezoelectric patches on the lines of stiffness matrix in section 2.1.1.

Modelling of damping is one of the complex parts of system modelling, as the mechanism of energy loss is very complicated. Friction at the joints and structural damping are important from manipulator analysis point of view. Effect of joint stiffness and structural damping assumed to be negligibly small in comparison to the control effects of the piezo actuator.

2.1.3 Piezoelectric Patch Modelling

The bending moment produced from the ith piezoceramic actuator bonded on flexible link surface due to the control input voltage $V_{Ci}(t)$ can be obtained by considering force equilibrium in the axial direction [13]. The moment M_i applied by the piezo actuator on the flexible link about its neutral axis is determined by,

$$M_i = -\varepsilon_p E_p t_p b\left[t_f + t_p/2 + t_b - D_n\right] = c_i V_{ci}(t) \tag{9}$$

where, t and b are thickness and width respectively. ε_p is the strain induced in the piezoceramic due to voltage applied across it, which is given by, $\varepsilon_p = d_{31} V_{ci}(t)/t_p$, where d_{31} is the piezoelectric strain constant. The constant c_i implies the bending moment per unit applied voltage, which is calculated from the geometric and material properties of the substrate link and piezo patch combination [14]. D_n is the distance from the bottom of the piezofilm sensor to the neutral axis of the combined element.

2.1.4 Final Matrices after Assembly

After getting the elemental characteristic matrices, assembly of elemental matrices in global co-ordinate system is the next step of FE analysis. The joint between two links need to be modelled properly during assembling the elemental matrices. The sharing node at the joint is having common translational degree of freedom while rotation is different corresponding to each node of either element. Also the elemental matrices derived in $R_l(O\,x\,y)$, need to be transferred to the common inertial frame $R(OXY)$ before assembly. It is carried out using standard procedures [12].

The final equation for the system after assembly of all the elements is of the form,

$$[M]\{\ddot{U}\} + [C]\{\dot{U}\} + [K]\{U\} = -[M]\{\ddot{U}_r\} \tag{10}$$

The coefficient matrices, $[M]$, $[C]$ and $[K]$ are functions of instantaneous linkage geometry.

Subspace iteration approach [12] is used for solving the Eigen value problem to get the natural frequencies and mode shapes.

2.1.5 Validation of FE model

Finite element formulation and code is validated comparing the results with an example given in [16]. The first three, configuration dependent natural frequencies of the 4-bar mechanism are given in table 1. (The system parameters used for the simulation are given in table 2 of [16]). The results generated from current analysis match very well with the literature.

Table 1: First three natural frequencies (rad/s) of flexible 4-bar mechanism. Values in parentheses are from [16].

Crank angle (deg)	1st natural frequency	2nd natural frequency	3rd natural frequency
90	293 (295)	317 (319)	687 (691)
180	298 (300)	311 (314)	571 (574)

3 Controller

Proportional and derivative (PD) controller is used for actuating the piezoceramic actuator for suppressing the vibration of flexible link. The control voltage V_C generated by applying a PD control law to piezo sensor voltage V_S is given by,

$$V_C = -K_p V_S - K_d\, dV_S/dt \tag{11}$$

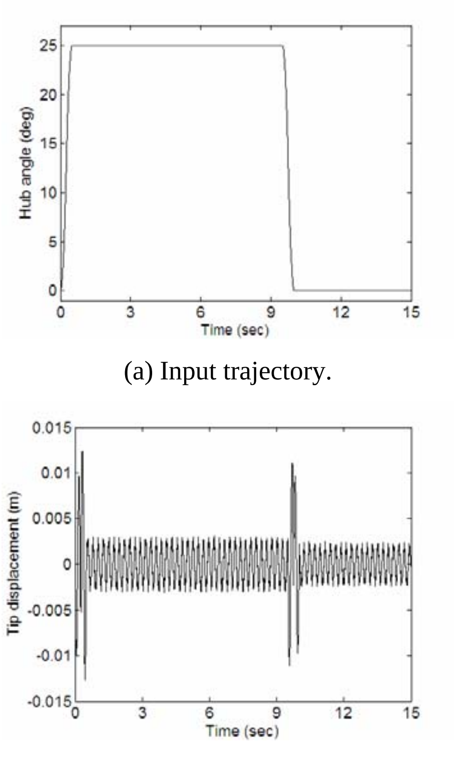
where K_p and K_d are the proportional and derivative gain respectively. The V_S is calculated from [15],

$$V_S = \frac{-k_{31}^2 D_d b}{g_{31} C} \sum_{i=i_s}^{i_f} f\left(x\right) v'' dx \tag{12}$$

where D_d is the distance from the beam neutral axis to the sensor surface and $f\left(x\right)$ is the distribution shape function of the sensor. The sensor is mounted between elements i_s and i_f. Also, k_{31} is electro-mechanical coupling factor and g_{31} is piezoelectric voltage constant and C is the capacitance of sensor. FE code and controller is implemented using MATLAB® 7.[1]

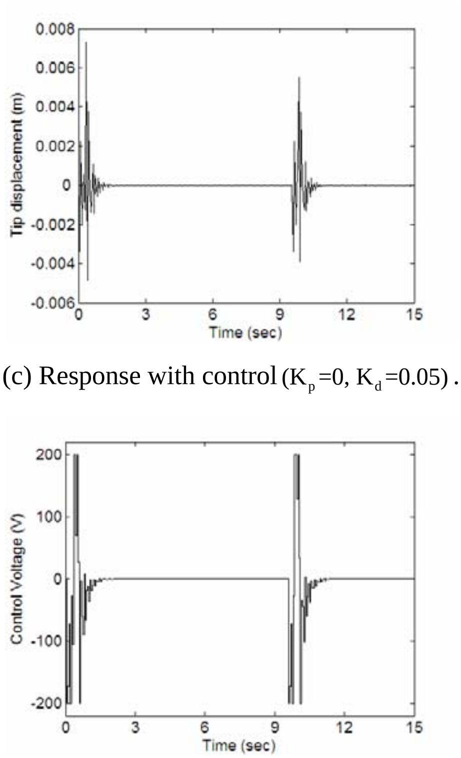
4 Results and Discussion

In order to check the feasibility of improving positioning accuracy using piezoelectric actuator, simulations are carried out on single-link and five-link manipulators. The vibration signal is measured using piezo-film sensor. Simple pick and place trajectory as shown in figure 3a is used. The manipulator is required to reach the commanded position of 25^0 and return to the initial position of 0^0 with 10s interval. In order to simplify the problem, the control of actuator for rigid body motion is not modelled in this study; instead realistic trajectory resulting after controlling the rigid body actuator is considered for giving rigid body acceleration as excitation. As shown in figure 3a, the trajectory reaches the desired position in 0.5s which is achieved by constant acceleration in the first 0.25s and deceleration in the next 0.25s.



(a) Input trajectory.

(b) Response without control.



(c) Response with control $(K_p = 0,\ K_d = 0.05)$.

(d) Control voltage.

Figure 3: Tip response and control effort for single-link manipulator without tip mass (dimensions L=30cm, b=25mm, t=0.5mm and material props Eb=65x109N/m2, ρ =2890 Kg/m3)

The properties of piezoelectric material used are given in table 2. The control voltage amplitude is restricted to +/-200V considering electric field limitation for given thickness of PZT. Piezo sensor/actuator locations are chosen nearer to hub as these locations are effective for controlling first mode which is generally dominant in the response [17].

Table 2: Material properties and geometric parameters of piezoelectric materials.

Property/Parameter	**PVDF**[2]	**PZT**[3]
E (N/m^2)	2.0 x 10^9	6.6 x 10^{10}
ρ (Kg/m^3)	1780	7500
d_{31} (m/V)	23x10^{-12}	-190x10^{-12}
Thickness (mm)	-	0.35

4.1 Single-Link Manipulator

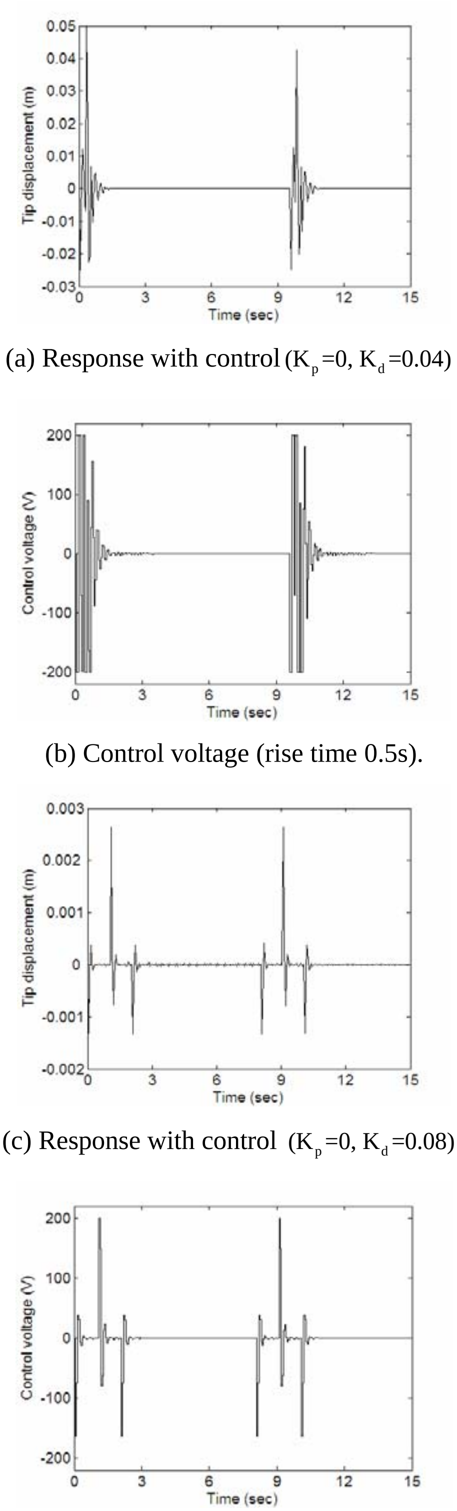
Simulations are carried out for both with and without tip mass cases to simulate the end-effector. Tip mass equal to the weight of the link is assumed to be mounted at free end of manipulator. The collocated sensor and actuator are of length 4cm mounted on either side of link in such a way that their centre line

[1] MATLAB®7.0, The MathWorks, Inc., Massachusetts, USA

[2] Measurement Systems Inc. 2002

[3] Piezosystems Inc. 2002

is at 9cm from the hub. Figures 3 and 4 show responses at tip of the manipulator and control efforts for the cases with and without tip mass respectively. System parameters are given in the caption of figure 3.



(a) Response with control $(K_p=0, K_d=0.04)$.

(b) Control voltage (rise time 0.5s).

(c) Response with control $(K_p=0, K_d=0.08)$.

(d) Control voltage (rise time 2s).

Figure 4: Tip response and control effort for single link manipulator with tip mass.

The observations from figures 3 and 4 can be summarised as follows:

1. In all the cases considered, the end-effector vibration level reduced to almost zero within 1s.
2. PZT actuators could damp out residual vibrations but could not control the jerks.
3. The end-effector vibration level is much higher when its mass is included in the model. Correspondingly, control efforts are more for this case.
4. Though PD control is implemented, only derivative gain has been found to be effective for vibration reduction.
5. From figures 4a and 4c, vibration can be reduced to a great extent by increasing the rise time of the desired trajectory. Obviously control efforts are lower for larger rise time.

4.2 Parallelogram Five-Link Manipulator

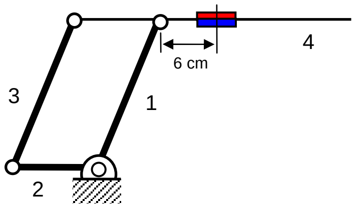
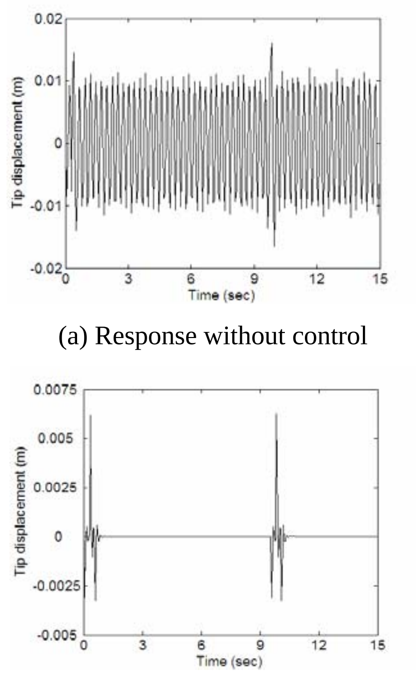


Figure 5: Parallelogram five-bar manipulator (L_1=0.33m, L_2=0.136m, L_3=0.33m and L_4=0.5m, $A_{rigid}=3x10^{-4}m^2$, $I_{rigid}=1x10^{-8}m^4$, $E_{rigid}=2x10^{11}N/m^2$, ρ_{rigid}=7840, $A_{flex}=2x10^{-5}m^2$, $I_{flex}=10.67x10^{-11}m^4$, $E_{flex}=65x10^9N/m^2$, ρ_{flex}= 2890).



(a) Response without control

(b) Response with control $(K_p=0, K_d=0.05)$.

Figure 6: Five-bar manipulator tip response.

For simplicity, only link 4 is considered flexible while links 1, 2 and 3 are rigid as shown in figure 5.

Collocated PZT and PVDF patches of length 4cm are mounted on either side of flexible link, whose centres lie at 6cm from the joint of links 1 and 4. The responses at free end of link-4 are given in figure 6. The tip mass is not considered for this case. System parameters are given in the caption of figure 5. Active vibration control using PZT actuators has been found very effective for improving the positioning accuracy of five-bar manipulator.

5 Conclusion

Active vibration control of single and multi-link manipulator using PZT actuator is demonstrated by implementing PD control with FE approach. This approach effectively controls the vibration level thus improving the tip positioning accuracy. But it could not limit the jerks during manoeuvring. The performance can be enhanced by effectively controlling the rotary actuators, which produces the rigid body motion of manipulator links in conjunction the piezoelectric actuators. In slewing, the rise time plays a critical role.

6 References

[1] T. Bailey and J.E. Hubbard Jr., “Distributed piezoelectric-polymer active vibration control of a cantilever beam”, *Journal of Guidance*, 8(5), pp 605-611 (1985).

[2] A. Baz and J, Ro, “Vibration control of rotating beams with active constrained layer damping”, *Smart Materials and Struct.*, 10, pp 112-120 (2001).

[3] S.B. Choi and M.S. Han, “Letter to the editor: vibration control of a rotating cantilevered beam using piezoactuators: experimental work,” *Journal of Sound and Vibration*, 277, pp 436-442 (2004).

[4] M.R. Kermani, M. Moallem and R.V. Patel, “Study of system parameters and control design for a flexible manipulator using piezoelectric transducers”, *Smart Materials and Struct.*, 14, pp 843-849 (2005).

[5] D. Sun, J.K. Mills, J. Shan and S.K. Tso, “A PZT actuator control of a single-link flexible manipulator based on linear velocity feedback and actuator placement”, *Mechatronics*, 14, pp 381-401 (2004).

[6] H.C. Shin and S.B. Choi, “Position control of a two-link flexible manipulator featuring piezoelectric actuators and sensors", *Mechatronics*, 11, pp 707-729 (2001)

[7] S.B. Choi, C.C. Cheong, B.S. Thompson and M.V. Gandhi, “Vibration control of flexible linkage mechanisms using piezoelectric films”, *Mech. Mach. Theory*, 29(4), pp 535-546 (1994).

[8] A. Midha, A.G. Erdman and D.A. Frohrib, “Finite element approach to mathematical modeling of high speed elastic linkages”, *Mech. Mach. Theory*, 13(6), pp 603-616 (1978).

[9] D.A. Turcic and A. Midha, “Generalized equations of motion for the dynamic analysis of elastic mechanism systems”, *ASME J. Dynam. Systems, Measmt Control*, 106(4), pp 243-254 (1984).

[10] S. Nagarajan and D.A. Turcic, “Lagrangian formulation of the equations of motion for elastic mechanisms with mutual dependence between rigid body and elastic motions, Part 1: Element level equations and Part 2: System equations”, *ASME J. Dynam. Systems, Measmt Control*, 112(2), pp 203-223 (1990).

[11] P. K. Subrahmanyan and P. Seshu, “Dynamics of a flexible five bar manipulator”, *Computers & Structures*, 63(2), pp 283-294 (1997).

[12] K.J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ (1982).

[13] T. Bailey, J.E. Hubbard Jr., “Distributed piezoelectric-polymer active vibration control of a cantilever beam”, *J. Guidance, Control, Dynamics*, 8(5), pp 605-611 (1985).

[14] S.B. Choi and H.C. Shin, “A hybrid actuator scheme for robust position control of a flexible single link manipulator”, *J. Robotic Systems*, 13(6), pp 359-370 (1996).

[15] S. Miller and J. Hubbard Jr., “Observability of a Bernoulli-Euler beam using PVF2 as a distributed sensor”, *Proceedings 7th Conf. Dynamics and Control of Large Structures, VPI and SU*, Blacksburg, VA, pp 375-390 (1987).

[16] D.A. Turcic and A. Midha, “Dynamic analysis of elastic mechanism systems. Part I: Application”, *ASME J. Dynam. Systems, Meas. Control*, 106(4), pp 249-254 (1984).

[17] K.D. Dhuri and P. Seshu, “Piezo actuator placement and sizing for good control effectiveness and minimal change in original system dynamics”, *Smart Materials and Struct.*, (under review) (2006).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Path Planning for Obstacle Avoidance and Vision Control for Person Guiding with a Mobile Robot using Multiple Sensors

Jinpyo Hong, Youjun Choi, Kyihwan Park
Gwangju Institute of Science and Technology,
Gwangju, Republic of Korea.
jphong@gist.ac.kr, ychoi@gist.ac.kr, khpark@gist.ac.kr

Abstract

A mobile robot in the public space should have the ability to detect the object whichever it is moving object or not. Especially, since our developed mobile robot will be used in the market, it requires abilities to avoid the obstacle and guide the person into the merchandise location. The main contributions of our approach include: 1) Assumed that the path between the shelves is straight and the obstacle exists, we can find the movable empty passageway by the segment estimation algorithm and longest distance estimation algorithm using a laser measurement system(LMS); and 2) For guiding the person to merchandise location, we developed the simple vision system for face detection using the one USB camera and ultrasonic sensor. Our proposed approach has been successfully tested for the task of person guiding.

Keywords: path planning, longest distance estimation algorithm(LDEA), vision control,pan tilt camera

1 Introduction

In recent years, numerous studies have attempted to develop the mobile robot used in the public spaces such as post office, museum and so on. Especially, many services are needed for the help of the human in the large market. When a person is shopping in the market, the person can not exactly know where the merchandise to be bought is located since there are many products and their locations are frequently changed. Therefore, our mobile platform has been developed for the purpose of guiding a customer to a merchandise location in the large market.

There are many issues to be considered to develop a mobile robot for the purpose of using at the market.: Localization, navigation, map building and so on. For guiding the customer to the merchandise location, it is necessary to find the passageway where the mobile can move without collision. Also, since the customer may not follow the mobile robot, the vision system for tracking the customer always is needed. Therefore, in this paper, we will focus on the two issues.

When performing the guiding task, we need to track the customer because he or she may not follow the mobile robot for looking around other products. Therefore, in this paper, for solving this problem, we propose the new obstacle avoidance algorithm using a laser range finder and a vision system using a camera and an ultrasonic sensor.

A vision system in order to track the face of the person has been developed using the stereo camera system. This method has the strong point that it covers a wider angular range of the view. However it requires the camera calibration process to get the precision and the more vision data process [1]. In research [2], the face detection and tracking was carried out by velocity controller with skin segmentation method using one pan tilt camera but, this method is not flexible when the distance between the pan tilt camera and a moving object is changed because the 3D depth information can not be known. Therefore, we propose a simple vision system using one USB camera and a ultrasonic sensor that are operated independently on PC and microprocessor.

Another proposed issue of this paper is fixed obstacle avoidance algorithm. In past years, various kinds of obstacle avoidance algorithms have been proposed for the robot navigation [3]. To accomplish a navigation of mobile robot, a robot must make decision to move to empty space from obstacle. Especially, at the market, it is more important because a location of obstacles, i.e. market shelves, are changed by needs frequently. Assumed that every obstacles are rectangle shape, we propose the fast path planning method for fixed obstacle avoidance since the market shelf are almost rectangle.

2 The Outline of Guiding the Person to Merchandise Location

When the mobile robot guides the person to the merchandise location, it needs to look around the environments for obstacle avoidance. With LMS, we can obtain the distance data of the environments and generate the movable empty passageway to reach the merchandise location. At the same time, the mobile robot should always detect the face of the customer by pan-tilt camera control because it can know whether the customer follows the mobile robot. With the status of the face detection, the mobile robot guides the person to the movable empty passageway and repeatedly generate the movable empty passageway. If the face detection is failed, the mobile robot must turn around toward the direction of the customer until it is succeeded.

In this paper, we developed independently the longest distance estimation algorithm to generate the movable empty passageway, face detection algorithm and pan-tilt camera control.

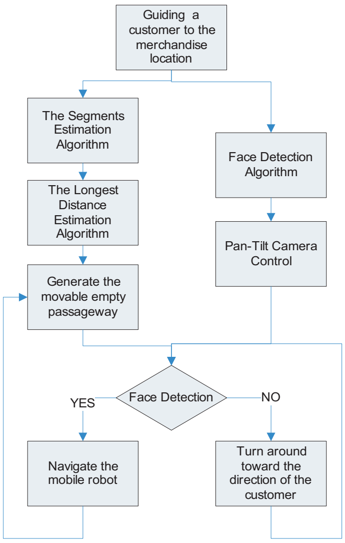

Guiding a customer to the merchandise location
The Segments Estimation Algorithm
The Longest Distance Estimation Algorithm
Generate the movable empty passageway
Face Detection Algorithm
Pan-Tilt Camera Control
Face Detection
YES
NO
Navigate the mobile robot
Turn around toward the direction of the customer

Figure 1: The outline of guiding the person to merchandise location.

3 Path Planning for Obstacle Avoidance

It is necessary to generate a path for the fixed obstacle avoidance at the market since the location of market shelves are changed frequently. Therefore, we suggests the longest distance finding algorithm using Laser Measurement System(LMS) to avoid a collision between the robot and market shelve this paper.

3.1 Data Acquisition by Laser Measurement System

The Laser Measurement System(LMS) we have used this paper has the maximum scanning distance of 30 metres approximately. It has 0.5 degree resolution 0 degree to 180 degree, the interval is 0.5 degree. Therefore the ith element of the array is the distance of ith degree from origin of LMS as shown in table 1.
The primary raw data is transferred to a Single Board Computer(SBC) on the mobile robot as an array of Measured Data[i], which stands for the obstacle distance in the ith scanning angle of the LMS. A plotted map with taw data from LMS is shown as figure 2.

Table 1: Scanning data array from 0 to 180 degree.

Angle[i]	**0**	**...**	**180**	**...**	**359**
Measured Data[i]	37	...	284	...	148

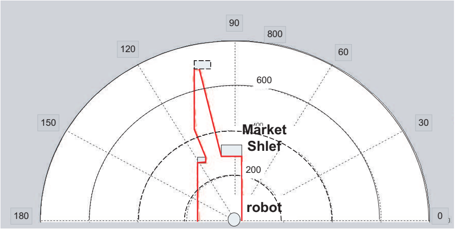


Figure 2: The scanning map of LMS.

3.2 The Segments Estimation Algorithm

The first process of the longest distance estimation algorithm concerns the laser scan data acquisition we have mentioned before. Therefore it is necessary to recover the segments estimation that forms the actual visibility region from the point data that LMS gives [4]. We get poly-lines with the laser scanning data obtained as an ordered list of polar coordinates (r,θ) where market shelves may be found. Thus to execute the segments estimation, we have to convert from polar coordinate to rectangular coordinate by using following equation (1).

$$\begin{pmatrix} MD[i]_x \\ MD[i]_y \end{pmatrix} = MD[i]\begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix} + \begin{pmatrix} R_x \\ R_y \end{pmatrix} \tag{1}$$

where, MD is Measured Data, R_x and R_y are current x and y coordinate of robot, respectively.
To detect an edge of market shelves, at least two segments which belongs to the same straight line will be needed as shown in figure 3. The distance between two segments which is going to be passage is calculated by following equation (2):

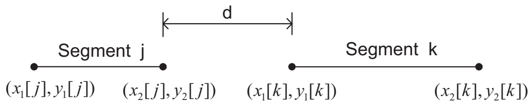


Figure 3: Market shelf representation by segments.

$$d = \sqrt{(x_2[j]-x_1[k])^2+(y_2[j]-y_1[k])^2} \tag{2}$$

where d is a distance between two segments.

If an angle of robot is moved by θ from the origin, the scanning map by measured data from LMS is also rotated by θ. To find out market shelf, we need to check whether measured elements are on the same object or not by calculating slope of continuous elements. However a cost of this method is too high because of a division operation. If each point from LMS is rotated as much as angular difference between the global and local reference frame, then each point is moved to easy coordinate which makes it easy to find out line segments without division operation because the slope of continuous points is going to be parallel to x or y axis. Therefore, to reduce the cost, we check the moving angle θ first. After that, we equate a rotation transformation operation by multiplying $2X2$ transformation matrix which produces rotation about the origin to make zero-degree-scanned data from the origin. Hence, we can find out each segment using geometrical criterion method.

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \qquad (3)$$

To estimate a length of each segment, we have used geometrical criterion. Among measured elements if a distance between y coordinate of two continuous elements p_i, p_{i-1} is less than defined threshold value ζ_1, we assume that these elements belong to the same object in figure 4. Similarly, if distance between x coordinate of two continuous elements p_i, p_{i+1} is less than ζ_2, we assume that these elements belong to the same object. The selection of all line segments x_j that contribute to the same line can be done in a threshold-based manner according to following equation (4).

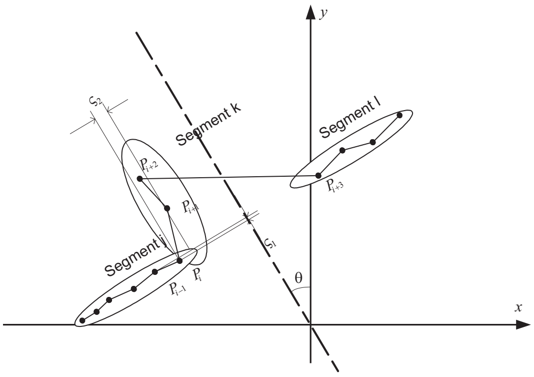


Figure 4: Estimation of segements.

$$(p_{i+1} - p_i)(p_{i+1} - \overline{p_i})^T \leq \zeta_i \qquad (4)$$

where ζ_i is a threshold value and $\overline{p_i}$ is the representation of the reference line, respectively.

3.3 The Longest Distance Estimation Algorithm(LDEA)

It is necessary to find out an movable path to move the robot at the desired position. Therefore, we suggest the longest distance estimation algorithm. First, this algorithm finds out the longest distance between the robot and line segment which is done by proposed segments estimation algorithm. After that, it checks whether robot can reach to the found longest distance or not. This algorithm keeps doing these process until it finds out the longest movable distance. The proposed the longest distance estimation algorithm(LDEA) is shown in figure 5:

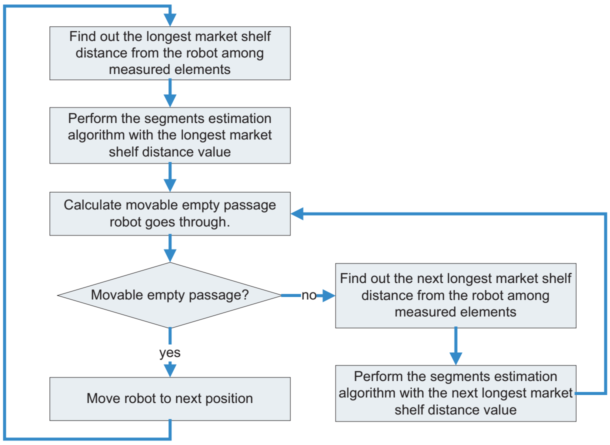

Find out the longest market shelf distance from the robot among measured elements
Perform the segments estimation algorithm with the longest market shelf distance value
Calculate movable empty passage robot goes through.
Movable empty passage?
no
Find out the next longest market shelf distance from the robot among measured elements
Perform the segments estimation algorithm with the next longest market shelf distance value
yes
Move robot to next position

Figure 5: The longest distance finding algorithm.

4 Pan-Tilt Camera Control with Vision and 3D Depth Sensor

We can obtain the two dimensional information from single camera. In order to get the distance from the camera to the object, it is utilized to compare the measured object size with the object size which is measured in the known distance. Since this method requires previously measured many data about objects, many data base is needed and it takes long time to search the detected object. On the other hand, stereo vision system using two cameras don't need many data base but it requires the specific vision data processing algorithm to get the 3D depth information. Therefore, we propose the vision system using the one camera and ultrasonic sensor. Since 3D depth information can be obtained from the ultrasonic sensor, our proposed vision system is more speedy and simple than other systems.

4.1 Pan-Tilt Camera

Although the camera can be controlled by the mobile robot body's rotation movement, it is more efficient to separate the control functions between mobile robot and pan tilt camera in view of complexity and safety. Therefore, we designed the pan tilt camera mechanism to detect the face of the person in the limited range with two degrees of freedom.

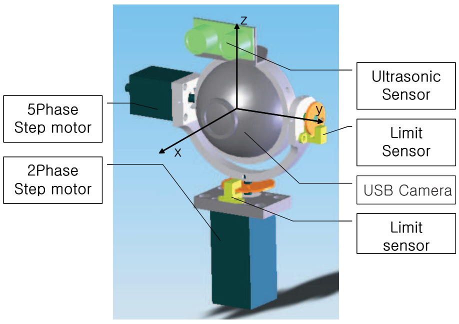


Figure 6: Pan tilt camera.

The pan tilt camera mechanism is shown as figure 6. It has two degrees of freedom at y axis and z axis and operates in the range of ± 50 degrees. It is controlled by geared step motor with high resolution and power. Since it is controlled by vision data processing and 3D depth sensor, the motor does not need a feedback sensor like as an encoder.

4.2 Face Detection Algorithm

In general, the USB camera has 30 frame rate when it has no load to process the vision data. For improving the performance of vision control, the face detection algorithm is the one of the important property. we selected the cascade of boosted classifiers working with haar-like features(CBCH) as the face detection algorithm. It has the characteristics which have rapid calculation time and high precision since its classifier is so simple composed of addition and subtraction operation. We realized the algorithm with open source, OPEN CV library. The face detection result is shown as figure 7.

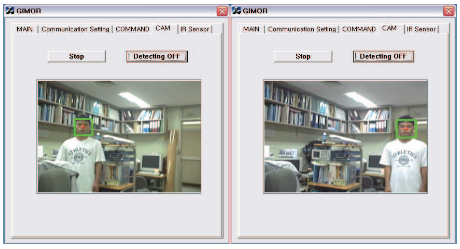


Figure 7: Face detection using CBCH.

The green line rectangle indicates the face detection region in figure 7 and the vision coordinate of the face is calculated as the centre of the rectangle.

4.3 Pan-Tilt Camera Control Scheme

The overview for controlling the pan-tilt camera using an USB camera and ultrasonic sensor is shown as figure 8. The camera is initialized in the predefined position and the face detection using CBCH algorithm is performed. If the face detection is failed, the camera is hold without movement and otherwise, we can find out the control reference input by the geometric relation from vision data and ultrasonic data.

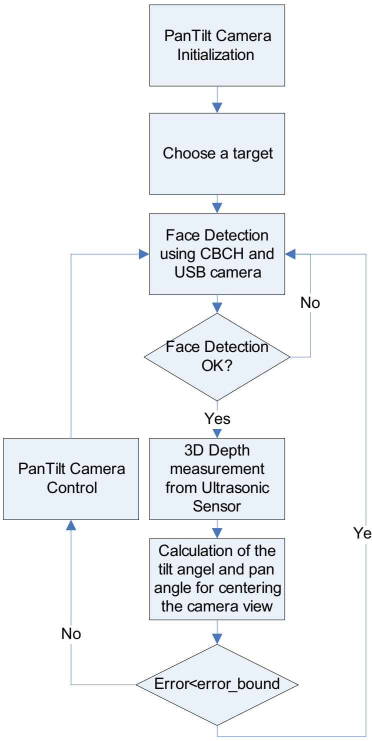

PanTilt Camera Initialization
Choose a target
Face Detection using CBCH and USB camera
Face Detection OK?
No
Yes
3D Depth measurement from Ultrasonic Sensor
PanTilt Camera Control
Calculation of the tilt angel and pan angle for centering the camera view
Error<error_bound
No
Yes

Figure 8: An overview of our proposed pan-tilt camera control algorithm.

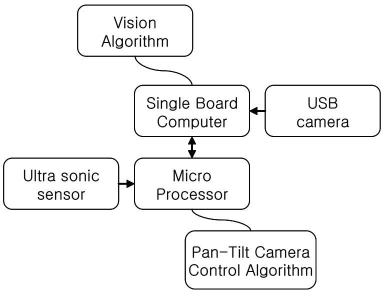


Figure 9: Hardware architecture for controlling the pan-tilt camera.

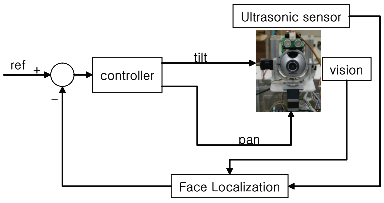


Figure 10: Control block diagram.

The hardware architecture is shown in figure 9. A single board computer(SBC) is used for vision data processing from the USB camera and a microprocessor operates the ultrasonic sensor separately for reducing the calculation time. Since the pan tilt coordinates of the customer's face are obtained from the SBC, this face localization data is transferred to the microprocessor through RS232 communication. Finally, we can control the pan tilt camera by combining the datum in the microprocessor as shown in figure 10.

4.4 The Geometric Relation Between Two Sensors Data

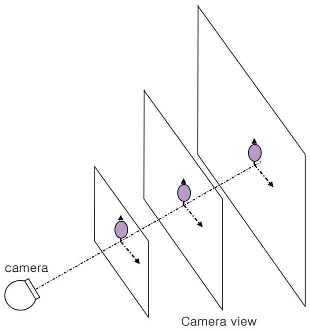


Figure 11: The size of camera view according to the distance between the camera and the object.

The vision data from the camera has the meaning of different real distance according to the distance between the camera and the object. As shown in figure 11, the view size of pan tilt camera has a characteristic to become larger or smaller according to the measured distance. Therefore, we need to get the coefficient to convert the pixel to real distance by experimental results. we obtained the pixel size raw data about real size by our developed vision program in figure 12. we did the experiment increasing the distance from the camera to the object. As you can see in figure 12, there is a linear characteristic affected by lens distortion or camera performance. Since it is similar to a straight line, we got the straight line equation by using the least-squares regression method [7].

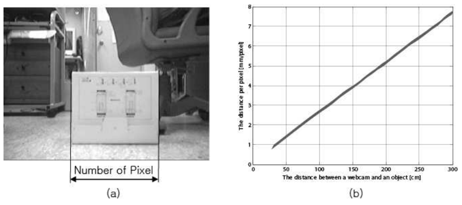


Figure 12: The coefficient graph to convert the pixel to real distance.

$$y = a_0 + a_1 x \tag{5}$$

$$a_1 = \frac{n\sum x_i y_i - \sum x_i y_i}{n\sum x_i^2 - (\sum x_i)^2} \tag{6}$$

$$a_0 = \bar{y} - a_1 \bar{x} \tag{7}$$

where $\bar{y}$ and $\bar{y}$ are the means of y and x, respectively.

The meaning of x is the displacement measured by ultrasonic sensor and that of y is the coefficient to convert pixel size into the real size according to x. For controlling the pan tilt camera, we should derive the correlation among a vision coordinate,a camera coordinate and angels of tilt and pan. As shown in figure 13, when the view is located on the camera, we can assume that the vision coordinate and the camera coordinate are the same direction in x axis and the opposite direction in y axis. Therefore, we can derive the relation of two coordinates as follows.

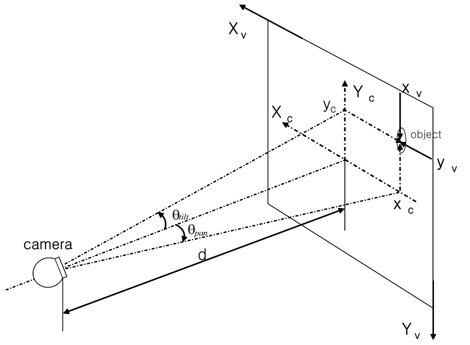


Figure 13: The relation between the vision coordinate, camera coordinate and the angles of tilt and pan.

$$x_c = x_v - x_{pixelsize}/2 \tag{8}$$

$$y_c = -y_v + y_{pixelsize}/2 \tag{9}$$

where c is a camera coordinate index and v is a vision coordinate index, and $x_{pixelsize}$ and $y_{pixelsize}$ indicate vision sizes.

Finally , we got the camera angles to control the pan tilt camera from the right triangles. The equations of the pan angle and the tilt angle are represented by equations (10) and (11).

$$\theta_{pan} = \arctan(\frac{x_c}{d}) \tag{10}$$

$$\theta_{tilt} = \arctan(\frac{y_c}{d}) \tag{11}$$

5 Experimental Results

Since we have suggested navigation algorithm for the purpose of using at the market as a guidance robot, an shelf-like object is set in the narrow passage for a test environment as shown in figure 14. The size of the test environment is 1.8 metres by 20 metres. We assume that there are no people in the test environment except client and every market shelf have rectangular shape. A local map of the test environment is generated by LMS. The figure 15 is a result of the LDEA algorithm test. It shows that the mobile robot estimates the longest

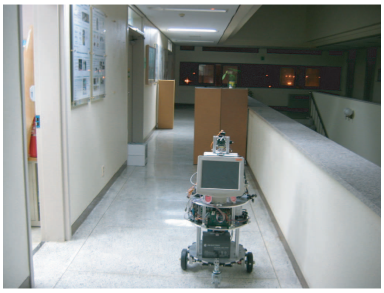


Figure 14: Test environment.

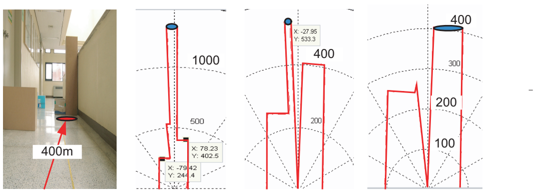


Figure 15: The LDEA algorithms test.

distance which goes through according to the location of each shelf.

For guiding the person, we have experiment about customer's face tracking with our proposed pan-tilt camera. Figure 16a shows that the coordinates of the face are converged on that of the camera view plane(320*240 pixels). And figure 16b shows that the customer's face is located at the four corners on the view screen before face tracking control. Our proposed vision system has a good performance to track the face of the person though it is composed of the one camera and ultrasonic sensor.

6 Conclusions

In this paper, we have presented the path planning algorithm and pan tilt camera control method for guiding the customer to the merchandise location. For navigate the mobile robot, we have found the movable empty passageway utilizing the LEDA. In order to control the pan tilt camera, we developed the pan tilt camera mechanism with step motors, USB camera and ultrasonic sensor. The geometric relation has been derived from the vision data and ultrasonic data required to track the face of the person in the centre of image plane. These algorithms has been tested successfully for the task of person guiding independently.

7 References

[1] H.S. Kwon, Y.R. Yoon, J.B. Park and A.C. Kak, " Person tracking with a mobile robot using two uncalibrated independently moving cameras", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp 2877-2883 (2005)

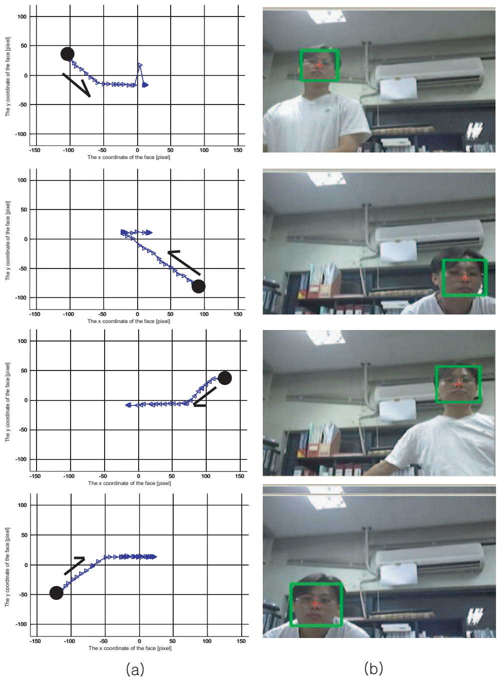


Figure 16: The pan tilt camera control.

[2] L. Jordao, M. Perrone, J. P. Costeira, "Active Face and Feature Tracking",*Proceedings International Conference on Image Analysis and Processing 1999*, pp 31-38 (1999)

[3] B. Hu, H. Jin, K. He and G. Chen,"Research on the Strategy of Obstacles Avoidance of Outdoor Mobile Robot", *Proceeding of the 2004 American Control Conference*, Boston, Massachusetts, pp 2362-2367 (2004)

[4] R. Siegwart and R. Nourbakhsh, "chapter 4. Perception",R. Siegwart and R. Nourbakhsh, *Introduction to autonmous mobile robots*, MIT Press, Cambridge, Massachusetts, pp 159-163

[5] J. Canou, G. Mourioux, C. Novales and G. Poisson, "A local map building process for a reactive navigation of a mobile robot", *Proceedings of the 2004 International Conference on Robotics and Automation*, New Orleans, LA, pp 4839-4844 (2004)

[6] T. Yamane, Y. Shirai and J. Miura, "Person tracking by integrating optical flow and uniform brightness regions", *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Lueven, Belgium, pp 3267-3272 (1998)

[7] S.C. Chapra and R.P. Canale, "least squares regression",S.C. Chapra and R.P. Canale, *Numerical methods for engineers with software and programming application*, Mc Graw Hill, International Edition, Seoul, pp 440-445

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Emotion Recognition using the Frequency Ranges Sensitive to Emotion

Kyung Hak Hyun, Eun Ho Kim, Yoon Keun Kwak
Department of Mechanical Engineering
Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
ykkwak@kaist.ac.kr

Abstract

In human-robot interaction, detecting human state is an important issue. There are several values to estimate in the human state including emotional state. To date, study on emotion recognition has focused on detecting the values of pitch, formant, or cepstrum from the variation of speech according to changing emotions. A correlation of emotion and frequency was proposed and used for emotional speech features. However, the values of emotional speech features vary by not only emotions but also speakers. Because each speaker has unique frequency characteristics, it is difficult to apply the same manner to different speakers. Therefore, in the present work we considered the personal characteristics of speech. To this end, we analysed the frequency characteristics for a user and chose the frequency ranges that are sensitive to variation of emotion. From these results, we designed a personal filter bank and extracted emotional speech features using this filter bank. This method showed an overall improvement of 90% although there are differences among individuals.

Keywords: emotion, recognition, frequency range, filter

1 Introduction

Emotional speech recognition involves automatically identifying the emotional or physical state of a human being from voice. The importance of emotion recognition for human computer interaction is widely recognised [1]. Although the emotional state does not alter linguistic content, it is an important factor in human communication, as it provides feedback information in many applications. In human-machine interactions, the machine can be made to produce more appropriate responses if the state of emotion of the person can be accurately identified. In particular, language understanding can be further improved if the emotional state of the speaker can be extracted, and this in turn will enhance the accuracy of the system. Current automatic translation algorithms focus on the semantic component of speech. Additional useful information could be obtained if the emotional state of the speaker could also be identified and presented, such as in non-face-to-face situations. Other applications of an automatic emotion recognition system include tutoring, alerting, and entertainment [2].

1.1 Previous Works

Making a machine recognise emotions from speech is not a new idea. The first investigations were conducted in the mid-1980s using statistical properties of certain acoustic features [3][4]. In 2000, emotional speech recognition was employed by therapists as a diagnostic tool in medicine [5]. Presently, most research is focused on finding powerful combinations of classifiers that advance the classification efficiency in real-life applications. As an example, ticket reservations, so-called “SmartKom”, can recognise customer frustration and change their response accordingly [6][7].

1.2 Purpose of Work

Humans have a unique vocal tract system, speaking habits, and different characteristics for each emotion. Furthermore, nobody can speak in the same way every time (intra-speaker variability). On the other hand, humans can distinguish differences among speakers even when they utter the same words or sentences (inter-speaker variability).

Due to intra-speaker variability, the spectrograms of speech always change. However, owing to characteristics of the vocal tract system and speaking habits, variation of acoustic parameters are small. However, the inter-speaker variability is larger than intra-speaker variability. This can help in identifying the speaker, but makes it difficult to recognise emotions in a speaker independent manner. Hence, the results of emotion recognition in speaker independent systems are below 60% [8][9].

However, more accurate rates are needed for practical use. Therefore, in this study, we analysed the frequency characteristics of speakers and applied a personal filter bank from the results.

1.3 Outline

In section 2, we define the target emotions to be recognised. The emotions are Neutrality, Joy, Sadness, and Anger. In addition, an emotional speech database, which is used in training and evaluating the recogniser, is introduced. In section 3, we explain the speech feature, Log Frequency Power Coefficient

(LFPC), which is used for emotion classification. We analyse the frequency characteristics using the LFPC and design a personal filter bank from the analysis. We then show the emotion recognition rates obtained using the proposed method and compare then with results from a previous method in section 4. Finally, conclusions are presented in section 5.

2 Target Emotions

Before discussing emotion recognition technology, it is necessary to define the emotions that are considered in this study. As noted above, emotion is subjective, and hence the boundary between emotions is ambiguous, making emotion recognition very difficult. Thus, in the present study, only four emotions, anger, joy, sadness, and neutrality, are considered. These target emotions have previously been explained in a previous work [3]. The number of emotions comes from Russell's circumplex model of affect which is in a two-dimensional space of pleasure-displeasure and degree of arousal. [4]. In this model, Russell sorted affect words into 2 axes, as presented in figure 1. Hence, in the present work, we made four groups by 2 axes and then defined the representative emotion in each group. Although the classification of joy, anger, sadness, and neutrality requires verification in terms of its appropriateness to represent emotions of each quadrant, we could use at least four emotions for primary emotions. Therefore, we assumed that there are four fundamental emotions. Thereupon, if the fundamental emotions could be recognised well, then the emotion recognition system developed here could be applied to complex emotions.

3 Emotional Speech Database

The Korean database [10] was used in our experiment. The database was recorded in the framework of the G7 project in Korea. In this database five male speakers and five female speakers were recorded.

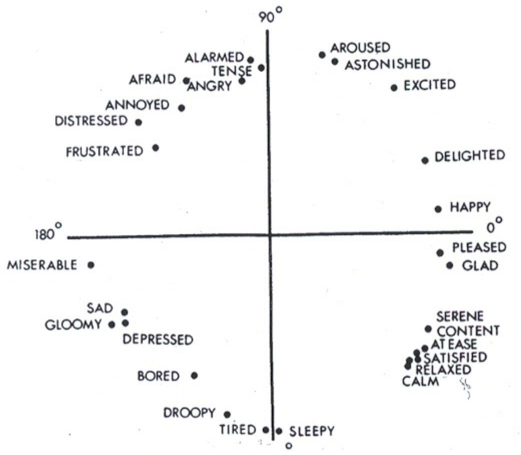


Figure 1: A circumplex model of effect.

3.1 Corpus of Database

Each corpus contains several phonemes. The database contains 45 sentences.

The corpora contain short, medium, and long sentences that are context independent. The sentences are chosen upon consideration of the followings:

1. The sentence is able to pronounce in several emotional states.
2. The sentence can express the emotions naturally.
3. The database should contain all phonemes of Korean.
4. The database should contain several dictions such as honorific words.

3.2 Speech Materials

The corpora were recorded over four basic emotions. The recorded emotions are neutrality (N), joy (J), sadness (S), and anger (A). These emotions are defined in the previous section.

The database contains speech recorded in three iterations. All speech was recorded four times and the worst record among those was discarded. The aim of this step is to filter clumsy wording and maintain consistency of the speech.

The database contains 5400 sentences. The subjective evaluation tests for all corpora were performed.

3.3 Recording Conditions

The subjects were amateur actors and actresses who have practiced emotional expression and they were selected according to their ability of expression. The recordings were made in a silent experimental environment with DAT. The sampling frequency was 48 kHz with quantisation of 16 bits. Feature calculation used recordings that were decimated to a sampling rate of 16 kHz using an eight-order Chebyshev filter.

3.4 Subjective Evaluation Test

Table 1: Human performance.

Recog. (%)	Neutrality	Joy	Sadness	Anger
Neutrality	83.9	3.1	8.9	4.1
Joy	26.6	57.8	3.5	12.0
Sadness	6.4	0.6	92.2	0.8
Anger	15.1	5.4	1.0	78.5
Overall	78.2			

Subjective evaluation tests were made for the database. The subjective evaluation test included 30 listeners. The listeners were engineering students from Yonsei University in Korea.

Each listener decided which emotion corresponded to each utterance. The samples were played randomly. 10 listeners made decisions for each utterance.

The results of the subjective evaluation test showed, on average, 78.2% accuracy. The confusion matrix of the test is shown in table 1. The accuracy was highest for the emotion of sadness at 92.2%, whereas joy was the worst at 57.8%. Many errors were made in discriminating between joy and anger and between neutrality and anger. These results are caused by the difference in the basis for neutrality.

4 Personal Filter Bank Design

4.1 Log Frequency Power Coefficients

Human beings perceive audible sound from 20 Hz to 20 kHz. Furthermore, human perception is not linear to physical frequency, and hence there is a different unit for audible sound frequency, that is, Mel [11].

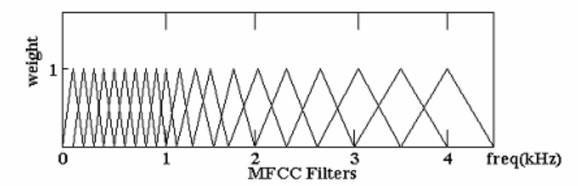


Figure 2: The filter bank of human auditory system.

Table 2: Centre frequency and bandwidth of log frequency filter banks.

Filter	Centre Frequency	Bandwidth
1	127 Hz	54 Hz
2	192 Hz	75 Hz
3	281 Hz	104 Hz
4	406 Hz	145 Hz
5	579 Hz	202 Hz
6	820 Hz	280 Hz
7	1155 Hz	389 Hz
8	1620 Hz	541 Hz
9	2267 Hz	753 Hz
10	3167 Hz	1046 Hz
11	4417 Hz	1454 Hz
12	6154 Hz	2021 Hz

The human auditory system has several sensitive frequency ranges. Under 1,000 Hz, humans feel the pitch is linear to physical frequency; however, when the frequency is increased above 1,000 Hz, the auditory system becomes insensitive. In other words, as frequency changes become larger, humans can distinguish changes of pitch.

Accordingly, we can regard the human audible system as a filter bank, as illustrated in figure 2. Therefore, a log frequency filter bank can be regarded as a model that follows the varying auditory resolving power of the human ear for various frequencies.

In Tin Lay Nwe's study [12], a filter bank is designed to divide speech signals into 12 frequency bands that match the critical perceptual bands of the human ear. The centre frequencies f_i and bandwidth b_i for a set of 12 band-pass filters are derived as follows:

$$b_i = C \tag{1}$$

$$b_i = \alpha b_i \qquad 2 \le i \le 12 \tag{2}$$

$$f_i = f_1 + \sum_{j=1}^{i-1} b_j + \frac{(b_i - b_1)}{2} \tag{3}$$

where C is the bandwidth, f_1 is the centre frequency of the first filter, and alpha is the logarithmic growth factor. To make use of the information of the fundamental frequency, the frequency of the lower band is set at 100 Hz. Hence, C=54Hz and f =127Hz.

The centre frequencies and bandwidths of 12 filter banks, which were proposed by Tin Lay Nwe, are presented in table 2 and the scheme of extracting the LFPC is presented in figure 3.

4.2 Filter Bank Analysis

In the analysis of frequency characteristics of a user, we attempted to identify specific frequency ranges that are sensitive to changes of emotion and robust to changes of context. To this end, we compared the rates of emotional speech recognition by changing the filter bank. The results are presented in figure 4 and figure 5. Both show normalised values on the y-axis in a range between 0 and 1 to verify the effect of each filter. In figure 4, the recognition rates are for speech that was filtered by a one order filter. Hence, we can estimate which filter is most useful with respect to recognising the emotion. Figure 5 shows the results for speech that was filtered by an eleven order filter bank. Thus, we can identify the worst filter.

It is possible to select the best filter or abandon the worst filter on the basis of the above results. In this regard, the results demonstrate that accepting or rejecting a specific filter from the filter bank can affect the emotional recognition results.

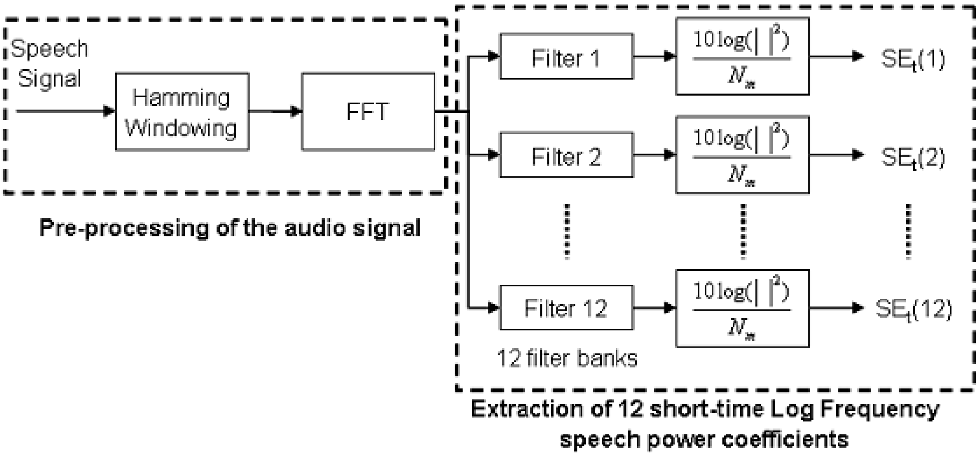

Speech Signal
Hamming Windowing
FFT
Pre-processing of the audio signal
Filter 1
Filter 2
Filter 12
12 filter banks
$\frac{10\log(|\ |^2)}{N_m}$
SE$_t$(1)
SE$_t$(2)
SE$_t$(12)
Extraction of 12 short-time Log Frequency speech power coefficients

Figure 3: The scheme of extracting LFPC.

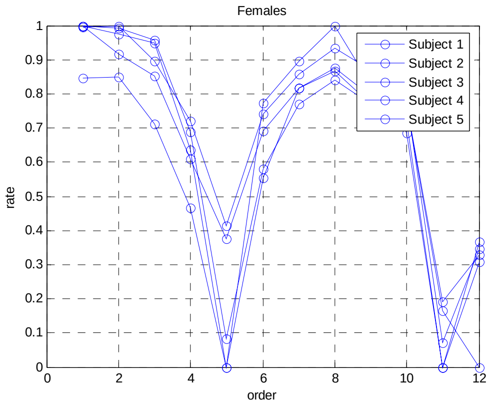


Figure 4a: One filter selection in females.

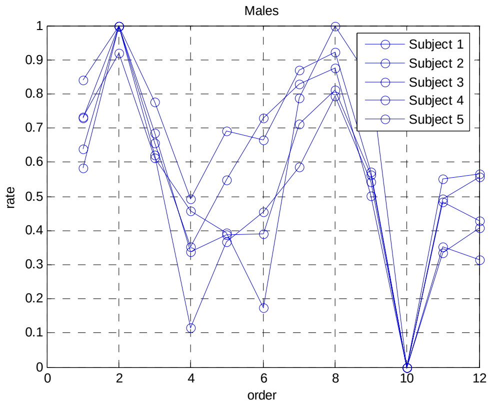


Figure 4b: One filter selection in males.

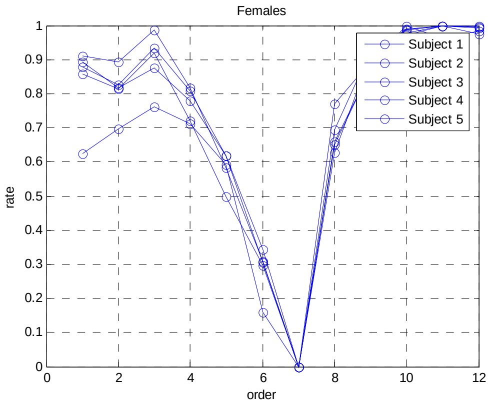


Figure 5a: One filter exception in females.

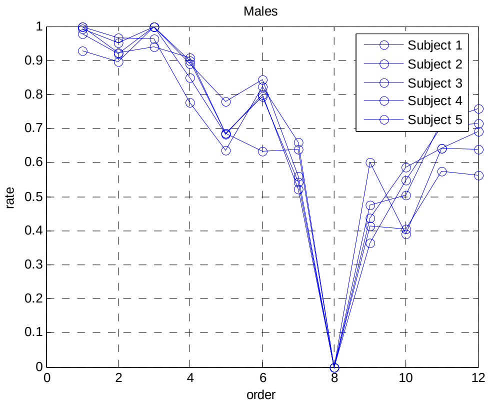


Figure 5b: One filter exception in males.

However, they do not indicate the best set of filters to recognise the emotion, because each filter of the filter bank is not independent of the other filters. According to the study of Rabiner and Juang [13] and a hypothesis of Tin Lay Nwe, each filter shares frequency ranges with its neighbour filters, as presented in figure 2.

Therefore, a filter cannot be chosen independently from the filter bank and should be chosen under consideration of the effects of other filters.

4.3 Design of the Personal Filter Bank

4.3.1 Design Method

From the analysis presented above, we find that there are different frequency ranges that can help in the recognition of emotion in speech. Hence, we compared the results obtained using a full order filter bank with those derived using a selected filter set.

The full order filter bank was selected from Tin Lay Nwe's study [12]. For comparison, we assessed several filter sets in order to identify the best set in terms of performance. At this step the full order is twelve and each filter can be selected independently. Thus, it is necessary to compare 4095 cases (2^{12}-1) for one person.

Finally, we examined the results obtained using a Bayesian Classifier and chose the best set for each person. A summary of this procedure is presented in figure 6. In this figure, full order means the classifier uses the full information from filters 1~12, whereas order choicer means the classifier adaptively uses the information among the filters.

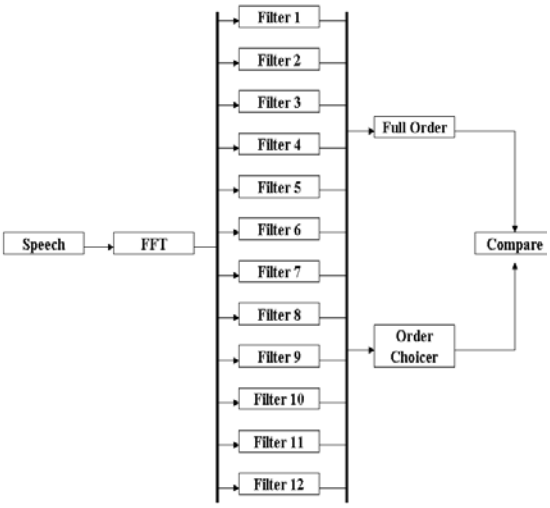


Figure 6: Flow chart of the adaptive filter bank method.

Table 3: Adaptive filter orders and the improvement rate.

Subject	Adaptive filter order	Improvement (%)
Female 1	3,4,5,7,9,10,11,12	97
Female 2	2,5,6,7,8,9,10,11,12	93
Female 3	4,6,7,9,11,12	98
Female 4	1,4,5,6,8,10,11	94
Female 5	3,5,6,7,8,9,10,11,12	84
Male 1	3,4,5,6,9,10,11,12	86
Male 2	4,5,6,7,10,12	82
Male 3	2,3,4,10,11	85
Male 4	1,3,4,6,8,9,10,11,12	79
Male 5	2,5,6,7,8,10,11	96

Table 4: Comparison result with PCA.

Females				
Subject 1	Subject 2	Subject 3	Subject 4	Subject 5
Principal component analysis				
88.44	87.15	81.50	80.20	91.91
Proposed method				
91.72	89.51	86.27	84.08	93.67

Males				
Subject 1	Subject 2	Subject 3	Subject 4	Subject 5
Principal component analysis				
85.65	85.10	83.68	71.24	90.44
Proposed method				
88.66	88.08	87.27	73.64	94.00

4.3.2 Evaluation

To evaluate the designed filter bank, we compared the results of the designed filter bank with those of the full-order filter bank. We performed 100 iterations for the designed filter bank for cross-validation in order to verify the results statistically. For the cross-validation, the training and test data was randomly chosen from the database; 120 training data were chosen randomly and 300 test data were chosen not only randomly but also differently from the training data for the evaluation. The results of each subject are presented in table 3.

We found that the designed filter bank method sometimes decreased the recognition rate. However, the degree of decrease was very small. From table 3, however, 80~90% of cases showed an increase in the recognition rate for 100 repetitions of the experiment for each subject.

Finally, the recognition rates of all subjects are presented in table 4. In addition, we present a comparison with the principal component analysis (PCA) results. Hence, we find that the proposed method is superior to the PCA method, as the latter can only find the principal component, which is good for representation, whereas the proposed method finds the frequency ranges that are sensitive to change of emotion.

5 Conclusion

In this work, we consider the correlation of speech frequency with emotion and develop a personal filter bank for emotion recognition. We verified the superiority of the proposed method through comparison with PCA results. The strength of the proposed approach lies in selecting frequency ranges that are sensitive to the change of emotion. Although the improvement varied on a case by case and person to person basis, a roughly 2~5% overall improvement in recognition rate was obtained.

For future work, we will make a demonstration programme in MATLAB to verify this system working properly via a microphone. After verification, we will load this algorithm to a service robot named “silver-robot”, which is being developed and supported by ministry of Commerce, Industry and Energy of Korea.

6 Acknowledgements

This research was performed for the Intelligent Robotics Development Programme, one of the 21st Century Frontier R&D Programmes funded by the Ministry of Commerce, Industry and Energy of Korea.

7 References

[1] M. Pantic and L. Rothkrantz, “Toward an affect-sensitive multimodal human-computer interaction”, *Proc. of the IEEE*, 91, pp 1370-1390 (2003).

[2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz and J.G. Taylor, “Emotion recognition in human-computer interaction,“ *IEEE Sig. Proc. Mag.*, 18(1), pp 32-80 (2001).

[3] R. van Bezooijen, *The Characteristics and Recognizability of Vocal Expression of Emotions*, Foris, Drodrecht, Netherlands (1984).

[4] F.J. Tolkmitt and K.R. Scherer, “Effect of experimentally induced stress on vocal parameters”, *J. Exp. Psychol.: Human Percept. Perform*, 12(3), pp 302-313 (1986).

[5] D.J. France, R.G. Shivavi, S. Silverman, M. Silverman and M. Wilkes, “Acoustical properties of speech as indicators of depression and suicidal risk”, *IEEE Trans. Biomed. Eng.*, 7, pp 829-837 (2000).

[6] J. Ang, R. Dhillon, A. Krupski, E. Shriberg and A. Stolcke, “Prosody-based automatic detection of annoyance and frustration in human-computer dialog”, *Proc. Intl. Conf. on Spoken Language Processing (ICSLP2002),* Denver, CO, pp 2037-2040 (2002).

[7] F. Schiel, S. Steininger and U. Turk, “The Smartkom multimodal copus at BAS”, *Proc. Language Resources and Evaluation* (2002).

[8] S. McGilloway, R. Cowie, E. Douglas-Cowie, S. Gielen, M. Westerdijk and S. Stroeve, “Approaching automatic recognition of emotion from voice: a rough benchmark”, *Proceedings ISCA Workshop on Speech and Emotion*, Belfast (2000).

[9] N. Esau, B. Kleinjohann, L. Kleinjohann and D. Stichling, “MEXI: machine with emotionally extended intelligence”, *Proceedings Hybrid Intelligent Systems: Design and Application (HIS2003)*, Melbourne, pp 961-970 (2003).

[10] B.S. Kang, *Text Independent Emotion Recognition Using Speech Signals*, MSc Thesis, Yonsei University, Seoul (2000).

[11] T.F. Quatieri, *Discrete-Time Speech Signal Processing Principles and Practice*, Prentice Hall, Upper Saddle River, NJ (2002).

[12] T.L. Nwe, S.W. Foo and L.C. DeSilva, “Speech emotion recognition using hidden Markov model”, *Speech Communication*, 41(4), pp 603-623 (2003).

[13] L.R. Rabiner, B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, NJ (1993).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Adjustable Bipedal Gait Generation using GA Optimised Fourier Series Formulation; Real-time Gait Adjustment

L. Yang[1], C.M. Chew[1], A.N. Poo[1], T. Zielinska[2]
[1]Control and Mechatronics Lab, Mechanical Engineering
National University of Singapore, Singapore
[2]Institute of Aircraft Engineering and Applied Mechanics
Warsaw University of Technology, Warsaw, Poland
yanglin@nus.edu.sg, mpeccm@nus.edu.sg, mpepooan@nus.edu.sg, teresaz@meil.pw.edu.pl

Abstract

This paper discusses the parameter online adjustment functions which further construct a closed-loop locomotion control on the basis of the walking gait generation method Genetic Algorithm Optimised Fourier Series Formulation (GAOFSF). Via adjusting one or two parameters, the GAOFSF method is verified applicable to the autonomous adjustment of bipedal walking gaits on uneven terrains in real-time. Motions under transitions have been dynamically simulated, showing the validity of the gait online adjustment functions.

Keywords: real-time, gait adjustment, genetic algorithm, Fourier series

1 Introduction

In this paper, the gaits for bipedal locomotion control which have been generated using the Genetic Algorithm Optimised Fourier Series Formulation (GAOFSF) approach [1, 2] are investigated with a view towards exploring how easily, and to what extent, they can be adjusted in real time during walking in response to external disturbances or to desired changes in walking pace, stride and walking patterns.

The GAOFSF approach is conceptually inspired by the biological CPG (Central Pattern Generator) [3]. In previous research works on CPG [4, 5], the analyses were entirely conducted in the mathematical domain and there is no equivalent robot posture stability criterion to assist the elaborated CPG model. Therefore, there is no clear physical relationship between the adjustable parameters in the CPG and the resulting gait. As such, it takes much effort, mostly through trial and error, to be able to obtain suitable parameters for the CPG mathematical models, i.e. Van der Pol equation [4] and Neural Oscillator [5].

Comparing the previous ZMP-based [6,7,8] and CPG-based work [4,5,9,10,11,12] that have been done with the recently proposed GAOFSF approach gives the characteristics as shown in table 1. From the table, it can be seen that the GAOFSF approach applies the CPG concept using the ZMP stability criterion. This clarifies the feedback pathway in the gait generator and ensures the stability of the gait as well. Referring to [1,2], the framework of the GAOFSF is, briefly:

1. Prescribe the knee and hip joint trajectory by Truncated Fourier Series (TFS) functions, which are formulised as equation (1) and (2), with different time-shift values (figure 1) and unknown amplitudes. These TFS functions are composed to be the general walking pattern generator for the biped robot.

2. Mathematically express all the motion objectives and constraints so as to ensure an overall optimal performance and a valid walking motion.

3. Use the evolutionary algorithm, i.e. Genetic Algorithm, to search for values for all the unknowns for the GAOFSF model to give an optimal walking gait satisfying multiple objectives such as good robustness, stability, smooth motion and low foot ground impact.

4. Check the validity of the off-line generated motion according to the objectives and constraints set.

5. Elaborate key parameters having clear physical meanings for motion adjustment in real-time.

Steps 1 to 4 have been discussed in our published papers [1, 2]. In this paper, we focus on the discussion for step 5 and motion transitions. Section 2 discusses the stride-frequency adjustment. Section 3 presents how to adjust the walking step-length. Section 4 gives the motion pattern modification in real-time supervised by regressed functions. Section 5 presents the transition scheme for walking on uneven terrains and its results verified by dynamic simulations.

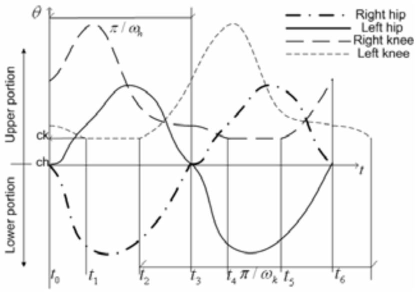


Figure 1: Typical shape of hip and knee joint trajectories.

Table 1: Properties of ZMP, CPG and GAOFSF based approaches.

	ZMP	**CPG**	**GAOFSF**
Relationship between mathematic model and dynamics	Clear	Unclear	Clear
Real-time gait adjustment without motion database	Sophisticated	Applicable	Applicable
Trajectory generation	Offline	Online	Offline generated; Online adjustable
Coordinate	Cartesian space	Inner coordinate	Joint space
Assumptions and constraints	Hard assumptions and constraints	Not applicable	No assumption; Soft motion constraints
Parameter online adjustable	Difficult to derive	Hard to predict	Simple and predictable

2 Stride-Frequency Adjustment

In the GAOFSF approach [1, 2], hip and knee joint trajectories are represented by the TFS model, as in equations (1) and (2) respectively.

$$\theta_{rh},\theta_{lh}=\begin{cases}\theta_h^+=\sum_{i=1}^{n}R\cdot A_i\sin i\omega_h(t-t_h^+)+c_h, & \text{positive angle;}\\ \theta_h^-=\sum_{i=1}^{n}R\cdot B_i\sin i\omega_h(t-t_h^-)+c_h, & \text{negative angle;}\end{cases}\tag{1}$$

$$\theta_{rk},\theta_{lk}=\begin{cases}\theta_{k1}=\sum_{i=1}^{n}R\cdot C_i\sin i\omega_k(t-t_k)+c_k,\\ \theta_{k2}=c_k\geq 0\end{cases}\tag{2}$$

where A_i, B_i, and C_i are constant parameters, R is a scaling constant, and t_h^+, t_h^- and t_k are all time-shift values (refer to figure 1). Therefore, the fundamental frequency in the TFS represents the step frequency directly. Simply adjusting ω_h (ω_k has a fixed relationship with ω_h) the stride frequency, the gait can be expanded or squeezed with respect to the time.

The objectives of adjusting stride frequency ω_h are:

1. To reduce ankle joint effort by regulating walking posture for transitions in response to external perturbations.

2. To maintain stable walking when there are excessive external perturbations.

To illustrate point 1, one experiment was conducted. An external push force of 110N was applied to the body trunk link from 5.44s to 5.53s when the stride-frequency ω_h =3.2. From dynamic simulation results, stable walking can be achieved without the need to update the walking rhythm ω_h. However, to attempt to reduce the ankle torque effort, the gait frequency, ω_h, was increased from 3.2 to 5. Figure 2 shows that a significant reduction in ankle joint effort can be achieved. Similarly, the stride-frequency ω_h can always be updated gradually according to the current state situation until the desired rhythm is achieved. It can also be applied in order to achieve a lower torque effort at the stance ankle when walking without external forces applied.

For point 2, a dynamic simulation was performed to study the effect of the application of an external impulse, in the form of a 160N force acting for 0.1s, on the robot walking on a flat-terrain. If the walking rhythm is not adjusted to accommodate for this external impulse, the robot's rhythmic motion will be disrupted resulting in instability. The stick diagram obtained from the dynamic simulation for this situation is as shown in figure 3. However, if in response to the external impulsive force, the walking pace is appropriately increased, the robot can maintain balance and continue with its stable walking.

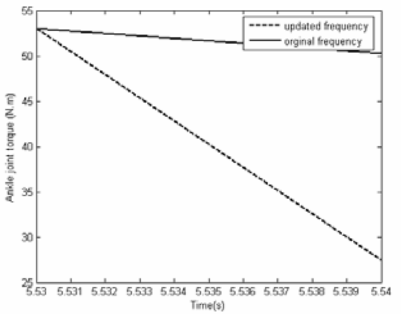


Figure 2: Stance ankle joint torque under situations of updating and not updating walking rhythm ω_h.

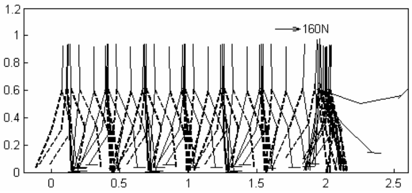


Figure 3: A disrupted walk under a pushing force.

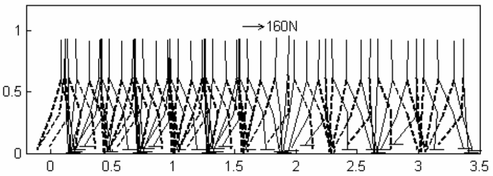


Figure 4: A successful walking under the same external pushing force.

Table 2: Stride-frequency range on different terrains.

	ω_h (min)	ω_h (max)
Flat terrain	3.2	7
10° uphill	2	6
10° downhill	4	8

Figure 4 shows the results of the dynamic simulation of such an example. In this case, the same impulsive force was applied but in response, the stride frequency, ωh, was increased from 3.2 to 5 and the step length increased from R=1 to R=1.1. After the implementation in dynamic simulations, the safe range of the flat-terrain walking, uphill and downhill walking on the 10º slope is shown in table 2.

The following figures 5 and 6 are adopted as an example to show the dynamic simulations of the ensured slowest walking and fastest walking on 10º up-slope terrain. The sampling time is 0.15s.

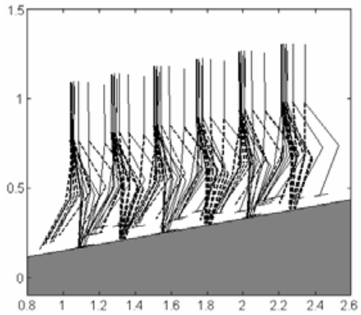


Figure 5: Slowest uphill walking.

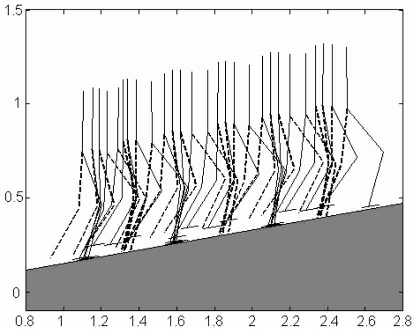


Figure 6: Fastest uphill walking.

3 Step-Length Adjustment

The change of stride is able to give a stable transition when human walk to avoid an obstacle or under perturbations. From equation (1) and (2), it can be noted that, using the TFS formulation, step-length can be easily adjusted by changing the value of just the scaling parameter *R*. The change does not only change the stride but also change the swing height. Therefore, it also helps to avoid the ground obstacles on some level. However, there is no guarantee that this will result in stable locomotion. To investigate this, the value of *R* was varied in the joint angle trajectories obtained earlier by GAOFSF for walking on all the three types of terrains: flat, 10º upslope and 10º down-slope. The stability of the resulting gaits was then investigated through dynamic simulations using Yobotics!.

The use of step-length adjustment is mainly for walking under perturbations which cause the CG position away from the desired one. If the CG is forward, the step-length allows to be adjusted bigger and vice versa. Table 3 shows the safely achievable step-length range of the flat-terrain, 10º uphill and downhill motions. Furthermore, just as the aforementioned, it helps to stride obstacles of a range of heights. The following figure 7 and figure 8 are taken as an example to show the stick diagram of the dynamic simulations of the downhill walking with the necessarily minimum step-length and maximum step-length. It can also be observed the swing height is much more obvious of the bigger-step walking.

Table 3: Step-length range.

	R **(min)**	***R*** **(max)**
Flat terrain	0.5	1.2
10ºuphill	0.5	1.3
10ºdownhill	0.7	1.1

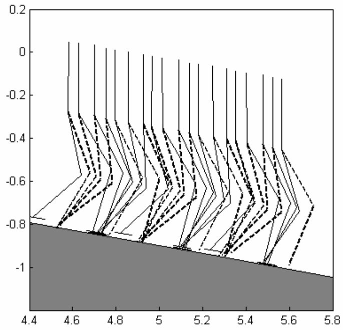


Figure 7: Small-step down-slope walking.

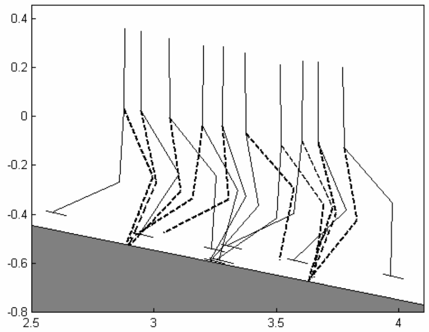


Figure 8: Big-step down-slope walking.

4 Pattern Adjustment

By a further observation of human gaits, human is supposed to walk with a natural relationship between two thighs, and the thigh, shank of the same side, to reduce the inner muscle forces when walking on rough-terrains. The supposed natural relationship can be conceived by modelling the two thighs and the thigh, shank of the same side as two virtual spring-damper systems which have their optimal elastic behaviour themselves. Therefore, it gives the intuition that the GAOFSF can also mimic such a character. The simplest way to keep a fixed link motion relationship is keeping the (A_i, B_i, C_i, t_1, t_2) same referring to figure 1.

If the leg motions are exactly the same w.r.t. the slope plane, referring to figure 9 and the geometrical knowledge, the following relationships can be established as:

$$\triangle OAB \cong \triangle OA'B'$$

$$\because |\theta_l'(t)| = |\theta_l(t)| - |\alpha| \qquad \theta_l(t), \theta_l'(t): \text{ possitive direction}$$

$$|\theta_l(t)| + |\theta_r(t)| = |\theta_l'(t)| + |\theta_r'(t)|$$

$$\therefore |\theta_r'(t)| = |\theta_r(t)| + |\alpha| \qquad \theta_r(t), \theta_r'(t): \text{ negative direction}$$

$$\therefore \begin{cases} \theta_l'(t) = \theta_l(t) - \alpha \\ \theta_r'(t) = \theta_r(t) - \alpha \end{cases} \qquad \alpha > 0$$

$$\because \theta(t) = \theta_{rh}, \theta_{lh} = \begin{cases} \theta_h^+ = \sum_{i=1}^{n} R \cdot A_i \sin i\omega_h (t - t_h^+) + c_h \\ \theta_h^- = \sum_{i=1}^{n} R \cdot B_i \sin i\omega_h (t - t_h^-) + c_h \end{cases}$$

$$\theta'(t) = \theta_{rh}', \theta_{lh}' = \begin{cases} \theta_h^+{}' = \sum_{i=1}^{n} R \cdot A_i \sin i\omega_h (t - t_h^+) + c_h - \alpha \\ \theta_h^-{}' = \sum_{i=1}^{n} R \cdot B_i \sin i\omega_h (t - t_h^-) + c_h - \alpha \end{cases}$$

$$\therefore c_h' = c_h - \alpha$$

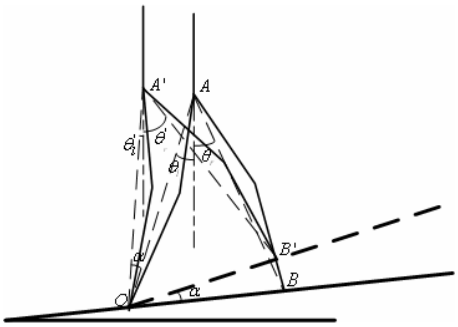


Figure 9: Geometrical illustration of walking pattern adjustment.

Therefore, even if simply only adjust the hip offset value according to $c_h' = c_h - \alpha$, as long as the resulting ZMP trajectory is still within the supporting polygon, this direct adjust still can secure a stable walking on the *OB'* plane. However, the range for the pattern adjustment is relatively restricted.

Inspired of the above and also considering the fact that human walking does not only keep the natural relationship, but also adjusts its leg posture pattern w.r.t the terrain surface, we feel that (A_i, B_i, C_i, t_1, t_2) actually needs to be changed while keeping the natural fixed relationship between links. However, adjusting (A_i, B_i, C_i, t_1, t_2) in real-time is not feasible in the GAOFSF method unless a motion database is provided. Therefore, instead of learning (A_i, B_i, C_i, c_h, c_k, t_1, t_2) all the time to store a motion database, adjusting (c_h, c_k) only in GA is much more efficient. The following discusses how to simply adjust (c_h, c_k) only to achieve the motion pattern adjustment and to compensate the side effect from the unchanged (A_i, B_i, C_i, t_1, t_2) as far as possible.

Firstly, the 10° uphill and downhill optimised motions are chosen as the references for the fixed relationship between links for going up and going down motions respectively to ensure a changeable range considering the geometrical analysis. Then, loosen the soft motion constraints in GA, i.e. the step-length constraint, but keep the (A_i, B_i, C_i, t_1, t_2) same and tune the (c_h, c_k) only. Since amplitudes of TFS model (A_i, B_i, C_i) play a great role on determining the walking stride, even though the constraint of step-length is fully loosened, it will not give a very different step-length compared with the one of the target walking ($\pm 10^o$ slope-walking). If necessary, to further compensate the effect from the fixed (A_i, B_i, C_i, t_1, t_2), a trunk pitch angle can be added to the computation.

From the results given by GA for different slope walking, the regressed cubic function for c_h and c_k versus the ground inclination is like equation (3):

$$\begin{cases} c_h = 6.3e^{-005}\alpha^3 - 1.2e^{-003}\alpha^2 - 1.8e^{-002}\alpha + 0.039 \\ c_k = -1.2e^{-004}\alpha^3 + 2.5e^{-003}\alpha^2 + 2.6e^{-002}\alpha + 0.12 \end{cases} \tag{3}$$

for going up motions. Figure 10 shows the resulting ZMP trajectories of motions on 2 and 15-degree up-slope terrains. It can be noted that the difference of ZMP trajectories of those 2 slopes are constrained small. Therefore, the feasible stride-frequency and walking step-length on different terrains are planned in a comparable range which reflects a good compensation to the unchanged (A_i, B_i, C_i, t_1, t_2).

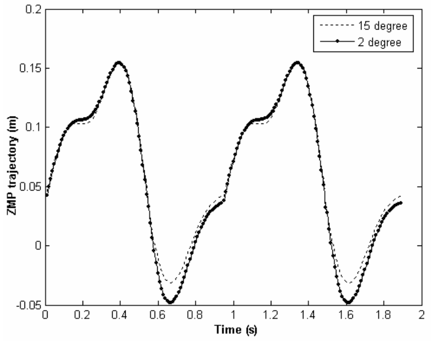


Figure 10: ZMP trajectories of motions on 2 and 15 degree up-sloped terrains.

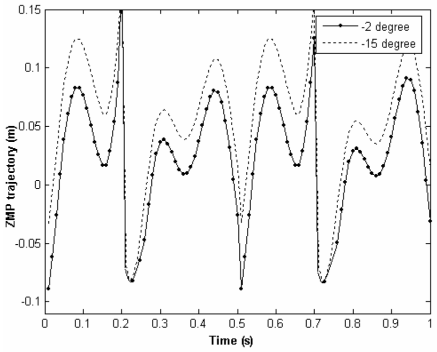


Figure 11: ZMP trajectories of motions on -2 and -15 degree down-sloped terrains.

For the going down motions, the regressed c_h and c_k function versus the terrain slope is like equation (4):

$$\begin{cases} c_h = 1.7e^{-005}\alpha^3 + 3.6e^{-004}\alpha^2 + 3.9e^{-003}\alpha - 0.22 \\ c_k = 1.2e^{-005}\alpha^3 - 4.5e^{-005}\alpha^2 - 2e^{-002}\alpha + 0.69 \end{cases} \tag{4}$$

Figure 11 shows the ZMP trajectories of motions on the -2 and -15-degree down-slope terrains. Similarly, the difference of the ZMP trajectories are also constrained small.

The resulting safe range of stride-frequency on different down sloped terrains is shown in table 4. The necessary and safe step-length range thus can be interpolated according to table 3.

Table 4: Stride-frequency range on different slopes.

slope	**2º**	**5º**	**7º**	**10º**	**12º**	**15º**
ω_h (min) $\omega_{h(\max)}$	2.7 6	2.7 6	2.6 6	2.6 6	2.6 6	2.6 6
slope	**-2º**	**-5º**	**-7º**	**-10º**	**-12º**	**-15º**
ω_h (min) $\omega_{h(\max)}$	3 8	3 8	3.2 8	3.5 8	4 8	4.5 8

From the above regressed equations (3) and (4), it can be noted the stance leg tends to be straightened for even slopes but to be bent more on steep slopes. This actually conforms to human walking gaits.

5 Transitions

In this section, walking transitions are discussed. Figure 12 illustrates motion feedback information, i.e. θ_h, θ_{st} and ground inclinations that are looked at.

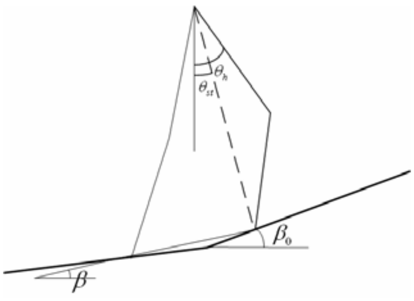


Figure 12: An illustration of the connecting phase.

To smooth the motion transition between two different terrains, or to sustain the walking stability under perturbations, a suitable connecting phase is very important. Here, a connecting phase P_t which is to indicate a proper transition is defined like equation (5)

$$P_t = \theta_{st} + 0.5(\theta_h - \theta_{st}) \tag{5}$$

This is because during transitions, stance leg is more important to the posture stability compared with what swing leg does. However, if only θ_{st} is concerned, only the posture phase has been considered. However, if the robot is under the case that the stance leg is bent obviously, the considered transitional phase will cause a big jerk. Furthermore, if the moment of inertia under such a case is bigger, it can cause the instability. Similarly, if only the hip joint angle θ_h is considered, there also is bias for the transitional phase, under which, the walking posture is not well described. Besides the P_t defined, a time-shift value is also necessary to be found out to indicate the very smooth connecting phase in the updated reference motion. Here, the whole period of stance phase is divided into 20 portions, P_i (i=1,2…20) according to (6)

$$\begin{aligned} &if\,(P_t < P_{i+1}\;\&\&P_t \geq P_i) \\ &shift\_t = 0.05i \cdot 0.5T; \end{aligned} \tag{6}$$

$$\begin{array}{ll} if\,(P_t < P_{\min}) & if\,(P_t > P_{\max}) \\ \{shift\_t = 0; & \{shift\_t = 0.5T; \\ \text{update }(R);\} & \text{update }(R);\} \end{array}$$

T is the walking cycle period. $P_{\min}$ and $P_{\max}$ are the maximum and the minimum transitional phase values respectively. During transitions, as long as either toe or heel of the swing foot touches the ground, loosen the swing foot ankle joint so as to comply the foot with the new terrain and thus give feedback information of the new slope. Motion transitions on rough-terrains are generally covered by the following 2 situations.

1. $\beta \leq \beta_0$: take β

2. $\beta > \beta_0$: take β_0

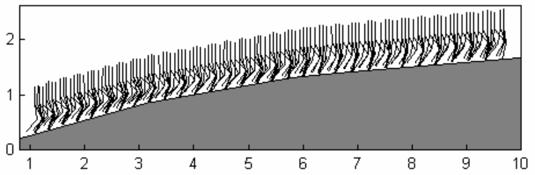


Figure 13: Motion on rough terrain 1.

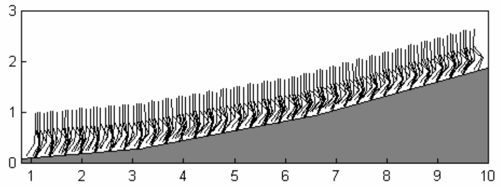


Figure 14: Motion on rough terrain 2.

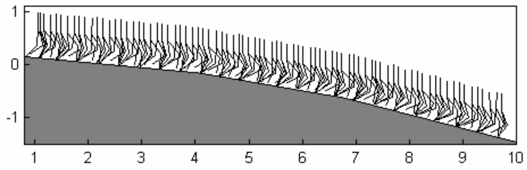


Figure 15: Motion on rough terrain 3.

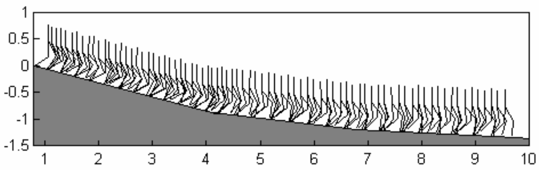


Figure 16: Motion on rough terrain 4.

From simulation results (figures 13 to 16), interestingly, the time shift value *shift_t* of the multi-slope uphill and multi-slope downhill motions are all zero, implying the start phase of the updated motion is very similar to the end phase of the last motion. It reflects adjusting c_h and c_k value gives a very consistent motion on rough-terrains (up and down).

When motion trajectories are updated completely, say the whole (A_i, B_i, C_i, c_h, c_k, t_1, t_2) is updated, the *shift_t* will work obviously on the motion transition. It indicates a proper connecting phase when reference trajectories update. Figure 17 to figure 20 show the transitions between flat-terrain and $\pm 10^o$ sloped terrain walking motions. Here, all stride-frequency are fixed. Therefore the controller forces the robot to quickly adapt to a nominated stride-frequency. If under such a case, the transition is still achievable, with the stride-frequency online adjustment, the transition will be even improved. The *shift_t* values of the following 4 walking transitions are 0, 0, 0.5s, 0.1s respectively.

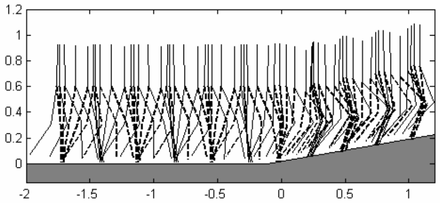


Figure 17: Motion from flat to up-slope terrain.

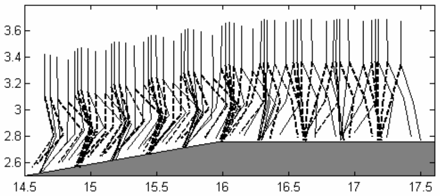


Figure 18: Motion from up-slope terrain to flat terrain.

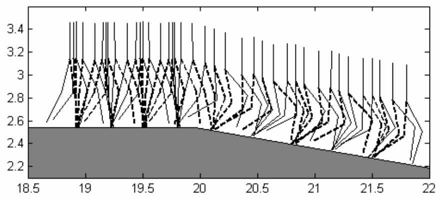


Figure 19: Motion from flat to down-slope terrain.

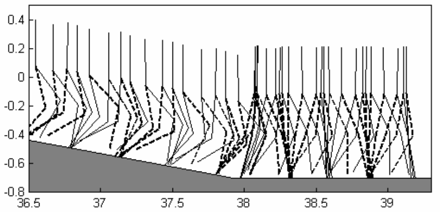


Figure 20: Motion from down-slope to flat terrain

6 Conclusion

By adjusting key parameters in the GAOFSF, three basic gait adjustment functions -- walking rhythm, walking stride and pattern -- have been achieved. Therefore, the GAOFSF can adjust (c_h, c_k, R, ω_h) to mimic a CPG function automatically. Although the adjusted motions have not been optimised considering objective functions [1, 2], they still ensure a similar walking rhythm and stride range of the optimised 10° uphill and downhill walking. Therefore, the compensation for the unchanged (A_i, B_i, C_i, t_1, t_2) is good enough to be accepted. The stable transitions show the pattern adjustment by tuning c_h and c_k is geometrically consistent. The *shift_t*, *R* and the selected feedback give smooth transitional motions on rough-terrains. The autonomous adjustment of (R, ω_h) will be presented in our following research papers.

7 References

[1] L. Yang, C.M. Chew, A.N. Poo and T. Zielinska, “Adjustable bipedal gait generation using genetic algorithm optimized Fourier series formulation”, *Proc. IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, Beijing, pp 4435-4440 (2006).

[2] L. Yang, C.M. Chew, T. Zielinska and A.N. Poo, “Reliable and adjustable biped gait generation for slopes using a GA optimized Fourier series formulation”, *CISM Courses and Lectures, RoManSy 16, Robot Design, Dynamics, and Control,* 487, pp187-194 (2006).

[3] S. Grillner, “Locomotion in vertebrates: central mechanisms and reflex interaction”, *Physiological Reviews*, 55, pp 247-304 (1975).

[4] M.S. Dutra, A.C. de P. Filho and V.F. Romano, “Modeling of a bipedal locomotor using coupled nonlinear oscillators of Van der Pol” *Biological Cybernetics*, 88(4), pp 286-292 (2004).

[5] Y. Fukuoka, H. Kimura and A.H. Cohen, “Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts”, *The International Journal of Robotics Research*, 22(3-4), pp 187-202 (2003).

[6] R.K. Rastogi, Task Oriented Stable Gait Synthesis in Biped Locomotion, Masters Thesis, Indian Institute of Information Technology, Allahabad (2005).

[7] J.H. Park, “Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motion of biped robots”, *Fuzzy Sets and Systems*, 134, pp 189-203 (2003).

[8] H. Lim and A. Takanishi, “Compensatory motion control for a biped walking robot”, *Robotica*, 23, pp 1-11 (2005).

[9] M.M. Williamson, “Neural control of rhythmic arm movements”, *Neural Networks*, 11, pp 1379-1394 (1998).

[10] S. Miyakoshi, M. Yamakita and K. Furuta, “Juggling control using neural oscillators”, *Proc. Intl Conf on Intelligent Robots and Systems*, Munich, pp 1186-1193 (1994).

[11] *Users Guide for Yobotics! Menu*, Yobotics, Inc., http://yobotics.com/index.html (2000-2003).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Robust Terrain Classification with a Convolutional Neural Network

Michael Happold and Mark Ollis
Applied Perception, Inc.,
Cranberry Township, Pennsylvania, United States
happ@appliedperception.com, mollis@appliedperception.com

Abstract

We propose a novel method for classifying terrain in unstructured, natural environments for the purpose of aiding mobile robot navigation. This method operates on range data provided by stereo without the traditional preliminary extraction of geometric features such as height and slope, replacing these measurements with 2D histograms representing the shape and permeability of objects within a local region. These histograms are then classified by means of a convolutional neural network trained to categorise terrain according to its level of traversability for a small mobile robot. With its shift invariance, the convolutional net provides a degree of robustness to noisy ground plane estimates, an Achilles heel of outdoor robotics. The system has been tested both on a small mobile robot and datasets from a wide variety of environments.

Keywords: autonomous systems, vision systems for robotics, artificial neural networks

1 Introduction

The complexity and irregularity of natural environments present a formidable challenge to robot vision systems. This is particularly the case because such vision systems must operate in real-time in order to be of any use for practical applications. Therefore, whether using imagery or range data, it is important to be able to reduce the complexity of the scene to a small set of compact representations.

A common method of extracting geometric features from range data is to separate points belonging to an approximate ground plane in the vicinity of the robot from points possibly belonging to obstacles [1]. Once these two classes of points are determined, features such as the maximum height and the slope within a local region can be easily and rapidly computed. Often, these two measurements alone are employed by hand-coding appropriate thresholds that encapsulate the robot's capabilities. This methodology, however, tends to be quite sensitive to noise, especially with respect to the ground plane estimate. And it is almost a certainty that there will be noise in this estimate.

Even given a reliable ground plane estimate, it is rather uncertain that such a limited set of geometric measurements can accurately represent the diversity of natural terrain types and their traversability in relation to a particular mobile robot. The tendency has been to ignore this diversity and force the environment into a binary representation. Depending on how one sets the thresholds on the feature measurements, this approach tends to produce either overly-aggressive or overly-conservative behaviour. What is desirable, rather, is to keep the robot safe while exploiting the full range of its capabilities.

In light of these two concerns, it is not surprising that recent efforts in terrain classification have explored one of two paths. The first is to inject learning into the system to improve the estimate of the ground plane beyond even robust fitting techniques [2]. The second is to enhance the terrain representation with features that are not dependent on the ground plane and capture physical characteristics beyond extent and orientation. One popular feature is density, which in a voxelised world is expressed by a ratio of the number of range points falling within a voxel to the sum of this number and the number of range points whose rays pass through the voxel [3]. Its purpose is to represent the penetrability of an obstacle, and thus possibly distinguish between vegetation and rocks. The use of density has certainly improved terrain classification.

An alternative to this paradigm is to allow a learning system to automatically determine which features are relevant for classifying terrain. The first significant work in this area fed raw imagery to a neural network, tagged with a driver's steering decision recorded along with the images, to produce a steering decision for an autonomous vehicle [4]. Though this represented a milestone in vision-guided robot navigation, translating this success to unstructured environments has proven difficult. Recent work [5] has attempted just such a domain shift, employing a convolutional neural network to learn steering decisions from imagery in off-road terrain, with results that suggest a great deal of promise. Merely learning steering angles from imagery, however, is not sufficient for navigation in complex environments where backing up might be required. Furthermore, methods for computing the 3D positions corresponding to classified image pixels without the use of explicit range measurements are very imprecise and often rely on a flat ground plane assumption.

In our work, we also apply a convolutional neural network in order to allow the system to learn what features are appropriate. However, rather than using raw imagery as input, our classifier operates on 2D histograms of 3D points within local regions as well as 2D histograms of rays associated with points outside of these regions (the permeability of a region). The ground plane estimate remains, though it is used only for approximate placement of the window governing the histogram extraction. The 2D histograms are extracted from a database of stereo range data collected with a small mobile robot in diverse environments. The network learns to mimic human classification of the terrain in labelled regions, extracting its own features from the histograms. It is able to learn complex classifications that easily thwart systems using simple geometric features, and is extremely robust to inaccurate ground plane measurements due to its shift invariance. Because the system is operating on 3D data, placing our terrain classifications into a map becomes trivial. The trained network has been incorporated into the robot's terrain classification system, where it runs in real-time and demonstrates performance superior to classifiers that rely on geometric features such as height and slope.

2 Approach

Our system operates in four stages. First, 2D histograms of 3D points and pass throughs are extracted from evenly-spaced regions in the point cloud provided from the stereovision system. These histograms are then range-compensated to account for the decrease in point density as range increases. They are then fed to a convolutional neural network trained to classify the histograms into one of four traversability classes. Finally, the output of the network is converted to a traversability cost and placed in a map for the robot's planning system to operate on.

2.1 2D Histogram Extraction

We have chosen 2D histograms because they provide a compact representation of the three-dimensional structure and appearance of regions in a scene. They are accumulated from the projections of 3D points onto 2D slices through the range data, or from the intersections of rays passing through a 2D window to points beyond the window's defined region. Point projection histograms have been used to pick out symmetric structures such as concertina wire from vegetation by comparing the histograms of points on either side of a 2D slice using normalised correlation [6]. We are unaware of any prior use of pass-through histograms.

We use an occupancy grid [7] to partition the range data into columns of 20 cm width x 20 cm length and infinite height. A 2D histogram is computed by first positioning its horizontal centre at the horizontal centre point of a column and setting its vertical position to be such that the ground plane intersects the window at the column's centre 20 cm above the window's lowest point. The window is divided into 14x80 (width x height) cells of size 2 cm x 2 cm, and thus represents a 0.28 m x 1.6 m planar region. A window centred on one column will thus overlap slightly with two neighbouring columns. For the sake of speed, we can eliminate this overlap, expanding our columns to 28 cm x 28 cm, reducing the number of histograms we compute and classify.

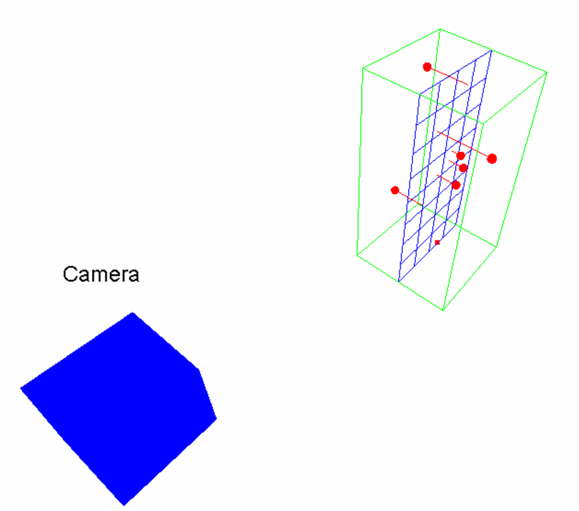


Figure 1: Point projection onto *PW* histogram.

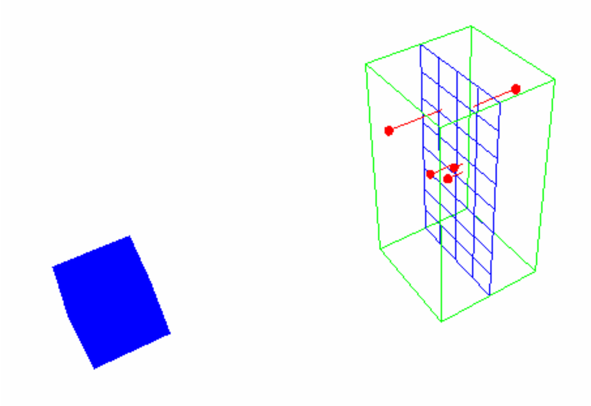


Figure 2: Point projection onto *OW* histogram.

A human expert hand-labels selected columns in the training data with one of four class labels representing increasing cost of traversal, with the highest cost understood to represent untraversable terrain. Three histograms are extracted for each selected region in the training data. One of these derives from a 2D window (*PW*), shown in figure 1, that is parallel to the vector from the reference stereo camera to the region centre and the world *Z* (up) vector and whose centre is aligned with the centre of the region. Points that lie within 14 cm on either side of this window are projected onto their corresponding cell, creating a histogram of the point count. A second histogram is created from a 2D window (*OW*), shown in figure 2, that is orthogonal both to the first plane and the horizontal vector from stereo reference camera to a point directly above the region centre, i.e., having the same *Z* value as the reference camera. Again, points within 14 cm on either side of the window are projected onto its cells and tallied. The final histogram (*RW*), shown in figure 3, represents the counts of the rays from the reference camera to 3D points that pass through this second window beyond its 14 cm inclusion region. Figure 4 shows the three histograms extracted from a column of range data.

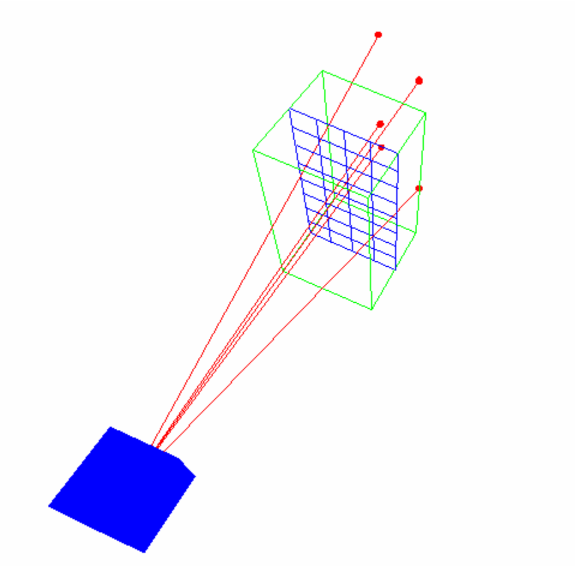


Figure 3: Ray intersection for *RW* histogram.

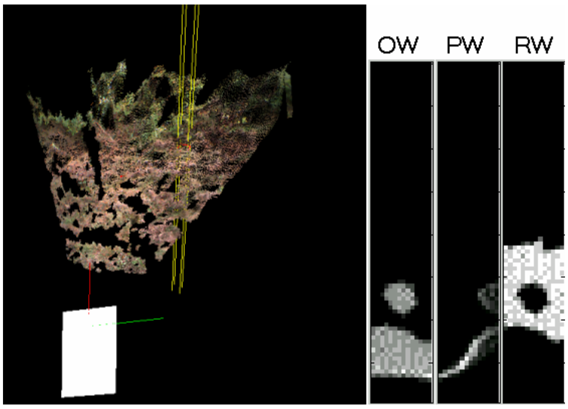


Figure 4: Example of histogram extraction from range data of a meadow.

2.2 Range Invariance

An important invariance that needs to be accounted for in the system is invariance to range. While the angle between two pixel rays remains the same with increasing range, the Euclidean distance between two 3D points similar distances along their respective rays increases with range. This implies that the point count histograms are not invariant with range, becoming less dense as range increases.

Growing the size of the histogram windows and their respective cells as range increases would account for the sparseness of range data at greater distances, but would in turn introduce a scale variability as objects would cover smaller regions of the windows as they grow. Instead, we choose to scale the point and ray intersection counts within cells by a function of range. The area of a 2D region subtending the stereo field of view is directly proportional to the square of the range. The number of pixel rays intersecting a 2D window of fixed size is therefore indirectly proportional to the square of its range from the stereo camera. This suggests that we ought to adjust histogram counts by the square of the range R . In order to test the efficacy of this method, we compare classification after adjusting histogram counts by a factor of R , $R\sqrt{R}$, and R^2 .

2.3 Convolutional Neural Network

Convolutional neural networks were developed to relieve designers of classification systems of the need to determine an optimal set of feature extractors as well as to handle the scale and shift invariance of inputs such as handwritten characters [8]. What distinguishes a convolutional neural network is the introduction of convolutional and subsampling layers. Convolutional layers consist of feature maps, each with its own set of weights shared among units within a map. The shared weights act as a trainable convolutional mask, and each unit within a feature map computes the response of a local receptive field within the input to the mask. A trainable bias, shared by all units within a map, is added to the response and a nonlinearity is applied. Because each unit within a map applies the same mask, the maps can be viewed as feature detectors that are invariant to position within the input. Maps in convolutional layers are often connected to multiple input maps, and weight sharing may occur across these multiple connections as well.

Further invariance to scale is introduced by the subsampling layers, which typically follow convolutional layers. Again, the subsampling layers consist of maps with a shared weight, and units within a map are connected to local receptive fields in the previous layer. However, unlike with convolutional layers, these fields do not overlap, so the subsample layer downsamples the previous layer. Each map in a subsample layer is typically connected to a single input map from the previous layer.

We trained six-layer convolutional neural networks on various combinations of these histograms, with the most basic taking the *OW* point count histogram as input. The convolutional layers learn shift invariant feature extractors, which removes the need to guess which features should be derived from the histograms. The shift invariant nature of the feature extraction is also essential to increasing the system's robustness to ground plane estimates. Training is performed with a variant of RPROP [9] suitable for convolutional networks. RROP uses an adaptive, weight-specific update value for changing weights that eliminates the possibly harmful effects of the magnitude of the partial derivative $\partial E / \partial w_{ij}$. Only the sign of the partial derivative is considered.

Backpropagating error can be tricky due to the border effects of convolution (an input unit on the boundary might only be connected to a single unit in the next layer, while interior input units will have multiple connections). The method of pushing gradients rather than pulling them was employed for ease of development [10]. A gradient is pushed as follows:

$$g_{j+i}^{L} = g_{j+i}^{L} + w_i^{L+1} g_j^{L+1} \qquad (1)$$

Here g_{j+i}^{L} is the gradient for the ($j+i$)th unit in the Lth layer. Thus, for each unit j in the next layer $L+1$ and each weight i, we backward propagate the gradient to a unit in layer L defined by j and i. This formulation allows us to ignore differences in the number of connections to border units.

The architecture for the simplest network is illustrated in figure 5, showing a single map for each layer and a half-sized input. In many ways, it mimics that of LeNet5 [8], which is used for character recognition. The full input to our network is a 14x80 histogram. The first hidden layer *C1* consists of six convolutional maps, each with its own set of 5x5 weights shared among units within a map. The next hidden layer *S1* consists of six subsample maps, one for each convolutional map in the previous layer. Each unit in a subsample map is connected to a unique 1x2 set of units in its corresponding convolutional map. The third hidden layer *C2* consists of 16 convolutional maps, each connected to a unique subset of the previous layer's maps with which it convolves a 3x3 set of weights unique to each convolutional map but shared across connections to the subsample maps. The fourth hidden layer *S2* consists of 16 subsample maps, each connected to a single map in the previous layer and subsampling 2x4 regions. The fifth hidden layer *F1*, containing 100 units, is fully connected to every unit of every map in the fourth hidden layer. Finally, the output layer is fully connected to the fifth hidden layer.

Two more complex networks use either the *PW* or *RW* histograms as additional input to a larger network. The first convolutional layer is expanded to have twelve maps, six connected to the *OW* input and six to the new histogram input. Subsequent layers are expanded correspondingly, although we limit the second convolutional layer to 21 maps rather than doubling it to 32 in order to limit computational complexity.

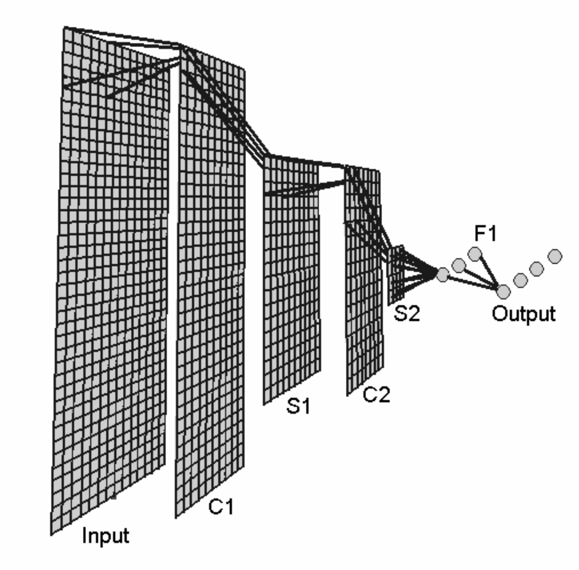


Figure 5: *OW* architecture showing input of half height (14x40).

2.4 Cost Conversion and Terrain Maps

We convert the output of the network to a cost by first scaling the four outputs to sum to unity. We then multiply each output by a value representing a cost between 25-255 corresponding to its cost class (25, 100, 150, 255). These costs were chosen to reflect aspects of the planner supplied by Carnegie Mellon with its robot (all costs below 90 are considered to be of cost 45, and 255 is considered lethal). We introduce an adaptive threshold on our costs so that costs greater than 230 map to the lethal value of 255. This threshold is adjusted downward by bumper hits on the robot. Costs are then placed in their appropriate map cell according to the position of the histograms that were classified. The map consists of 20 cm x 20 cm cells. The planner runs D* on the map to determine a global path.

3 Results

3.1 Platform

The data collection and testing platform is a small outdoor mobile robot developed for the Learning Applied to Ground Robotics (LAGR) program by Carnegie Mellon University and shown in figure 6. It is a differential drive vehicle with rear caster wheels. A WAAS enabled GPS provides position updates at 1 Hz, which are combined with odometry and the output of an inertial measurement unit (IMU) in an extended Kalman filter (EKF). Sensors include two IR range finders and two stereo heads. The stereo baseline for each stereo pair is 11cm and the focal length is 4mm. The two heads are rotated toward the centre of the vehicle by 20 degrees and tilted downward by 15 degrees. The horizontal FOV spanned by both stereo heads is 101 degrees. Stereo data is computed using the supplied PointGrey Triclops library.

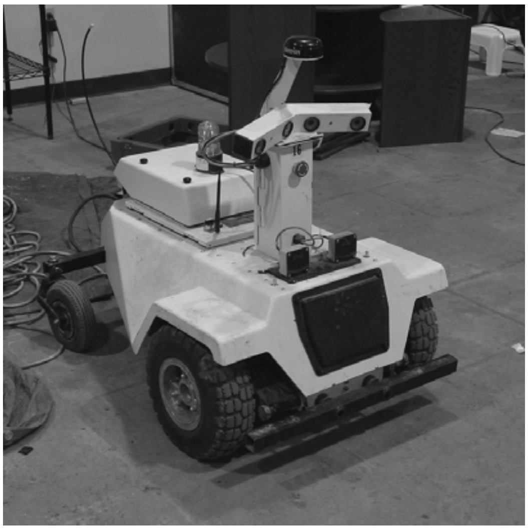


Figure 6: LAGR vehicle.

3.2 Training and Classification Results

For our dataset, we have extracted and labelled 15777 examples drawn from stereo data from eight different locations throughout the United States. Terrain types include forest, meadow, brush and hills. The data were collected over four seasons and at varying times of the day. This dataset includes terrain that is very difficult to discern using height or slope: rocks and logs on the one hand (not traversable), and tall grass (traversable) and bushes (partially traversable) on the other.

To evaluate the effectiveness of convolutional networks operating on 2D histograms, we compute the five-fold cross validation error [11] for the *OW* and the combined *OW+PW* classifiers using R^2 range compensation, and compare these results with those from a standard multi-layer perceptron trained on height and slope alone. Results are shown in figure 7, demonstrating a significant improvement in classification error when using a convolutional neural network on 2D histograms. There is also a noticeable improvement shown by the *OW+PW* classifier over the single histogram network.

To determine the importance of range compensation, we again compare five-fold cross validation error rates, this time for a single network architecture (*OW+PW*) with varying range compensation applied to its input. There is a mild but noticeable improvement as the degree of range compensation increases, with R^2 range compensation producing the best results.

Finally, we compare the two two-histogram combinations---*OW+PW*, and *OW+RW*---using R for range compensation. Figure 8 shows the five-fold cross validation scores for each, with the pass-through histogram providing better results.

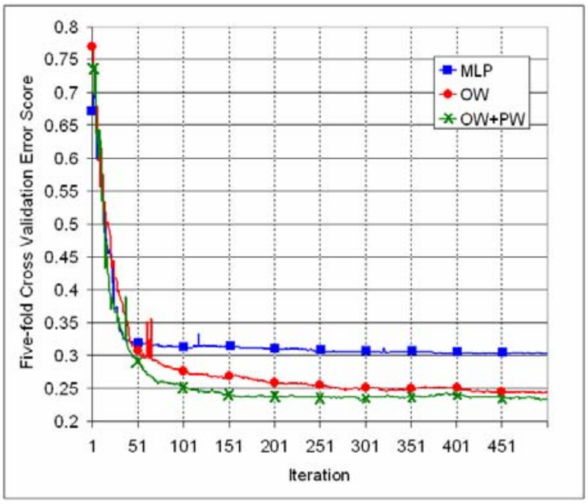


Figure 7: Comparison of an MLP using height and slope and convolutional networks using histograms.

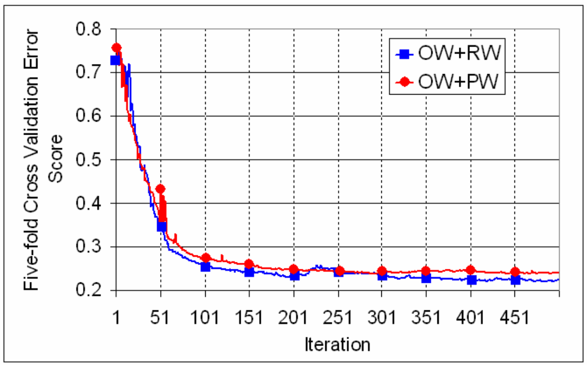


Figure 8: Comparison of two histogram convolutional networks with *R* range compensation.

3.3 Field Experiments

We have also tested the effectiveness of the convolutional neural network trained on *OW* histograms for driving in rugged terrain that is not represented in the network's training data. This network was fielded on the LAGR robot shown above. Only *OW* histograms were used because of the computational constraints of the platform (one 2GHz Pentium M is available for each stereo head). The entire vision system cycled at 1.5 Hz. The LAGR vehicle's speed is capped at 1.3m, and the vision system's cycle rate was found to be sufficient for this speed.

The test course was a winding path approximately 150m long through woods in western Pennsylvania. The straight line to the goal diverges from the path, bringing the vehicle amongst thin trees and fallen logs. The course difficulty exceeds that of most LAGR test courses. Nonetheless, the vehicle navigated the course smoothly, reaching the goal with only one bumper hit, which was on a log no more than 10 cm in height. Figure 9 provides representative samples of the course and the corresponding mapping to cost output from the neural network. Two thin trees are clearly picked out as very high cost. The average speed on the course was approximately 0.8 m/s. Subsequent runs showed similar performance; the only undetected obstacles were logs < 10 cm tall.

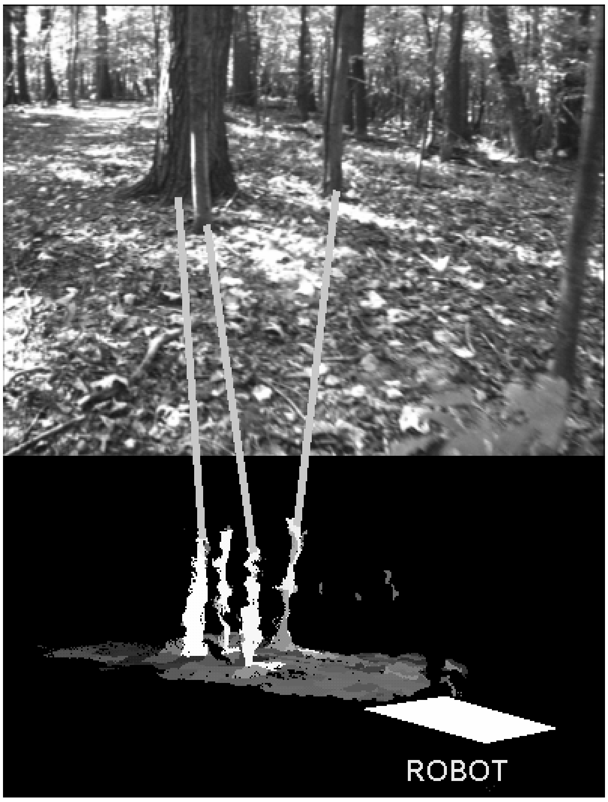


Figure 9: Correspondence between scene and greyscale encoding of cost for 3D data.

4 Conclusion

We have presented a novel method for terrain classification for mobile robot navigation that eliminates the need for guessing which geometric features are relevant. Instead, the system learns to mimic a human's classification scheme while learning to extract the appropriate features from 2D histograms of 3D range data. We also introduce the notion of a 2D pass-through histogram. Which aspects of these histograms are important for terrain classification are determined through the feature learning inherent in convolutional neural networks. The trained system outputs a class judgement, which is then converted to a traversability cost and placed in a map. The fact that we are operating on range data enables us to accurately localise the system's terrain classifications, an important gain over previous work. The value of this system has been demonstrated on a small mobile robot in outdoor, unstructured terrain.

5 Acknowledgements

This research has been financially supported through the DARPA LAGR program. The views and conclusions contained in this document are solely those of the authors.

6 References

[1] A. Talukder, R. Manduchi, A. Rankin and L. Matthies, “Fast and reliable obstacle detection and segmentation for cross-country navigation”, *Proceedings of the IEEE Intelligent Vehicles Symposium (IV2002),* Versailles, pp 610-618 (2002).

[2] C. Wellington and A. Stentz, “Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetation”, *Proceedings of the International Conference on Field and Service Robotics (FSR2003),* Tsukuba, pp 49-54 (2003).

[3] A. Kelly, O. Amidi, M. Happold, H. Herman, T. Pilarski, P. Rander, A. Stentz, N. Vallidis and R. Warner, “Toward reliable off road autonomous vehicles operating in challenging environments”, *Proceedings of the International Symposium on Experimental Robotics (ISER2004),* Singapore, pp 125-134 (2004).

[4] D. Pomerleau, “Alvinn: an autonomous land vehicle in a neural network”, *Proceedings Advances in Neural Information Processing Systems 2 (NIPS1989),* Denver, pp 305-313 (1989).

[5] Y. LeCun, U. Muller, J. Ben, E. Cosatto and B. Flepp, “Off-road obstacle avoidance through end-to-end learning”, *Proceedings Advances in Neural Information Processing Systems 18 (NIPS2005),* Vancouver, pp 739-746 (2005).

[6] N. Vandapel and M. Hebert, “Finding organized structures in 3-d ladar data”, *Proceedings Army Science Conference (ASC2004),* Orlando, pp 9-16 (2004).

[7] A. Elfes, “Occupancy grids: a stochastic spatial representation for active robot perception”, *Proceedings 6th Conference on Uncertainty in AI (UAI1990),* Cambridge, pp 60-70 (1990).

[8] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, 86(11), pp 2278-2324 (1998).

[9] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: the rprop algorithm”, *Proceedings of the IEEE International Conference on Neural Networks (ICNN1993),* San Francisco, pp 586-591 (1993).

[10] P. Simard, D. Steinkraus and J. Platt, “Best practices for convolutional neural networks applied to visual document analysis”, *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR2003),* Edinburgh, pp 958-962 (2003).

[11] R. Duda, P. Hart and D. Stork, *Pattern Classification* (2nd edition), John Wiley and Sons, New York (2001).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

A GA Based Controller for a Mobile Inverted Pendulum

Mark Beckerleg and John Collins
Engineering Research Institute
Auckland University of Technology, Auckland, New Zealand
m.beckerleg@aut.ac.nz

Abstract

Previous controllers for a two wheeled mobile inverted pendulum have used standard PID techniques. This paper presents a novel method which uses genetic algorithms to evolve the control system for a computer simulation of a mobile inverted pendulum. The heart of the controller is a lookup table that relates the current pendulum's angle and angular velocity to the motor direction and torque settings required to maintain balance of the pendulum. The lookup table is modified by a genetic algorithm using tournament selection, two-point crossover and creeping mutation with a 1% possibility of mutation occurring. It is shown that controllers capable of balancing the pendulum for time periods greater than three minutes could evolve within 200 generations.

Keywords: Evolutionary computation, genetic algorithm, mobile inverted pendulum.

1 Introduction

Mobile inverted pendulums have moved from university research to commercial reality with the advent of the Segway. With the commercialisation of the pendulum and model applications such as remotely controlled helicopters, gyroscopes and accelerometers have come down to affordable levels enabling universities and students to use these in project applications. The use of these devices and the pendulum model can also be modified for other fields such as bipedal locomotion [1] or game playing such as soccer [2].

Mathematical models describing the behaviour of the pendulum have been developed enabling the construction of PID controllers which provide excellent balancing performance [3, 4]. Other researchers have investigated fuzzy logic [5-7] and neural network [8, 9] controllers to maintain the pendulum balance.

This paper investigates the techniques of evolutionary computation and genetic algorithms to evolve the controller for the pendulum. Evolutionary processes have been extensively researched for robotic applications such as robotic navigation using evolutionary computation, evolvable hardware and evolutionary artificial neural networks. Genetic algorithms have not previously been used to control a mobile inverted pendulum.

The main difficulty in the field of evolutionary robotics is the large search space and time required to evolve a controller. Several methods have been used to try to overcome this by either reducing the chromosome and thus the search space [10], or by using subsumption architecture where individual behaviours of the robot are evolved independently of each other [11, 12]. In this application we use subsumption where the first behaviour to evolve is initially balancing. Future behaviours such as navigation or autonomy can be independently evolved.

The Auckland University of Technology has built a mobile inverted pendulum as part of a research project for evolutionary computation and evolvable hardware. The pendulum has been balanced with a controller using standard PID control software based on previous research. The hardware comprises two parts: a motherboard which has the physical components such as gyroscopes, accelerometers, motor drivers and wheel sensors required to balance the pendulum; and a daughter board which contains the processor used for control.

Three types of daughter boards have been created with three different processor cores, an Atmel 8 bit microcontroller, an ARM 32 bit processor and an Altera Cyclone II FPGA. The daughter boards have been created as a medium to provide research into evolutionary computation and several robotic platforms have been developed which interface to these boards. This will allow research into evolvable artificial neural networks, evolvable fuzzy logic and evolvable hardware controllers. The first stage of the research is to create a simulation of the pendulum to allow testing of genetic algorithms.

2 Mathematical Modelling

The pendulum is an example of a non-holonomic robot with three degrees of freedom (DOF), two planar motions and one tilt-angular motion, but with direct control in only the planar motions. Thus the control of the planar motion must work in such a way as to control the tilt/angular motion. As shown in figure 1, the pendulum can rotate around the z axis (tilt); this is described by its angle θ_p and its angular velocity ω_p. The pendulum can move on its x axis described by its position x and its velocity υ. The pendulum can also rotate around the y axis described by the angle δ and angular velocity δ'.

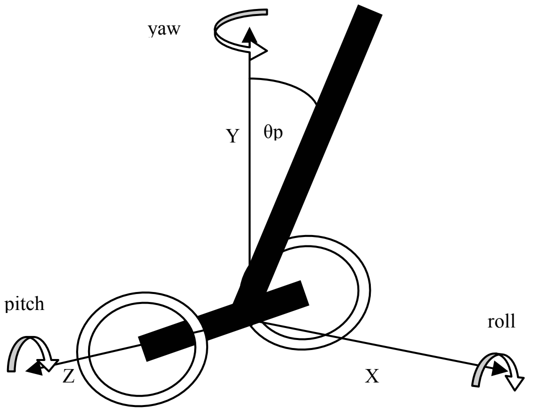


Figure 1: Physical model of a pendulum.

For this simulation the two wheels are driven together so there will be no yaw and the pendulum is on a flat surface so there will be no roll. The important parameters are the pendulum's angle, angular velocity and the linear displacement along the x-axis.

It can be shown that the equations (1) and (2) describing the motion of the system are

$$\left(M_p + 2M_w + \frac{2I_w}{r^2}\right)\ddot{x} + M_p l \cos\theta_p \ddot{\theta}_p = \frac{2T}{r} + M_p l \dot{\theta}_p^2 \sin\theta_p \quad (1)$$

$$M_p l \cos\theta_p \ddot{x} + I_p \ddot{\theta}_p = M_p g l \sin\theta_p - 2T \quad (2)$$

The computer simulation solves equations in quantised time intervals to determine the pendulum's current acceleration $\ddot{x}$, horizontal displacement $\dot{x}$, angle θ_p and angular velocity $\dot{\theta}_p$.

It is assumed that the motor torque T varies linearly with applied motor voltage and motor speed.

3 Genetic Algorithm

3.1 Chromosome

In evolutionary computation terms the chromosome is a possible solution or solutions to a problem. A single solution is termed an individual and a group of solutions is termed a population. In regard to the pendulum the solution is the required motor direction and torque for a given angle and angular velocity. The motor driver for the pendulum is an H-bridge driver which is controlled by an eight bit number. The number is a linear representation of the motor direction and torque with 0 representing the maximum reverse torque, 125 motor stopped and 250 the maximum forward torque. The number step size is 25 allowing eleven possible torque settings.

The chromosome is a two dimensional lookup table which converts the angle and angular velocity of the pendulum to the required motor direction and torque. The x-axis represents the pendulum's angle, ranging ±18 degrees from vertical. The angle step size is 3 degrees. The y-axis represents the angular velocity of the pendulum, ranging ±30 degrees/second. The angular velocity step size is 5 degrees/second.

The population size is 100 with the initial population randomly generated.

3.2 Reproduction and Selection

Chromosome reproduction uses a two point crossover scheme where two points within the chromosome are randomly chosen and the gene code within these two points is swapped between the two parents. Creeping mutation is also used, where the gene is replaced with a value within 10% of the non-mutated value. The probability of a mutation occurring is 1% and every mutation will randomly change 10 genes within the individual's chromosome.

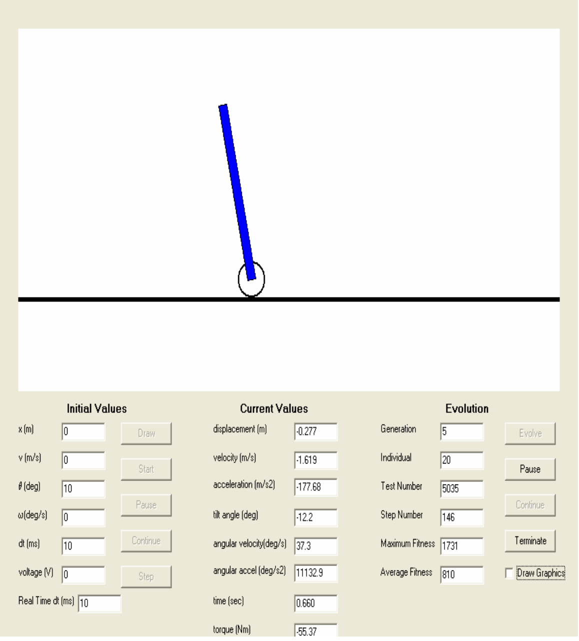
Tournament selection is used to select the best individuals. This process selects a subgroup of individuals from the population and only the fittest individual is retained. For this example the subgroup size is two, which has poor selection intensity but maintains the widest gene pool. An added advantage is that tournament selection requires no sorting of the population.

3.3 Fitness

The fitness is determined by the length of time that the pendulum remains within a vertical range of $\pm 18^0$ and remains within ± 0.5 metres of its start position.

4 Simulation

The simulation shown in figure 2 has a diagrammatic representation of the pendulum and numerical displays of the initial and current pendulum values of horizontal displacement, horizontal velocity, angle, angular velocity, individual balancing time and motor torque settings. The pendulum starting angle, current generation, individual, average fitness and maximum fitness are also displayed. These parameters are also saved to memory for future analysis. The simulation can be set to run in real time where the motion of the pendulum can be observed, or it can be speeded up: this allows faster evolution but the pendulum can not be observed.



Initial Values

x (m) 0
v (m/s) 0
θ (deg) 10
ω(deg/s) 0
dt (ms) 10
voltage (V) 0
Real Time dt (ms) 10

Current Values

displacement (m) -0.277
velocity (m/s) -1.619
acceleration (m/s2) -177.68
tilt angle (deg) -12.2
angular velocity(deg/s) 37.3
angular accel (deg/s2) 11132.9
time (sec) 0.660
torque (Nm) -55.37

Evolution

Generation 5
Individual 20
Test Number 5035
Step Number 146
Maximum Fitness 1731
Average Fitness 810
□ Draw Graphics

Figure 2: Simulation GUI.

5 Results

5.1 Simulation and GA Modifications

When the simulation was first run it was found that under certain conditions the controller would stop evolving. To prevent this both the simulation and the fitness criteria needed to be modified. These modifications are described next.

5.1.1 Variation of the Start Angle

It was found only a limited part of the controller array would evolve. Under investigation it was determined that the pendulum would learn to move from its initial offset starting angle and then maintain its balance. However the pendulum never evolved how to balance from other start conditions. To overcome this, the same individual is tested over a range of starting positions. The fitness was modified to accumulate the total time that the pendulum remained upright.

5.1.2 Horizontal Drift

The requirement is for the pendulum to maintain a balance within a limited horizontal displacement. However it was found that the pendulum would find a balance point with a vertical offset giving a constant horizontal motion. In order to penalise this characteristic, the individual undergoing fitness evaluation was terminated when the pendulum moved ± 0.5 metres from the initial horizontal starting position. Thus the fitness for each individual is determined by its ability both to balance the pendulum and to remain stationary.

5.1.3 Individual Not Leaving a Test

It was found that individual solutions were capable of kicking the pendulum upright from its initial starting position and then stabilising it within the ±0.5 metre range. This is a good result for one individual at one starting point; however, the fitness evaluation would never end and other start angles for that individual and other individuals in the population would never evolve. To overcome this, all individuals are terminated after a 5 minute interval.

5.2 Partial Evolution

It was found that some parameters of the array related to a high pendulum angle with high angular velocities did not evolve to the motor torque value expected from the mathematical model, i.e. moving to a maximum forward or reverse torque. This is because at these positions the pendulum is not able to maintain balance, even with the motor at maximum torque. Therefore the resulting fitness will allow suitable evolution of those parts.

5.3 Fitness

Five simulations were run for a maximum of 200 generations and the best and average fitness of each simulation was recorded. A typical result for evolution using tournament selection is shown in figure 3 with an initial large gap between the best and average fitness, with a gradual improvement of the average fitness with each generation. The best individual would have been expected to reach 300 seconds, but instead the best individual plateaus at 150 seconds with only small improvements after that. It is likely that the evolution was moving into local maxima. This could possibly be avoided by increasing the mutation rate.

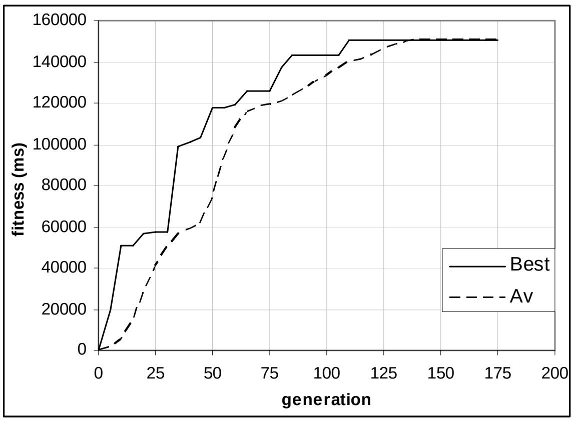


Figure 3: Average and best fitness.

Figures 4 to 5 graph the recorded best and average fitness of the five simulations. It can be seen that the best fitness stays constant for several generations and then increases in a large step when a better individual has been reproduced. The average fitness of the population has a gradual increase every generation as the individuals slowly improve towards the best individual.

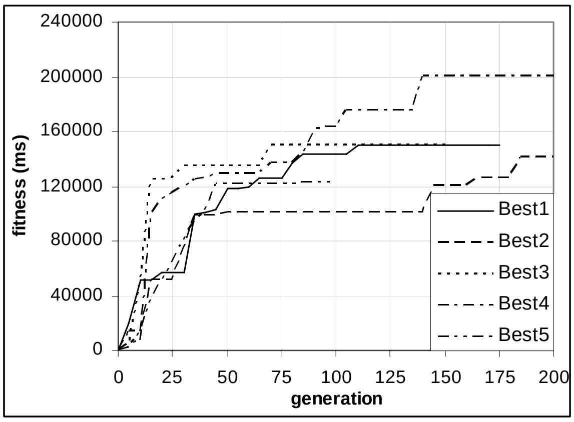


Figure 4: Best fitness.

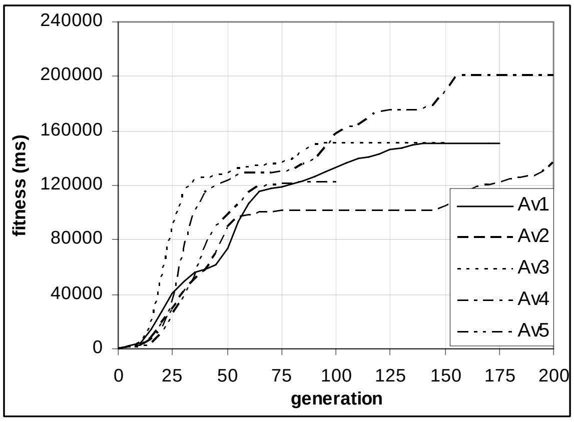


Figure 5: Average fitness.

6 Conclusions

This paper has described a method where a simulation of a mobile inverted pendulum controller can be evolved to a point where an acceptable level of balance is achieved. It was found that the GA produced a controller capable of balancing the pendulum for two minutes within 70 generations.

7 Future Research

The future aims of this project are to evolve the pendulum controller using both evolutionary computation and evolvable hardware. This can be researched using the three daughter boards with a range of processors and FPGAs. The AUT mobile pendulum has been built with trainer wheels which can be set so that it will start at a ±18^0 position.

A background software routine will be designed to reset the pendulum back to a starting position and angle. It will allow individuals to be tested and will take back control of the pendulum if it moves outside permitted limits, allowing the pendulum controller to evolve without human interaction. The background software will either perform the genetic algorithms if enough processing power is available, or it will transmit the data to a PC where the genetic algorithm can be performed.

8 Nomenclature

The following variables represent the pendulum characteristics on which the model is dependent.

g = acceleration of gravity

I_p = pendulum's moment of inertia around the z axis

l = distance to pendulum's centre of mass from the z axis

I_w = moment of inertia of the wheel around the z axis

M_p = mass of the pendulum

M_w = mass of the wheel

r = radius of the wheel

T = motor torque

x = horizontal displacement of the wheels

θ_p = angle of pendulum relative to the vertical axis

9 References

[1] K.K. Noh, J.G. Kim and U.Y. Huh, "Stability experiment of a biped walking robot with inverted pendulum," *Industrial Electronics Society (IECON) 2004, 30th Annual Conference of IEEE*, Busan, Korea, 3, pp 2475-2479 (2004).

[2] B. Browning, P.E. Rybski, J. Searock and M.M. Veloso, "Development of a soccer-playing dynamically-balancing mobile robot," *Proceedings IEEE International Conference on Robotics and Automation*, New Orleans, pp 1752-1757 (2004).

[3] F. Grasser, A. D'Arrigo, S. Colombi and A.C. Rufer, "JOE: a mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, 49, pp 107-114 (2002).

[4] Y. Kim, S. Kim and Y. Kwak, "Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot," *Journal of Intelligent and Robotic Systems*, 44, pp 25-46 (2005).

[5] X.M. Ding, P.R. Zhang, X.M. Yang and Y.M. Xu, "The application of hierarchical fuzzy control for two-wheel mobile inverted pendulum," *Electric Machines and Control*, 9, pp 372-375 (2005).

[6] S.S. Ge, Y.K. Loi and C.Y. Su, "Fuzzy logic control of a pole balancing vehicle," *Proceedings 3rd International Conference on Industrial Automation*, Montreal, pp 10-17 (1999).

[7] L. Ojeda, M. Raju and J. Borenstein, "FLEXnav: a fuzzy logic expert dead-reckoning system for the Segway RMP," *Proceedings Unmanned Ground Vehicle Technology VI*, International Society for Optical Engineering, Orlando, FL, USA, pp 11-23 (2004).

[8] S.S. Kim, T.I. Kim, K.S. Jang and S. Jung, "Control experiment of a wheeled drive mobile pendulum using neural network," *Proceedings 30th Annual Conference of IEEE Industrial Electronics Society (IECON 2004)*, 2-6 November 2004, Busan, South Korea, pp 2234-2239 (2004).

[9] T.C. Chen, T.J. Ren and C.H. Yu, "Motion control of a two-wheel power aid mobile," *Proceedings SICE Annual Conference 2005*, August 8-10 2005, Okayama, Japan, pp 2589-2593 (2005).

[10] I. Harvey, "Artificial evolution: a continuing SAGA", *Proceedings Evolutionary Robotics: From Intelligent Robotics to Artificial Life (ER 2001)* International Symposium, Tokyo, 2217, pp 94-109 (2001).

[11] R.A. Brooks, "A robot that walks: emergent behaviors from a carefully evolved network," *Proceedings IEEE International Conference on Robotics and Automation*, Arizona, pp 692-694 (1989).

[12] D. Cliff, I. Harvey and P. Husbands, "Evolutionary robotics," *IEE Colloquium on Design and Development of Autonomous Agents*, London, UK, pp 1/1-1/3 (1995).

Biomechatronics: Emerging Technologies, Applications, and Challenges

Winncy Y. Du[1] and Scott W. Yelich[2]
[1]Department of Mechanical and Aerospace Engineering
San Jose State University, San Jose, USA
[2]S&W Consultants, Santa Clara, USA
wdu@email.sjsu.edu, swyelich@sbcglobal.net

Abstract

This paper is a review of recent advances in the emerging field of biomechatronics – an interdisciplinary study of biology, mechanics, and electronics. The scope of biomechatronics spans botany, zoology, bionics, robotics, nervous systems, engineering cybernetics, materials, and control, with the objectives to understand the mechanisms behind the behaviour of biological organisms, and apply methods and systems found in nature to design and advance medical devices and life-support systems. The current fundamental and applied research on biomechatronics include modelling biophysics, kinematics and dynamics of humans/animals, development of prosthetic-orthotic devices and human assistive devices, study and advancement of nervous systems and engineering cybernetics. The technical challenges in biomechatronics, such as cost of high tech biomechatronic devices, design and manufacturing requirements, and clinical issues, are also discussed in the paper.

Keywords: biomechatronics, mechanics, electronics, prosthetics, orthotics, nervous systems, biology.

1 Introduction

Biomechatronics has been recognised as one of the 10 emerging technologies that could change the world [1]: *biomechatronics, airborne networks, quantum wires, silicon photonics, metabolomics, magnetic-resonance force microscopy, universal memory, bacterial factories, enviromatics,* and *cell-phone viruses*. In the next decade, biomechatronics is expected to play an increasingly influential role in the advancement of medical devices and life-support systems.

Biomechatronics is the interdisciplinary study of *biology*, *mechanics*, and *electronics* [2], with emphasis on the interactivity of biological organs with electromechanical devices and systems. The earliest biomechanical device, which can be traced back to early Greek theatrical simulations of living beings, conveyed a sense of life through motion, and motion remains a hallmark of artificial beings [3]. Primitive Biomechatronic devices have existed for decades, such as heart pacemakers, hearing aids, prostheses, stents, and implanted defibrillators and neural probes. Scientists foresee many new and innovative biomechatronic devices in the future including pancreatic pacemakers for diabetics, mentally controlled electronic muscle stimulators for stroke and accident survivors, micro-cameras and microphones wired into the brain to aid the blind and deaf, and microchips implanted under a person's skin to confirm health history and identify.

Biomechatronics is also the scientific and technical foundation for other emerging fields of research, medical application, and industry, including micro/nano technologies and manufacturing processes, artificial intelligence, biosensors and actuators, bio-electronic packaging, fibre optic imaging, and new materials. One of the major focuses of biomechatronics is to study and model the nature of living systems, which are highly efficient, optimal, and adaptive and serve as the best models for engineering design. Many of today's innovative designs and products have benefited from biomechatronic research findings. For instance, sonar/radar/medical ultrasound imaging techniques have resulted from imitating the echolocation of bats. The most important contribution of biomechatronics is in the advancement of healthcare and medicine. These advancements will (1) provide better quality of life, (2) improve clinical outcomes, (3) reduce costs, (4) encourage compliance, and (5) ensure greater patient satisfaction [4].

The field of biomechatronics is rapidly advancing with significant support from research funding and widespread collaboration among universities, government research organisations, and corporations. The growth in miniaturisation of electronics, micro-/nano- fabrication, biocompatible materials, CMOS technology, low power consumption components and chips, wireless communications, and wireless power transfer has also spurred advancements in biomechatronic systems. In 2002 US medical devices alone generated about $57.6 billion revenue. It continuously increases at annual growth rate of 8%.

This paper is organised as follows: section 1 is an introduction; section 2 presents the basis of biomechatronics; section 3 explores the state-of-the-art R&D in biomechatronics; section 4 identifies the economic, technical, and clinical challenges facing biomechatronics; and section 5 is a summary.

2 Basis of Biomechatronics

2.1 Definition

Biomechatronics is the interdisciplinary study of biology, mechanics, and electronics [2], see figure 1. *Biology* is the study of life and living creatures, including their structure, function, growth, origin, evolution, and distribution. *Mechanics* is a branch of physics concerned with motion and the forces that cause the motion. It includes study of the mechanical properties of materials, design, construction, and use of machinery or mechanical structures, as well as the functional and technical aspects. *Electronics* is a branch of physics that studies the emission and effects of electrons and the use of electronic devices, including use and design of electronic circuits, controlling, interface, processing and distribution of information/signals, and the conversion and distribution of electric power. Biomechatronics emphasises the interactivity of biological organs with electromechanical devices and systems.

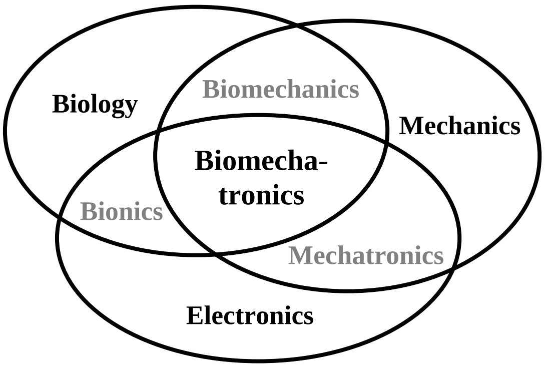


Figure 1: Biomechatronics: integration of biology, mechanics and electronics.

2.2 Objectives

With profound understandings of how biological systems work, behave and interact, and knowledge of how to simulate these biological systems using mechanical structures combined with electronics, biomechatronics is focused on two primary objectives, to:

1. Guide the design and fabrication of novel, high performance bio-inspired machines and systems, for many different potential applications.
2. Develop novel devices that can better act on, substitute parts of, and assist human beings, such as in diagnosis, surgery, prosthetics, rehabilitation and personal assistance.

2.3 Scope

Biomechatronics involves the following areas:

Study of Plants (Botany): An example includes studying the *Lotus Effect* of the water lily (a flower that never gets wet or dirty, figure 2), to design dirt- and water-repellent paint and coatings, and self-cleaning glass, fabric, and tile. A number of new products are already available using lotus-effect surfaces, including roof shingles, car paint, and a building facade paint. For example, StoLotusan paint by Sto Ltd in UK (http://www.sto.co.uk/).

Figure 2: Lotus - a flower that never gets wet or dirty.

Study of Animals (Zoology): Research has found that a bat can determine an object's size, shape, direction, distance, and motion by producing sounds of 30 kHz and higher. Study of the echolocation of bats has improved sonar, radar, and medical ultrasound imaging (figure 3 [5]).

Figure 3: A bat can determine an object's size, shape, direction, distance and motion.

Study of Humans (Anatomy): Knowledge of human anatomy led to the development of the artificial heart to save lives (figure 4).

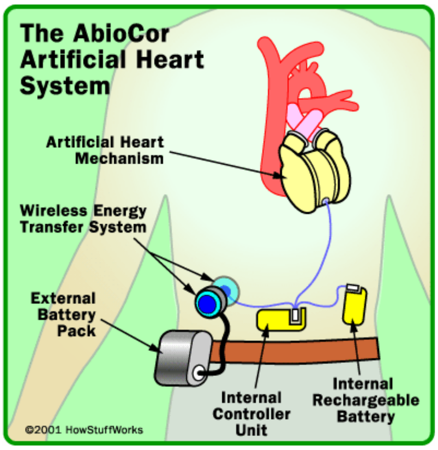


Figure 4: An artificial heart system.

Investigation of New Technologies and Biocompatible Materials: An exploration of new manufacturing methods, intelligent systems, nano-bio robotics, sensors, internal diagnostics, molecular probes, and new biocompatible coating materials, is essential to increase the speed, accuracy and functionality of biomechatronic devices.

Development of Implant-on-a-chip Technology: Many biomechatronic devices involve integration of various mechanical and electronic components that must have minimum sizes, extremely low power consumption, and maximum life (e.g. more than 10 years). Implant-on-a-chip technology provides a future solution.

2.4 More Examples of Biomechatronics Products

A typical biomechatronic device, a C-leg prosthesis (by *Otto Bock HealthCare*, Minneapolis, Minnesota), is shown in figure 5 [6]. It consists of a carbon fiber frame (1), providing the supporting structure and housing for hydraulics with servo-motor (2) and electronics (3), distal tube clamp (4), upper joint section with pyramid for socket connection (5), and shin tube adapter with integrated strain gauges (6).

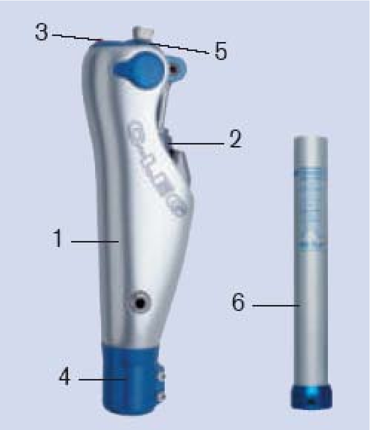


Figure 5: C-leg: knee-inspired structure design + mechanical components + electronics.

Figure 6 shows another example of a biomechatronic device, a micro-cricket robot (figure 6a) and its major components (figure 6b) developed by Case Western Reserve University.

Figure 6a: A micro-cricket robot.

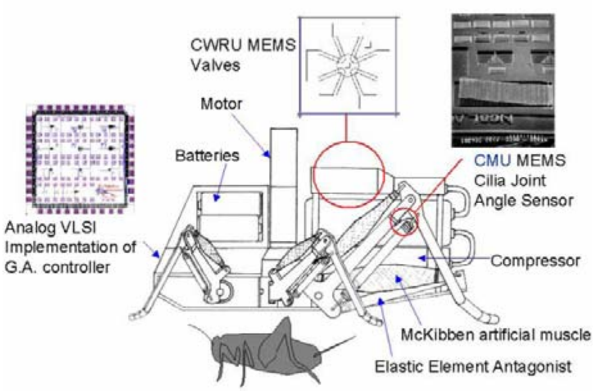

CWRU MEMS Valves
Motor
Batteries
Analog VLSI Implementation of G.A. controller
CMU MEMS Cilia Joint Angle Sensor
Compressor
McKibben artificial muscle
Elastic Element Antagonist

Figure 6b: Major components of a micro-cricket robot.

3 State-of-the-Art R&D in Biomechatronics

3.1 Research Scope and Objectives

The general scope of biomechatronic research spans the fundamental, theoretical and experimental challenges posed by the application of Mechatronics in Medicine and Biology. The main focus of biomechatronics is to analyse biological systems from a biomechatronic point of view, to understand the scientific and engineering principles underlying their extraordinary performance. This leads to the following research objectives, to:

1. Understand and model the mechanisms behind the behaviour of biological organisms.
2. Expand the boundaries of current technology by incorporating insights from Biology.

3.2 Research Subjects

3.2.1 Modelling Biophysics, Kinematics and Dynamics of Humans/Animals

Biophysical models of muscles and joints, and of the kinematics and dynamics of human/animal motion are useful in the design and analysis of innovative biomechatronic systems. Examples of research subjects are:

1. Modelling of the human soft tissue: the human soft tissue can be modelled as

   $$K(\lambda) = dT/d\lambda = \alpha\beta e^{\alpha(1-\lambda)} \qquad (1)$$

   where α and β are tissue parameters, λ is the compression length ratio, T is the uniaxial compression stress, and K is the tissue stiffness. The model is useful in selecting silicon rubber materials that have similar behaviour to replace the real soft tissue for haptic surgical research [7].
2. Dynamic model of vertebrate skeletal muscle (VSM): dynamic properties of VSM can be

modelled as an actuator whose output force is a function of length, velocity, and level of activation. At maximal activation, this relationship can be described by a parabolic, dimensionless equation:

$$F_L / F_{L_0} = c_1 (L / L_0)^2 + c_2 (L / L_0) + c_3 \qquad (2)$$

where F_L is the isometric force at muscle length L, F_{L_0} is the isometric force at the muscle's "resting" length L_0. This model is the basis to design bio-inspired actuators and sensors [8].

3.2.2 Development of Prosthetic and Orthotic Devices

Applied research on the development of microprocessor-controlled prosthetic and orthotic devices has been based on the following fundamental research: innovative socket configurations, improved knee mechanisms, new suspension options, novel materials (for thinner, lighter, and stiffer components), biomechanics of locomotion, and soft tissue mechanics (to develop low cost dynamic transducers).

Recent advancements in prosthetic technology have been significant and they could develop into an effective alternative for organ transplants in the future. While the biggest developments were witnessed in prosthetic legs, the advances in arm and hand prosthetics have led to superior microprocessor-controlled devices as shown below:

1. Microprocessor-controlled Knee: a joint that uses nanotechnology to function more like a biological knee (Massachusetts Institute of Technology).
2. Dextra Artificial Hand: a device that enables a person to use existing nerve pathways to control individual computer-driven mechanical fingers. (Rutgers University and Nian-Crae Inc.).
3. Prosthetic Arm: an artificial limb that can communicate with the human body's neural network (University of Adelaide).

As for the future of prosthetics, new research is speeding the development of brain controlled devices that may allow amputees and paralytics to use their limbs. Soon these so-called brain-computer interfaces (BCI) may also allow people completely paralysed by neurodegenerative diseases or trauma to regain some movement. In the future, microprocessors could become standard devices for developing adaptive prostheses. Sensors and microprocessors have allowed excellent control through which a prosthesis can be adjusted according to the movement patterns of each individual.

3.2.3 Assistive Technology and Devices

Research on the development of assistive technology has been complimentary to the development of biomechatronic prosthetic devices. Assistive robots can aid people with disabilities in performing daily living tasks, or performing tasks that are too difficult or dangerous for humans, and even enable doctors to perform delicate surgery. Several surveys [9] have identified key areas in which assistive devices can help: (1) Eating and drinking, (2) Personal hygiene, (3) Work and leisure, (4) Mobility, and (5) General reaching and moving. Examples include, the:

1. Weston Wheelchair: a wheelchair mounted robot arm that assists the elderly and disabled (Bath Institute of Medical Engineering, UK).
2. Project Cyborg: a silicon chip transponder implanted into the forearm that operates doors, lights, computers and other electronic devices (University of Reading [10]).
3. Lower Extremity Exoskeleton: a mechatronic device attached to the lower body to effectively take the load off people's backs when walking (U.C. Berkeley).
4. Da Vinci Surgical System: an assistive robot that allows surgeons to perform delicate surgical procedures for a variety of cardiac operations that were previously impossible without open exposure (Intuitive Surgical, Inc, Mountain View, CA).

3.2.4 Nervous Systems and Engineering Cybernetics

Nervous Systems and Engineering Cybernetics relate to neuroscience, robotics, interface and sensor technology, communication, dynamic systems and control. The research efforts mainly focus on real-time modelling of neurons, nervous system simulation, and interfacing electronics to the nervous system – either using the potential of microtechnology and microelectronics, or the electroneurophysiological and neuro-anatomic possibilities and constraints. The key research issue is how to contact nerve fibers as selectively and efficiently as possible.

The field of neutral stimulation is experiencing fast-paced innovation on two fronts, namely *electrostimulation* and *physical neural rehabilitative therapy*, with the aid of assistive and rehabilitative robots. Ongoing research is focused on developing neutral electrodes to tap nerve signals and connect them in order to operate prosthetics.

In recent years, data from neurobiological experiments have made it increasingly clear that biological neural networks, which communicate through pulses, use the timing of the pulses to transmit information and perform computation. This realisation has stimulated a significant growth of research activity in the area of pulsed neural networks, ranging from neurobiological modelling and theoretical analyses, to algorithm development and hardware implementations [11]. Such research is motivated both by the desire to enhance understanding of information processing in biological networks, as well as by the goal of developing new information processing technologies. Following are a few examples.

1. Study of the human neuromuscular-skeletal system to assist impaired human motor control and find possible methods of interfacing with the human body at all hierarchical levels of the human motor system.
2. Development of the Cochlear Implant, a small complex electronic device (surgically placed under the skin behind the ear) that can help to provide a sense of sound to a person who is profoundly deaf or severely hard of hearing (Cochlear Corporation).
3. Development of the Neurobot, a robot that is equipped with an artificial neural network that enables advanced learning and adaptation by evolutionary knowledge processes using a simulated neural network or integrating with an organic neural network (Carnegie Mellon University, University of California San Diego, Neurobotix.com).
4. Investigation of methods of introducing electrical signals to the pathway leading to the visual cortex of the brain. *Optobionics* (Wheaton, Illinois) developed such a method: bypassing the entire eye, a grid of electrodes is implanted directly onto the visual cortex of the brain, with wires exiting the skull in a small bundle. This grid can be stimulated by signals received from an externally worn camera. Such a system was successfully demonstrated at 20/400 (in a person with no normal light perception).
5. Several methods are being developed for support and training of the impaired neuromuscular functions by interaction with the neuromuscular system, e.g. in people with spinal cord injury and stroke. These methods include *Functional Electrical Stimulation* (FES) to activate muscles [12] or influence the central nervous system, electrotactile interfacing, *Electromyography* (EMG) and the measurement of body movements with inertial sensors.

4 Economic, Technological and Clinical Challenges

4.1 Cost of High Tech Biomechatronic Devices

This emerging technology can be cost prohibitive. An above-the-knee biomechatronic prosthetic device can cost more than $20,000 (source: Kevin Carroll, vice president, Prosthetics, Hanger Corp.). A Weston wheelchair mounted robot arm costs about $28,000 (http://staff.bath.ac.uk/mpsmrh/robot/westonspec.htm). A ceramic-on-ceramic hip is at least $7,500 and a metal-on-metal hip is about $5,500 [13].

4.2 Design and Manufacturing Issues

The following design and manufacturing issues have been identified in developing biomechatronic devices: (1) liability, (2) biocompatibility, (3) reliability, (4) packaging, (5) system traceability - 100%, (6) moisture resistance, (7) mechanical shock, (8) vibration, (9) fatigue, (10) thermal shock, (11) battery life, (12) electromagnetic interference (EMI), (13) corrosion, (14) sterility and (15) miniaturisation.

4.3 Clinical Issues

The development of new biomechatronic devices for human use is requiring more resources, more time, and more patience than ever [14]. Each person who wants to market Class I, II and some III devices[1] intended for human use in the U.S. must submit a 510(k) to the FDA (Food and Drug Administration) at least 90 days before marketing (called *Premarket Notification*) unless exempt from 510(k) requirements. All 510(k) applications for Class II and Class III devices now require clinical trials to justify that *safety* and *efficacy* of the device is similar to previously marketed devices. Further, these clinical trials must be performed in a *randomised, prospectively controlled* manner, meaning some patients receive treatment by prior devices, and others with new devices. An approval process can take years, depending on product type and intended use.

Since adequate clinical trials are now more specifically defined and more frequently required, several critical issues need to be addressed when designing a biomechatronic device for human use:

1. Similarity of the method in the new device compared to the one in previously marketed devices.
2. Proper approach to patient selection, including means to assure similar features in treatment and control groups.

[1] Defined by FDA: A *Class I device* presents minimal potential harm to the user and are often simpler in design, which are subject only to general controls. A *Class II device* is those for which general controls alone are insufficient to assure safety and effectiveness, thus it is also subject to special controls. Existing methods are available to provide such assurances. A *Class III device* is one for which insufficient information exists to assure safety and effectiveness solely through the general or special controls. Such a device needs premarket approval, a scientific review to ensure the device's safety and effectiveness.

3. Different phases of device testing as defined by FDA and limitation of clinical testing.
4. Malpractice risk in clinical trials and redundant safety design.
5. Familiarity with *FDA regulations: 21 CFR Part 820 - Quality System Regulation*, (http://www.fda.gov/cdrh/gmp/gmp.html) for the medical device and biotech industries.

In Europe, medical products cannot be commercialised in any European Union country without the CE Mark, since June 1998. The Medical Dispositives Directive (93/42/CEE) brings under control the Essential Requirements, which guarantees the efficiency and safety for products manufactured within these contexts. Besides these requirements, it is also necessary for the manufacturer to maintain an updated quality certified system based on International Standards (ISO9001/EN46001).

5 Summary

Biomechatronics, the synergetic integration of biology with mechanical and electronic systems, is playing an increasingly influential role in the advancement of healthcare and medicine. There is a growing market that will continue to grow for the foreseeable future. The scope of biomechatronics spans botany, zoology, bionics, robotics, nervous systems, engineering cybernetics, and control. Its main research focus includes understanding and modelling biological organisms and systems, and developing innovative mechatronic devices (e.g. prosthetic-orthotic devices, joint implants, artificial hearts, neurological disorder diagnostic devices, neuro-rehabilitation robots). Biomechatronics is also the scientific and technical foundation for other emerging fields of research, medical application, and industry, including micro/nano technology, artificial intelligence, biosensors and actuators. Compared to other emerging technologies, biomechatronics still faces many challenges including miniaturisation, cost, clinical and ethical issues, and barriers to commercialisation.

6 Acknowledgements

The authors thank Mr. Matt Murphy, Vice-President of Adept Technology, Inc. (Livermore, California) for sharing his expertise and insights in medical device design and manufacturing, and his knowledge of regulations and quality control on medical devices for the preparation of this paper.

7 References

[1] Technology Review: An MIT Enterprise, "10 Emerging Technologies", http://www.technologyreview.com/read_article.aspx?id=14407&ch=infotech, visited on 11/18/2006.

[2] Whatis?com: The Leading IT Encyclopedia and Learning Centre, "Biomechatronics", http://whatis.techtarget.com/definition/0,,sid9_gci528845,00.html, visited on 11/18/2006.

[3] S. Perkowitz, *Digital People: From Bionic Humans to Androids*, Joseph Henry Press, Washington, DC (2004).

[4] R.G. Bushko, "Defining the future of health technology biomechatronics", *Studies in Health Technology and Informatics*, 80, pp 3-18 (2002).

[5] Wikipedia: The Free Encyclopedia, "Bat", http://www.anotherstateofmind.net/bat%20necklace.JPG, visited on 11/18/2006.

[6] *The Electronic 3C100/3C105 C-Leg ® Knee Joint System: Instructions for Use*, Otto Bock, Minneapolis (2006).

[7] J. Longnion, J. Rosen, M. Sinanan and B. Hannaford, "Effect of geared motor characteristics on tactile perception of tissue stiffness," *Proceedings Studies in Health Technology and Informatics: Medicine Meets Virtual Reality*, Newport Beach, CA, pp 286-292 (2001).

[8] B. Hannaford, K. Jaax and G. Klute, "Bio-inspired actuation and sensing," *Autonomous Robots*, 11(3), pp 267-272 (2001).

[9] S. Prior, "An electric wheelchair mounted robotic arm: a survey of potential users", *Journal of Medical Engineering and Technology*, 14, pp 143-154 (1990).

[10] K. Warwick, "I, Cyborg", University of Reading, http://www.kevinwarwick.com/ICyborg.htm, visited on 11/18/2006.

[11] C.M. Bishop, *Pulsed Neural Network*, MIT Press Boston (1998).

[12] J. Kameyama, M. Sakurai and Y. Honda, "Control of shoulder movement by FES", *Proceedings Annual International Conference of IEEE Engineering in Medicine and Biology Society*, 13(2), pp 871-872 (1991).

[13] A.S. Brown, "Hip new world", *Mechanical Engineering*, 128 (10), pp 28-32 (2006).

[14] S.A. Guelcher and J.O. Hollinger, *An Introduction to Biomaterials*, CRC Press/Taylor and Francis Group, Boca Raton (2006).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

A Novel Heuristic Approach to Real-time Path Planning for Nonholonomic Robots

H. Mahjoubi and F. Bahrami
Control and Intelligent Processing Centre of Excellence,
School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
h.mahjoubi@ece.ut.ac.ir, fbahrami@ut.ac.ir

Abstract

This paper introduces a local path planning and navigation algorithm for nonholonomic robots. Using a novel representation method, a reduced-dimension and precise search space is provided for the algorithm. The characteristics of this search space accelerate the performance of the algorithm, making it suitable for real-time applications. A heuristical weighting approach is developed for the path planning module. Incorporating the effect of direction, this part is responsible for detection of an appropriate sub-goal within the sight of the model. Special potential fields are applied to navigate the model toward this point. To tackle the local minima problem, the wall following method is effectively adapted. The results of simulations in concave, maze-like and unstructured environments have shown that the proposed method is highly robust.

Keywords: environment representation, heuristical path planning, motion control, nonholonomic robots.

1 Introduction

The problem of path planning is a well-known problem in the field of robotics. The objective is to find collision-free trajectories for a system that can move in unknown environments without human intervention to fulfil a given task. Specifically, this system must be able to travel between a start and a goal configuration while avoiding static and/or dynamic obstacles in the environment.

Development of robust techniques for autonomous navigation is an important challenge in robotics. Currently, many new applications demand for truly autonomous robots that can operate in real-world and unstructured environments. Maintenance robots operating in hardly accessible or hazardous zones [1] and semi-autonomous vehicles for disabled people [2] are two of such applications.

While proceeding in an unknown environment, the robot may face many unexpected situations. The path planner must be able to handle such situations in real-time. Thus, the algorithm to be used must have a relatively high speed. One of the most effective factors in the speed of the path planner is the method used to represent the surrounding environment. Based on the selected method, the representative data (or the search space) can vary in structure and dimensions. These two characteristics of the search space have a direct effect on the speed of the path planning algorithm. An irregular structure as in *object oriented maps* [3], *free space maps* [3] or *quad-trees* [4] may decrease the speed of the algorithm by introducing complications in the storage and access to different parts of the representative data. Whereas, regular representations such as *composite-space* (*CS*) *maps* [5] are more convenient. In this method, the space is discretised into a grid of rectangular cells (or voxels) and each cell is marked as an obstacle or a non-obstacle. The speed of the algorithm is reversely related to the number of cells [6]. Here, one may decrease the dimensions of the search space by choosing larger cells in order to increase the speed of the path planner, which will also result in decreasing the precision of the representation. Hence, there is a trade off between the speed of decision making and the precision of the selected path. To overcome this problem, we have proposed a novel representation method [6] which will be discussed more thoroughly in section 2.

Many different algorithms can be used with the proposed representation method in order to solve the problem of path planning. Graph search methods [7], A* (A-star) algorithm [8], potential fields [9], fuzzy approaches [10] and genetic algorithm [6, 11] are some of these approaches. In this paper, we have developed a heuristic weighting approach – inspired by A* search algorithm – which is shown to be both robust and fast when used with the mentioned representation method. This Algorithm will be described in section 3.

A robust path planner must be able to handle the local minima problem. Virtual target [12], virtual obstacle [13], Distbug [14] and the method proposed by Huang and Lee [15] are some of the available approaches to handle this problem. Among them, the last two methods use the wall following algorithm which is shown to be highly effective. The representation method used in this paper allows us to detect and follow a desired wall in a relatively simple manner. The details are discussed in section 4.

In section 5 the results of simulations will be presented. Section 6 includes the discussions and conclusions.

2 Environment Representation

2.1 Basic Concepts

In [6] it was shown that in conventional *CS-map* method there is a direct relation between the number of cells and the precision of the representation. However, a large search space will result in slower performance of the path planner and thus, it can not be used for real-time applications. Therefore, in [6] a new method of representing the environment was introduced that reduces the dimension of the search space, while keeping the precision as high as possible.

In this paper, only two dimensional environments will be considered. Each environment will be treated as a binary 2D image in which the black parts represent the obstacles. An example is shown in figure 1.a. In the representation of the environment, two main constraints were integrated. First, the desired path must keep a minimum distance from each obstacle in order to provide safe passage for the simulated nonholonomic robot. To do this, we use the morphological operand "*dilation*" [16] on the described binary image. This results in expansion of obstacles with the desired amount of safety margin. Figure 1.b shows the result of performing this operation on figure 1.a. The grey areas are the expansions of the obstacles.

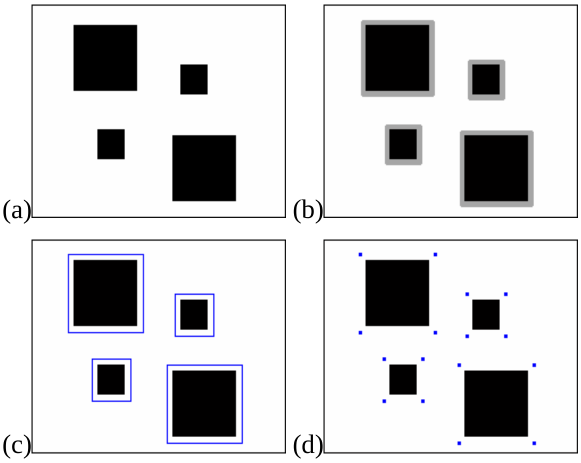


Figure 1: (a) typical 2D environment - black parts represent the obstacles; (b) result of performing dilation on figure 1a - grey parts are the expansions of the obstacles; (c) boundaries of the expanded obstacles; (d) corner points of these boundaries, extracted by analysing the slope variations.

The second constraint is the length of the path. The shortest path is always a series of lines whose ends are the corner points of the boundaries of the obstacles [6]. Thus, the second step will be to find the boundaries of the expanded obstacles. This can be easily done in a binary image (figure 1.c). Note that this boundary is the circumferential polygon of an obstacle and not necessarily its convex hull. By calculating the slope variations along these boundaries, the corner points will be found. Knowing which corner points are in the direct sight of each specific corner point, one can simply use these points to find the optimal path between any two locations.

2.2 Local Representation of the Surrounding Environment

As mentioned before, the main goal of this paper is to develop a local path planner for real-time applications. Usually, the environment is not completely known and our knowledge is limited to the surroundings of the current position within a certain range (a frame). Thus, in order to make any local decision, the algorithm must first model the surrounding environment.

In the environment shown in figure 2.a, local surroundings of a given point – marked with + – are enclosed within a circle of radius *R*. Thus, the modelled robot can only sense the parts which are shown in figure 2.b. After applying the two steps mentioned in section 2.1 on this image, points which are shown in figure 2.c will be resulted. These points will be used for local path planning in the current frame. Here, *R* is assumed to be 3 meters.

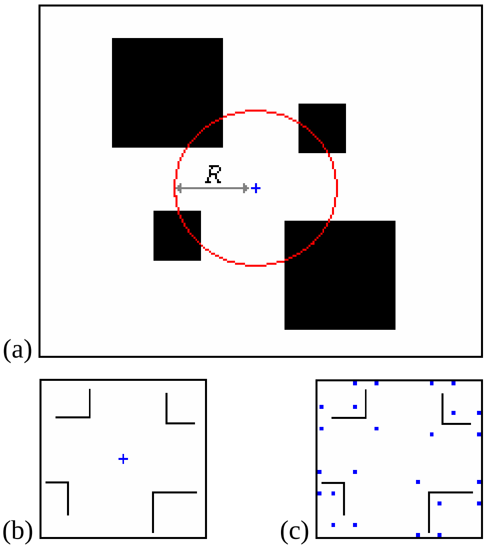


Figure 2: (a) the local surroundings of a certain point are enclosed within a circle of radius *R* = 3 metres; (b) the edges of the obstacles within the local frame which are in direct sight of the reference point; (c) representation of the local frame.

3 Path Planning and Motion Control

3.1 Local Path Planner

To start the path planning procedure, the start and goal configurations must be known. Here, the coordinates of goal are specified with respect to that of a reference point – centre of the frame – on the simulated nonholonomic robot. Once this is done, the path planner will start the navigation task.

To introduce the effect of the goal, we will find the point where the connecting line of the reference and the goal positions crosses over the boundary of the current local frame. This point (*D*) will be added to the group of previously found corner points.

All points – including the reference – are treated as nodes. Starting from the reference point, a cost will be calculated for every node *n* which is in the direct sight of the current node (*n-1*):

$$f(n) = [g(n) + h(n)][1 + \frac{c \times |\omega_o(n)|}{300}] \qquad (1)$$

Where *f(n)* is the calculated cost for node *n*, *g(n)* the Euclidian distance between nodes *n* and *n-1*, *h(n)* the Euclidian distance between nodes *n* and *D*, and $\omega_o(n)$ the deviation between the heading direction of the modelled robot and the direction of the node *n* with respect to the reference node (in degrees). Parameter *c* is assumed to be 1 if node *n* is in direct sight of the reference node, otherwise it is zero.

After calculating the cost of all possible nodes, the one with the lowest cost will be indexed as the current node and the above steps will be repeated for it. Once this node lies on the boundary of the local frame, this phase will be finished. The specified series of points will represent the optimal path in the current local frame.

3.2 Motion Control and Navigation

The developed path planning algorithm was originally meant to be used for navigation of an automatic wheelchair. Thus, the model is a nonholonomic robot with turning constraints. To control the motion of this model and in order to find the position of the new reference point, a potential field approach has been used.

First, the series found in section 3.1 will be searched for points which are in the direct sight of the reference point. The last point found with this condition will serve as an attractor. If the deviation of this attractor from the heading of the model is less than 15 degrees, the attractive force will be calculated as follow:

$$attraction = 7.5 \times \Delta\psi_a (e^{-0.4 \times d_a} + 0.4) \qquad (2)$$

Where, $\Delta\psi_a$ is the deviation between the heading of the model and the direction of the attractor with respect to the reference point (in degrees) and d_a is the Euclidian distance between the attractor and the reference position (in meters).

To calculate the repulsive force, the nearest obstacle point to the reference position is found in the representation phase. Using these coordinates for the repeller, the repulsive force will be:

$$repulsion = 5000 \times \Delta\psi_r e^{(-6.5 \times \frac{\pi}{180}|\Delta\psi_r| - 0.0016 \times d_r)} \qquad (3)$$

Where, $\Delta\psi_r$ is the deviation between the heading of the model and the direction of the repeller with respect to the reference point (in degrees), and d_r is the Euclidian distance between the repeller and the reference position (in meters).

Both equations (2) and (3) are biologically inspired force fields [17]. However, the original parameters' values were set so that the model can mimic human motion. Here, we have adjusted these values so that the generated paths are more similar to those generated by real wheelchair users. Using these two forces, a heading will be determined. Depending on the desired speed (1.5 m/s) and time step for each frame (0.1 s), the reference point will be moved along this direction (15 cm).

Since the model has turning constraints, the above steps will result in improper paths when the deviation of attractor from the heading of the model is more than 15 degrees. In this case, the model will be allowed to select an appropriate turning centre (turning radius is assumed to be 55 cm). Then, it will turn around this point with a step of 6° (angular speed is assumed to be 60 °/s). The reference point will be changed correspondently (figure 3).

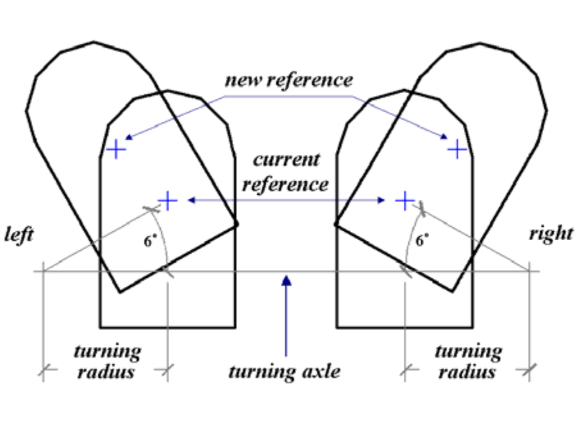


Figure 3: The turning procedure of the modelled nonholonomic robot.

Finally, if the distance between the new reference point and the goal is less than 10 cm, the process will be finished. Otherwise, a new frame centred at this point will be formed and a new step will begin.

4 Handling Local Minima Problem

There are times that the described algorithm in section 3 leads the model into local traps. In figure 4, one of such situations is shown. In such cases, the algorithm provides the model with an attractor. However, after moving toward this attractor for several steps, the algorithm offers a new attractor which is in the opposite direction of the previous one. The same thing happens after moving toward this attractor for several steps and finally, the model will end up in oscillating between these two attractors.

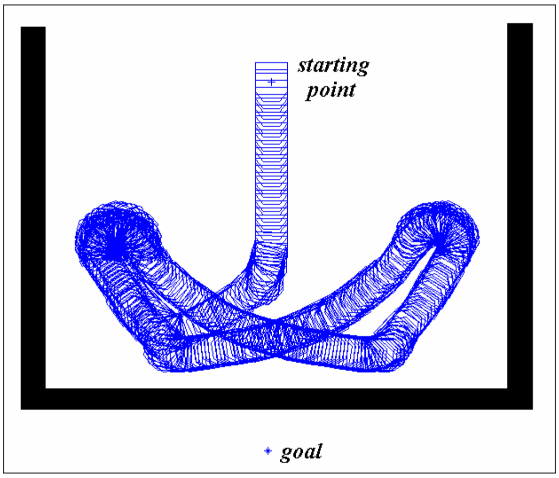


Figure 4: The performance of the algorithm when facing a local trap. Here, no escape strategy is included.

There are many approaches to solve this problem. Among them, the wall following algorithm is shown to be more effective [14, 15]. In addition, as it will be described shortly, this method is highly compatible with the representation method proposed in [6]. Here, the detection and leaving criterions of the wall following method will be defined only based on the direction of the attractor and the goal.

As it was mentioned earlier, a local trap occurs whenever the directions of new and old attractors are almost opposite. Here, a threshold of 140 degrees is chosen to be the sign of local minima. Upon detecting a local trap, the model will move toward the old attractor. Then, the obstacle between the reference point and the goal will be detected and all other obstacles will be ignored. Thus, the representation result will only include the obstacle responsible for the local minima condition. Here, the point D (section 3.1) will not be included in the search space. In addition, equation (1) will be changed as follow:

$$f(n) = [1 + \frac{c \times |\omega_o(n)|}{300}] \tag{4}$$

By doing so, the effect of the goal will be omitted and the model is forced to move along the boundary of the trapping obstacle. The weight of direction in equation (4) guarantees that there won’t be any rollbacks.

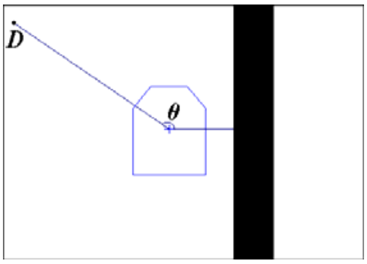


Figure 5: While in the wall following mode, the angle ***θ*** is used by the algorithm to find the appropriate leaving position.

Although point *D* is not included in the search space, it will be determined in every step. This point and the detected repeller along with the reference point will form an angle as shown in figure 5 (angle ***θ***).

The algorithm terminates the wall following mode whenever ***θ*** exceeds 145 degrees. At this point, everything will be set to its original state and normal path planning procedure will be used.

5 Simulation Results

All parts of the described algorithm are implemented in MATLAB 7.0 on a 3 GHz, P4 PC with 512 MB onboard RAM.

An 80 cm × 110 cm nonholonomic robot is simulated. The safety margin of the obstacles is assumed to be 60 cm. The normal speed and the time step are set to 1.5 m/s and 0.1 s, respectively.

Figures 6 to 10 show the generated trajectories for five different situations. As can be seen, all trajectories are smooth and collision-free. In addition, the wall following module has enabled the model to escape from the local trap of figure 8 and to navigate through the maze-like environments of figures 9 and 10.

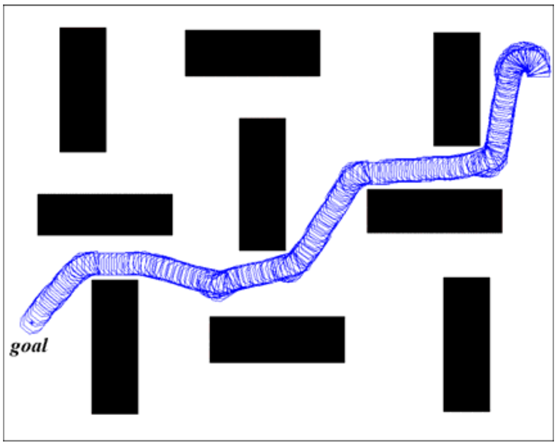


Figure 6: The generated path in a normal environment with rectangular obstacles.

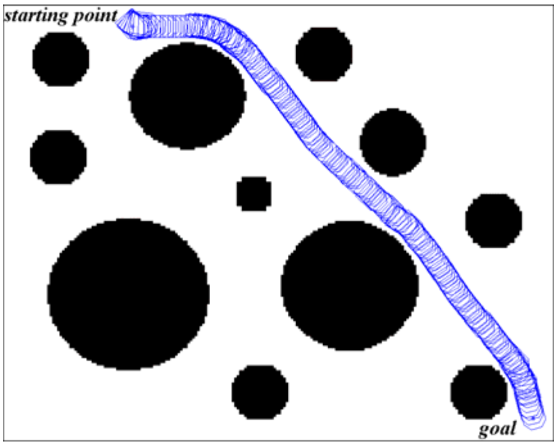


Figure 7: The generated path in a normal environment with circular obstacles.

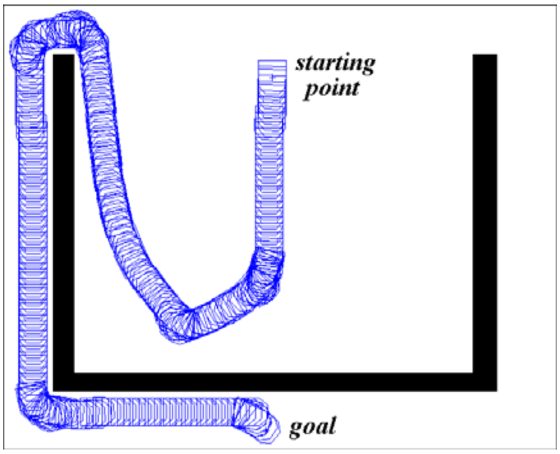


Figure 8: Providing an escape strategy, the algorithm has successfully handled the local minima of figure 4. Note that walls have no curvature.

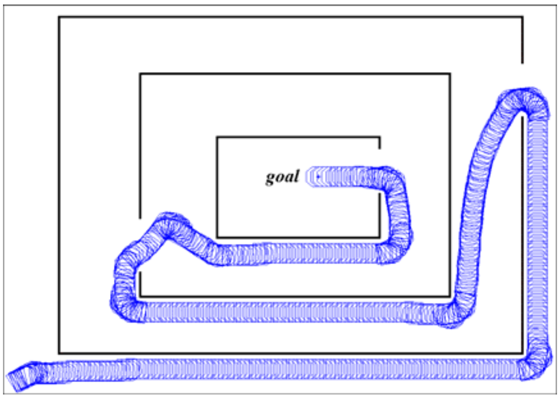


Figure 9: The generated path in a maze-like environment. Walls have no curvature.

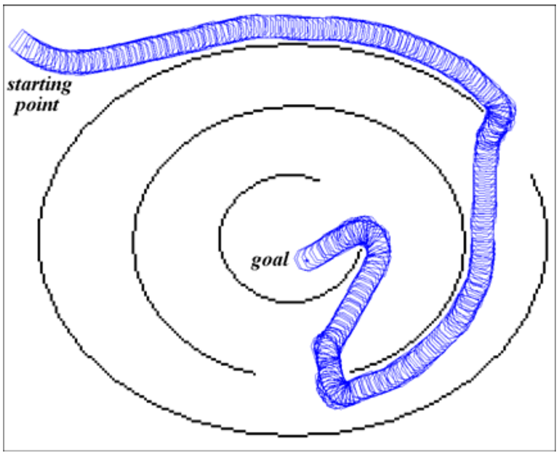


Figure 10: The generated path in a maze-like environment. Walls have a significant curvature.

To determine the performance speed of the algorithm in each step, we have used the profile function of MATLAB. The largest time required for simulation of one step was 52 ms which was observed near obstacles with curved boundaries (as in figures 7 and 10). It is also worthy to note that on average, 38 ms of this time was used for representation of the local frame.

6 Discussion

Representation of the environment with the corner points of expanded obstacles' boundaries is the basic idea of section 2. Comparing to other common representation methods, the resulting search space of the proposed approach is considerably smaller and yet, it can precisely determine the shape and position of the surrounding obstacles for the path planning algorithm. All one has to do is to provide a list of directly accessible corner points for every corner point which is encountered during the search for a path.

The heuristical path planner of section 3.1 and the motion controller of section 3.2 are responsible for navigation of the model in the represented frame. As it can be seen in section 5, the use of directional weighting and the bio-inspired potential fields have lead to smooth and safe trajectories which can be traversed easily by a nonholonomic robot.

To avoid local minima and dead-ends, we have used the wall following approach. As it is explained in section 4, this method can be adapted easily by the proposed path planning and representation algorithms. Omission of insignificant obstacles and simplification of the cost function – equation (1) – are the only required adjustments in this case. Once a local minima is detected and these changes are applied, all extra required effort is controlling of the leaving criterion at the end of each step. After eliminating the threat, only the normal cost function must be restored and again, the representation must include every obstacle within the range of sight.

Some typical simulation results have been presented in section 5 along with a brief statistic on the execution time of each step. The only effective parameter in the speed of the proposed algorithm is the size of the search space. Obviously, any factor that increases the number of search space members will decrease the speed of this algorithm. A curved obstacle is such a factor since it has many corner points in its boundary. This is why we had the largest step simulation time in environments similar to those in figure 7 or 10. However, this effect is not highly significant since the largest step simulation time for a simple environment like the one in figure 6 is about 46 ms. Therefore, the system may respond to any stimulant within about 50 ms. This delay can be reduced even more if we use a parallel structure instead of the serial one which is used in current simulations. Thus, the real-time performance of the algorithm will be ensured.

The presented path planner is shown to be robust for navigating unknown static environments. It can also handle sudden changes in the static configuration of the model's surroundings. We are currently trying to add dynamic obstacle avoidance modules to this scheme. Another deficiency of our approach is the lack of the ability to judge on memory. There were times when the model was trapped in a global minima and visited certain points for several times. We believe that this problem can be solved by memorising the observed parts of the environment. Once a global trap is detected, a global path planner can be used along with this memory to find an escape strategy. The representation process will be applied to the whole available memory of the current environment. Depending on the size of the memory, the size of the search space may vary which will most probably increase the step execution time. Since this task can be done once per global trap, such a drawback should not be too troublesome.

7 References

[1] B. Cho, S. Byun, C. Shin, J. Yang and J. Oh, “KeproVt: underwater robotic system for visual inspection of nuclear reactor internals”, *Nuclear Engineering and Design*, 231(3), pp 327-335 (2004).

[2] T. Rofer and A. Lankenau, “Architecture and applications of the Bremen autonomous wheelchair”, *Information Sciences*, 126(1/4), pp 1-20 (2000).

[3] P.J. McKerrow, *Introduction to Robotics*, Addison Wesley, New York (1991).

[4] A. Yahja, S. Singh and A. Stentz, “An efficient on-line path planner for outdoor mobile robots”, *Robotics and Autonomous Systems*, 32(2/3), pp 129-143 (2000).

[5] S. Bandi and D. Talmann, “Space discretization for efficient human navigation”, *Computer Graphic Forums*, 17(3), pp 195-206 (1998).

[6] H. Mahjoubi, F. Bahrami and C. Lucas, “Path planning in an environment with static and dynamic obstacles using genetic algorithm: a simplified search space approach”, *Proceedings of IEEE World Congress on Computational Intelligence (WCCI)*, Vancouver, Canada, pp 8652-8658 (2006).

[7] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, New Jersey (1995).

[8] S. Koening and M. Likhachev, “Fast replanning for navigation in unknown terrain,” *IEEE Transactions on Robotics*, 21(3), pp 354-363 (2005).

[9] J. Borenstein and Y. Koren, “The vector field histogram - fast obstacle avoidance for mobile robots”, *IEEE Journal of Robotics and Automation*, 7(3), pp 278-288 (1991).

[10] H. Maaref and C. Barret, “Sensor-based navigation of a mobile robot in an indoor environment”, *Robotics and Autonomous Systems*, 38(1), pp 1-18 (2002).

[11] L. Tian and C. Collins, “An effective robot trajectory planning method using genetic algorithm”, *Mechatronics*, 14, pp 455-470 (2004).

[12] W.L. Xu, “A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behavior-based mobile robot”, *Robotics and Autonomous Systems*, 30(4), pp 315-324 (2000).

[13] F.G. Pin and S.R. Bender, “Adding memory processing behavior to the fuzzy behaviorist approach: resolving limit cycle problems in mobile robot navigation”, *Intelligent Automation and Soft Computing*, 5(1), pp 31-41 (1999).

[14] I. Kamon and E. Rivlin, “Sensory-based motion planning with global proofs”, *IEEE Transactions on Robotics and Automation*, 13(6), pp 814-822 (1997).

[15] H.P. Huang and P.C. Lee, “A real-time algorithm for obstacle avoidance of autonomous mobile robots”, *Robotica*, 10, pp 217-227 (1992).

[16] R.C. Gonzalez and R.E. Woods, *Digital Image Processing* (2nd edition), Prentice Hall, New Jersey (2002).

[17] B.R. Fajen and W.H. Warren, “A dynamical model of visually guided steering, obstacle avoidance and route selection”, *International Journal of Computer Vision*, 54(1/2), pp 13-34 (2003).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Generation of Multiple Cubic Spiral Paths for Obstacle Avoidance of a Car-like Mobile Robot using Evolutionary Search

Kao-Ting Hung[2], Jing-Sin Liu[1], Yau-Zen Chang[2]
[1]Institute of Information Sciences, Academia Sinica, Nangkang, Taipei, Taiwan, ROC
[2]Department of Mechanical Engineering, Chang Gung University, Tao-Yuan, Taiwan, ROC
kattin@st.cgu.edu.tw, liu@iis.sinica.edu.tw, zen@mail.cgu.edu.tw

Abstract

Cubic spiral is a curve whose direction is a cubic polynomial of arc length. It has been used as a path primitive for nonholonomic car-like mobile robot with restricted turning capability in free environment [1], [2]. In this paper, we investigate the use of cubic spiral segments for car-like mobile robots moving from a given start configuration to a given goal configuration in an environment cluttered with known static obstacles. We propose a parallel genetic algorithm (PGA) based on the island model to generate multiple collision-free paths in parallel via sets of a predefine number of intermediate configurations connected by collision-free cubic spirals. The architecture of PGA is simple: it consists of multiple islands, where subpopulation of each island runs a simple genetic algorithm independently with the same fixed-length real-coded chromosome representation of cubic spiral path segments, basic genetic-operators, one subgoals manipulation operator, and maintains diversity of subpopulations by synchronous migration among the islands and a big bang activity. The generation of multiple collision-free paths by PGA is via the following strategy: (i) the generated collision-free paths are similar to an already found collision-free path, then they are eliminated and replaced by randomly generated paths for later evolution, (ii) otherwise, they are extracted as new alternative collision-free paths. Simulation results of mobile robot for a set of different start orientations show the effectiveness of this evolutionary approach to evolve a more rich set of collision-free paths.

Keywords: obstacle avoidance, smooth path planning, car-like mobile robot, parallel genetic-algorithm (PGA)

1 Introduction

Path planning of mobile robots with different designs has been extensively and intensively studied in recent years. A lot of methods, for example: artificial potential field method, probabilistic roadmap method, either global or local, offline or online path planning in complex environments based on a variety of path primitives have been proposed [11]. In general, a path is made up of a set of local curve pieces. Path planning problem can be formulated as an optimisation problem subject to motion constraints, which constrain the differential geometry of the set of feasible paths [17]. In particular, genetic algorithm (GA) based path planning for a mobile robot [3]-[7], [14], [15] has attracted more attentions, mainly for shortest piecewise-linear path in grid environments with obstacles modelled as fully occupied connected grids. The advantage of GA based path planner lies in the parallel problem solving capability and the minimum requirements regarding the problem formulation. However, implicitly assumed in this kind of problem formulation is the mobile robot can instantaneously turn from any orientation to arbitrary orientation, without any constraint of turning radius (i.e. no consideration of nonholonomy of robot motion). In addition, piece-wise linear paths does not allow smooth tracking by mobile robots, and thus for practical use, need smoothing or refinement [17]. Besides, most existing evolutionary path planners contain many special design (problem-specific knowledge based) operators to “repair” a longer or colliding line segment. The solution techniques are often engineered for a specific context. These complicated the overall design of GA-based path planner and reduce the flexibility of GA.

For smooth path, not only the smooth motion along a local curve pieces but also the smooth transition between the path segments (i.e. orientation of current end configuration is the orientation are guaranteed for mobile robot. B-spline is a good candidate for smooth path due to its local and global control property and ease of computation of points and derivative on it [16]. In this paper, cubic spiral is chosen as the primitive for generating smooth path of better curvature control (than the more well-known B-spline). Cubic spiral has been employed to mobile robot path generation for the first time by [1]. Its two drawbacks: (1) overly long path for close locations with large changes in orientation, or very distant goals, (2) lengthy paths with no use of the reverse motion capability are remedied by [2]. However, the obstacle avoidance capability of the cubic spiral path planning methods of [1], [2] is very limited. In this paper, we aim to enhance the obstacle avoidance capability of car-like mobile robot in static environment using cubic spiral as path primitive. To this aim, the parallel genetic algorithm (PGA) based on the island model, embedded with specially designed problem-specific operations, is designed to generate multiple paths composed of collision-free cubic spiral path segments linking from a given start configuration to a given goal configuration through a predefined number (depending on the complexity of the environment and task) of intermediate configurations found by evolutionary search without cross the static obstacles or walls. Multiple paths, which allow more path options for improving the success of robot path planning, are generated. Our evolutionary method is simple and general to be potentially applied to dynamic environment.

The remainder of this paper is organised as follows. Cubic spiral path planning method [1] is briefly reviewed. Then our evolutionary smooth path generation method based on multi-island PGA would be detailed in section 3. Some comparisons to show the effectiveness of specially designed operators and experiments to demonstrate the performance of multiple paths generation are conducted in static rectangular environments cluttered with different polygonal obstacles configuration in section 4. Finally, we make conclusions and describe the future works in section 5.

2 Preliminaries

This section contains two parts. One part summarises from [1] some materials related to cubic spiral and its planning method. The other part briefly introduces the genetic algorithms related to this work.

2.1 Review of Cubic Spiral and its Planning Method

By definition, a cubic spiral is a set of curves that the direction function θ is a cubic polynomial of arc length. For an arbitrary configuration q, $[q]$ denotes its position (x, y), and (q) its direction θ. The definition of *symmetric* was described in detail in [1].

A *symmetric mean q* of any configuration pair (q_1, q_2) is a configuration that (q_1, q) and (q, q_2) are both symmetric pairs. All symmetric means of a configuration pair (q_1, q_2) forms a circle if $(q_1) \neq (q_2)$ or a line segment connecting q_1 and q_2 if $(q_1) = (q_2)$ (i.e. parallel configurations) (proposition 3, [1]). The symmetric mean of two given configurations q_1 and q_2 can be defined via the position ratio γ (figure 1):

Non-parallel case: $(q_1) \neq (q_2)$

Define the centre of the circle that go through the given configurations q_1 and q_2.

$$p_c = (x_c, y_c) = \left(\frac{x_1 + x_2 + c(y_1 - y_2)}{2}, \frac{y_1 + y_2 + c(x_2 - x_1)}{2} \right)$$

where $q_1 = (x_1, y_1, \theta_1)$, $q_2 = (x_2, y_2, \theta_2)$, and $c = \cot((\theta_2 - \theta_1)/2)$. Thus the position of symmetric mean $[q_s]$ can be defined as:

$$[q_s] = (x_c + r \cdot \cos(\beta_1 + (\beta_2 - \beta_1) \cdot \gamma), y_c + r \cdot \sin(\beta_1 + (\beta_2 - \beta_1) \cdot \gamma))$$

where β_1 and β_2 represent the orientations from p_c to q_1 and q_2 respectively. The orientation of the symmetric mean can be defined according to the position of symmetric mean [1].

Parallel case: $(q_1) = (q_2) = \theta$

$$q_s = (x_1 + (x_2 - x)_1 \cdot \gamma, y_1 + (y_2 - y)_1 \cdot \gamma, \beta - (\theta - \beta))$$

where $[q_1] = (x_1, y_1)$, $[q_2] = (x_2, y_2)$, and

$$\beta = \tan^{-1}\left({y_2 - y_1} \middle/ {x_2 - x} \right)$$

It is noted that a cubic spiral can connect two symmetric configurations. The local path planning method using cubic spiral can connect two given configuration q_1 and q_2: If q_1 and q_2 are symmetric, connect these two configurations with a cubic spiral directly. Else, connect these two configurations through a selected symmetric mean using two cubic spirals.

For a configuration pair (q_1, q_2), the size d is the distance between the two points $[q_1]$ and $[q_2]$, and the deflection angle α the angle between the two orientations (q_1) and (q_2). Let $q \equiv (x, y, \theta)$ represent the configuration of a car-like mobile robot (modelled as a unicycle) where (x , y) is the Cartesian position and θ is the orientation. The $q(s) = (x(s), y(s), \theta(s))$ at arc length s of a car-like mobile robot is defined by the integral equations describing the path starting from the initial configuration (x_0, y_0, θ_0)

$$\begin{cases} \theta(s) = \theta_0 + \int_0^s \kappa(t) dt \\ x(s) = x_0 + \int_0^s \cos(\theta(t)) dt \\ y(s) = y_0 + \int_0^s \sin(\theta(t)) dt \end{cases} \tag{1}$$

where s is defined as 0 at the initial point (x_0, y_0).

2.2 Multi-island PGA

For simple genetic algorithm (SGA), the conflict between speedup of convergence rate and avoidance of local minimum is embodied in the selection pressure and population diversity. PGA based on island model is a parallelisation scheme of genetic algorithms that can potentially reduce the execution time. This scheme divides the population into several communicating subpopulations each evolving via SGA in an island with a common pool serving as a migration centre (figure 2), created by random sampling from each island. A fraction of subpopulations (called migration rate) of individuals of each island is randomly selected to send to the common pool, and gathered. Then the common pool redistributes the individuals randomly onto the different islands. The size of the common pool equals the migration size (the number of individuals that migrate) of each island times the number of islands.

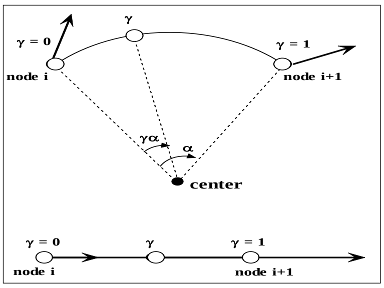


Figure 1: Symmetric mean represented by a position ratio γ. (a) non-parallel case: the symmetric mean is a circle; (b) parallel case: the symmetric mean is a line segment.

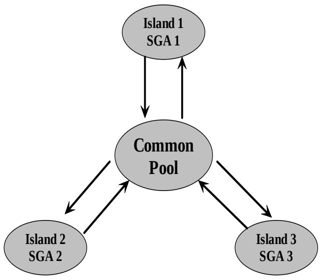


Figure 2: Migration policy of PGA based on island model.

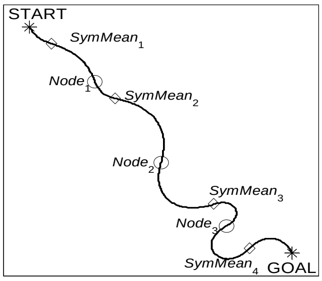


Figure 3: Path representation.

Migration introduces diversity into a population, and the selection pressure increases monotonically with higher migration rates. It is possible to raise selection pressure in subpopulations of some islands, and concurrently to augment the population diversity in subpopulations of other islands by migration scheme.

3 Evolutionary Path Planning Method

For path planning of mobile robot using evolutionary search, in this paper two kinds of operators are applied to the path planning problem during evolution: one is the standard genetic operators; the other is a local path refinement operators which is highly specific to infeasible path segments (subgoals manipulation operator). Furthermore, to extract a more rich set of feasible paths from generation to generation, a filtering mechanism and partial reinitialisation (called “big bang”) is also devised.

3.1 Attributes of Evolutionary Process

Individual/path representation: For path planning, it is most natural to use real-coded chromosome, which directly represents the path as a vector of real parameters to avoid decoding operation [8]. In this paper, a path is composed of cubic spiral segments connecting the given initial and goal configurations through a prespecified number of ordered intermediate points. Let the number of intermediate points of a path be *N*, then *N*+1 pairs of configurations need to be connected by cubic spirals, or in total *N*+1 symmetric means are to be determined for these segments. As the symmetric mean is located on a circle or a line connected by two given configurations, it can be represented by a position ratio, as depicted in figure 1. Additionally, the chromosome consists of an additional gene representing the position ratio (i.e. symmetric mean) between the last intermediate subgoal and Goal configuration. Therefore, a path is encoded as a chromosome of 4N+1 genes by a sequence of ordered cubic spirals. Concatenating the representation of each subpath, a path is represented by:

$$(\gamma_1 x_1 y_1 \alpha_1, \gamma_2 x_2 y_2 \alpha_2, \cdots, \gamma_N x_N y_N \alpha_N, \gamma_{N+1})$$

Figure 3 illustrates the cubic spiral path representation. Each path is composed of a set of adjacent subpaths, which are two cubic spirals connecting two configurations through their respective symmetric mean.

Fitness: In this paper, we consider 2D, static and known rectangular environments with (convex or concave) polygonal and wall-like obstacles. The intrinsic cost of a candidate path is the total number of intersected sampled points of a path with all existing obstacles, or how many sampling nodes of path outside the boundary of map. Since the path of the car-like mobile robot is governed by the equations of motion (1), we could compute the intrinsic cost of each path according to [12]. The cost of the optimum path would be zero since all path segments need to be collision-free. The path planning problem is to search for solutions that result in a minimum intrinsic cost. Actually, a rank-based fitness assignment strategy is adopted in this research, thus the problem is then formulated as searching for chromosomes/paths to maximise the fitness.

Genetic operators: The genetic operators (includes crossover, mutation, and selection) modifies the population of candidate paths to create a new (and hopefully more feasible) population of paths. Real coded GA requires a special design of crossover and mutation operators to work with real parameters to ensure that the parameters stay within the specified upper and lower limits. The detail can be referred in [9, 10]. Please note that we implement the well-known roulette wheel selection.

Subgoals manipulation operator: For path planning problem, it is important to consider how to elaborate an infeasible path into a more acceptable path. To this aim, we further design a local path refinement operator, subgoals manipulation operator, operating on the infeasible paths segments that cross the obstacles in order to accelerate the evolution to find out the collision-free paths. The operator locally manipulates those nodes in a predefined neighbourhood, so that this local refinement of path shape occurs in the sun-regions. Here the manipulation is primarily based on the mutation operator. Figure 4 shows the diagram of this operator. The second path segment of original path (solid path) crosses an obstacle, and the operator randomly shifts the configuration of Node 2 to a new neighbouring location to obtain new path (dashed path), very possibly becoming collision-free.

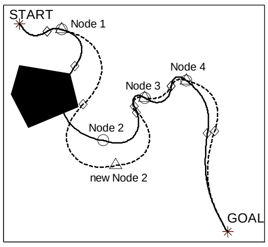


Figure 4: The subgoals manipulation operator.

Filtering and big bang activity: Despite of the introduction of the subgoals manipulation operator, another difference with the traditional PGA is filtering and big bang activity. Since GA needs to retain better solutions to reproduce the new population, it might lead GA to search the same region repetitively. To avoid the situation that the population contains mostly similar paths, our PGA incorporates a filtering mechanism for a continual extraction of distinct feasible paths from the evolutionary search process: For each generation, the feasible paths are chosen from the population and they are compared with the contents of the external file (the file is initially empty). Those that are not similar to the paths of the external file are picked up to the external file without further evolution. Furthermore to retain the already found feasible paths and create new feasible paths in the current and later evolutions, those that are similar to the paths within the external file are eliminated, and an equal number of randomly generated chromosomes are put into the population to form a new population for further evolution. This strategy of restart search, called big bang activity, avoids a regeneration of similar feasible paths that have already been generated, thus accelerating the search process and increasing the diversity of paths generated.

Given two generated paths with the same start and goal, a similarity measure of two paths/chromosomes is introduced as follows to recognise similar paths. Given two paths linking the same start and goal configurations, and N intermediate configurations, let $({}^m x_1, {}^m y_1),\ldots,({}^m x_N, {}^m y_N)$ denote the positions of all subgoals, ${}^m\alpha_1,\ldots,{}^m\alpha_N$ the variation of orientations, and ${}^m\gamma_1,\ldots,{}^m\gamma_N$ the position ratios in the m-th path (or chromosome). Comparison of two paths is made according to their position, deflection angle, and position ratio of all control nodes that defined as follows:

$$\begin{cases} NormP_i = \left\| \left({}^m x_i, {}^m y_i\right) - \left({}^n x_i, {}^n y_i\right) \right\| \big/ MapDist \\ \quad NormA_i = \left| {}^m\alpha_i - {}^n\alpha_i \right| \big/ \pi & , i = 1 \sim N \\ \quad NormR_i = \left| {}^m\gamma_i - {}^n\gamma_i \right| \big/ 0.6 \end{cases}$$

where *MapDist* equals the diagonal distance of the map, *m* and *n* are the indices of two paths/chromosomes in a specific (sub)population. The four shape variables of a cubic spiral path segment are normalised by their respective ranges: *MapDist*, π, and 0.6 are allowable ranges of the path variables during evolution. If the maximum of these normalised terms is larger than a predefined threshold (0.1 in all the simulations of the paper), we say that the two paths are distinct (dissimilar).

3.2 Evolutionary Path Planner Based on Multi-island PGA

The initial subpopulations for the islands are created by randomly generated *x*-monotone positions, deflection angles within a range of –π/2 to π/2, and position ratios of symmetric means in the range of 0.2 to 0.8, of all intermediate points within the predefined ranges. The subpopulations of each island have constant population size (100 individuals per generation in this paper).

4 Experimental Results

In this section, simulations are conducted for investigating the effectiveness of the specially designed operations for 50 independent runs. Performance of multiple paths generation of our evolutionary planner is shown for rectangular environment that consists of six polygonal obstacles with different initial configurations.

4.1 Effectiveness of Subgoals Manipulation Operator and Big Bang

A. Effectiveness of subgoals manipulation operator for locally repair the infeasible segments

For validating the necessity of subgoals manipulation operator, we perform a series of repeated runs of two SGAs, one without (SGA-1) and the other with (SGA-2) the subgoals manipulation operator for validating its effectiveness. The parameters setting for this comparison can be found in Table 1. Figure 5 shows the following: (i) In 50 runs, SGA-1 has a success rate of 50% and SGA-2 has a success rate of 68% that can find a collision-free path within 200 generations. It is clear that the subgoals manipulation operator helps raising the probability of finding a feasible path more efficiently. (ii) Among 50 runs, SGA-1 and SGA-2 have 30% and 48%, respectively, that can find a collision-free path within the first 40 generations (below the horizontal dotted line in figure 5). In both cases, SGA is more likely to premature in early generations, i.e. so-called genetic drift, where there are largely unexplored or under-explored regions of search space, indicating that SGA rapidly converged in an area of search space.

Table 1: Parameters setting for comparison of effectiveness of subgoals manipulation operator.

	SGA-1	SGA-2
No. intermediate points	5	
Population size	100	
Evolutionary generations	200	
Crossover rate, mutation rate	0.71, 0.01	
No. manipulations	none	20/generation

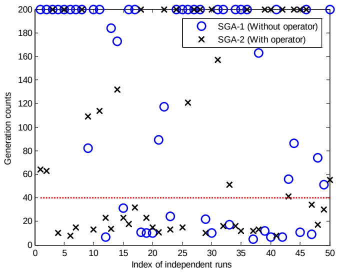


Figure 5: Generation that first feasible path is found, recorded for 50 runs of SGA with or without the subgoals manipulation operator.

B. Effectiveness of big bang

Figure 6 shows that the number of distinct collision-free paths found by the 3-island PGA (defined as Table 2) with big bang activity is mostly more than or at least equal to that without big bang activity in each of 50 independent runs with the same initialisation in both schemes.

Table 2: Parameter settings for 3-island PGA.

Crossover rate, mutation rate	(0.71, 0.01) (0.1, 0.9) (0.4, 0.3)
No. migrations	6/island
No. manipulations	20/generation

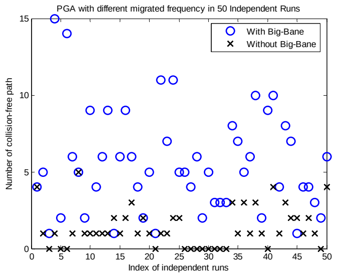


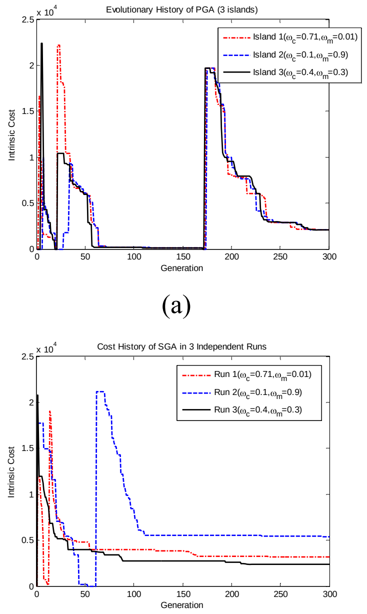
Figure 6: Number of distinct feasible paths of 3-island PGA with or without big bang activity in 50 independent runs.

4.2 Comparison of the PGA and SGA

Both the subgoals manipulation operator and the big bang activity are implemented in this comparison. Table3 shows the parameters setting for three-island PGA and three separate SGAs. Each island/SGA has a different crossover and mutation pair of rates in this comparison. Figure 7 illustrates the evolution of the best cost of three subpopulations of PGA and 3 independent runs of SGA in each island, whose performance strongly depends on the crossover and mutation rates. For PGA, solutions are obtained for each simulation, while no collision-free path is found for SGA within the maximum number of generations in the worst case. PGA based on island model shows significant improvements to finding more feasible paths.

Table 3: Parameter setting for comparison of PGA and three SGAs.

	3-island PGA	3 independent SGAs
Population size	100/ island	300/run
No. generations	300	
No. migrations	6/ island	none
No. manipulations	20s/generations	60
Big bang	Yes	



(a)

(b)

Figure 7: Evolution of intrinsic cost: (a) PGA (b) 3 separate SGAs.

4.3 Generation of Multiple Paths using PGA

Figure 8 shows the searching results with varied start orientations that would cover the relative different situations between the start and goal. In all three cases performed, PGA shows its adaptation to successfully generate at least 5 collision-free paths. It is likely that different runs of PGA may generate different number of possibly different feasible paths. This is due to the experience acquired during the exploration phase of evolution can affect the later convergence/exploitation outcome.

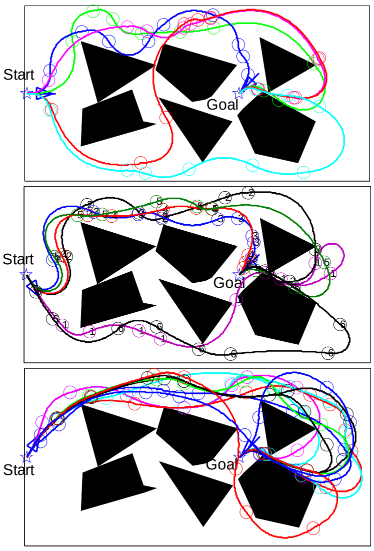


Figure 8: Multiple-path planning (4 intermediate nodes) for environment-I as start orientation is varied.

5 Conclusion and Future Work

In this paper, a PGA based on island model is applied to explore potential feasible paths for a car-like mobile robot. Our approach is effective to generate multiple alternative ordered sequences of a pre-specified number of intermediate collision-free configurations that makes a car-like mobile robot go through an obstructed environment with a directed path. Actually, subgoals manipulated operator, filtering and big-bang activity, as justified by simulations, effectively improve the evolutionary search capability.

For the incorporation of more criteria of paths, like curvature constraint and length minimisation [2], can be investigated in the future, actually, the multi-objective path optimisation problem is our work in progress. Furthermore, it's also a challenging task to plan a smooth path in the dynamic environments.

6 Acknowledgements

This research was supported by National Science Council, ROC, under contract no. 94-2212-E-001-001.

7 References

[1] Y.J. Kanayama and B.I. Hartman, "Smooth local path planning for autonomous vehicles", *Int. J. Robot. Res.*, 16(3), pp 263-283 (1997).

[2] T.C. Liang, J.S. Liu, G.T. Hung and Y.Z. Chang, "Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral", *Robotics and Autonomous System*, 52, pp 312-335 (2005).

[3] C. Hocaoglu and A.C. Sanderson, "Planning multiple paths with evolutionary speciation", *IEEE Transactions on Evolutionary Computation*, 5(3), pp 169-191 (2001).

[4] P. Wide and H. Schellwat, "Implementation of a genetic algorithm for routing an autonomous robot", *Robotica*, 15, pp 207-211 (1997).

[5] M. Gemeinder and M. Gerke, "GA-based path planning for mobile robot systems employing an active search algorithm", *Applied Soft Computing*, 3, pp 149-158 (2003).

[6] J. Xiao, Z. Michalewicz, L. Zhang and K. Trojanowski, "Adaptive evolutionary planner/navigator for mobile robots", *IEEE Transactions on Evolutionary Computation*, 1(1), pp 18-28 (1997).

[7] A. C. Nearchou, "Path planning of a mobile robot using genetic heuristics", *Robotica*, 16, pp 575-588 (1998).

[8] A.H. Wright, "Genetic algorithms for real parameter optimization", in G. Rawlins (ed.), *Fundamentals of Genetic Algorithms*, Morgan Kaufmann, pp 205-218 (1991).

[9] K.A. De Jong, *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, PhD Thesis, University of Michigan, Ann Arbor, MI (1975).

[10] H.K. Lam, F.H. Leung and P.K.S. Tam, "Design and stability analysis of fuzzy model-based nonlinear controller for nonlinear systems using genetic algorithm", *IEEE Trans. System, Man and Cybernetics*, Part B, 33, pp 250-257 (2003).

[11] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publisher, New York (1991).

[12] K. Konolige, "A gradient method for real-time robot control", *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Takamatsu, Japan, pp 639-646 (2000).

[13] Z. Skolicki and K. De Jong, "The influence of migration sizes and intervals on island models", *Genetic and Evolutionary Computation Conference 2005 (GECCO'05)*, Washington, DC, pp 1295-1302 (2005).

[14] Y. Hu and S.X. Yang, "Knowledge based genetic algorithm for path planning of a mobile robot", *Proc. IEEE Int. Conf. Robotics and Automation (ICRA'04)*, New Orleans, pp 4350-4355 (2004).

[15] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of an autonomous mobile robot", *IEEE Int. Conf. Computational Intelligence in Robotics and Automation (CIRA'97)*, California, pp 138-141 (1997).

[16] P.N. Azariadis and N.K. Aspragathos, "Obstacle representation by bump-surfaces for optimal motion-planning", *Robotics and Autonomous Systems,* 51, pp 129-150 (2005)

[17] G. Dudek and S. Simhon, "Path planning using learned constraints and preferences", *IEEE Int. Conf. Robotics and Automation (ICRA'03)*, Taipei, pp 2907-2913 (2003)

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Evolving Crowd Motion Simulation with Genetic Algorithm

Chih-Chien Wang, Tsai-Yen Li
Computer Science Department
National Chengchi University, Taipei, Taiwan
g9022@cs.nccu.edu.tw, li@nccu.edu.tw

Abstract

The problem of generating realistic crowd motion for a robot fleet in robotics or a virtual crowd in computer animation has attracted much attention in recent years. Previous work has succeeded in using virtual forces to simulate the motions of virtual creatures, such as birds or fish, in a crowd. However, the way in which to set up the virtual forces in order to generate the motions desired still remains a problem to be solved on a case-by-case basis. In this paper, we propose to use a genetic algorithm to generate an optimal set of weighting parameters for composing virtual forces according to the given environment and desired behaviour in movement. We have designed a list of measurement functions for the comparison of the fitness function used in the genetic algorithm. Experiments in simulation were conducted for several environments and different types of behaviours. The results show that compelling examples can be generated with the parameters found automatically in this approach.

Keywords: crowd motion simulation, genetic algorithm, robot formation, multiple robot system

1 Introduction

Formation control for multi-robot systems is a classical robotic problem that has attracted much attention in the literature [1]. In recent years, due to the changes in the entertainment industry, especially in computer animation, the techniques of simulating the behaviours of virtual crowd also has created potential applications in contexts such as a virtual mall, digital entertainment, etc. The techniques have also been used in scientific studies such as simulating crowd movement for transportation or urban planning.

According to the contexts of the application, the motion of a crowd can be structured in many ways with various degrees of organisation. For example, in most robotic applications, the goal of formation control is usually to form a specific shape by means of referencing methods [13]. However, it is more common in computer animation applications to ask for a less structured crowd movement so long as it looks realistic according to our everyday experience. For example, one may desire to simulate a moving crowd as a cohesive group with a consistent orientation or a crowd following their leader/goal as closely as possible. We call these desirable visual effects the *movement behaviours* of a crowd. In addition, the types of agents involved in the simulation may include a flock of birds, a school of fish, or a crowd of people. Among these agents, simulating a human crowd is the most challenging because of the complex behaviour that humans possess. We will call these agents *robots* in the rest of this paper.

Most of the current systems adopt a local control approach. One of the common approaches to this problem in computer animation adopts the virtual force model [13], whereby the movement of each robot is affected by virtual forces computed according to its spatial relation between itself and other robots or with objects in the environment. However, each designer must still face the problem of how to choose the most appropriate forces and select the best weights. In addition, there is no objective way to evaluate the result.

In this paper, we propose to model the problem of generating good examples of the behaviour of the movement of crowds by designing an appropriate parameterisation and evaluation model and we adopt the genetic algorithm [6,8,10] to search for an optimal set of parameters. The parameterised virtual-force model is then evaluated by an array of measures to describe the desired crowd behaviour. We have conducted experiments to generate parameter sets for several common crowd behaviours in various environments of different spatial structures. We believe that the use of this system will be able to result in the automation of the time-consuming tuning of the parameters required in the creation of the simulation of a behaviour desired within a given environment.

2 Related Work

Crowd motions can be generated by simulation or by planning. Due to the complexity involved in such a system, simulation with distributed local computation is the most common approach. However, some work plans the motion for a large crowd or uses a hybrid approach in which only the motions of a few key robots (called leaders) are planned. We will first review the related work in simulation and then in planning.

2.1 Crowd Motion Simulation

The approaches to simulate crowd motion can be classified into three categories each at different levels of abstraction: *particle system*, *flocking system*, and *behavioural system*. Bouvier [2], Brogan and Hodgins [3,4], used a physical-based particle system to simulate a crowd of athletes such as runners or bikers competing in a field. Reynolds [13,14] sought to apply virtual forces to individual robots to create steering behaviours for the whole crowd. Tu and Terzopoulos [5] used rule-based finite-state machine to construct a cognitive model of fish and succeeded in creating several interesting behaviours including flocking. Muse and Thalmann [12] used behaviour rules to design virtual characters and used scripts to construct complex social interactions with various levels of control. They used the theory in social psychology to model crowd movement behaviours. In addition, Saiwaki et al. [15] used the principle of chaos to design a virtual crowd. However, the great variations observed on the trajectory make it difficult to control the result of simulation in a precise way.

2.2 Motion Planning for Crowd

Several approaches have used the techniques of motion planning to generate motions for multiple robots. However, as the complexity of the problems in motion planning grow exponentially according to degrees of freedom in the system, the problem of planning the motion for a crowd of robots by use of a systematic search seems to be intractable. Therefore, most approaches use decoupled planning whereby crowd motions are generated sequentially by planning only for one robot at a time. In addition, in order to reduce the degree of complexity, only the motions of a few key robots are planned, and the other robots use local rules to follow the leaders. For example, in [7], the authors have proposed a decoupled planning approach to generate the motions for leaders while taking the number of followers into account. The rest of the crowd follows the leaders with the virtual force model proposed in [13]. In contrast, the authors in [8] proposed a centralised approach to tackle the motion planning problem for a crowd of over 200 robots. However, since reaching the goal is the only criterion, the behaviour of the crowd during the course of the movement remains difficult to control.

3 Design of Movement Model and Virtual Forces

In this research, we have adopted the virtual force model proposed in [13] as the way to affect the movement of each individual robot. We will first describe the movement and perception of a robot and then the virtual force model.

3.1 Design of Movement Model for an Individual Robot

We assume that the robots move under the influence of virtual forces proposed in [14], and that they must respect maximal speed limits in translation and rotation. In addition, in our system, a robot is given a view angle of 330 and a constant view distance that is 20 times the size of the robot, as shown in figure 1. Since the virtual forces are computed locally, only the robots or obstacles that are within the range of view have effect on the computation of the virtual forces.

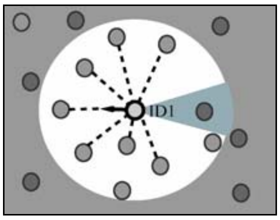


Figure 1: View range of a robot and its neighbours.

In the simulation system, each individual robot computes the virtual forces described in the next subsection according to its position relative to other robots or environmental obstacles within its range of view. The computed virtual forces are used to update the next configuration of the robot. However, the new configuration is not guaranteed to be collision-free. In the case of being in-collision, the system makes use of various local strategies to avoid a collision.

3.2 Design of Virtual Forces

The movement in our crowd simulation system is driven by five types of virtual forces have been used in this work: *separation*, *alignment*, *cohesion*, *following*, and *collision*. The first three were originally proposed in [13] and computed according to the objects in the range of view while the other two were added in [7].

Separation force (F_{sep}): the repulsive force is computed proportionally according to the distance between the robot and other neighbouring robots within the range of view. The resultant force is the summation of all of the forces exerted by each individual robot. The effect of the force is to maintain a safe distance between the robots, and the strength desired should be different for the different types of behaviour required for different environments.

Alignment force (F_{align}): A robot uses this force to align its velocity and orientation with other neighbouring robots. An average velocity for the robots within the range of view is computed first. The alignment force is then computed according to how the velocity of the current robot deviates from the average one.

Cohesion force (F_{coh}): The centre of the robots within the range of view is first computed according to their position vectors. Then the cohesion force is computed based on the vector between the centre and the current position of the robot. This force has the effect of keeping the robots as a group.

Following force (F_{fol}): A moving crowd follows a specific goal, which may not be a physically existing leader. The following force is an attractive force that drives the crowd to its goal. This force is computed according to the distance from the goal, which could also be moving. However, since there could be obstacles in the environment, the vector connecting their positions may not reflect the correct direction and distance for the following action. However, it is often found that robots can easily get stuck at some local cavity under the attractive following force. Therefore, we have adopted the model proposed in [7] to compute a collision-free following path by making the robot head to a point along the trace of the leader that does not cause collisions with obstacles, as illustrated in figure 2.

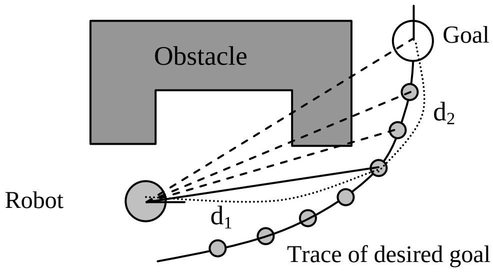


Figure 2: Following the trace of a desired goal.

Collision force (F_{col}): The repulsive collision force is exerted by the environmental obstacles when a condition of collision may be predicted after certain period of time. The direction of this force is opposite to that of the direction of the movement of the robot, and this force takes effect to resolve collisions when the robot is close to the obstacle boundary.

Our system uses a linear combination of the normalised component forces described above to compute the final virtual force as shown in (1).

$$\begin{aligned} F &= s_1 * F_{sep} + s_2 * F_{align} + s_3 * F_{coh} \\ &+ s_4 * F_{fol} + s_5 * F_{col} \end{aligned} \tag{1}$$

The set of weights (s_1, s_2, s_3, s_4, s_5) determines how the forces are composed to affect the behaviour of the robot. The appropriate setting of the weights remains as an empirical problem for the designer, which is the reason that we propose to automate the search process for an optimal solution with the genetic algorithm as described in the next section.

4 Evolution of Crowd Motion Behaviour

In a crowd simulation, the trajectory of a crowd may vary greatly according to the scenario and the environment in which the crowd is situated. For example, in an outdoor open space, people tend to keep a larger distance between themselves, and in a narrow passage, in contrast, people may be jam-packed together as they seek to reach a goal. Therefore, the designer is required to tune the weights of the component forces in order to achieve the results desired within the framework of using virtual forces to simulate the motions of a crowd. The complexity of the parameter space and the time-consuming process of developing the simulation make it a difficult optimisation problem for human or for machine.

We propose to solve the problem with the use of a genetic algorithm, commonly used to solve the problem of searching for an optimal solution in a large search space [9,10]. In a genetic algorithm, the parameters that control the performance of a system are first encoded, and fitness functions for evaluating the solutions are provided by the designer of the system. The evaluation of a population in a generation with the fitness function allows for the selection of good parameter sets to be used in the continuing evaluation in the subsequent generations. Evolution operations such as crossover and mutation are also often used in the evolution process to find global optimal solutions instead of the local optima. In the following subsection, we will describe in more detail how we encode the crowd simulation problem and how we design the fitness functions for evaluating different type of crowd behaviours.

4.1 Problem Formulating with a Genetic Algorithm

The set of weights described in the previous section are used as the genes for encoding. Each of genes is encoded into a bit string of length 10. Since we have five parameters (genes) in our system, the total length of the chromosome is 50 bits. In the current system, the population is set to 200. Each chromosome is tested in the simulation and evaluated using the fitness function. In this work, we have chosen the roulette wheel selection mechanism to select the samples that will be able to survive in the next generation. In this selection mechanism, each sample occupies a region that is proportional to the value returned by the fitness function. That is, the sample that performs better occupies a larger region in the wheel and will have a higher probability of being selected. Then, sample points are randomly selected in the wheel, and the selected samples will be put into the crossover pool for the crossover operation. We have also chosen the one-point crossover mechanism for the crossover operation, where a random cut point is selected. In each generation, we also perform the mutation operation on samples selected with the probability of 0.01. The mutation operation switches only one randomly selected bit of the selected sample.

4.2 Definitions of Fitness Functions

Fitness functions provide the basis for the selection of good samples that can survive in the next generation. The desired fitness function may also be different for different scenes and different types of behaviour. However, instead of designing a specific fitness function for each type of behaviour, we designed several elementary fitness functions that can be used to compose the final fitness function of a behaviour. The system allows the designer to use a linear combination of the elementary fitness functions to specify in what form the desired behaviour should appear. Most of the elementary fitness functions are computed based on the spatial relation between a robot and its neighbours. The neighbours of a robot are defined as the *k*-nearest robots, where *k* is specified by the user. Following Miller [11], we set the value of *k* as seven in the current system. Next, we describe the elementary fitness functions used in this work.

1. Inter-robot distance (*G_m*): For each robot, the inter-robot distance is defined as the average distance of the *k*-nearest neighbours around the robot itself. The relative distance to robot *i* is denoted as r_i, and their average distance between each other is denoted as R_i as shown in (2). For a given frame, the difference between the average and the user-specified value is the main performance index. This value is normalised by the quantisation factor Q_d to make it fall in the interval I_f=[0,1], as shown in (3). The overall system performance is computed as the average of the movement of all robots over the whole path as shown in (4)

$$R_j = \frac{\sum_{i=1}^{i=k} r_i}{k} \tag{2}$$

$$G_j = \frac{Q_d}{\left|R_j - R_e\right|}, \text{ s.t. } 0 < G_j \le 1 \tag{3}$$

$$R_m = \frac{\sum_{j=1}^{j=N} G_j}{N}, G_m = \frac{\sum_{m=1}^{m=L} R_m}{L} \tag{4}$$

2. Distance to the goal (*G_g*): The calculation is similar to that for inter-robot distance described above except for that the distance is computed with respect to the goal instead of each robot. In the interest of saving space, we do not repeat the formula here. In addition, instead of being a physically existing object, the goal could also be a designated position at a point behind a possibly moving leader. Again, the value returned by this fitness function is also normalised to the interval I_f.

3. Number of collisions (*G_c*): Collisions are considered an undesirable effect in crowd simulation. Therefore, the larger the number of collisions, the smaller the value that this fundamental fitness function should return. Assume that N_c denotes the number of robots that are in collision with other robots or obstacles, and that *N* denotes the total number of robots in the simulation. Then, the collision ratio s_j is defined as the percentage of robots that are in collision as shown in (5). The overall elementary fitness function is then defined as the average of this ratio in movement over the whole path as shown in (6).

$$s_j = \frac{N - N_c}{N} \tag{5}$$

$$G_c = \frac{\sum_{j=1}^{j=L} s_j}{L} \tag{6}$$

4. Consistency in orientation (*G_a*): This elementary fitness function is defined to measure consistency in orientation with relation to that of neighbouring robots. The average difference in the orientation of a robot with respect to other neighbouring robots is computed in (7). The consistency in orientation for a single robot, as shown in (8), is defined as the complement of the difference in orientation. The fitness function for the whole crowd is then defined as the average of all robots in movement over the whole path as shown in (9).

$$A_j = \frac{\sum_{i=1}^{i=k} \left|\theta_j - \theta_i\right|}{k}, i \ne j \tag{7}$$

$$B_j = (1 - \frac{A_j}{\pi}) \tag{8}$$

$$B_m = \frac{\sum_{j=1}^{j=N} B_j}{N}, G_a = \frac{\sum_{m=1}^{m=L} B_m}{L} \tag{9}$$

5. Consistency in Distance (*G_d*): The consistency of the distances between one robot and another neighbouring robot is defined as the standard deviation of these distances. The average distance and standard deviation are first computed according to (10) and (11). The consistency in distance for a robot is then defined as the inverse of the standard deviation multiplied by the quantisation factor Q_σ (12). In addition, we need to ensure that the value of the consistency will always fall into the range I_f. The overall performance index is computed as the average of the consistency in difference between all robots over the whole path as shown in (13).

$$R_{mean} = \frac{\sum_{i=1}^{i=k} r_i}{k} \tag{10}$$

$$\sigma_j = \frac{\sqrt{\sum_{i=1}^{i=k} (R_i - R_{mean})^2}}{k} \tag{11}$$

$$F_j = \frac{Q_\sigma}{\sigma_j}, \text{ s.t. } 0 < F_j \le 1 \tag{12}$$

$$F_t = \frac{\sum_{j=1}^{j=N} F_j}{N}, G_d = \frac{\sum_{t=1}^{t=L} F_t}{L} \tag{13}$$

The overall fitness function (F_{sum}) for the evaluation of the performance of a crowd simulation is computed based on a linear combination of the elementary fitness functions defined above. The formula is shown in (14).

$$G_{\mathrm{sum}} = S_m * G_{\mathrm{m}} + S_g * G_{\mathrm{g}} + S_c * G_{\mathrm{c}} + S_a * G_{\mathrm{a}} + S_d * G_{\mathrm{d}} \quad (14)$$

A designer makes use of the weights (S_m, S_g, S_c, S_a, S_d), according to the nature of the desired behaviour, to compose the final fitness function from the elementary ones. We assume that these weights are more intuitive to set compared to the weights in (1) and that they should be the same for the same desired behaviour of crowd movement. However, the optimal weights in (1) may be scene specific, as will be shown in the next section.

5 Experimental Design and Results

We implemented the simulation system and the genetic algorithm in Java. In the experiments that we report in this section, the total length of the simulation and the desired distance between the robots are specified by the user. In addition, only one group is considered at a time.

5.1 Scene and Behaviour Variations

Generally speaking, it is difficult to classify environments or define typical scenes. Nevertheless, we have defined and tested three types of scenes in our current experiments including an open space without obstacles (E1), a scene with a narrow passage (E2), and a scene cluttered with small obstacles (E3).

Three types of behaviours were tested in our experiments: *group moving* (B1), *following* (B2), and *guarding* (B3). The group moving behaviour refers to keeping the crowd moving with a given inter-robot distance in a group. In the case of the following behaviour, the objective is to pursue a possibly moving goal as closely as possible, and for the guarding behaviour, the crowd is supposed to surround the possibly moving goal, which could also be another leader robot.

For each different type of behaviour, we used the elementary fitness functions to compose the final fitness function used in the genetic algorithm. We assume that these weights are used to express the designer’s intention for an ideal behaviour and should be independent of the environment. Nevertheless, the user still needs to give the inter-robot distance in order for the simulation to converge into the desirable behaviours.

5.2 Experimental Results

We conducted experiments with the genetic algorithm to acquire the set of parameters for the desired behaviour for each given environment. The set of weighting parameters generated by the system are given in table 1. In order to validate the parameters, we used the parameters obtained for E1 to run simulations with E2 and E3. The overall scores G_{sum} returned by the fitness function are 295, 150, and 262, respectively. When we used the optimal parameters generated for E2 and E3, respectively, to run the experiments again, the scores improved to 271 and 284, respectively. Although the scores in the cluttered environments (E2 and E3) are not as good as might have been anticipated from the result for E1, the scores were greatly improved when the optimal parameters were used. This experiment reveals that the optimal weighting parameters for the virtual force are scene dependent.

Table 1: Optimal weights generated by the genetic algorithm for various environments and behaviours (B: Behaviour, E: Environment)

	s_1	s_2	s_3	s_4	s_5
B1-E1	0.42	0.03	0.15	0.84	0.44
B1-E2	0.87	0.31	0.99	0.49	0.16
B1-E3	0.65	0.12	0.19	0.01	0.12
B2-E1	0.53	0.15	0.77	0.36	0.62
B2-E2	0.90	0.39	0.80	0.06	0.16
B2-E3	0.95	0.21	0.93	0.07	0.04
B3-E1	0.94	0.42	0.90	0.14	0.59
B3-E2	0.98	0.43	0.95	0.75	0.71
B3-E3	0.97	0.32	0.63	0.09	0.27

The results of the simulation are illustrated in figures 3 to 5. A convex hull (in red) is computed to illustrate the boundary of a crowd (20 robots) in each example. In figure 3, we show an example of the following behaviour (B2) for the crowd with a small inter-robot distance in a space cluttered with obstacles (E3). In figure 4, we show an example of group movement (B1) where the desired inter-robot distance is set to a higher value, and the crowd needs to pass the narrow passage (E2) in order to reach the goal. In figure 5, we show the case of body guarding a specific leader robot depicted in red (B3) in an open space (E1). The crowd succeeds in surrounding the leader along the path while they are moving.

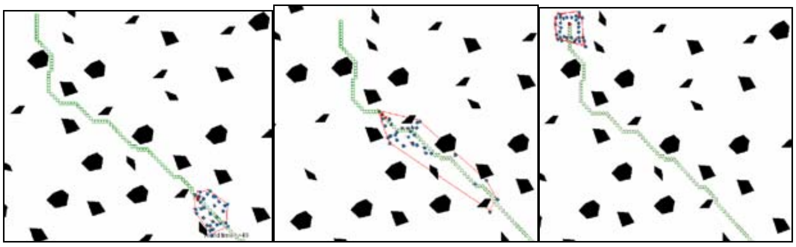

(a) (b) (c)

Figure 3: Example of simulation results on the following behaviour (B2) with a small inter-robot distance in a space cluttered with small obstacles (E3)

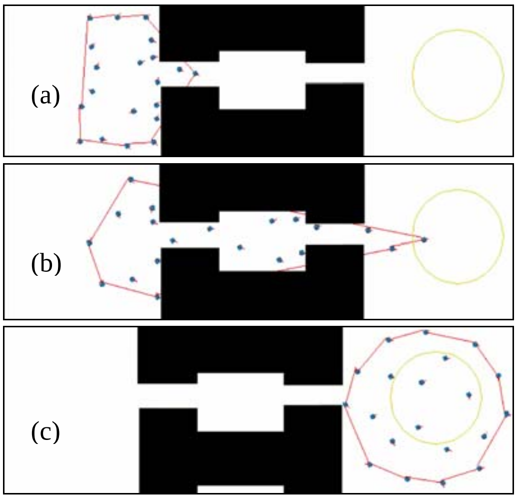


Figure 4: Example of simulation results for the group moving behaviour (B1) with a large inter-robot distance passing a narrow passage (E2).

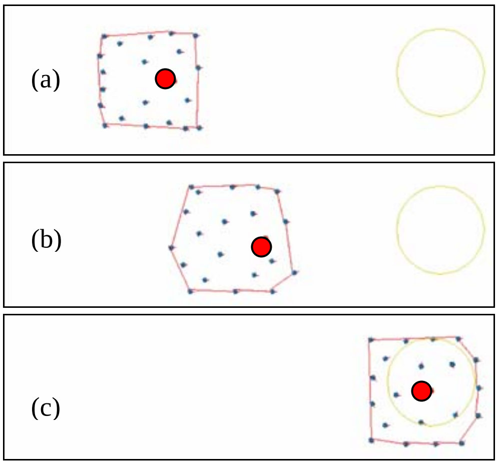


Figure 5: Example of simulation results for the guarding behaviour (B3) in open space (E1).

6 Conclusion

As the applications of virtual crowd simulation have increased in recent years, the demand for better simulation tools for creating desirable results have also risen. In this paper, we have proposed to formulate the problem as one of the optimisation of the setting of the parameters for virtual forces by the use of a genetic algorithm. The fitness functions used in the genetic algorithm are composed according to the behaviours desired from elementary ones designed for evaluating a specific aspect of simulation. Our preliminary experiments reveal that the genetic algorithm is a good way to automate the time-consuming process of generating the optimal set of parameters for virtual forces for a given scene and desired behaviour in movement. In the future, we will conduct experiments with more types of environments and behaviours in order to shed some light on the limitations of using this type of virtual force mechanism to simulate the motions of a crowd.

7 Acknowledgement

This work was supported by National Science Council under contract NSC94-2213-E-004-006.

8 References

[1] T. Balch and R.C. Arkin, “Behaviour-based formation control for multirobot teams,” *IEEE Transactions on Robotics and Automation*, 14(6), pp 926-939 (1998).

[2] E. Bouvier, E. Cohen and L. Najman, “From crowd simulation to airbag deployment: particle systems, a new paradigm of simulation,” *Journal of Electronic Imaging*, 6(1), pp 94-107 (1997).

[3] D.C. Brogan and J. Hodgins, “Group behaviors for systems with significant dynamics,” *Autonomous Robots*, 4, pp 137-153 (1997).

[4] D.C. Brogan, R.A. Metoyer and J.K. Hodgins, “Dynamically simulated characters in virtual environments,” *IEEE Computer Graphics and Applications*, 18(5), pp 58-69 (1998).

[5] J. Funge, X. Tu and D. Terzopoulos, “Cognitive model: knowledge, reasoning, and planning for intelligent characters,” *Proc. of ACM SIGGRAPH*, Los Angeles, pp 29-38 (1999).

[6] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA (1992).

[7] T.Y. Li, Y.J. Jeng and S.I. Chang, “Simulating virtual human crowds with a leader-follower model,” *Proc. of 2001 Computer Animation Conf.*, Seoul, Korea (2001).

[8] T.Y. Li and H.C. Chou, “Motion planning for a crowd of robots,” *Proc. of the 2003 Intl. Conf. on Robotics and Automation*, Taipei, Taiwan (2003).

[9] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA. (1998).

[10] G. Mitsuo and C. Runwei, *Genetic Algorithms and Engineering Design*, John Wiley and Sons, New York (1997).

[11] G.A. Miller. “The magical number seven, plus or minus two: some limits on our capacity for processing information,” *Psychological Review*, 63, pp 81–97 (1956).

[12] S.R. Musse and D. Thalmann, “Hierarchical model for real time simulation of virtual human crowds,” *IEEE Trans. on Visualization and Computer Graphics*, 7(2), pp 152–164 (2001).

[13] C.W. Reynolds, “Flocks, herds, and schools: a distributed behavioural model,” *Computer Graphics*, pp 25-34 (1987).

[14] C.W. Reynolds, “Steering behaviours for autonomous characters,” *Proc. of Game Developers Conf.*, San Jose (1999).

[15] N. Saiwaki, T. Komatsu, T. Yoshida and S. Nishida, “Automatic generation of moving crowd using chaos model,” *Proc. of IEEE Int. Conf. on System, Man and Cybernetics*, pp 3715-3721 (1997).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Planning of Unpredictable Trajectories for Surveillance Mobile Robots

Luiz S. Martins-Filho[1], Elbert E.N. Macau[2], Ronilson Rocha[3]
[1]Department of Computer Science and [3]Department of Control and Automation,
Federal University of Ouro Preto, Ouro Preto, MG, Brazil
[2]Associate Laboratory for Computating and Applied Mathematics,
National Institute for Space Research, São José dos Campos, SP, Brazil
luizm@iceb.ufop.br, elbert@lac.inpe.br, rocha@em.ufop.br

Abstract

This paper discusses a peculiar trajectory planning strategy for surveillance mobile robots. The proposed trajectory planner is based on the typical behaviour of chaotic dynamical systems. This specific kind of mission requires fast scanning of the robot workspace using trajectories resembling non-planned motion for external observers. To obtain these features, a chaotic behaviour is imparted to the robot motion using a trajectory-planner based on the well-known conservative Standard map (also called Taylor-Chirikov map). In the paper, we detail the trajectory planning strategy and briefly describe the robot modelling and the motion control. Concluding the text, we present the results and analysis of numerical simulations.

Keywords: mobile robots, motion planning, surveillance robots, kinematic control, chaos

1 Introduction

This paper addresses issues on mobile robots motion planning and control to accomplish specific terrain surveillance missions [1]. Mobile robotics remains as an important research subject, in spite of a long time of cumulated results and advances, in consequence of its ever-increasing economical and technological relevance and the variety of applications' domains [2].

These surveillance missions require a motion control system comprising some basic navigation competencies, like planning and reacting. Moreover, additional motion requirements are strongly desirable: high unpredictability of the robot trajectories and fast workspace scanning. To satisfy these special demands, we propose a trajectory planner based on dynamical behaviour of chaotic systems, working as an auxiliary module within a closed-loop locomotion control scheme. The motion planning results on a trajectory passing through a sequence of objective points, i.e. the robot must reach a planned sequence of target locations.

Studies concerning the interactions between mobile robotics and chaos have been developed more recently. An integration between the robot motion and the Arnold chaotic system, for instance, is proposed to impart chaotic behaviour to mobile robots in [3]. In [4], an extension of this same control strategy, applying different chaotic systems on integration with the robot kinematic, is presented. This same principle of systems integration is used in [5], associating a Van der Pol equation with the obstacle and the target. In [6], an open-loop control approach is proposed to produce unpredictable trajectories, using Lorenz chaotic system to command the wheels. Another kind of chaos application appears in [7]: solving local minima problems in path-planning through virtual force field.

1.1 Surveillance Robots

An important issue on surveillance mission for mobile robots is the planning of patrolling path. Actually, the robot motion strategy is a crucial factor for the success of any robot surveillance mission.

For accomplishing the very first mission requirement, the scanning of the whole robot workspace, an intuitive strategy is to command the robot motion to cross every terrain cell (supposing the terrain partitioned in unitary area cells), covering systematically the entire terrain. A quite obvious procedure for the complete covering is an exhaustive and systematic scan using parallel straight trajectories. Nevertheless, an eventual smart intruder could easily understand this strategy, and therefore it would avoid the sentry robot and stay unperceived by the security system. Many other simplistic strategies could be proposed, but still falling into the category of predictable behaviours.

Evidently, using a group of surveillance robots, the patrolling planning becomes a different and easier problem, where each robot receives a scan task for a subarea of the partitioned terrain, optimising the team actions and ensuring the fast and complete terrain coverage [8]. In [9], another approach is proposed, applying genetic algorithms to maximise the terrain scan in a very intricate strategy, and with consequent high computational costs.

More traditional motion planning approach is based on human operators commanding the robot, or the multi-robots team, to visit a sequence of terrain regions in an arbitrary way, or in basis of the operator suspections or conjectures about the possibilities of an intruder presence (e.g. in [8]).

In terms of unpredictability, the trajectories produced using random motion can be considered very erratic and impossible to be understood by the intruder. This random-like planning, as proposed in [10], guarantees the complete terrain scan. As a consequence, this strategy attends the two main patrol mission requirements: unpredictable trajectories and whole terrain covering. Nevertheless, this strategy presents a disadvantage: a complementary difficulty for the robot control functions involving its localisation. This localisation function complexity can be verified in the sensor fusion schemes [10].

In this work, we intend to exploit the dynamical behaviour of a conservative chaotic system to accomplish the main surveillance mission requirements.

2 Trajectory Planning

Our objective is to obtain a deterministic chaotic behaviour of the robot motion, in such a way that the trajectories become unpredictable, i.e. the knowledge of the robot system state during an arbitrary time does not allow to predict its subsequent trajectory. That is a consequence of the initial conditions dependence or sensibility, and the continuous frequencies spectrum, that characterises a non-periodic behaviour. Despite its nonperiodicity, the obtained trajectories pattern tends to a sort of repetition.

This behaviour is an attractive issue for the considered mobile robots missions since it provides the desired trajectory unpredictability and fast terrain scanning. We intend to impart the chaotic behaviour to the robot using Poincaré sections. A Poincaré section defines a dynamic system's phase space with arbitrary dimension, and composes a map featuring the chaotic behaviour. Differential equations generate a discrete map through the flow intercepting a Poincaré section (or return map) at each unity of time.

A well-known two-dimensional coupled return map in phase space, called Standard map, represents a physical system defining a kicked rotor, and is very useful for studying the chaotic motion basic features. This area-preserving map was firstly proposed by Bryan Taylor (and independently obtained by Boris Chirikov) to describe the dynamics of magnetic field lines on the kicked rotor [11]. The experimental scheme of this system is composed by a rotor subjected to a driving force that comes at exactly even intervals in time, provided by a periodic pulse. The simple scheme consists of kick occurrence on a constant direction and intensity. The dynamics consequence of this forcing action depends of the system position and speed at the kick application instant. The mathematical model of a Poincaré section of this system is given by the following map equations

$$\begin{cases} x_{i+1} &= x_i + K \sin y_i \\ y_{i+1} &= y_i + x_{i+1} \end{cases} \tag{1}$$

Here x is a periodic configuration variable (angular position) and y is the momentum variable (angular speed), both computed $\bmod(2\pi)$. The map is parameterised by K, that represents the strength of the nonlinear kick. Standard map is so interesting mainly because it can show chaos under very basic conditions.

The terrain coverage using the Standard map is analysed by simulating the iterative map and verifying if the coverage properties of the sequence of planned target positions satisfy the mission requirements. For these tests, we define a 100×100 (in normalised measurement unit) square terrain. The map simulation begins with an arbitrary initial position, and considers the gain value $K = 7$. The results for partial target locations planned for 10,000 iterations are shown in figure 1, where the uniform distribution of the points over the terrain can be observed. We define a performance index for the terrain coverage based on 10,000 square unit cells (1×1), and we analyse the visited cells percentage for the robot target locations planning (this analysis does not consider the robot trajectories between two target locations). This visitation index, plotted in terms of its time evolution, is shown in figure 2 (where index=1 represents 100% of cells visited).

These results are similar to a uniform distribution obtained by random numbers generation, however with very different construction nature, as we will discuss in section 4. The time evolution of visitation index allows us to conclude that the complete scan is ensured. Moreover, the robot will also visit the terrain cells during the displacement between two consecutive planed positions, as a consequence, the effective visitation index taking into account the executed trajectory will grow faster.

3 Robot Motion Control

We adopt here a typical differential motion robot with two degrees-of-freedom. The locomotion system is composed by two independent active wheels, and a passive wheel working as a sort of free steered wheel. The robot is equipped with proximity sensors

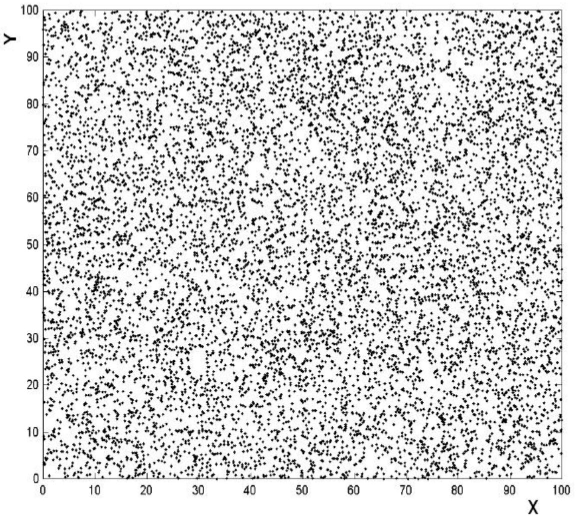


Figure 1: The terrain showing the subgoals locations planned by the Standard map (10,000 planning iterations).

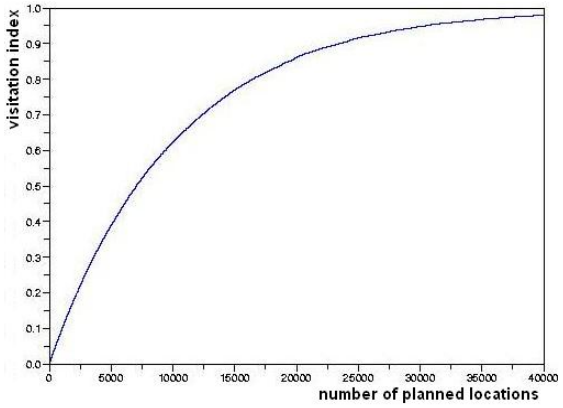


Figure 2: Time evolution of visited cells index (40,000 iterations).

capable of obstacles detection providing short-range distances (e.g. sonar or infrared devices), and with specialised sensors for detection and recognition of targets. The robot body is considered rigid, remaining on the horizontal plane. The motion resulting of a kinematic control of the active wheels can be described in terms of linear velocity $v(t)$ and robot direction $\theta(t)$. This model describes the motion of a fixed robot point, and the body rotation around this point, given by the angular velocity $\omega(t)$. The motion control is done by providing the two active wheels velocities or, equivalently, the input variables $v(t)$ and $\omega(t)$. The mathematical model of this motion considers three state variables: the robot position and its orientation $(x(t), y(t), \theta(t))$ [12]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

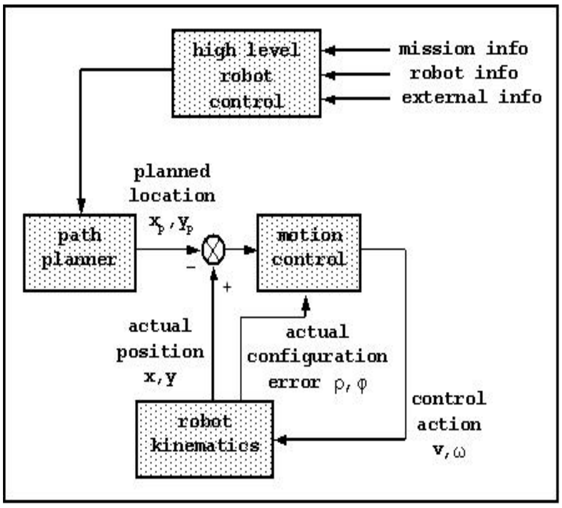


Figure 3: Overall robot control including a high level control module, the proposed path planner, the motion control, and the interaction with the physical robot.

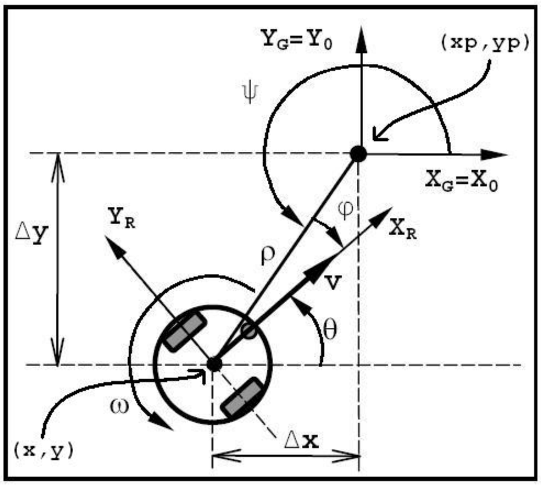


Figure 4: Configuration of the kinematic control showing explicitly the error variables ρ and φ.

This equation describes a nonholonomic nonlinear system. The control problem for this class of systems has been deeply studied, and diverse adequate solutions are available such as in [13]. We adopted one solution involving a state feedback controller, which is an appropriate approach to produce a trajectory described by a sequence of coordinates (x_p, y_p). It means that the target locations planning is performed by a specialised robot module, independent of the motion control module, as shown in the scheme depicted in figure 3.

This control law was proposed in [13], considering the geometric configuration described in figure 4. The robot is posed at an arbitrary position and orientation (x, y, θ), and the target position is defined by the robot trajectory-planner. A configuration error vector is defined by $e = [\rho \;\; \varphi]^T$, where ρ and φ define the next target relative position.

The robot kinematic model, equation (2), is described on the absolute reference frame, fixed on the workspace. We define φ as the angle between the direction from the robot centre to the desired position and the axis X_R of the body reference frame. The variable ρ describes the distance between current and desired positions, and ψ is the angle between the direction to the desired position and the axis X_0. We simplify the model, reducing the problem to a two-degrees-of-freedom one, using polar coordinates and the following coordinate change: $\rho = (\Delta x^2 + \Delta y^2)^{1/2}$ and $\varphi = \pi + \theta - \psi$. The new mathematical description of the robot motion is given by

$$\begin{bmatrix} \dot{\rho} \\ \dot{\varphi} \end{bmatrix} \begin{bmatrix} -\cos\varphi & 0 \\ \frac{1}{\rho}\sin\varphi & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

This new motion model is not defined at $x = y = 0$, i.e. at the origin of $X_G Y_G$ frame ($1/\rho \to \infty$ for $\rho \to 0$). This point is only achieved when the robot reaches the goal location, and it can be avoided switching the target to the next planned location before arriving too close to the singularity. The feedback control law, defining the system inputs v and ω, is given by:

$$\begin{aligned} v &= k_1 \rho \cos\varphi \\ \omega &= -k_1 \sin\varphi \cos\varphi - k_2 \varphi \end{aligned} \tag{4}$$

The proof of the system stabilisation, and the consequent convergence to the target position, applying this nonlinear control law, is given by Lyapunov stability analysis. Considering a Lyapunov function V defined in terms of ρ and φ as follows:

$$V = \frac{1}{2}(\rho^2 + \varphi^2) \tag{5}$$

taking its derivative, and using the equation (3), we obtain

$$\dot{V} = \rho\dot{\rho} + \varphi\dot{\varphi} = \rho(-v\cos\varphi) + \varphi(\omega + \frac{v}{\rho}\sin\varphi) \tag{6}$$

To analyse this expression, we substitute the control variables by the control laws defined in equation (4):

$$\dot{V} = -k_1(\rho\cos\varphi)^2 - k_2\varphi^2 \tag{7}$$

From equation (7), considering exclusively positive values for k_1 and k_2, we can see that $\dot{V} \leq 0 \ \forall \ (\rho, \varphi)$. This is a sufficient condition for the asymptotic convergence of equation (3).

We do not consider in this work the problem of obstacle avoidance, nevertheless a simple solution can be easily implemented, for instance, using the BUG2 algorithm proposed in [14]: the robot follows the obstacle contour, but departs immediately when it is able to move directly toward the target.

A second motion control approach, called here discontinuous control law, is defined as a sequence of two control steps: an initial rotational maneuver around the robot centre to orientate the robot head towards the goal position, followed by a straight trajectory motion directly to the planned objective point.

Using these two feedback control laws, we will now test the proposed trajectory planning strategy, and analyse the unpredictability and the terrain coverage of the trajectories obtained by the simulated robot motion.

4 Numerical Simulations Results

The robot motion was simulated numerically using the mobile robot kinematic model discussed above, applying a motion control to track the trajectory composed of a planned sequence of objective points. In the case of different terrain shapes, the planning process could fit the interest area inside a square Standard map, excluding the points planned outside the terrain.

Another remark: the robot can percept an object or event when performing the surveillance task inside the sensor range region. The dimensions of this perception field depend on the properties of the device used to perceive external objects. In this work, we do not take into account this extra area covered by the sensors, consequently the terrain scan analysis presented here considers the worst case.

Considering the crucial requirements of a typical surveillance mission, and the main ideas developed in our approach, we can configure different scenarios for the numerical simulations. These scenarios could present different ways of obtaining the construction of the mobile robot trajectories through the combination of the motion control laws and the determination of the regions to be visited by the patrol robot. We will compose here two scenarios.

In the first case, we consider the approach proposed in this work, i.e. we adopt the continuous control law that can produce smooth trajectories avoiding undesirable and unnecessary maneuvers, and their consequent control switches, and we apply a planning strategy based on the Hamiltonian chaotic system (the conservative Standard map) to establish a sequence of objective locations or, in other words, a sequence of terrain regions to be visited by the robot. The results of numerical simulations for this scenario can be seen in figure 5.

In the second scenario, a random strategy is applied to determine the sequence of objective points. This strategy consists of taking a random sequence of locations (coordinates x_p and y_p) uniformly distributed in the patrolled space. This random planning provides a similar result in terms of terrain coverage, as discussed in section 2. For the robot motion control, we adopt

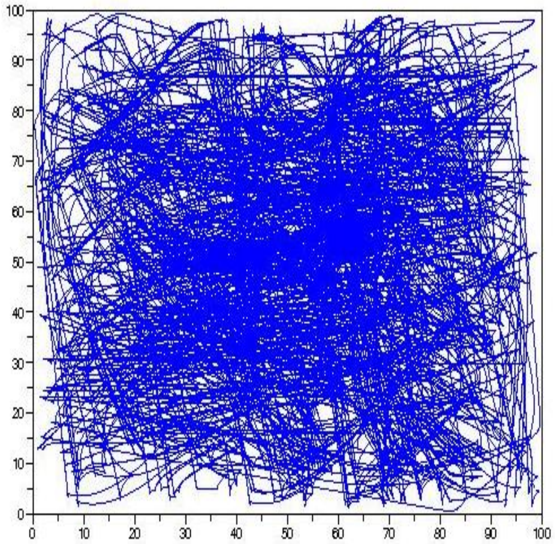


Figure 5: The mobile robot trajectory evolution considering the first simulation scenario.

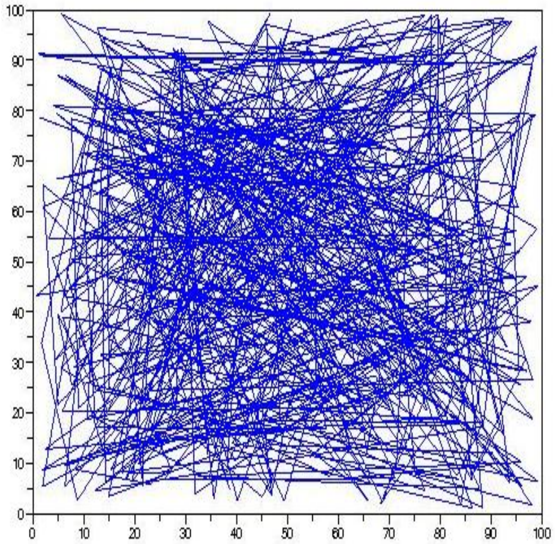


Figure 6: The mobile robot trajectory evolution considering the third simulation scenario.

the discontinuous control law based on a rotational maneuver around the robot's centre to orientate its head towards the next desired position, followed by a straight trajectory directly to the target point. The results for this case are shown in figure 6.

Discussing the advantages and disadvantages of the two main aspects composing these scenarios, i.e. the control laws and visitation planning, the differences can be remarked and a qualitative conclusion about the proposed approach can be established.

Examining the options of control laws, continuous or discontinuous, it is quite obvious that the continuous one offers advantages: the smoother trajectories reduce control switches and maneuvers; moreover, they contribute to perform unpredictable trajectories for external observers, reinforcing the erratic aspect. In the opposite case, the discontinuous control composed of two subsequent maneuvers (pure rotation followed by linear displacement) produces piece-wise straight trajectories that could be predicted by an external observer trying to avoid the sentry detection range.

The choice between chaotic and random visitation planning can result, in a first analysis, in very similar results in terms of terrain scan and surveillance space coverage. However, there is a fundamental advantage of chaotic trajectories over the random walk: the deterministic nature of the sequence of objective points, i.e. the robot mission supervisor can know in advance the sequence of future robot positions.

Considering the chaotic planning approach, the robot navigation system maintains complete control and knowledge of the planned positions sequence for the reason that it's a deterministic system and its dynamical behaviour is precisely defined and deeply studied. In terms of navigation competencies, this path-planning determinism represents an important advantage over navigation based on a sort of random walk trajectory. For instance, this determinism can facilitates the frequent robot localisation procedure, which is a crucial function because the knowledge of the robot position with appropriate precision is a very necessary information for the robot itself and also for the mission operation centre. In a robot localisation procedure, sensors data assimilation algorithms take advantage of good previous information to estimate the current position with better accuracy. The importance of this knowledge is associated to the executed trajectory supervision, the information of the terrain scan process, the better precision of the eventual intruder or target localisation, and so on.

The results presented in section 2 (figures 1 and 2), showing the spacial distribution of planned points positions to be reached by the robot motion, confirm that the entire terrain will be visited and verified by the sentry robot. When moving between two subsequent planned positions, the robot will certainly visit other terrain cells, possibly passing multiple times by diverse terrain regions, which consists an advantage for the mission of search a mobil intruder trying to scape of the sentry detection. The erratic aspect of the robot motion can be easily verified by visual inspection of the obtained trajectories, that allows us to conclude qualitatively that the main mission requirements (trajectories unpredictability and complete terrain scan) are supplied by the proposed strategy.

5 Conclusion

The strategy proposed in this paper, tested through numerical simulations, can obtain appropriate results suited for planning the motion of surveillance mobile robots. The planned sequence of objective points trajectories satisfy the main requirements of fast scanning of the workspace. The resulting chaotic motion behaviour ensures high unpredictability of the robot trajectories, resembling a non-planned, quite erratic motion from external observers' point of view. Moreover, this work makes us believe that applications of special dynamical behaviours of nonlinear systems on solutions for mobile robots motion problems represent an attractive interdisciplinary interface for researchers of both scientific domains, reinforcing the interesting perspectives of future works.

This work will follow with quantitative analysis of the obtained robot trajectories, including the terrain coverage estimation, the evaluation of intruders detection probability, and the analysis of the executed trajectories dynamics using specific tools of Chaos Theory.

6 Acknowledgements

The authors acknowledge the financial support of Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Brazil); Fundações de Amparo à Pesquisa dos Estados de Minas Gerais and São Paulo - FAPEMIG (Minas Gerais, Brazil) and FAPESP (São Paulo, Brazil), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES (Brazil).

7 References

[1] H.R. Everett, "Robotic security systems", *IEEE Instrumentation and Measurement Magazine*, 6(4), pp 30-34 (2003).

[2] B. Rooks, "Robotics outside the metals industries", *Industrial Robot: An International Journal*, 32(3), pp 205-208 (2005).

[3] Y. Nakamura and A. Sekiguchi, "The chaotic mobile robot", *IEEE Transaction on Robotics and Automation*, 17(6), pp 898-904 (2001).

[4] A. Jansri, K. Klomkarn and P. Sooraksa, "On comparison of attractors for chaotic mobile robots", *Proceedings of Annual Conference of the IEEE Industrial Electronics Society*, Busan, Korea, pp 2536-2541 (2004).

[5] Y. Bae, "Target searching method in the chaotic mobile robot", *Proceedings of IEEE Digital Avionics Systems Conf.*, Salt Lake City, USA, pp 12.D.7 - 12.1-9 (2004).

[6] L.S. Martins-Filho, R.F. Machado, R. Rocha and V.S. Vale, "Commanding mobile robots with chaos", in J.C. Adamowski, E.H. Tamai, E. Villani and P.E. Miyagi (eds), *ABCM Symposium Series in Mechatronics*, Vol. I, ABCM, Sao Paulo, pp 40-46 (2004).

[7] C. Choi, S.-G. Hong, J.-H. Shin, I.-K. Jeong and J.-J. Lee, "Dynamical path-planning algorithm of a mobile robot using chaotic neuron model", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, USA, pp 456-461 (1995).

[8] J. Feddema, C. Lewis, and P. Klarer, "Control of multiple robotic sentry vehicles", in G.R. Gerhart, R.W. Gunderson, C.M. Shoemaker (eds), *Proceedings of SPIE - Unmanned Ground Vehicle Technology*, 3693, Orlando, USA, pp 212-223 (1999).

[9] T. Emmanuel, L. Fagegaltier and A. Liegeois, "Motion planning for a patrol of outdoor mobile robots", *Proceedings of of European Robotics and Intelligent Systems Conference*, Malaga, Spain, pp 130-139 (1994).

[10] T. Heath-Pastore, H.R. Everett and K. Bonner, "Mobile robots for outdoor security applications", *Proceedings of American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems*, Pittsburgh, USA, pp 1-16 (1999).

[11] A.J. Lichtenberg and M.A. Lieberman, *Regular and Stochastic Motion*, Springer-Verlag, Berlin (1983).

[12] R. Siegwart and I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, MIT Press, Cambridge, MA (2004).

[13] S.-O. Lee, Y.-J. Cho, M. Hwang-Bo, B.-J. You and S.-R. Oh, "A stable target-tracking control for unicycle mobile robots", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, pp 1822-1827 (2000).

[14] V. Lumelsky and T. Skews, "Incorporating range sensing in the robot navigation function", *IEEE Transactions on Systems, Man and Cybernetics*, 20(5), pp 1058-1068 (1990).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Flock Traffic Navigation Based on Negotiation

Carlos Astengo-Noguez, Ramón Brena-Pinero
Center for Artificial Intelligence,
ITESM, Campus Monterrey, México
castengo@itesm.mx, ramon.brena@itesm.mx

Abstract

Every day people face traffic congestion in urban areas, raising travel time and stress issues in drivers. Beyond traffic lights synchronisation, emerging location-based technologies such as GPS and cellular communications suggest some futuristic traffic coordination schemes. Agent-based coordination has been proposed to automate car flow in efficient ways, but at the cost of generating dangerous situations. In this paper we present an original traffic coordination method, where vehicles group in "flocks", just as many animal species travel in nature, in order to increase efficiency and safety. Rational negotiation is used here to take decisions about whether or not to join cars together. We present preliminary experiments showing the feasibility of our flock navigation scheme in a very simplified city represented by a square grid.

Keywords: multiagent, flocks, navigation, path planning, rational negotiation

1 Introduction

Every day people face traffic congestion in big cities. Currently, traffic is controlled through traffic lights, static signs and, in a few places, electronic boards that give information about traffic flow, road accidents or weather-related situations. Traffic congestion describes a condition in which vehicle speeds are reduced below normal, increasing drive times and causing vehicle queuing. It occurs only when the demand is greater than the roadway capacity [1].

Since 1982 congestion levels have risen in cities of all sizes, indicating that even the smaller areas are not able to keep pace with rising demand. There are several statistics that point to worsening congestion levels. Congestion extends to longer periods of the day, more roads, affects more of the travel and creates more extra travel time than in the past [2]. According to the US Federal Highway Administration [3] congestion levels have risen to levels experienced by the next largest population group every 10 years; in 2001, cities of between 500,000 and one million people experienced the congestion of cities between one and three million in 1992.

Reducing total congestion saves time and fuel and leads to decreased vehicle emissions [3]. The Texas Transportation Institute estimates that in 75 of the largest US cities in 2001, $69.5 billion dollars are wasted in time and fuel costs [4]

Since 1868, when the first traffic light was built in England, no other innovation has been developed to manage the traffic problem. Of course, multilevel intersections avoid the problem of coordinating traffic, but they are extremely costly solutions that are unfeasible in many situations.

Emerging location-based technologies, using the Global Positioning System (GPS) and cellular communications technologies, allow us to envision some futuristic coordination-based traffic handling mechanisms that could be feasible with current research-level technology [5, 6].

Some agent-based automated vehicle coordination mechanisms have been proposed [2] but they lead to intricate traffic merging. Indeed, in high density traffic, alternating individual cars at intersections could produce very dangerous situations; even assuming the protocol itself is flawless, any factor outside the protocol, like a blown tyre or mechanical failure, would be a recipe for disaster. Human lives could not be exposed to such a risk.

In this paper we propose that vehicles could navigate automatically in groups called "flocks". Birds, animals and fish seem to organise themselves effortlessly by travelling in flocks, herds and schools that move in a very coordinated way [7]. A flock consists of a group of discrete objects moving together. In the case of bird flocks, evidence indicates that there is no centralised coordination, but a distributed control mechanism.

Flock traffic navigation allows the coordination of intersections at the flock level, instead of at the individual vehicle level, making it simpler and far safer. To handle flock formation, we use coordination mechanisms issued from multiagent systems, which are systems composed of multiple interacting computing agents [8]. Agents are computer systems situated in some environment with autonomous action and interacting capability (cooperation, negotiation, coordination, etc).

After this motivating introduction, in the following section we make a technical presentation of our proposal. Then we present an experimental framework and some experimental results, followed by a discussion and revision of related work, ending with a conclusion.

2 Flock Navigation

Taylor and Jefferson [7] point out that nature is organised in four levels of structure: the molecular level, cellular level, organism level and population-ecosystem level. The higher levels have traditionally been expressed formally as systems of differential equations without much success. They write: “there are simply no mathematical tools for dealing with equations systems of that complexity”.

One of the most fundamental and successful insights from the field of artificial life has been the development of an alternative population modelling paradigm as sets of co-executing computer programmes [7].

Dyer [9] notes that “migration and navigation are very complex behaviours and require integration of multiple coordinated and carefully sequenced sensory /motor behaviours.” The execution of any of these behaviours requires a high degree of intelligence and computation. Animal decision-making is also complex and state-dependent.

Birds coordinate their migration behaviour in flocks. We pretend to use a similar strategy to model vehicular traffic.

In the city context, flock navigation means that vehicles join together in groups (flocks) to travel through intersections. But as individual cars can have different departure and destination points, we can see that they would travel together for only a part of their entire travel. This means that the following problems have to be solved:

- How to find appropriate travel partners;
- Where the travelling partners should join and where they should split;
- How different flocks coordinate in order to avoid collisions and improve efficiency.

We intend to solve these problems in the context of rational decision-making. This means that for each flock member it is convenient to travel with partners, compared with the possibility of travelling alone, and that if no gain can be shown they could travel alone.

Further, determination of joining and splitting points with respect to the flock would be determined on a rational basis as well.

Flock navigation, as presented in this paper, may seem to require automated driving cars, but actually flock navigation could just as well be suggested to human drivers, who would decide whether to follow the recommendation or not.

We think the rational perspective has some advantages over, say, a physics particle perspective where vehicles would simply obey laws regardless of whether or not their behaviour agrees with their interests. In a human society, if drivers have the option of driving alone or in automated flocks, they will only do the latter if it agrees with their interests. So we have to design social mechanisms in such a way that they are individually rational.

We will assume a city with the following rules:

- As for birds, there are no traffic lights (or an equivalent control mechanism) at intersections;
- Social rules give larger flocks priority when arriving at intersections;
- As a result of the previous rule, average speed will be a function of flock size. We use this assumption in the experimental framework (section 3).

In this paper we will not present a protocol for intersection control, which is supposed to be like the one described in [2], but alternating flocks instead of individual vehicles. In this paper the focus will be on the negotiation mechanism design, such that it complies with individual rationality. We present specific methods for answering the first two questions mentioned above, that is, how to find partners and how to agree on common travel with partners.

2.1 Joint Travelling: The “Bone” Diagram

Let us analyse the restricted situation in which two vehicles could travel either alone or together. As noted earlier, the advantage of travelling together is that the intersection control gives preference to groups over individuals, so that average speed is greater when travelling in a group of at least two cars.

This restriction to two vehicles, besides being for simplification, is related to an idea of initiating flocks with just two members and later populating them with additional members. The viability of this approach is supported by observation of nature: birds usually start flocks with only two birds, which communicate with complex visual and acoustic patterns [9]. Duet-singing birds produce complex antiphonal and polyphonic music [10].

Two partner candidates search for a common point (meeting point). From it they travel together until they reach a certain point (splitting point) from which they no longer benefit from travelling together. From this last point they travel on their own.

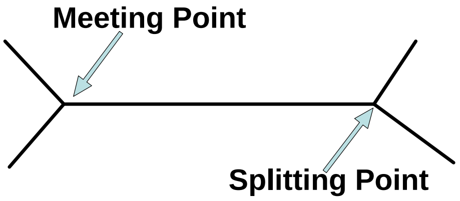


Figure 1: Bone diagram.

The plot for their complete trip looks like a "bone" structure (figure 1). The "bone structure" establishes the basis for negotiation among agents and reduces the flock formation problem to a searching problem.

The steps for deciding whether or not to travel together, and if yes, calculating a joint trip using the "bone structure", are presented in the following algorithm, where we assume that the speed of cars travelling alone is one block per unit time:

```
Input: Two potential travel partners $a_1$ and $a_2$ with starting and ending points $s_i$ and $e_i$ respectively for $i = 1, 2$.
1.  For each agent $a_i$, calculate the distance $d_i$ from the starting point to the ending point as a Manhattan distance.
2.  Find the closest equidistant intersection to the starting points, which will become the meeting point m. Distances from starting points to m will be $d_i (s_i)$ for $i = 1, 2$.
3.  Do the same for the ending points, which will be the splitting point $s$. Distances from ending points to $s$ will be $d_i (e_i)$ for $i = 1, 2$.
4.  Calculate the distance from $m$ to $s$, which is the joint travelling distance $j$.
5.  Make $j: = j/k$, where $k$ is a "speeding factor", which is a social benefit from travelling together imposed by the environment.
6.  Each agent calculates the total travelling time if they agree to travel together
    $ti = d_i (s_i)) + j + d_i (e_i)$
7.  if $ti <= di, i = 1, 2$ then
8.      Agents agree to travel together
9.  else
10.     Agents will travel alone as there is no benefit in joining together
11. end if
```

2.2 Travelling Partner Candidates

Finding appropriate travelling partners can be a complex problem as populations increase. We investigated simple ways of limiting the number of possible partners with whom it could be beneficial to negotiate for an eventual trip together.

One possible way is to find an upper limit in terms of the starting and ending compatibility points of two given possible partners. Of course, the closer their starting and also their ending points, the greater the possibility of a beneficial joint trip. We formalise these ideas in the following.

Assume a simple model of one lane, one direction (figure 2).

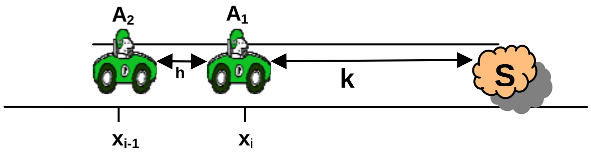


Figure 2: One lane.

Agent A_1 is at position x_i and A_2 is at position x_{i-1}. h represents the distance between them and k represents the distance to a hypothetical splitting point S. (i.e. because this is a one dimensional analysis S must be the goal for at least one of the agents).

Individually, agent A_1 must travel a distance of k units in order to reach its goal. Without losing generality, assume S is the goal for $A_{1;}$ Suppose constant speeds (1 distance unit / 1 time unit). k will also represent the time for A_1 to reach its goal.

Now, assume there is a second agent A_2 who wants to reach the same point S. A_2 is at a distance of h units from A_1.

Assume there is a social bonus ($b>0$) for travelling together, which means that the "system" allows the joint travellers to increase speed by a factor b; this is the selfish benefit of travelling together.

Let us review some cases:

a) Case 1: Agent A_1 will wait for A_2 to be at $x_{i.}$ The travelling time for agent A_1 is now $h + k/b$, so A_1 agrees to wait so long as the following inequality remains true:

$$h + \left[\frac{k}{b}\right] \leq k \qquad (1)$$

b) Case 2: If two directions are allowed, then agent A_1 moves to the left in order to meet agent A_2 earlier. Because we are assuming constant speed, they will reach the geometrical centre at $h/2$. The new agreement point will comply with:

$$\frac{h}{2} + \left[\frac{k + \frac{h}{2}}{b}\right] \leq k \qquad (2)$$

There fore,

$$h \leq 2k \frac{[b-1]}{[b+1]} \qquad (3)$$

Thus, cooperation is convenient for agent A$_1$ if the distance to agent A$_2$ is not greater than 2k *[b-1] / [b+1]*. A similar analysis must be done for agent A$_2$.

Now assuming a more complex situation, both agents are in the cartesian plane $\mathbf{R}^2$ (figure 3). They travel with constant velocity and they can have the social bonus if they decide to cooperate.

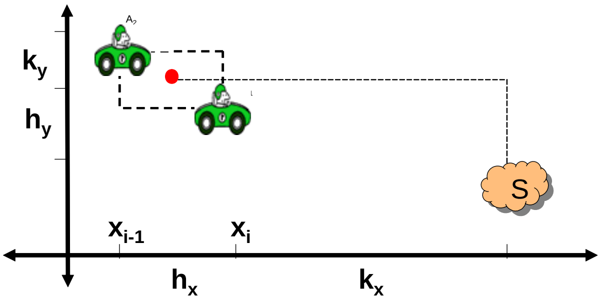


Figure 3: Two agents in the Cartesian plane.

If both agents agree to meet at the intermediate point, then the agreement will be based on:

$$\frac{h_x}{2}+\frac{h_y}{2}+\left[\frac{k_x+\frac{h_x}{2}+k_y+\frac{h_y}{2}}{b}\right]\leq k_x+k_y$$

$$(h_x+h_y)\leq 2(k_x+k_y)\frac{[b-1]}{[b+1]} \tag{5}$$

Again agent A1 will agree if the Manhattan distance is no greater than $2(k_x+k_y)\frac{[b-1]}{[b+1]}$.

In the experimental framework (section 3), if both agents agree to cooperate we will set the social bonus to b=2 in order to double the agents' velocities. In addition, we will decide that hx+hy be at most five blocks.

2.3 Negotiation Method

In this section we describe how two agents evaluate the convenience of travelling together, using the standard notion of individual rationality mentioned above.

Each agent knows their own initial and ending position and calculates the time that it will take to reach their goal individually.

Then they start to broadcast for partners in a fixed region. The *n* agents in the region share their information and for each pair a bone structure is constructed, from which they can calculate the new time taking into consideration the social bonus.

In order to address the question of how to find appropriate travel partners, each agent compares its individual time against the cooperation time for travelling with a partner.

A timetable matrix is formed as follows: when individual time is better than joint time, mark with a zero the intersection of row *i* and column *j* that represents a null relation between agents A_i and A_j.

Otherwise write at the intersection *i,j* the complete travel time that agent A_i would take travelling with agent A_j (i.e. individual time to reach the meeting point + time travelling together with their respective social bonus along bone-structure + individual time to reach their target).

Now, suppose that agent A_i finds several candidates for cooperation. The agent must then search in the timetable matrix which agent will be best to deal with (i.e. which one gives the minimum overall travel time).

Remember that two strategies s_i and s_j are in Nash equilibrium if neither partners has an incentive to choose another move than the agreed one [8].

Suppose agent A_i finds that cooperating with agent A_j will give the minimum overall time. The next step is to ask if agent A_i also gives the minimum global time for Agent A_j. If it is true, we find a Nash equilibrium strategy. So, they must travel together. The agreement between these two agents makes a virtual agent that is open for more partners, but is not allowed to change their common bone structure.

Now suppose that agent A_j still represents the best strategy for agent A_i but the converse is not true. Agent A_i is then placed in a special set called a Pareto set, and stays in it waiting to know if at the end of the first round A_j finds a partner with Nash equilibrium; if this is true, then agent A_i will recalculate its travel time knowing that it must agree with the bone structure of the other two agents (negotiation with virtual agents as in [11]).

Given a set of alternative allocations and a set of individuals, a movement from one allocation to another that can make at least one individual better off, without making any other individual worse off, is called a Pareto improvement or Pareto optimisation [8].

If agent A_i finds that the new bone structure for the others two agents is not better than travelling alone, it can search again in the table for the second best partner. If in the end it cannot find any partner it must travel alone without the social bonus.

With this procedure we can also address the second question, where travelling partners should join and split. The answer depends on the negotiation. If they are in a Nash equilibrium they will travel the whole bone-structure together, but if they are in the Pareto set they can share the bone structure while they find new partners trying to reach a Nash equilibrium state with other agents.

With this approach, individual agents can drop the flock either because they want to or because they have a problem (engine, tyre, etc). So flocks become greedy in that if one member abandons the flock, the other members still want to have the social bonus.

3 Experimental Framework

In this section, we evaluate the performance of our flocking model. There could be several performance measures such as fuel consumption or pollution emissions, but for the moment, the only performance figure will be the time saved in a trip from a starting to an ending point.

Our experiments are now settled in this way: N agents will be generated randomly with uniform probability but limited to an area of five blocks. Their targets are generated randomly around the whole city (20 x 20 blocks). For each experiment, the simulator generates starting points and target points for each agent.

Individual agents must then calculate their own individual path from starting to ending point using A* algorithm and using a traditional Manhattan distance as a metric.

The result will be an estimated arrival time. Agents then calculate hypothetical meeting and splitting points; they are hypothetical because they have not decided yet to travel together, they are just evaluating the possibility. Finally, travel times are calculated according to the algorithm in section 2.1.

We assume in this report that agents travel at the same speed. Agents receive a cooperation proposal and based on their estimated arrival time they choose to cooperate or not.

4 Experimental Results

In order to test these ideas, we ran simulations varying the number of agents in the negotiation process and looking for: a) the total time saved, b) its variance as a measure of dispersion and c) the computational time it takes to calculate it.

Each experiment was replicated one hundred times. As explained in section 2.1, agents were generated uniformly random in a 5x5 block area and targets were generated in a 20x20 city sector.

The total time savings are shown in table 1.

Table 1: Total time savings.

Number of Agents	Saved Time (ST)	Variance of ST	Computational Time (CT)	Variance of the CT	Maximun ST	minimum ST
2	3.17	34.7688	0.0149	0.0004	20	0
5	12.29	44.1423	0.0146	0.0003	30	0
10	24.175	78.3251	0.0222	0.0004	51	6
25	42.15	141.1389	0.0769	0.0003	80	19
50	64.705	252.1217	0.2489	0.0005	119	37
75	85.41	319.613	0.5258	0.0006	138	48.5
100	100.915	398.374	0.9116	0.0005	144.5	44.5
250	204.13	1320.3	5.4622	0.00001	308.5	130.5
500	366.575	2172.4	21.7039	0.00001	468.5	264

Looking at the graphs (figures 4 and 5) and analysing table 1, it seems that our model works poorly when there are few vehicles (1-5) with which to negotiate.

The performance of the model improves when we work with more than 10 vehicles. But when we reach more than 100 vehicles the computational time takes more than a second (on a Centrino/Windows XP/Matlab 7.0 PC, 1.10 GHz), which can be seen as a hazardous procedure because of the velocity involved (one vehicle at 100 km/hr travels almost 28 m ≈ ¼ block in one second).

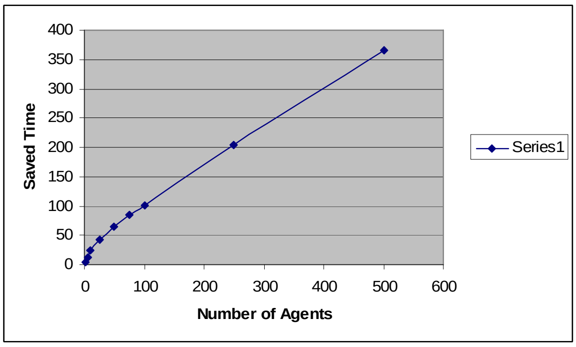


Figure 4: Number of agents and total saved time.

If the time units were measured in minutes we could see an important benefit for vehicles travelling in flocks. Because of the city geometry and the Manhattan metric, several paths can be taken with the same performance (the starting point and the target point form a convex hull, and any path in it gives the same benefit) and this gives us some freedom to avoid collisions.

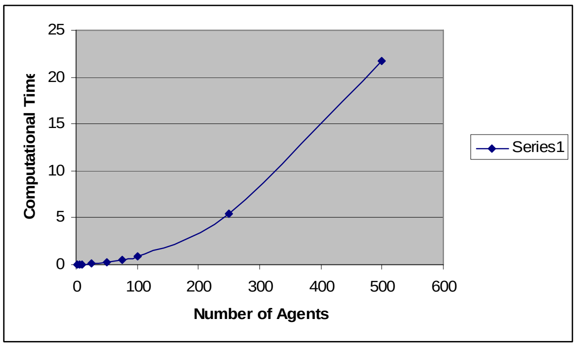


Figure 5: Number of agents and total computational time.

An extra set of 100 experiments were done with 100 agents in a larger city of 105x105 blocks. Vehicles were generated uniformly in the 5x5 blocks area at the centre of the city. The results were quite similar (measured as the individual average profit). On average 216 time units were saved with a variance of 1398. The maximum time saved was 330.5 unit times and the minimum was 141 unit times; the computational time was 0.8961 sec.

From the experiments presented here, we can see that flock navigation can produce substantial travelling time savings.

5 Discussion and Related Work

Our experimental results show that navigation in flocks could reduce travel times for users assuming a traffic control system that promotes group travelling over individual travelling. This is of course early experimentation. We think that in a reasonably big city the odds for finding travel partners increase, compared to small cities. But it is precisely in big cities that traffic problems are worse, so the number of vehicles accounts to the positive side here.

The application of multiagent technology in traffic problems is not new; several works have been done on this [12, 1, 13, 14], many of them related to traffic lights synchronisation.

Recently, work such as Dresner and Stone [2] has shown that it is possible to manage traffic without a traffic light, at least in one intersection. But we think Dresner and Stone's proposal implies a very dangerous scenario where traffic is interleaved. Indeed, even assuming their protocol has no flaws, the least mechanical or other malfunction would mean terrible accidents. In our flock navigation proposal traffic lanes are not interleaved, but flocks simply alternate, which is of course less prone to accidents.

6 Conclusions and Future Work

Our simulations indicate that flock navigation of autonomous vehicles could substantially save time for users and make traffic flow faster. We claim that multiagent coordinated flock navigation is a completely original proposal.

Future work includes extending flocks to arbitrary sizes in experiments, as well as calculating more measures in experiments, such as robustness, social benefits and others. It would be possible to characterise in precise ways parameters such as the relative speed of cars (assuming that lower average relative speeds with respect to the closest cars means lower risk) or even to simulate the malfunctions themselves.

7 References

[1] T.D. Wang and C. Fyfe, "Traffic jams: an evolutionary investigation", *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 04)*, Beijing, pp 381-384 (2004).

[2] K. Dresner and P. Stone, "Multiagent traffic management: an improved intersection control mechanism", *Proceedings 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, Utrecht, ACM Press, New York,, pp 471-477 (2005).

[3] Federal Highway Administration, "Traffic congestion and reliability, linking solutions to problems", Final Report 2003, http://www.ops.fhwa.dot.gov/congestion_report_04/, visited on 1/3/2005.

[4] D. Schrank and T. Lomax, "Annual urban mobility report", http://www.pittsburghregion.org/public/cfm/library/reports/2003UrbanMobilityStudy.pdf, visited on 15/4/2005.

[5] S. Biswas, R. Tatchikou and F. Dion, "Vehicle-to-vehicle wireless communication protocol for enhancing highway traffic safety", *IEEE Communications Magazin*e, 44(1), pp 74-82 (2006).

[6] L. Wischholf, A. Ebner and H. Rohling, "Self-organizing traffic information system based on car-to-car communication", *1st International Workshop on Intelligent Transportation (WIT 2004)*, Hamburg, pp 49-53 (2004).

[7] C.E. Taylor and D.R. Jefferson, "Artificial life as a tool for biological inquiry", *Artificial Life*, 1(1-2), pp 1-14 (1994).

[8] M. Wooldridge and M.J. Wooldridge, *Introduction to Multiagent Systems*, John Wiley and Sons, New York (2002).

[9] G.V.N. Powell, "Structure and dynamics of inter-specific flocks in a neotropical mid-elevation forest", *The Auk*, 96(2), pp 375-390 (1979).

[10] K. Lorenz, *Studies in Animal and Human Behavior*, *Vol II*, Harvard University Press, Cambridge, MA (1971).

[11] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory", *IEEE Transactions on Automatic Control*, 51(3), pp 401-420 (2006).

[12] V.R. Tomas and L.A. Garcia, "A cooperative multiagent system for traffic management and control", *Proceedings 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, Utrecht, ACM Press, New York, pp 52-59 (2005).

[13] M. Wiering, J. van Veenen, J. Vreeken and A. Koopman, *Intelligent Traffic Light Control*, Technical Report UU-CS-2004-029, Institute of Information and Computing Sciences, Utrecht University (2004).

[14] D. de Oliveira, A.L.C. Bazzan and V. Lesser, "Using cooperative mediation to coordinate traffic lights: a case study", *Proceedings 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, Utrecht, ACM Press, New York, pp 463-470 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

University Robotics Education with Fabrication Experiences of Simple Six-axis Biped Robot

Yoshihiko Takahashi, Hirofumi Takagi, Yoshiharu Uchiyama, and Takumi Takashima
Department of System Design Engineering
Kanagawa Institute of Technology, Atsugi, Kanagawa, Japan
ytaka@sd.kanagawa-it.ac.jp

Abstract

We have developed a simple six-axis biped robot system to be used as teaching material for undergraduate university students wishing to study an intelligent robot. This system enables students to fabricate the entire robot system by themselves as they understand each individual step and study level from mechanical design to programming languages. The components used to fabricate the robot system are very affordable allowing more students access to the project. This paper will present the details of the step-by-step study scheme using basic circuit theory, the six-axis biped robot, and lectures.

Keywords: teaching materials, undergraduate university students, biped robot, step-by-step study scheme, mechanical design, programming languages

1 Introduction

Many university students studying engineering courses wish to fabricate an intelligent robot. It is very difficult, however, to fabricate such an intelligent robot for some undergraduate students, who have not yet completed key subjects such as mechanical design, electrical circuit design, control systems and programming languages. Undergraduate students do not have many development opportunities. It is therefore very important to prepare an opportunity to fabricate an intelligent robot for every student who wishes to develop an intelligent robot.

Many researchers have developed teaching materials of robot fabrication. The line trace robot, for example, was proposed by the Shibaura Institute of Technology [1] and Tokai University [2]. This is a system which enables students to fabricate an entire robot system from mechanism to programme. These systems are very simple and often used as teaching material for children [3]. We believe the line trace robot is suitable for lower level study, but the movement and mechanisms are much too simple for students wishing to study an intelligent robot. Researchers have reported that university students in engineering courses are more interested in studying a humanoid type biped robot [4, 5].

We therefore propose humanoid type biped robots as teaching materials for university students. The University of Tokyo is currently using a very expensive humanoid robot, the HRP2 [6], and The Shibaura Institute of Technology and ZMP Corporation have proposed the e-nuvo which is less expensive than the HRP2. Both of these robot systems have their own control software and hardware systems, and students are able to study intelligent, multi-degree of freedom (DOF) movement, however they can only use a standard control programme, not one student may design one of their own. Another downside of these models is that they do not allow the student to study mechanical or electrical circuit design.

The purpose of our research is to provide this intelligent robot system as teaching material which will allow students to fabricate the whole robot system from mechanism to software by themselves.

We presented a simple six-axis biped robot as a teaching material in which a university student can study robotics with step-by-step fabrication experiences [7, 8].

This paper presents a newly developed basic study circuit and an improvement of the six-axis biped robot to ensure and ease the step-by-step study and fabrication. Finally, we introduce the atmosphere of a university lecture in the process.

2 Target Components of Proposed Simple Six-axis Biped Robot

This section individually explains the details of the target components of our robot system.

1. The target robot system is a humanoid type biped robot with six motors.

Many students in engineering courses wish to develop a multi-DOF intelligent robot [4, 5]. A humanoid biped robot is an example of a multi-DOF intelligent robot. A student can perform intelligent motion by using a complete humanoid robot, however the humanoid type biped robot in our research uses only six motors in order to reduce the fabrication time and cost. The intelligent motion in this case implies the computer controlled multi-DOF motion using a multi-DOF robot. Six axes are the minimum required for a robot to walk a straight line. Once a student completes the six-axis system, they may proceed to building a 12-axis system in which the robot can turn left or right.

2. Students may fabricate the entire robot system by themselves: the mechanism, the control circuit, the software programme, etc.

It is essential to fabricate a whole robot system to understand robotics [1, 2, 3, 4, 6]. By using the proposed simplified humanoid biped robot, a student conducts the mechanical design using purchased geared toy motors and aluminium plates, and conducts the control circuit design using a PIC microcomputer and a motor driver. The control software will be programmed using C, the standard programming language for robot controllers.

3. Students perform their fabrication confirming their comprehension level at each step focusing on essential items and key subjects.

A student may study and fabricate each item in steps since the fabrication process is divided into small items. A student is required to study many key subjects to fabricate an intelligent robot including mechanical design, electrical circuits, control systems and software programming. The study of the key subjects has generally not been completed by undergraduates; therefore, students study only essential items necessary for our robot system. Feedback control is, for example, conducted using the output of a potentiometer in order to understand control systems.

4. Key components are affordable.

Among the most important factors for this teaching material is the fabrication cost. When using an expensive robot, students are unable to overhaul the mechanism, or many students have to use one robot system. With affordable parts, students are able to purchase all parts, and fabricate their robot systems individually. This also allows students to overhaul their own robot systems.

3 Basic Study by Basic Study Circuit

The basic study circuit is designed to ensure the step-by-step study scheme. Figure 1 illustrates a scheme using the basic study circuit in which a student studies: LED lighting control (step 1), motor ON/OFF control (step 2), A/D input (step 3), motor feedback control and communication between two PICs (step 4).

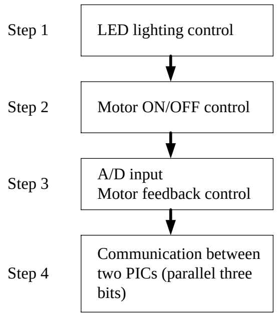


Figure 1: Study scheme of basic study circuit.

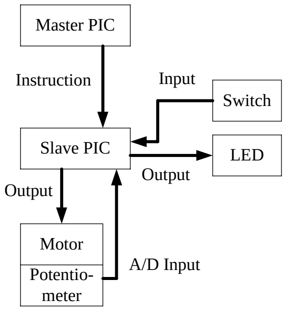


Figure 2: Control system of basic study circuit.

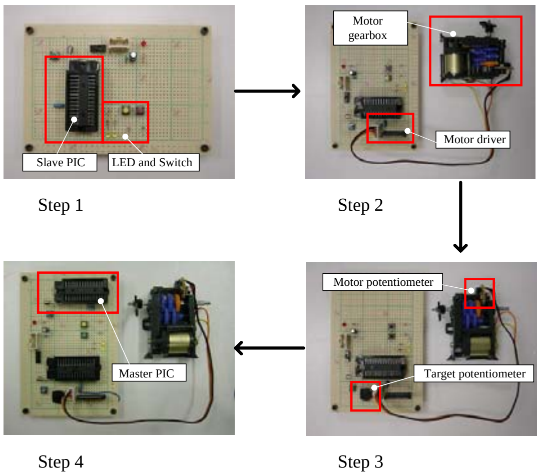


Figure 3: Fabrication sequence of basic study circuit.

Figure 2 shows the control system of the basic study circuit. A master PIC sends instructions to a slave PIC, which then receives the instruction. The slave PIC inputs switch information, and outputs signals to an

LED circuit. The slave PIC controls a motor system by outputting a signal to a motor, and by inputting an analogue signal from a potentiometer. Figure 3 demonstrates the fabrication sequence of the basic study circuit. A student fabricates the basic study circuit from step 1 to step 4 by adding necessary parts. In step 2, a motor gearbox and a motor are added. In step 3, two potentiometers are added. In step 4, a master PIC is added.

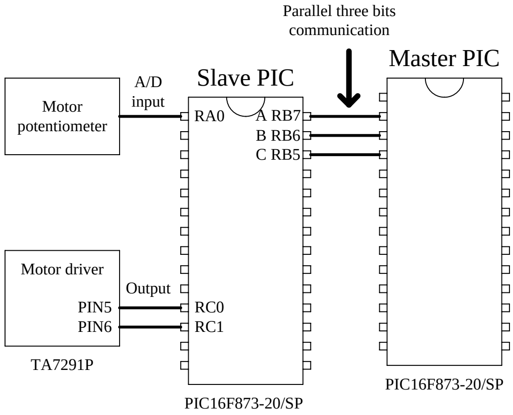


Figure 4: Communication scheme between the master and slave PICs.

Figure 4 shows the communication scheme between the master and slave PICs. Here, very simple parallel three-bit communication is proposed. Table 1 shows the three-bit communication in which slave input values are seven patterns from "000" through "111". The slave PIC controls the motor system depending on the slave input values. For example, the motor angle is controlled to 0 [deg], the centre position, when the slave input value is "000". Figure 5 shows the experimental results of motor control using the basic study circuit.

Table 1: Three-bit communication

	Slave input A B C	Desired value (max 1023)	Desired voltage[V]	Motor angle[deg]
(0)	0 0 0	500	2.44	0
(1)	0 1 1	350	1.71	-33
(2)	0 1 0	400	1.96	-22
(3)	0 0 1	450	2.20	-11
(4)	0 0 0	500	2.44	0
(5)	1 0 1	550	2.69	11
(6)	1 1 0	600	2.93	22
(7)	1 1 1	650	3.18	33

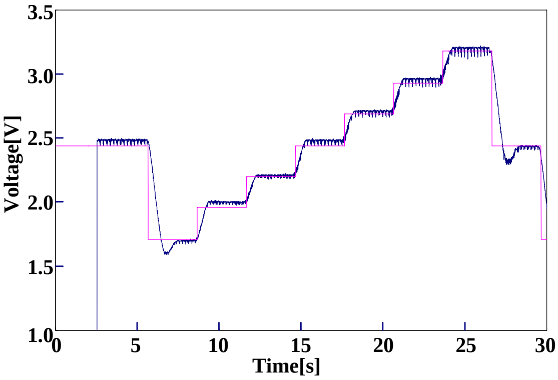


Figure 5: Experimental results of motor control.

4 Fabrication of Six-Axis Biped Robot

Students may proceed in their study to assembly of the six-axis robot after fabricating the basic study circuit. Figure 6 shows the control system of the six-axis biped robot. The control system utilises seven PICs where one PIC is used as the master PIC, and other six PICs are used as the slave PICs. One slave PIC controls one axis. The communication between the master PIC and the slave PICs, and the each axis control scheme is the same as the basic study circuit.

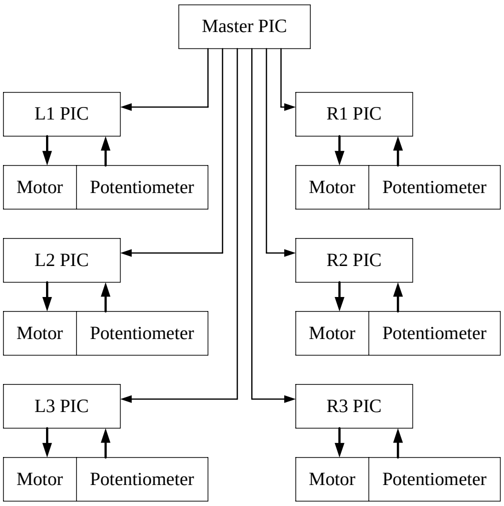
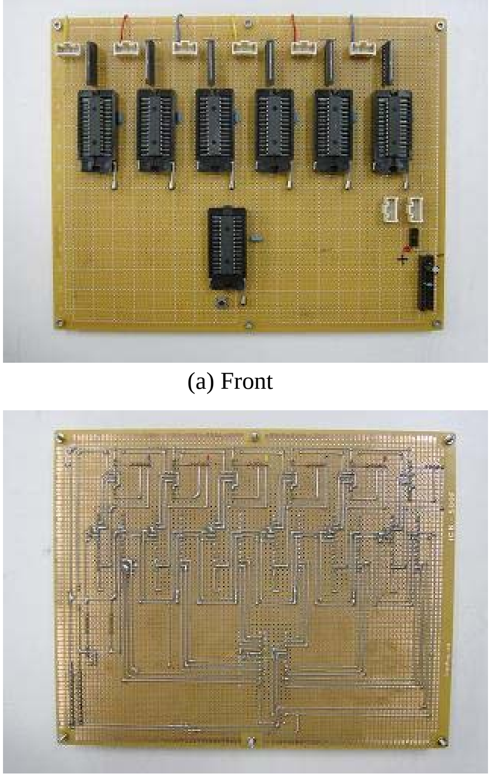


Figure 6: Control system of six-axis biped robot.

Figure 7 shows the fabricated control circuit of the six-axis biped robot. It is very easy to fabricate and check the circuit because all parts are arranged two dimensionally. Figure 8 shows the fabricated mechanism of the six-axis biped robot.

As affordability is one of the most important factors for the teaching material, affordable components are used in our robot. The following is the parts list for the fabrication of a robot mechanism and a control circuit.

1. Geared DC motor (6-speed gearbox H.E., TAMIYA); 2. PIC microcomputer (PIC16F873, Microchip Technology); 3. potentiometer (RDC501006A, gear type, ALPS Electric); 4. motor driver (TA7291P, Toshiba); 5. miscellaneous. The total fabrication cost of one robot system is less than US$250. Students are able to purchase all of the components by themselves.



(a) Front

(b) Back

Figure 7: Fabricated control circuit of six-axis biped robot.

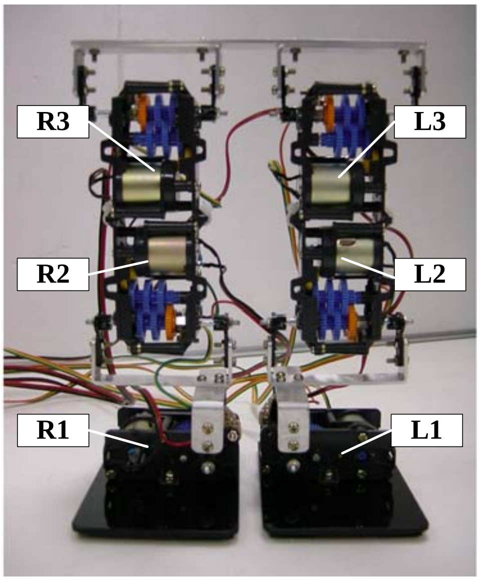


Figure 8: Fabricated mechanism of six-axis biped robot.

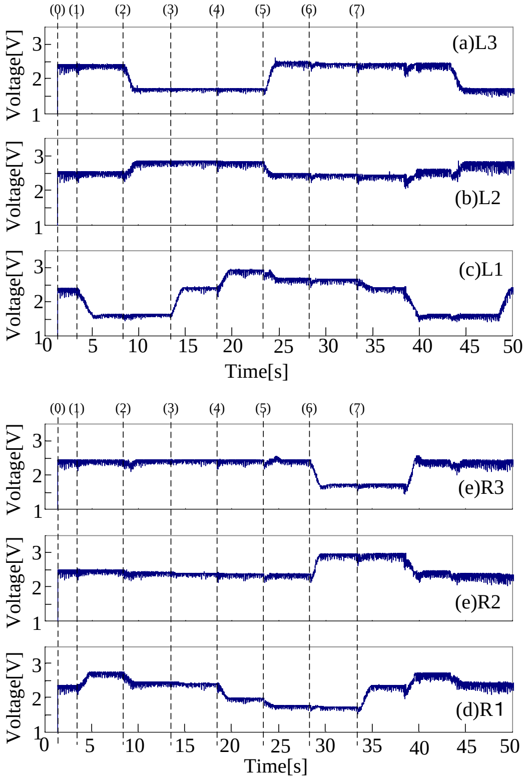


Figure 9: Experimental results of walking motion.

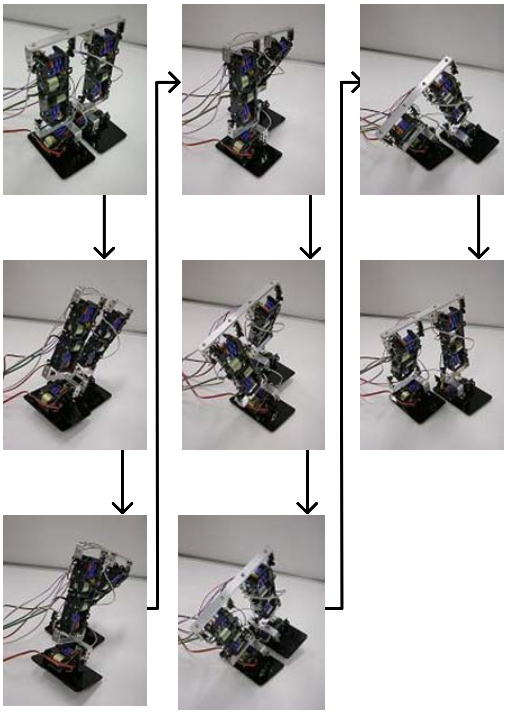


Figure 10: Photographs of walking motion.

Figure 9 illustrates the experimental results of walking motion of the six-axis biped robot, where in 0 the robot is standing upright, then it conducts one step walking (1 through 7). Figure 10 shows the photographs of the walking motion.

5 Lecture and Robot Contest using Six-Axis Biped Robot by Undergraduate University Students

The six-axis biped robot was used and evaluated in the "System Design Project" course in the Department of System Design Engineering at the Kanagawa Institute of Technology. There were 25 students in the course. They were all able to fabricate the robot system, and they all felt that they benefited from the project in their understanding of robot systems and robotics.

Figure 11: Photographs of robot contest.

A robot contest was also conducted. Figure 11 shows photographs from the robot contest. Students were evaluated on walking time along a 20-centimeter course, and their design report. The robot contest was an excellent goal for the students. Student evaluations confirmed that the robot system is suitable as teaching material.

6 Fabrication of Twelve-Axis Robot

Students may proceed their study to assembly of twelve-axis robot after fabricating the six-axis robot. Figure 12 shows the control system with four PICs where one PIC is used as the master PIC, and other three PICs are used as the slave PICs. One slave PIC controls four axes using parallel programming techniques. An I^2C communication scheme is used to communicate between the master PIC and the slave PICs. Figure 13 shows the twelve-axis biped robot.

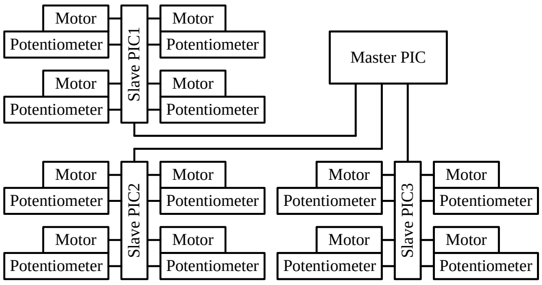


Figure 12: Control system of twelve-axis biped robot.

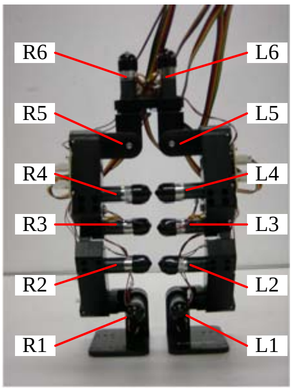


Figure 13: twelve-axis biped robot mechanism.

7 Conclusions

We have designed a small six-axis biped robot system aimed to be used as teaching materials at universities offering robotic studies. The system can be fabricated using affordable components; geared toy motors, PIC microcomputers, etc. The proposed fabrication scheme will facilitate students' understanding of robotics in simple steps.

The robot system was utilised in a course offered by the Department of System Design Engineering at the Kanagawa Institute of Technology and in a robot design contest. Students fabricated an entire robot system from mechanism to software by themselves and felt that they benefited their studies with this project. We are preparing further the next step self study teaching material of a high grade, twelve-axis robot which can turn left or right.

8 References

[1] Y. Tomimasu, M. Tokizawa, Y. Ando, M. Mizukawa, C. Kasuga and Y. Ogawa, "Development of the line trace robot as teaching materials for mechatronics education", *Proceedings of 2004 JSME Conference on*

Robotics and Mechatronics, Nagoya, Japan, 2P1-L2-36 (2004).

[2] N. Komine, K. Inagaki, T. Inaba and R. Matsuda, “Creative design practice: the subject of robot design for the first year students”, *Proceedings of 2004 JSME Conference*, Nagoya, Japan, 2P1-L2-35 (2004).

[3] S. Yura, N. Hiraoka, H. Sogo, S. Fujusawa, K. Shikama, S. Urushihara and H. Tarao, “Kids’ KARAKURI workshop at Takamatsu NCT”, *Proceedings of 2004 JSME Conference*, Nagoya, Japan, 2P2-L2-36 (2004).

[4] O. Yokoyama, M. Mizukawa and R. Sakai, “e-nuvo: development of education materials of bipedal locomotion robot”, *Proceedings of 2004 JSME Conference*, Nagoya, Japan, 2A1-L2-38 (2004).

[5] Y. Nakagawa, I. Kobayashi and K. Watanabe, “Education using small humanoid robot”, *Proceedings of 2004 JSME Conference*, Nagoya, Japan, 2P2-L2-35 (2004).

[6] K. Okada, M. Inaba and H. Inoue, “Mechano-informatics seminar for undergraduate students using the HRP2 humanoid robot”, *Proceedings of 2004 JSME Conference*, Nagoya, Japan, 2P2-L2-39 (2004).

[7] Y. Takahashi and M. Kohda, “Simple humanoid biped robot with PIC microcomputer for university education”, *Journal of Robotics and Mechatronics*, 17(2), pp 226-231 (2005).

[8] Y. Takahashi, “Development of biped walking robot kit for university education”, *Journal of the Robotics Society of Japan*, 24(1), pp 16-19 (2006).

[7] T. Endo, *Wakaru PIC Micon Seigyo*, Seibundo Shinkou Sya, Tokyo (2002).

[8] M. Suzuki, *Tanoshiku Dekiru C&PIC Seigyo Jikken*, Tokyo Denki Daigaku Syuppankyoku, Tokyo (2003).

[9] T. Gokan, *C Gengo ni yoru PIC Programming Nyumon*, Gijutsu Hyoronsya, Tokyo (2002).

Symbolic Formulation of Dynamics of a Six-Wheeled Mobile Robot

Angelo Amato, Alessandro Cammarata and Rosario Sinatra
Dipartimento di Ingegneria Industriale e Meccanica,
Università di Catania, Catania, Italy
acamma@diim.unict.it, rsintra@diim.unict.it

Abstract

This paper investigates the dynamic modelling of a simplified version of the mobile robot M6 [1], i.e., a six wheeled mobile robot used for volcanic exploration and surveying. Using the Natural Orthogonal Complement method, robot dynamics is developed also considering the interactions between the six spherical wheels and terrain, modelled as a warped smooth surface. Same simplifications, as contemporary contact of all the wheels with terrain or pure rolling contact on a unique point for each spherical wheel, are considered. Kinematics is completely solved and a symbolical formulation of the equations of motion is obtained. Finally, a numeric model, developed by Simulink, is implemented to verify the mathematical model.

Keywords: six-wheeled mobile robot, terrain contact, dynamics, NOC method

1 Introduction

Mobile robots assume particular importance in each task that humans can't accomplish due to inaccessibility or danger. These robots can be used for exploring and surveying hostile environments as volcanoes [1], [4] or for planetary exploration, as the rovers Spirit and Opportunity landed on Mars and developed by NASA. Robot motion is often provided to cameras or remote control and usually trajectory planning doesn't include study on dynamics. This lack is principally due to a difficulty in developing dynamic modelling of these nonholonomic systems [5]. Besides, taking into account terrain geometry, makes the study more difficult and computing very burden.

Starting from a previous paper [2], we focus on the symbolic deriving of a six-wheeled mobile robot dynamics by the Natural Orthogonal Complement method. Considering a generic warped smooth surface, terrain is included inside the model by the use of normal vectors passing from the contact points between wheels and surface.

2 Description of the Robot

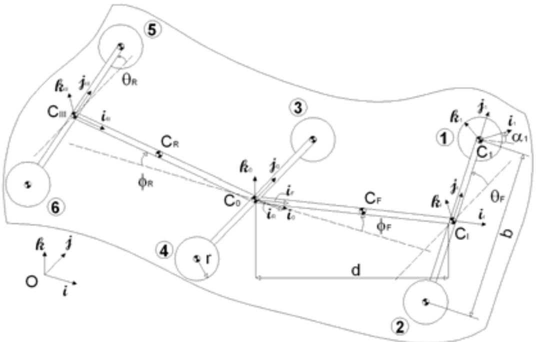


Figure 1: Representation of the mobile robot.

Table 1: Description of symbols.

$[\mathbf{i},\mathbf{j},\mathbf{k}]$	Inertial right-handed triad
$[\mathbf{i}_0,\mathbf{j}_0,\mathbf{k}_0]$	Right-handed triad describing the orientation of the central module
$[\mathbf{i}_F,\mathbf{j}_F,\mathbf{k}_F]$	Right-handed triad describing the orientation of the front axis
$[\mathbf{i}_R,\mathbf{j}_R,\mathbf{k}_R]$	Right-handed triad describing the orientation of the rear axis
$[\mathbf{i}_I,\mathbf{j}_I,\mathbf{k}_I]$	Right-handed triad describing the orientation of the front module
$[\mathbf{i}_{III},\mathbf{j}_{III},\mathbf{k}_{III}]$	Right-handed triad describing the orientation of the rear module
$[\mathbf{i}_j,\mathbf{j}_j,\mathbf{k}_j]$	Right-handed triad describing the orientation of the *j*-th wheel
r	Radius of each wheel
d	Length of the front and rear axes
b	Length of the front and rear modules
ϕ_F,ϕ_R	Pitch angles of the front and rear axes w.r.t. the central module
θ_F,θ_R	Roll angles of the front and rear modules w.r.t. the central module
α_j	Rotation angles of the *j*-th wheel

Figure 1 and table 1 show the robot and its geometrical parameters, respectively. It is the union of three modules, each one carrying two spherical wheels, connected by two axes: the front and the rear axis. Considering the central module, indicated with "zero", as reference for robot positioning and orienting, the remaining ones are located by the angles: $\phi_F, \phi_R, \theta_F, \theta_R$ that indicate, respectively, the pitch angles of the front or rear axis and the roll angles of the front or rear module. The mobile robot is made of eleven rigid bodies and each module has two degrees of freedom relative to its next module. So, in order to find the robot posture, we might calculate the central module pose and, subsequently, orientate the extremity modules with respect to it. Positioning and orienting of the central module are obtained after its mass centre vector $\mathbf{c}_0$ and its Euler-Rodriguez parameters $\mathbf{r}$ and r_0 are assigned [5],

$$\begin{aligned} &\mathbf{r} = \sin\frac{\psi}{2}\hat{\mathbf{e}},\ r_0 = \cos\frac{\psi}{2}; \\ &\mathbf{r}^T\mathbf{r} + r_0^2 = 1 \end{aligned} \tag{1}$$

where $\hat{\mathbf{e}}$ is the unit vector indicating the rotation axis and ψ is its associated rotation angle. Joining the angles $\phi_F, \phi_R, \theta_F, \theta_R$ and α_i, $i = 1,...,6$, the wheels rotation angles, the complete posture of the robot is obtained. The pose vector $\boldsymbol{\eta}$, defining position and orientation of the mobile robot, is so defined:

$$\boldsymbol{\eta} = (\mathbf{c}_0^T, \mathbf{r}^T, r_0, \phi_F, \phi_R, \theta_F, \theta_R, \alpha_1, ..., \alpha_6)^T \tag{2}$$

where, using (1), sixteen parameters upon seventeen are independent.

2.1 Computation of the Posture of the Robot

The Euler-Rodriguez parameters $\mathbf{r}$ and r_0 are used to define the rotation matrix $\mathbf{Q}_0$:

$$\mathbf{Q}_0 = [\mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0] = (2r_0^2 - 1)\mathbf{1} + 2\mathbf{r}\mathbf{r}^T + 2r_0\mathbf{R} \tag{3}$$

where $\mathbf{R}$ is the cross-product matrix (CPM) of vector $\mathbf{r}$ [5].

The two rotation matrices $\mathbf{Q}_F^0$ and $\mathbf{Q}_R^0$ express orientation of the front and rear axes with respect to (w.r.t.) the central module (figure 1):

$$\begin{aligned} \mathbf{Q}_F^0 &= \mathbf{j}_0\mathbf{j}_0{}^T + \cos\phi_F(\mathbf{1} - \mathbf{j}_0\mathbf{j}_0{}^T) + \sin\phi_F\mathbf{J}_0 \\ \mathbf{Q}_R^0 &= \mathbf{j}_0\mathbf{j}_0{}^T + \cos\phi_R(\mathbf{1} - \mathbf{j}_0\mathbf{j}_0{}^T) + \sin\phi_R\mathbf{J}_0 \end{aligned} \tag{4}$$

where $\mathbf{J}_0$ is the CPM of the unit vector $\mathbf{j}_0$. The rotation matrices $\mathbf{Q}_F$, $\mathbf{Q}_R$ are

$$\begin{aligned} \mathbf{Q}_F &= [\mathbf{i}_F, \mathbf{j}_F, \mathbf{k}_F] = \mathbf{Q}_0\mathbf{Q}_F^0 \\ \mathbf{Q}_R &= [\mathbf{i}_R, \mathbf{j}_R, \mathbf{k}_R] = \mathbf{Q}_0\mathbf{Q}_R^0 \end{aligned} \tag{5}$$

The mass centre vector $\mathbf{c}_F$ and $\mathbf{c}_R$ are easily defined once the vector $\mathbf{c}_0$ is given, i.e.

$$\begin{aligned} \mathbf{c}_F &= \mathbf{c}_0 + (d/2)\mathbf{i}_F \\ \mathbf{c}_R &= \mathbf{c}_0 - (d/2)\mathbf{i}_R \end{aligned} \tag{6}$$

Analogously, for the extremity modules we can derive,

$$\begin{aligned} \mathbf{Q}_I^F &= \mathbf{i}_F\mathbf{i}_F{}^T + \cos\theta_F(\mathbf{1} - \mathbf{i}_F\mathbf{i}_F{}^T) + \sin\theta_F\mathbf{I}_F \\ \mathbf{Q}_{III}^R &= \mathbf{i}_R\mathbf{i}_R{}^T + \cos\theta_R(\mathbf{1} - \mathbf{i}_R\mathbf{i}_R{}^T) + \sin\theta_R\mathbf{I}_R \end{aligned} \tag{7}$$

where $\mathbf{I}_F$ and $\mathbf{I}_R$ are the CPMs of the unit vectors $\mathbf{i}_F$ and $\mathbf{i}_R$, and then,

$$\begin{aligned} \mathbf{Q}_I &= [\mathbf{i}_I, \mathbf{j}_I, \mathbf{k}_I] = \mathbf{Q}_F\mathbf{Q}_I^F \\ \mathbf{Q}_{III} &= [\mathbf{i}_{III}, \mathbf{j}_{III}, \mathbf{k}_{III}] = \mathbf{Q}_R\mathbf{Q}_{III}^R \end{aligned} \tag{8}$$

The position vectors of the mass centres $\mathbf{c}_I$ and $\mathbf{c}_{III}$ are

$$\begin{aligned} \mathbf{c}_I &= \mathbf{c}_0 + d\,\mathbf{i}_F \\ \mathbf{c}_{III} &= \mathbf{c}_0 - d\,\mathbf{i}_R \end{aligned} \tag{9}$$

Finally, the orientation of each wheel is calculated. The first two wheels are joined to the front module, so, introducing the rotation matrices $\mathbf{Q}_1^I$ and $\mathbf{Q}_2^I$ expressing their orientation with respect to the front module,

$$\begin{aligned} \mathbf{Q}_1^I &= \mathbf{j}_I\mathbf{j}_I{}^T + \cos\alpha_1(\mathbf{1} - \mathbf{j}_I\mathbf{j}_I{}^T) + \sin\alpha_1\mathbf{J}_I \\ \mathbf{Q}_2^I &= \mathbf{j}_I\mathbf{j}_I{}^T + \cos\alpha_2(\mathbf{1} - \mathbf{j}_I\mathbf{j}_I{}^T) + \sin\alpha_2\mathbf{J}_I \end{aligned} \tag{10}$$

the final rotation matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are easily derived

$$\begin{aligned} \mathbf{Q}_1 &= [\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1] = \mathbf{Q}_I\mathbf{Q}_1^I \\ \mathbf{Q}_2 &= [\mathbf{i}_2, \mathbf{j}_2, \mathbf{k}_2] = \mathbf{Q}_I\mathbf{Q}_2^I \end{aligned} \tag{11}$$

Analogously, the rotation matrices $\mathbf{Q}_3^0$ and $\mathbf{Q}_4^0$ expressing the third and fourth wheel orientation with respect to the central module are

$$\begin{aligned} \mathbf{Q}_3^0 &= \mathbf{j}_0\mathbf{j}_0{}^T + \cos\alpha_3(\mathbf{1} - \mathbf{j}_0\mathbf{j}_0{}^T) + \sin\alpha_3\mathbf{J}_0 \\ \mathbf{Q}_4^0 &= \mathbf{j}_0\mathbf{j}_0{}^T + \cos\alpha_4(\mathbf{1} - \mathbf{j}_0\mathbf{j}_0{}^T) + \sin\alpha_4\mathbf{J}_0 \end{aligned} \tag{12}$$

Then the rotation matrices $\mathbf{Q}_3$ and $\mathbf{Q}_4$ become

$$\begin{aligned} \mathbf{Q}_3 &= [\mathbf{i}_3, \mathbf{j}_3, \mathbf{k}_3] = \mathbf{Q}_0\mathbf{Q}_3^0 \\ \mathbf{Q}_4 &= [\mathbf{i}_4, \mathbf{j}_4, \mathbf{k}_4] = \mathbf{Q}_0\mathbf{Q}_4^0 \end{aligned} \tag{13}$$

Analysing the rear module, the rotation matrices $\mathbf{Q}_5^{III}$ and $\mathbf{Q}_6^{III}$ of the fifth and sixth wheel can be derived as

$$\begin{aligned} \mathbf{Q}_5^{III} &= \mathbf{j}_{III}\mathbf{j}_{III}{}^T + \cos\alpha_5(\mathbf{1} - \mathbf{j}_{III}\mathbf{j}_{III}{}^T) + \sin\alpha_5\mathbf{J}_{III} \\ \mathbf{Q}_6^{III} &= \mathbf{j}_{III}\mathbf{j}_{III}{}^T + \cos\alpha_6(\mathbf{1} - \mathbf{j}_{III}\mathbf{j}_{III}{}^T) + \sin\alpha_6\mathbf{J}_{III} \end{aligned} \tag{14}$$

hence, from (13) the rotation matrices $\mathbf{Q}_3$ and $\mathbf{Q}_4$ assume the following expressions

$$\begin{aligned} \mathbf{Q}_5 &= [\mathbf{i}_5, \mathbf{j}_5, \mathbf{k}_5] = \mathbf{Q}_{III}\mathbf{Q}_5^{III} \\ \mathbf{Q}_6 &= [\mathbf{i}_6, \mathbf{j}_6, \mathbf{k}_6] = \mathbf{Q}_{III}\mathbf{Q}_6^{III} \end{aligned} \tag{15}$$

where $\mathbf{J}_I$ and $\mathbf{J}_{III}$ are the CPMs of the unit vectors $\mathbf{j}_I$ and $\mathbf{j}_{III}$, respectively.

Finally, the mass centres vectors of the wheels, fixed at the geometrical centres of each sphere, are derived as

$$\mathbf{c}_1 = \mathbf{c}_I + (b/2)\mathbf{j}_I; \qquad \mathbf{c}_2 = \mathbf{c}_I - (b/2)\mathbf{j}_I; \tag{16}$$

$$\mathbf{c}_3 = \mathbf{c}_0 + (b/2)\mathbf{j}_0; \qquad \mathbf{c}_4 = \mathbf{c}_0 - (b/2)\mathbf{j}_0; \tag{17}$$

$$\mathbf{c}_5 = \mathbf{c}_{III} + (b/2)\mathbf{j}_{III}; \qquad \mathbf{c}_6 = \mathbf{c}_{III} - (b/2)\mathbf{j}_{III} \tag{18}$$

2.2 Unit Normal Vector of the Ground n

Let us consider the ground as a smooth surface that is known analytically in the form [6],

$$z = \mu(x, y) \tag{19}$$

A generic point ground is located by the position vector $\mathbf{p}_i$, i.e.

$$\mathbf{p}_i = [x \quad y \quad \mu(x, y)]^T \tag{20}$$

Following the robot kinematic chains, the vector of (20) becomes,

$$\mathbf{p}_i(\boldsymbol{\eta}) \triangleq \mathbf{c}_i - r\mathbf{n}_i, \quad i = 1,...,6 \tag{21}$$

with $\mathbf{n}_i$ given by

$$\mathbf{n}_i = \left(\frac{\partial \mathbf{p}}{\partial x} \times \frac{\partial \mathbf{p}}{\partial y} \right) \Bigg/ \left\| \frac{\partial \mathbf{p}}{\partial x} \times \frac{\partial \mathbf{p}}{\partial y} \right\| \tag{22}$$

Naturally, (20) is correct if and only if each spherical wheel touches terrain into a unique point.

Besides, supposing that all the wheels are touching terrain at the same time, six kinematic constraints h$_i$ are added as

$$h_i(\boldsymbol{\eta}) \triangleq p_{iz} - \mu(p_{ix}, p_{iy}) = 0 \quad i = 1,...,6 \tag{23}$$

Once the pose of the central module is assigned by $\mathbf{c}_0$, $\mathbf{r}$ and r_0, a reduced nonlinear algebraic system $\mathbf{H}$ is introduced to obtain the unactuated joint angles $\phi_F, \phi_R, \theta_F, \theta_R$, where,

$$\mathbf{H} \triangleq [h_1 \quad h_2 \quad h_5 \quad h_6]^T \tag{24}$$

3 Twist Arrays Definition

The number of generalised independent velocities of the system (seven) doesn't coincide with the number of generalised independent coordinates (sixteen), due to the nonholonomic nature of the robot [5]. It's so possible to introduce a seven dimensional generalised independent velocities array $\mathbf{v}$, so defined:

$$\mathbf{v} = (\omega_{0l}, \dot{\alpha}_1, ..., \dot{\alpha}_6)^T \tag{25}$$

w.r.t. each angular or linear velocity is expressed.

The Natural Orthogonal Complement method requires that each *twist array* $\mathbf{t}_i = [\boldsymbol{\omega}_i^T, \dot{\mathbf{c}}_i^T]^T, i = 1,...,11$, is calculated, where $\boldsymbol{\omega}_i$ and $\dot{\mathbf{c}}_i$ are, respectively, the angular velocity vector and the mass centre velocity vector of each body. Besides, each twist array may be expressed with respect to the generalised independent velocities array $\mathbf{v}$ by the following expressions,

$$\mathbf{t}_i = \mathbf{T}_i \mathbf{v} \tag{26}$$

where $\mathbf{T}_i$ is the 6×7 dimensional *Twist Shaping Matrix* of each body.

The twist array $\mathbf{t}_0$ is so defined:

$$\mathbf{t}_0 = [\boldsymbol{\omega}_0^T, \dot{\mathbf{c}}_0^T]^T \tag{27}$$

Analysing the central module, the expression of the angular velocity vector $\boldsymbol{\omega}_0$ is derived

$$\boldsymbol{\omega}_0 = 2r_0\dot{\mathbf{r}} - 2\dot{r}_0\mathbf{r} + 2\mathbf{R}\dot{\mathbf{r}} \tag{28}$$

This vector can be decomposed, to facilitate calculation, into two components: one along and one normal to the unit vector $\mathbf{j}_0$, i.e.

$$\boldsymbol{\omega}_0 = \omega_{0l}\mathbf{j}_0 + \boldsymbol{\omega}_{0\perp} \tag{29}$$

Using (28), the angular velocity vectors of the third and fourth wheel are

$$\boldsymbol{\omega}_3 = \boldsymbol{\omega}_0 + \dot{\alpha}_3\mathbf{j}_0; \quad \boldsymbol{\omega}_4 = \boldsymbol{\omega}_0 + \dot{\alpha}_4\mathbf{j}_0 \tag{30}$$

Besides, the velocities of the mass centres of these wheels are obtained deriving w.r.t. time (17)

$$\dot{\mathbf{c}}_3 = \dot{\mathbf{c}}_0 + (b/2)\boldsymbol{\omega}_0 \times \mathbf{j}_0; \quad \dot{\mathbf{c}}_4 = \dot{\mathbf{c}}_0 - (b/2)\boldsymbol{\omega}_0 \times \mathbf{j}_0 \tag{31}$$

We assume a pure-rolling contact among wheel and terrain. Besides, introducing the unit vector $\mathbf{n}_i$ normal to the surface at the contact point of the *i*-th wheel with terrain, the vectors $\dot{\mathbf{c}}_i$, $i = 1,...,6$, may be written as

$$\dot{\mathbf{c}}_i = r\boldsymbol{\omega}_i \times \mathbf{n}_i \quad i = 1,...,6 \tag{32}$$

Subtracting (31) sidewise:

$$\dot{\mathbf{c}}_3 - \dot{\mathbf{c}}_4 = b\,\boldsymbol{\omega}_0 \times \mathbf{j}_0 \tag{33}$$

and substituting (32), (30) and (29) into (33) it is possible to compute $\boldsymbol{\omega}_{0\perp}$ [2], i.e.,

$$\begin{aligned} \boldsymbol{\omega}_{0\perp} &= \frac{1}{D_w}(\mathbf{1} - \mathbf{j}_0\mathbf{j}_0^T)[\dot{\alpha}_3\mathbf{n}_3 - \dot{\alpha}_4\mathbf{n}_4 + ... \\ &...+ \omega_{0l}(\mathbf{n}_3 - \mathbf{n}_4)]; \\ D_w &= \frac{b}{r} + \mathbf{j}_0^T(\mathbf{n}_3 - \mathbf{n}_4) \end{aligned} \tag{34}$$

At the same time, $\dot{\mathbf{c}}_0$ is computed adding (31), i.e.,

$$\begin{aligned} \dot{\mathbf{c}}_0 &= \frac{\dot{\mathbf{c}}_3 + \dot{\mathbf{c}}_4}{2} = \frac{r}{2}[\boldsymbol{\omega}_0 \times (\mathbf{n}_3 + \mathbf{n}_4) + ... \\ &...+ \mathbf{j}_0 \times (\dot{\alpha}_3\mathbf{n}_3 + \dot{\alpha}_4\mathbf{n}_4)]; \end{aligned} \tag{35}$$

The angular velocity vectors of the front and rear axis are,

$$\boldsymbol{\omega}_F = \boldsymbol{\omega}_0 + \dot{\phi}_F\mathbf{j}_0; \quad \boldsymbol{\omega}_R = \boldsymbol{\omega}_0 + \dot{\phi}_R\mathbf{j}_0 \tag{36}$$

while their mass centre velocities are

$$\begin{aligned}\dot{\mathbf{c}}_F &= \dot{\mathbf{c}}_0 + (d/2)\boldsymbol{\omega}_F \times \mathbf{i}_F;\\ \dot{\mathbf{c}}_R &= \dot{\mathbf{c}}_0 - (d/2)\boldsymbol{\omega}_R \times \mathbf{i}_R\end{aligned} \tag{37}$$

Considering the extremity modules, the angular velocity vectors $\boldsymbol{\omega}_I$ and $\boldsymbol{\omega}_{III}$ are

$$\boldsymbol{\omega}_I = \boldsymbol{\omega}_F + \dot{\theta}_F \mathbf{i}_F; \quad \boldsymbol{\omega}_{III} = \boldsymbol{\omega}_R + \dot{\theta}_R \mathbf{i}_R \tag{38}$$

Now, let us define the velocities $\dot{\mathbf{c}}_I$ and $\dot{\mathbf{c}}_{III}$, obtained deriving w.r.t. (9)

$$\begin{aligned}\dot{\mathbf{c}}_I &= \dot{\mathbf{c}}_0 + d\,\boldsymbol{\omega}_F \times \mathbf{i}_F;\\ \dot{\mathbf{c}}_{III} &= \dot{\mathbf{c}}_0 - d\,\boldsymbol{\omega}_R \times \mathbf{i}_R\end{aligned} \tag{39}$$

Finally, for the wheels of the extremity modules we have

$$\boldsymbol{\omega}_1 = \boldsymbol{\omega}_I + \dot{\alpha}_1 \mathbf{j}_I; \quad \boldsymbol{\omega}_2 = \boldsymbol{\omega}_I + \dot{\alpha}_2 \mathbf{j}_I \tag{40}$$

$$\begin{aligned}\dot{\mathbf{c}}_1 &= \dot{\mathbf{c}}_I + (b/2)\,\boldsymbol{\omega}_I \times \mathbf{j}_I;\\ \dot{\mathbf{c}}_2 &= \dot{\mathbf{c}}_I - (b/2)\,\boldsymbol{\omega}_I \times \mathbf{j}_I\end{aligned} \tag{41}$$

while for the rear module we have

$$\boldsymbol{\omega}_5 = \boldsymbol{\omega}_{III} + \dot{\alpha}_5 \mathbf{j}_{III}; \quad \boldsymbol{\omega}_6 = \boldsymbol{\omega}_{III} + \dot{\alpha}_6 \mathbf{j}_{III} \tag{42}$$

$$\begin{aligned}\dot{\mathbf{c}}_5 &= \dot{\mathbf{c}}_{III} + (b/2)\,\boldsymbol{\omega}_{III} \times \mathbf{j}_{III};\\ \dot{\mathbf{c}}_6 &= \dot{\mathbf{c}}_{III} - (b/2)\,\boldsymbol{\omega}_{III} \times \mathbf{j}_{III}\end{aligned} \tag{43}$$

3.1 Computation of the Unactuated Joint Velocities

In order to calculate the unactuated joint velocities $\dot{\phi}_F, \dot{\phi}_R, \dot{\theta}_F, \dot{\theta}_R$ let us write $\dot{\mathbf{c}}_I$ as

$$\begin{aligned}\dot{\mathbf{c}}_I &= \frac{\dot{\mathbf{c}}_1 + \dot{\mathbf{c}}_2}{2} = \frac{r}{2}[\boldsymbol{\omega}_I \times (\mathbf{n}_1 + \mathbf{n}_2) + \ldots\\ &\ldots + \mathbf{j}_I \times (\dot{\alpha}_1 \mathbf{n}_1 + \dot{\alpha}_2 \mathbf{n}_2)];\end{aligned} \tag{44}$$

obtained adding sidewise (41) and substituting (32) and (40). Comparing (39) and (44) we derive

$$\begin{aligned}&\frac{r}{2}[\boldsymbol{\omega}_0 \times (\mathbf{n}_3 + \mathbf{n}_4) + \mathbf{j}_0 \times (\dot{\alpha}_3 \mathbf{n}_3 + \dot{\alpha}_4 \mathbf{n}_4)] + \ldots\\ &\ldots + d[\boldsymbol{\omega}_0 \times \mathbf{i}_I - \dot{\phi}_F \mathbf{k}_F] = \frac{r}{2}[(\boldsymbol{\omega}_0 + \dot{\phi}_F \mathbf{j}_0 + \ldots\\ &\ldots + \dot{\theta}_F \mathbf{i}_I) \times (\mathbf{n}_1 + \mathbf{n}_2) + \mathbf{j}_I \times (\dot{\alpha}_1 \mathbf{n}_1 + \dot{\alpha}_2 \mathbf{n}_2)\end{aligned} \tag{45}$$

By dot-multiplying both sides of (45) by $\mathbf{j}_0$ or by $\mathbf{i}_F$ we, respectively, calculate $\dot{\theta}_F$ or $\dot{\phi}_F$. After calculations, we obtain

$$\begin{aligned}\dot{\theta}_F &= (\dot{\theta}_{Fl})\,\omega_{0l} + (\dot{\theta}_{F1})\,\dot{\alpha}_1 + \ldots\\ &\ldots + (\dot{\theta}_{F2})\,\dot{\alpha}_2 + (\dot{\theta}_{F3})\,\dot{\alpha}_3 + (\dot{\theta}_{F4})\,\dot{\alpha}_4\end{aligned} \tag{46}$$

and

$$\begin{aligned}\dot{\phi}_F &= (\dot{\phi}_{Fl})\,\omega_{0l} + (\dot{\phi}_{F1})\,\dot{\alpha}_1 + \ldots\\ &\ldots + (\dot{\phi}_{F2})\,\dot{\alpha}_2 + (\dot{\phi}_{F3})\,\dot{\alpha}_3 + (\dot{\phi}_{F4})\,\dot{\alpha}_4\end{aligned} \tag{47}$$

In the same manner it's possible to derive

$$\begin{aligned}\dot{\mathbf{c}}_{III} &= \frac{\dot{\mathbf{c}}_5 + \dot{\mathbf{c}}_6}{2} = \frac{r}{2}[\boldsymbol{\omega}_{III} \times (\mathbf{n}_5 + \mathbf{n}_6) + \ldots\\ &\ldots + \mathbf{j}_{III} \times (\dot{\alpha}_5 \mathbf{n}_5 + \dot{\alpha}_6 \mathbf{n}_6)];\end{aligned} \tag{48}$$

obtained adding sidewise (43) and substituting (32) and (42). Comparing (39) and (48) we have

$$\begin{aligned}&\frac{r}{2}[\boldsymbol{\omega}_0 \times (\mathbf{n}_3 + \mathbf{n}_4) + \mathbf{j}_0 \times (\dot{\alpha}_3 \mathbf{n}_3 + \dot{\alpha}_4 \mathbf{n}_4)] + \ldots\\ &\ldots - d[\boldsymbol{\omega}_0 \times \mathbf{i}_{III} - \dot{\phi}_R \mathbf{k}_R] = \frac{r}{2}[(\boldsymbol{\omega}_0 + \dot{\phi}_R \mathbf{j}_0 + \ldots\\ &\ldots + \dot{\theta}_R \mathbf{i}_{III}) \times (\mathbf{n}_5 + \mathbf{n}_6) + \mathbf{j}_{III} \times (\dot{\alpha}_5 \mathbf{n}_5 + \dot{\alpha}_6 \mathbf{n}_6)\end{aligned} \tag{49}$$

By dot-multiplying both sides of (49) by $\mathbf{j}_0$ or by $\mathbf{i}_R$ we, respectively, calculate $\dot{\theta}_R$ or $\dot{\phi}_R$, i.e.

$$\begin{aligned}\dot{\theta}_R &= (\dot{\theta}_{Rl})\,\omega_{0l} + (\dot{\theta}_{R3})\,\dot{\alpha}_3 + \ldots\\ &\ldots + (\dot{\theta}_{R4})\,\dot{\alpha}_4 + (\dot{\theta}_{R5})\,\dot{\alpha}_5 + (\dot{\theta}_{R6})\,\dot{\alpha}_6\end{aligned} \tag{50}$$

and

$$\begin{aligned}\dot{\phi}_R &= (\dot{\phi}_{Rl})\,\omega_{0l} + (\dot{\phi}_{R3})\,\dot{\alpha}_3 + \ldots\\ &\ldots + (\dot{\phi}_{R4})\,\dot{\alpha}_4 + (\dot{\phi}_{R5})\,\dot{\alpha}_5 + (\dot{\phi}_{R6})\,\dot{\alpha}_6\end{aligned} \tag{51}$$

The coefficients $\dot{\theta}_{Fi}, \dot{\phi}_{Fi}, \dot{\theta}_{Ri}, \dot{\phi}_{Ri}$, depending on $\boldsymbol{\eta}$ and **n**$_i$, have been omitted for brevity.

3.2 Computation of the Twist of the Central Module

Once the vectors $\boldsymbol{\omega}_0$ and $\dot{\mathbf{c}}_0$ are calculated, using (34), the twist shaping matrix $\mathbf{T}_0$ it is possible to obtain as

$$\mathbf{T}_0 = \begin{bmatrix} \boldsymbol{\omega}_{0l} & 0 & 0 & \boldsymbol{\omega}_{03} & \boldsymbol{\omega}_{04} & 0 & 0\\ \dot{\mathbf{c}}_{0l} & 0 & 0 & \dot{\mathbf{c}}_{03} & \dot{\mathbf{c}}_{04} & 0 & 0 \end{bmatrix} \tag{52}$$

with

$$\begin{aligned}\boldsymbol{\omega}_{0l} &= \mathbf{j}_0 + 1/D_w\,(\mathbf{1} - \mathbf{j}_0\mathbf{j}_0{}^T)(\mathbf{n}_3 - \mathbf{n}_4);\\ \boldsymbol{\omega}_{0i} &= (-1)^{i-1}/D_w\,(\mathbf{1} - \mathbf{j}_0\mathbf{j}_0{}^T)\mathbf{n}_i; \qquad i = 3,4\\ \dot{\mathbf{c}}_{0l} &= r/2\ [\boldsymbol{\omega}_{0l} \times (\mathbf{n}_3 + \mathbf{n}_4)];\\ \dot{\mathbf{c}}_{0i} &= r/2\ [\boldsymbol{\omega}_{0i} \times (\mathbf{n}_3 + \mathbf{n}_4) + \mathbf{j}_0 \times \mathbf{n}_i];\end{aligned} \tag{53}$$

3.3 Computation of the Twists of the Wheels 4, 5

Expressing (30) and (31) in terms of **v** the twist shaping matrices **T**$_3$ and **T**$_4$ are

$$\begin{aligned}\mathbf{T}_3 &= \begin{bmatrix} \boldsymbol{\omega}_{3l} & 0 & 0 & \boldsymbol{\omega}_{33} & \boldsymbol{\omega}_{34} & 0 & 0\\ \dot{\mathbf{c}}_{3l} & 0 & 0 & \dot{\mathbf{c}}_{33} & \dot{\mathbf{c}}_{34} & 0 & 0 \end{bmatrix};\\ \mathbf{T}_4 &= \begin{bmatrix} \boldsymbol{\omega}_{4l} & 0 & 0 & \boldsymbol{\omega}_{43} & \boldsymbol{\omega}_{44} & 0 & 0\\ \dot{\mathbf{c}}_{4l} & 0 & 0 & \dot{\mathbf{c}}_{43} & \dot{\mathbf{c}}_{44} & 0 & 0 \end{bmatrix}\end{aligned} \tag{54}$$

where

$$\begin{aligned}&\boldsymbol{\omega}_{3l} = \boldsymbol{\omega}_{0l};\ \boldsymbol{\omega}_{33} = \boldsymbol{\omega}_{03} + \mathbf{j}_0;\ \boldsymbol{\omega}_{34} = \boldsymbol{\omega}_{04};\\ &\dot{\mathbf{c}}_{3i} = \dot{\mathbf{c}}_{0i} + (b/2)\boldsymbol{\omega}_{0i}\times\mathbf{j}_0; \qquad i = l,3,4\end{aligned} \tag{55}$$

and

$$\begin{aligned}&\boldsymbol{\omega}_{4l} = \boldsymbol{\omega}_{0l};\ \boldsymbol{\omega}_{43} = \boldsymbol{\omega}_{03};\ \boldsymbol{\omega}_{34} = \boldsymbol{\omega}_{04} + \mathbf{j}_0;\\ &\dot{\mathbf{c}}_{4i} = \dot{\mathbf{c}}_{0i} - (b/2)\boldsymbol{\omega}_{0i}\times\mathbf{j}_0; \qquad i = l,3,4\end{aligned} \tag{56}$$

3.4 Computation of the Twists of the Front and Rear Axes

Considering, now, (36) and (37) in terms of $\mathbf{v}$ the twist shaping matrices $\mathbf{T}_F$ and $\mathbf{T}_R$ are

$$\begin{aligned}\mathbf{T}_F &= \begin{bmatrix}\boldsymbol{\omega}_{Fl} & \boldsymbol{\omega}_{F1} & \boldsymbol{\omega}_{F2} & \boldsymbol{\omega}_{F3} & \boldsymbol{\omega}_{F4} & 0 & 0\\ \dot{\mathbf{c}}_{Fl} & \dot{\mathbf{c}}_{F1} & \dot{\mathbf{c}}_{F2} & \dot{\mathbf{c}}_{F3} & \dot{\mathbf{c}}_{F4} & 0 & 0\end{bmatrix};\\ \mathbf{T}_R &= \begin{bmatrix}\boldsymbol{\omega}_{Rl} & 0 & 0 & \boldsymbol{\omega}_{R3} & \boldsymbol{\omega}_{R4} & \boldsymbol{\omega}_{R5} & \boldsymbol{\omega}_{R6}\\ \dot{\mathbf{c}}_{Rl} & 0 & 0 & \dot{\mathbf{c}}_{R3} & \dot{\mathbf{c}}_{R4} & \dot{\mathbf{c}}_{R5} & \dot{\mathbf{c}}_{R6}\end{bmatrix};\end{aligned} \tag{57}$$

with

$$\begin{aligned}&\boldsymbol{\omega}_{Fj} = \boldsymbol{\omega}_{0j} + \dot{\phi}_{Fj}\mathbf{j}_0;\ \boldsymbol{\omega}_{Fi} = \dot{\phi}_{Fi}\mathbf{j}_0; && j = l,3,4\\ &\dot{\mathbf{c}}_{Fi} = (d/2)\boldsymbol{\omega}_{Fi}\times\mathbf{i}_F; && i = 1,2\\ &\dot{\mathbf{c}}_{Fj} = \dot{\mathbf{c}}_{0j} + (d/2)\boldsymbol{\omega}_{Fj}\times\mathbf{i}_F;\end{aligned} \tag{58}$$

and

$$\begin{aligned}&\boldsymbol{\omega}_{Rj} = \boldsymbol{\omega}_{0j} + \dot{\phi}_{Rj}\mathbf{j}_0;\ \boldsymbol{\omega}_{Ri} = \dot{\phi}_{Ri}\mathbf{j}_0; && j = l,3,4\\ &\dot{\mathbf{c}}_{Rj} = \dot{\mathbf{c}}_{0j} - (d/2)\boldsymbol{\omega}_{Rj}\times\mathbf{i}_R; && i = 5,6\\ &\dot{\mathbf{c}}_{Ri} = -(d/2)\boldsymbol{\omega}_{Ri}\times\mathbf{i}_R;\end{aligned} \tag{59}$$

3.5 Computation of the Twists of the Front and Rear Modules

Writing (38) and (39) in terms of $\mathbf{v}$ the twist shaping matrices $\mathbf{T}_I$ and $\mathbf{T}_{III}$ are

$$\begin{aligned}\mathbf{T}_I &= \begin{bmatrix}\boldsymbol{\omega}_{Il} & \boldsymbol{\omega}_{I1} & \boldsymbol{\omega}_{I2} & \boldsymbol{\omega}_{I3} & \boldsymbol{\omega}_{I4} & 0 & 0\\ \dot{\mathbf{c}}_{Il} & \dot{\mathbf{c}}_{I1} & \dot{\mathbf{c}}_{I2} & \dot{\mathbf{c}}_{I3} & \dot{\mathbf{c}}_{I4} & 0 & 0\end{bmatrix};\\ \mathbf{T}_{III} &= \begin{bmatrix}\boldsymbol{\omega}_{IIIl} & 0 & 0 & \boldsymbol{\omega}_{III3} & \boldsymbol{\omega}_{III4} & \boldsymbol{\omega}_{III5} & \boldsymbol{\omega}_{III6}\\ \dot{\mathbf{c}}_{IIIl} & 0 & 0 & \dot{\mathbf{c}}_{III3} & \dot{\mathbf{c}}_{III4} & \dot{\mathbf{c}}_{III5} & \dot{\mathbf{c}}_{III6}\end{bmatrix}\end{aligned} \tag{60}$$

where

$$\begin{aligned}&\boldsymbol{\omega}_{Ij} = \boldsymbol{\omega}_{Fj} + \dot{\theta}_{Fj}\mathbf{i}_F; && j = l,1,\ldots,4\\ &\dot{\mathbf{c}}_{Ii} = \dot{\mathbf{c}}_{0i} + d\,\boldsymbol{\omega}_{Fi}\times\mathbf{i}_F; && i = l,3,4\\ &\dot{\mathbf{c}}_{Ik} = d\,\boldsymbol{\omega}_{Fk}\times\mathbf{i}_F; && k = 1,2\end{aligned} \tag{61}$$

and

$$\begin{aligned}&\boldsymbol{\omega}_{IIIj} = \boldsymbol{\omega}_{Rj} + \dot{\theta}_{Rj}\mathbf{i}_R; && j = l,3,\ldots,6\\ &\dot{\mathbf{c}}_{IIIi} = \dot{\mathbf{c}}_{0i} - d\,\boldsymbol{\omega}_{Ri}\times\mathbf{i}_R; && i = l,3,4\\ &\dot{\mathbf{c}}_{IIIk} = -d\,\boldsymbol{\omega}_{Rk}\times\mathbf{i}_R; && k = 5,6\end{aligned} \tag{62}$$

3.6 Computation of the Twists of the Wheels 1, 2, 5, 6

From (40) and (41) it follows that

$$\begin{aligned}\mathbf{T}_1 &= \begin{bmatrix}\boldsymbol{\omega}_{1l} & \boldsymbol{\omega}_{11} & \boldsymbol{\omega}_{12} & \boldsymbol{\omega}_{13} & \boldsymbol{\omega}_{14} & 0 & 0\\ \dot{\mathbf{c}}_{1l} & \dot{\mathbf{c}}_{11} & \dot{\mathbf{c}}_{12} & \dot{\mathbf{c}}_{13} & \dot{\mathbf{c}}_{14} & 0 & 0\end{bmatrix};\\ \mathbf{T}_2 &= \begin{bmatrix}\boldsymbol{\omega}_{2l} & \boldsymbol{\omega}_{21} & \boldsymbol{\omega}_{22} & \boldsymbol{\omega}_{23} & \boldsymbol{\omega}_{24} & 0 & 0\\ \dot{\mathbf{c}}_{2l} & \dot{\mathbf{c}}_{21} & \dot{\mathbf{c}}_{22} & \dot{\mathbf{c}}_{23} & \dot{\mathbf{c}}_{24} & 0 & 0\end{bmatrix}\end{aligned} \tag{63}$$

with

$$\begin{aligned}&\boldsymbol{\omega}_{1i} = \boldsymbol{\omega}_{Ii};\ \boldsymbol{\omega}_{11} = \boldsymbol{\omega}_{I1} + \mathbf{j}_I; && i = l,2,3,4\\ &\dot{\mathbf{c}}_{1j} = \dot{\mathbf{c}}_{Ij} + (b/2)\boldsymbol{\omega}_{Ij}\times\mathbf{j}_I; && j = l,1,\ldots,4\end{aligned} \tag{64}$$

and

$$\begin{aligned}&\boldsymbol{\omega}_{2i} = \boldsymbol{\omega}_{Ii};\ \boldsymbol{\omega}_{22} = \boldsymbol{\omega}_{I2} + \mathbf{j}_I; && i = l,1,3,4\\ &\dot{\mathbf{c}}_{2j} = \dot{\mathbf{c}}_{Ij} - (b/2)\boldsymbol{\omega}_{Ij}\times\mathbf{j}_I; && j = l,1,\ldots,4\end{aligned} \tag{65}$$

while from (42) and (43) we obtain

$$\begin{aligned}\mathbf{T}_5 &= \begin{bmatrix}\boldsymbol{\omega}_{5l} & 0 & 0 & \boldsymbol{\omega}_{53} & \boldsymbol{\omega}_{54} & \boldsymbol{\omega}_{55} & \boldsymbol{\omega}_{56}\\ \dot{\mathbf{c}}_{5l} & 0 & 0 & \dot{\mathbf{c}}_{53} & \dot{\mathbf{c}}_{54} & \dot{\mathbf{c}}_{55} & \dot{\mathbf{c}}_{56}\end{bmatrix};\\ \mathbf{T}_6 &= \begin{bmatrix}\boldsymbol{\omega}_{6l} & 0 & 0 & \boldsymbol{\omega}_{63} & \boldsymbol{\omega}_{64} & \boldsymbol{\omega}_{65} & \boldsymbol{\omega}_{66}\\ \dot{\mathbf{c}}_{6l} & 0 & 0 & \dot{\mathbf{c}}_{63} & \dot{\mathbf{c}}_{64} & \dot{\mathbf{c}}_{65} & \dot{\mathbf{c}}_{66}\end{bmatrix}\end{aligned} \tag{66}$$

with

$$\begin{aligned}&\boldsymbol{\omega}_{5i} = \boldsymbol{\omega}_{IIIi};\ \boldsymbol{\omega}_{55} = \boldsymbol{\omega}_{III5} + \mathbf{j}_{III}; && i = l,3,4,6\\ &\dot{\mathbf{c}}_{5j} = \dot{\mathbf{c}}_{IIIj} + (b/2)\boldsymbol{\omega}_{IIIj}\times\mathbf{j}_{III}; && j = l,3,\ldots,6\end{aligned} \tag{67}$$

and

$$\begin{aligned}&\boldsymbol{\omega}_{6i} = \boldsymbol{\omega}_{IIIi};\ \boldsymbol{\omega}_{66} = \boldsymbol{\omega}_{III6} + \mathbf{j}_{III}; && i = l,3,4,5\\ &\dot{\mathbf{c}}_{6j} = \dot{\mathbf{c}}_{IIIj} - (b/2)\boldsymbol{\omega}_{IIIj}\times\mathbf{j}_{III}; && j = l,3,\ldots,6\end{aligned} \tag{68}$$

4 Dynamics

Using the NOC method the dynamics equations assume the form [5],

$$\mathbf{I}(\boldsymbol{\eta})\dot{\mathbf{v}} + \mathbf{C}(\boldsymbol{\eta},\mathbf{v})\mathbf{v} = \boldsymbol{\tau}(\boldsymbol{\eta}) + \boldsymbol{\gamma}(\boldsymbol{\eta}) \tag{69}$$

where $\mathbf{I}$, $\mathbf{C}$, $\boldsymbol{\tau}$ and $\boldsymbol{\gamma}$ are, respectively, the 7×7 generalised inertia matrix, the 7×7 Coriolis and centrifugal-force matrix and the generalised active and gravity forces. These terms can be written as

$$\begin{aligned}&\mathbf{I} = \sum_i \mathbf{T}_i^T\mathbf{M}_i\mathbf{T}_i,\ \mathbf{C} = \sum_i (\mathbf{T}_i^T\mathbf{M}_i\dot{\mathbf{T}}_i + \mathbf{T}_i^T\mathbf{W}_i\mathbf{M}_i\mathbf{T}_i),\\ &\boldsymbol{\tau} = \sum_i \mathbf{T}_i^T\mathbf{w}_i^A,\quad \boldsymbol{\gamma} = \sum_i \mathbf{T}_i^T\mathbf{w}_i^G,\\ &i = l,F,R,I,III,1,\ldots,6\end{aligned} \tag{70}$$

where $\mathbf{M}_i, \dot{\mathbf{T}}_i, \mathbf{w}_i^A$ and $\mathbf{w}_i^G$ are the i-th 6×6 mass matrix, the derivative of the twist shaping matrix w.r.t. time, the active wrench and the gravitational wrench acting on the i-th body. Introducing, for each body, the mass m_i, the inertia matrix $\mathbf{I}_i$, the CPM $\boldsymbol{\Omega}_i$ of $\boldsymbol{\omega}_i$, the torques τ_i acting on each wheel and the gravity acceleration g, it yields

$$\mathbf{M}_j = \begin{bmatrix} \mathbf{I}_j & \mathbf{O} \\ \mathbf{O} & m_j \mathbf{1} \end{bmatrix}, \mathbf{W}_j = \begin{bmatrix} \mathbf{\Omega}_j & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix},$$

$$\mathbf{w}_0^A = [-(\tau_3 + \tau_4)\,\mathbf{j}_0^T \quad \mathbf{o}_3^T]^T,$$

$$\mathbf{w}_I^A = [-(\tau_1 + \tau_2)\,\mathbf{j}_I^T \quad \mathbf{o}_3^T]^T,$$

$$\mathbf{w}_{III}^A = [-(\tau_5 + \tau_6)\,\mathbf{j}_{III}^T \quad \mathbf{o}_3^T]^T, \tag{71}$$

$$\mathbf{w}_1^A = \mathbf{w}_2^A = [\tau_i\,\mathbf{j}_I^T \quad \mathbf{o}_3^T]^T,$$

$$\mathbf{w}_3^A = \mathbf{w}_4^A = [\tau_i\,\mathbf{j}_0^T \quad \mathbf{o}_3^T]^T,$$

$$\mathbf{w}_5^A = \mathbf{w}_6^A = [\tau_i\,\mathbf{j}_{III}^T \quad \mathbf{o}_3^T]^T,$$

$$\mathbf{w}_j^G = [\mathbf{o}_3^T \quad -m_j g \mathbf{k}^T]^T, \;\; j = l, F, R, I, III, 1, \ldots, 6$$

5 Numerical Simulation

In this section a numerical simulation, developed by a Matlab-Simulink model, is implemented.

Table 2: Geometrical and initial pose parameters.

r = 0.1 [m]	ρ = 2700 [kg/m^3]
b = 0.5 [m]	r$_{ai}$ = 0.04 [m]
d = 0.3 [m]	r$_{ae}$ = 0.05 [m]
$\mathbf{c}_0$ = [1,1,2.2]^T [m]	$\mathbf{r}$ = [0.1,-0.2,-0.1]^T [m*rad]
r$_0$ = -0.2 [rad]	

Table 2 shows the geometrical parameters and the initial posture vectors of the robot. Links are modelled as cylindrical bodies where r$_{ae}$ is the external radius, r$_{ai}$ is the internal radius and ρ is the density.

Figures 2 and 3 show the trajectories of the Euler-Rodriguez parameters and of the centre point $\mathbf{c}_0$, obtained considering all the torques, applied to the wheels, equal to -10 Nm.

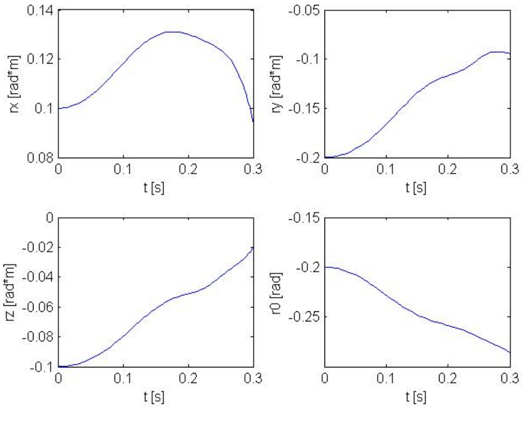


Figure 2: Euler-Rodrigues parameters: **r** and r$_0$.

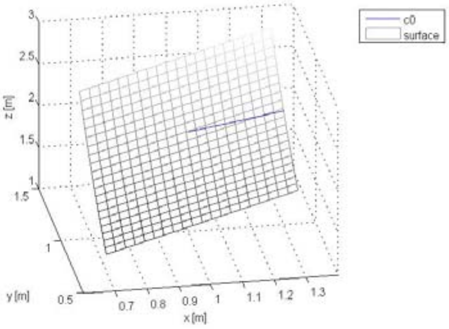


Figure 3: Trajectory of the centre point c$_0$.

6 Conclusions

The symbolic formulation of dynamics of a six-wheeled mobile robot here by the use of the Natural Orthogonal Complement was developed. The simplifications assumed in this work are: a) pure rolling contact between wheels and terrain, b) a unique contact point for each wheel and c) a contemporary contact of all the wheels. Non-holonomic nature and topology of the system and the coupling spherical wheels-terrain, make the computation very burden. Our future aim is to free this model from same restrictions. A robot with only five or four cylindrical deformable wheels in contact and is under study.

7 References

[1] M. Lacagnina, G. Muscato, S. Guccione and R. Sinatra, “Modelling and simulation of multibody mobile robot for volcanic environment explorations”, *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL*, Lausanne, Switzerland, pp 714-720 (2002).

[2] D.S. Nasrallah, J. Angeles and H. Michalska, “Modelling of an anti-tilting outdoor mobile robot”, *Proceedings of IDETC/CIE*, Long Beach, California (2005).

[3] K. Iagnemma, S. Dubowsky, *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers*, Springer Verlag, Heidelberg (2004).

[4] M. Lacagnina, G. Muscato and R. Sinatra, “Kinematics of a hybrid robot Wheeleg”, *Proceedings INES IEEE International Conference on Intelligent Engineering Systems*, Helsinki (2001).

[5] J. Angeles, *Fundamentals of Robotic Mechanical Systems*, Springer, New York (1997).

[6] J.C. Samin and P. Fisette, *Symbolic Modelling of Multibody Systems*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp 219-227 (2003).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Fruition of a Pre-Historical Cave Using a Mobile Robot

G. Cicirelli, A. Milella, A. Distante
Institute of Intelligent Systems for Automation,
National Research Council, Bari, Italy
cicirelli@ba.issia.cnr.it

Abstract

Earth holds a vast number of artistic treasures in its underground caves. A lot of these caves are still unexplored, others are known but very difficult to access. There is a great need to preserve these noteworthy sites from any form of damage or pollution. Recently new technological solutions have been proposed to allow not only the remote fruition of the artistic and historical treasures of caves that are inaccessible, but also the reconstruction of 2D and 3D maps of the environment which are very useful to study and characterise the site. In this paper we propose a mobile robot that can autonomously navigate the cave and acquire and send information to a remote console. The application of such a solution is advantageous because it does not need the installation of particular infrastructures in the site, minimising the risk of damaging the cave. We have performed an experimental session in the Grotta dei Cervi located at Porto Badisco along the Adriatic coast in southern Italy. The robot constructs a 2D map of the environment using data from the on-board laser range scanner and builds the 3D model of some zones of particular interest using a computer vision technique. These zones are identified by RFID tags.

Keywords: inspection robot, range sensing, planar map construction, RFID sensors, 3D reconstruction

1 Introduction

In the south of Italy, along the Adriatic coast, a cave holds a prehistoric treasure remarkable for its complexity and artistic and historical relevance. The Grotta dei Cervi (Stags' Cave) has on its walls a huge collection of paintings of hunting scenes, stags, men and small animal groups dated to the Middle Neolithic period. The long corridors of the cave are decorated with fascinating red and black wall paintings realised with red ochre and bat guano (figure 1).

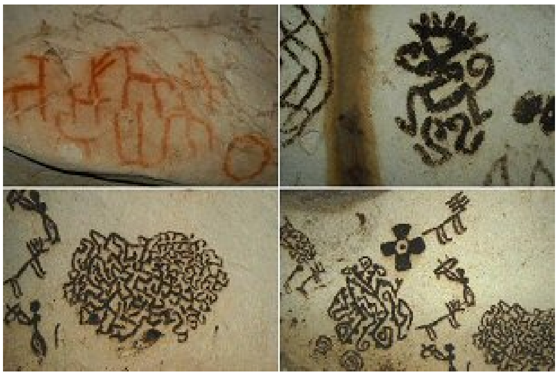


Figure 1: Some paintings on the walls of the cave.

Access to the cave is restricted to a few authorised people. Care must be taken to guarantee their safety and to prevent polluting elements from being introduced into this very particular and valuable environment. The application of a technological solution seems to be the best way to allow the (remote) fruition of the archaeological site and to satisfy the requirements of cave preservation and safety. By fruition, in this instance, we mean bringing the possibility of experiencing the wonders of the cave through the medium of the robot to any external point in the world. In this paper we propose using a mobile robot which is able to autonomously navigate the cave and to acquire useful data using its on-board sensors. This solution reduces the risk of damaging the cave because it does not require the installation of invasive infrastructures. The cave is formed by a series of narrow and twisting corridors. We have inspected one of these corridors.

In this paper we present an interesting integration of two different algorithms on the same appropriately equipped platform. The robot constructs the 2D map of the environment using a laser range scanner and builds the 3D model of some zones of particular interest using a computer vision technique. The combination of these two approaches allows us to obtain a complete knowledge of the environment in an automatic way. Several approaches can be found in the literature that deal separately with the two different problems of building 2D and 3D models of an environment. Few works combine the two different tasks. In [1] a mobile robot, equipped with a laser range scanner and a panoramic camera, is used as a security guard to survey large areas, then a 3D complete model of the environment is reconstructed. In [2], sonar and video data are used for the 3D reconstruction of an indoor environment. Our aim is to build a robot able to navigate in the site and to reconstruct 3D models of particular areas for the remote fruition of the site, at a lower computational cost and with higher accuracy. In addition, the explored environment is very singular and several difficulties had to be addressed during the experimental session.

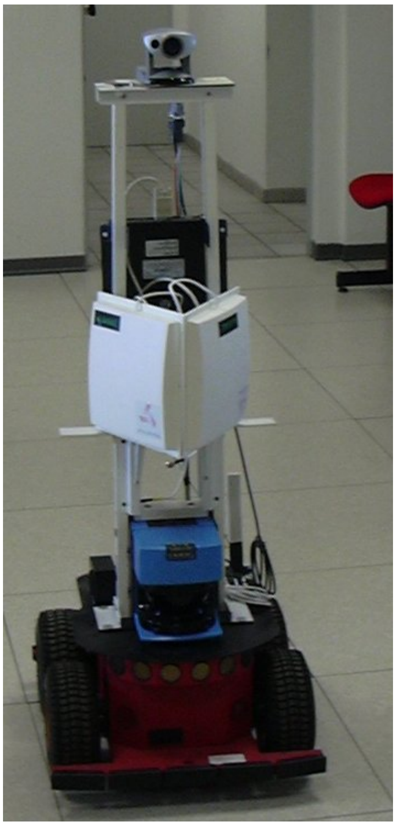


Figure 2: Pioneer 3AT platform equipped with Sick laser range finder, RFID reader and two antennas.

The mobile vehicle used for the experimental session is composed of the Pioneer 3AT mobile robotic platform from ActivMedia Robotics, equipped with a laser range finder, the RFID reader and two antennas, sonar sensors, a video camera, inclinometers, encoders, a compass and an antenna for wireless communication between the robot and a remote computer (figure 2). Its four tractor wheels can scale a 45° gradient and sills of 9cm. Sixteen forward and rear sonar sensors sense obstacles up to 7m away. The laser range finder is able to sense objects at a distance up to 80m away with a resolution of 0.5° (360 readings on 180°). On the vehicle a support with appropriate height is placed, carrying the illumination system and a pan-tilt-zoom camera. The structure is of aluminium and is light and robust. It is 1m high in order to acquire the wall paintings from an appropriate perspective. In fact, the most interesting wall paintings are between 1m and 2m above the ground. The laser and video data are used to construct the 2D map of the explored corridor of the cave and the 3D model of some zones of particular artistic and historical interest, respectively. First the planar map of the inspected environment is obtained using the laser scanner mounted on the robot platform and applying a scan matching algorithm. The high accuracy of about one millimetre error allows one to build a very accurate planar map. Using a computer vision technique we build the three-dimensional model of some zones of particular interest. The only geometrical constraint is the correspondence between each feature point in different images. The images must be acquired by the same camera with a fixed focal length. The interesting zones are identified using RFID (radio frequency identification) technology. Passive RFID tags are deployed in these areas and are useful both for navigation and recognition purposes. These tags do not require any external power source; they are activated by the electromagnetic field generated by the RFID antenna with a radius of approximately 2m and are detected by the RFID reader. Moreover each tag is univocally identified by an ID allowing recognition of the area of the environment where it is attached. Recent advances in the field of RFID technology have been allowing its use in a wide range of applications such as health-care, object identification, assistance to the disabled and mobile robot navigation [3, 4, 5].

In the following sections of the paper the methods used to build the 2D and 3D models of the environment will be discussed.

2 Planar Map Construction

Mobile robot localisation and environment mapping have been extensively studied in the robotics literature [6, 7, 8]. Several researchers have focused on the problem of simultaneous localisation and mapping (SLAM), in which a map of the environment is built and the robot is simultaneously localised therein [9, 10, 11]. Most approaches use precise distance sensors such as laser range finders [11, 12] for their high accuracy, but often the applicability is restricted to planar structured environments, such as offices and regular buildings. The laser readings have a high accuracy allowing the building of accurate planar maps especially when the robot moves in a plane. The cave environment presents a very rough terrain characterised by depressions and bumps. In this case the laser data that supply planar information must be integrated with the data from the inclinometer in order to obtain accurate information about the scanned environment. During the data acquisition procedure, the incremental errors typical of odometers must be considered and accounted for. To solve the SLAM problem, the scan matching algorithm proposed by Gutmann [13] was used. Gutmann’s algorithm, the combined scan matcher (CSM), is based on the integration of the IDC algorithm proposed by Lu and Milios [7] and that proposed by Cox [14]. Initially the CSM algorithm was tested in our laboratory in order to verify its accuracy. A map of the laboratory was constructed and an error analysis of the robot position in the map was carried out. This error analysis was done in our lab because the limited access to the cave prevented us from placing all the necessary instruments in the site. The corrected positions of the robot obtained after the application of the CSM were compared with the robot positions measured by using a theodolite station. The estimated error on robot positions is below 30cm. Note that in a robot navigation context this error can be considered negligible, since the estimated position remains inside the robot base. The CSM algorithm was also applied to the data acquired in the cave, obtaining a point map. The corrected planar map is shown in figure 3, where four zones of particular interest for the presence of prehistoric paintings are also indicated.

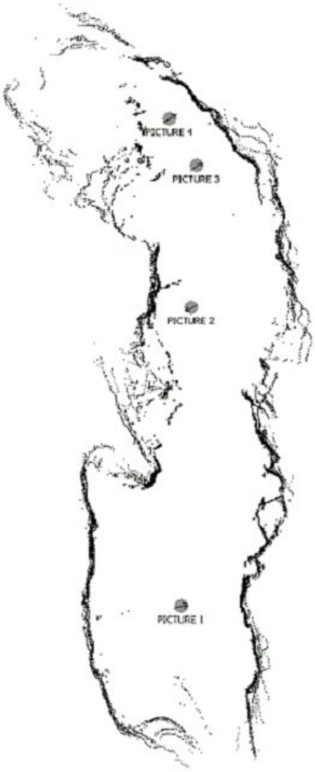


Figure 3: 2D map of the explored area of the cave obtained after the application of the scan-matching algorithm. The explored corridor is about 40x5m^2.

The map obtained using the laser scanner supplies new and useful, although still approximate, information about the structure of the cave; such information was not available before our visit. It is important to note that the structure of the cave, supplied by the map, is very important for the knowledge and fruition of the archaeological site because it describes the morphology of the whole environment, placing each painting in its context and facilitating a better understanding of its role and meaning. Furthermore, the planar map is very useful for the navigation of the robot inside the cave. In order to identify the areas of the cave that are rich in wall paintings, we use RFID (radio frequency identification) technology. RFID tags can be placed near those areas that we want to be recognised. Each tag is characterised by a unique code and so it can be used to refer to a particular painting, once the robot is in its proximity. After that the robot can acquire images of the paintings and send them to a remote computer station or can trigger the procedure for the 3D model reconstruction of the area in its vicinity. An experimental phase to prove the efficiency of the RFID sensors was carried out in our laboratory before the experimental session in the cave. First a detection model for the RFID antennas was generated by considering the frequency of detections. For this purpose the procedure proposed in [3] was used. The robot was rotated in front of a tag and the frequency was recorded. This was repeated at different distances and at with fixed angular resolution. In figure 4 the resulting histogram is shown; the distances are expressed in centimetres and the angular resolution is 3°. This detection model can describe the efficacy of the detection range of the robot as its distance and orientation vary with respect to the tags.

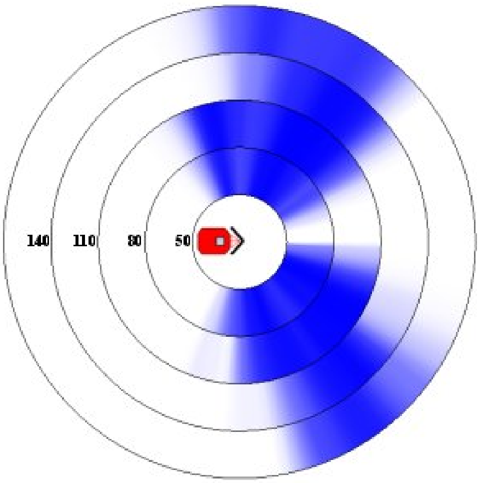


Figure 4: Detection fields for the RFID antennas.

By placing the RFID tags near the areas of interest the robot can plan the path to those regions and enable the visual fruition of the prehistoric paintings and the construction of a 3D model of the observed area. The position of the tags is learned during the data acquisition phase.

3 3D Model Reconstruction

Construction of 3D models from images, in the context of mobile robots exploring real indoor environments, is a thoroughly investigated topic in computer vision [1, 10, 15]. The major obstacle for extensive use of vision sensors is their computational complexity. But this problem is becoming less relevant with the increase in computing power of the available machines; therefore, the extraction of real-time full 3D information from cameras seems to be achievable in the near future. Reconstructing 3D models using computer vision techniques generally requires the extraction and matching of features (points, lines, target objects) [16, 1]. Moreover, it is important to determine the correspondence between features in different images, since the accuracy of the resulting model depends directly on the accuracy of the feature correspondence. The method described in this paper uses as geometrical constraint only the correspondence between corners in different images. A complex 3D scene is reconstructed using a set of three images acquired from three different viewpoints

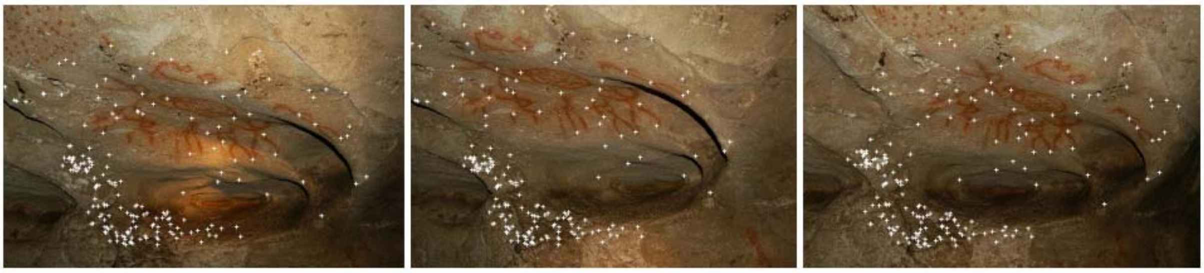


Figure 5: Three images of an area of the wall rich in paintings. The white crosses represent the correct point matches.

of the same scene. The only requirement is that the images must be acquired by the same camera with a fixed focal length.

After image acquisition, feature points that correspond to high curvature points (corners) are extracted in each image using the Harris corner detector [15]. The maximum number of corners to be extracted in each image is fixed a priori. A matching procedure is then applied to each pair of images. A classical correlation technique is first used to establish the matching candidates between two images by determining a correlation score for each pair of points. If the correlation score is higher than a given threshold, the related pair of points is considered a candidate match. To verify the candidate matches, a parameter counts the number of similar candidate matches found in the neighbourhood of each candidate matched point. The sum of these parameters for all candidate matches defines an energy function. The minimisation of the energy function through a relaxation technique solves the ambiguity problem [17]. After the determination of the corner correspondence for each pair of images, the set of correct matches for all three images is determined.

Figure 5 shows three images with matched points. Knowing the corresponding corners it is possible to determine the fundamental matrix and the intrinsic parameters of the camera [18]. At this point, all the necessary data to reconstruct the 3D scene are known. The 3D model is reconstructed through the application of the polygonal mesh technique. The 3D model of the scene has been made using VRML. Figure 6 shows the 3D model of the scene acquired in the cave.

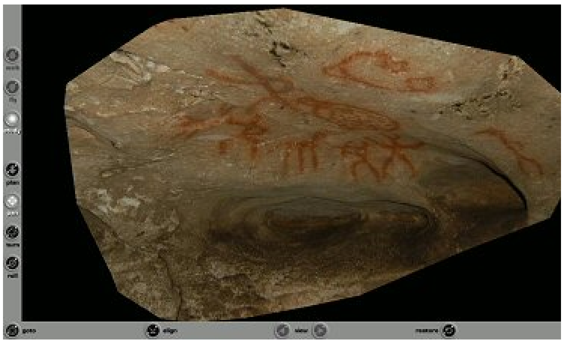


Figure 6: The reconstructed 3D model of the area of the cave shown in the images in figure 5.

4 Conclusions

This work focuses on a technological solution for the remote fruition of paintings in a cave that is inaccessible to all but the most expert speleologists. A properly equipped mobile robot platform has been used to reach this goal. The mobile robot, properly equipped, can navigate the cave and send useful information to a remote console. The robot is made up of a mobile robotic platform equipped with a laser range finder, sonar sensors, a video camera, inclinometers, odometers and a compass. Using all these devices, together with suitably integrated software modules, it is possible to build the two-dimensional map of the site and to recover the 3D structure of some zones of particular interest. The 2D map has been constructed using the data provided by the laser range finder which allows the construction of a planar map with very high precision. Zones of particular interest for the presence of prehistoric paintings are identified by using RFID tags. The three-dimensional structure of these zones has been reconstructed using three images acquired from different viewpoints by the on-board camera, which has a fixed focal length. An experimental session performed in the Grotta dei Cervi has been presented in the paper. The proposed approaches for the 2D map building and for the 3D model reconstruction are accurate and flexible. Although there are some structural limits, the technological solution presented seems to be suitable for making an archaeological cave available without damaging it. It allows the growth of knowledge of these kinds of site, it supports their study and improves the capability of monitoring and preserving very relevant archaeological treasures.

5 References

[1] P. Biber, H. Andreasson, T. Duckett and A. Schilling, "3D modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera", *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004),* Sendai, Japan, pp 3430-3435 (2004).

[2] J.M. Leiva, P. Martinez, E.J. Perez, C. Urdiales and F. Sandoval, “3D reconstruction of static indoor environment by fusion of sonar and video data”, *Proceedings International Symposium on Intelligent Robotics Systems (SIRS 2001),* Toulouse, France (2001).

[3] D. Hahnel, W. Burgard, D. Fox, K. Fishkin and M. Philipose, “Mapping and localization with RFID technology”, *Proceedings IEEE International Conference on Robotics and Automation*, New Orleans, pp 1015-1020 (2004).

[4] V. Kulyukin, C. Gharpure, J. Nicholson and G. Osborne, “Robot-assisted wayfinding for the visually impaired in structured indoor environments”, *Autonomous Robots*, 21(1), pp 29-41 (2006).

[5] T. Tsukiyama, “Navigation system for mobile robots using RFID tags”, *Proceedings International Conference on Advanced Robotics (ICAR)*, Coimbra, Portugal (2003).

[6] L.Y. Ip and A.B. Rad, “Incorporation of feature tracking into simultaneous localization and mapping building via sonar data”, *Journal of Intelligent Systems*, 39(2), pp 149-172 (2004).

[7] F. Lu and E. Milios, “Robot pose estimation in unknown environments by matching 2D range scans”, *Journal of Intelligent and Robotic Systems*, 18, pp 249-275 (1997).

[8] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte and A.Y. Ng, “Simultaneous mapping and localization with sparse extended information filters: Theory and initial results”, *Algorithmic Foundations of Robotics*, 7, pp 363-380 (2003).

[9] G. Grisetti and L. Iocchi, “Map building in planar and non-planar environments”, *Proceedings 2nd International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (SRMED)*, Lisbon, Portugal (2004).

[10] J.S. Gutmann and K. Konolige, “Incremental mapping of large cyclic environments”. *Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA-99)*, Monterey, CA (1999).

[11] C. Stachniss, D. Hahnel, W. Burgard and G. Grisetti, “On actively closing loops in grid-based fastSLAM”, *Advanced Robotics,* 19(10), pp 1059-1079 (2005).

[12] D.F. Wolf and G.S. Sukhatme, “Mobile robot simultaneous localization and mapping in dynamic environment”, *Autonomous Robots*, 19(1), pp 53-65 (2005).

[13] J.S. Gutmann and C. Schlegel, “Amos: Comparison of scan matching approaches for self-localization in indoor environments”, *Proceedings 1st Euromicro Workshop on Advanced Mobile Robots*, Milan, Italy, IEEE Computer Society Press, pp 1-7 (1996).

[14] I.J. Cox, “Blanche - an experiment in guidance and navigation of an autonomous robot vehicle”, *IEEE Transactions on Robotics and Automation*, 7(2), pp 193-204 (1991).

[15] V. Sequeira, K. Ng, E. Wolfart, J.G.M. Gonalves and D.C. Hogg, “Automated reconstruction of 3d models from real environments”, *ISPRS Journal of Photogrammetry and Remote Sensing*, 54, pp 1-22 (1999).

[16] J.K. Aggarwal and B.C. Vemuri, “3D model construction from multiple views and intensity data”, *Proceedings IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* Miami Beach, FL, pp 435-437 (1986).

[17] Z. Zhang, R. Deriche, O. Faugeras and Q. Luong, “A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry”, *Artificial Intelligence*, Vol. 78, no. 1-2, pp 87-119, (1995).

[18] R.I. Hartley, “Kruppa’s equations derived from the fundamental matrix”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2), pp 133-135 (1997).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Balance Control of a Biped Robot Using the ZMP State Prediction by the Kalman Estimator

Sangbum Park, Youngjoon Han, Hernsoo Hahn
School of Electronic Engineering
Soongsil University, Seoul, Korea
forcepsb@ssu.ac.kr, young@ssu.ac.kr, hahn@ssu.ac.kr

Abstract

This paper proposes the balance control scheme of a biped robot using the position of ZMP predicted by the Kalman filter. The mathematical model of the biped robot is generally approximated by 3D-LIPM (3D linear inverse pendulum mode), but it cannot completely express the robot's dynamics. The stability of the biped robot depends on whether the ZMP (zero moment point) position is in the stability region or out of. And the internal error between the robot mechanism and its model could affect the stability of a robot. Therefore, the balance control proposed by the paper not reduces the internal error, but also timely generates the proper control. The experiment of the proposed balance control is simulated on the virtual workspace where the biped robot may encounter with various difficulties.

Keywords: biped walking robot, balance control, Kalman filter

1 Introduction

Nowadays robots are expanding their application domain into house keeping and entertainment as well as industry automation. This trend has already given birth to humanoid robots such as ASIMO which has the appearance of a human being and can share the working space in the house without changing the structure. Humanoid robot is characterised in the technical viewpoint by its biped walking. It is due to the fact that the balance control of a biped robot is very difficult since the actuating joints are not fixed and thus easily fallen in unstable states.

The most popular method of maintaining the stable pose of the biped robot is to utilise the concept of ZMP (zero moment point) which was proposed by Vukobratovic [1]. It illustrated that the motion of a robot would be always stable if ZMP is located inside the stable region in the supported region which is confined as the contacting areas of the biped robot's feet with floor. The researches for stable walking of biped robot, based on the concept of ZMP accomplished thus far, can be categorised into two approaches: 1) generation of the walking pattern based on the dynamics of a robot and 2) balance control by correcting the errors included in the measured ZMP.

The methods of generating the walking pattern of the biped robot are proposed variously. For example, Erbatur applied the ZMP trajectory of human's walking pattern to biped robot [2], and Arakawa generated the optimal walking pattern of the biped robot by applying a training set of ZMP, which is proper to human walking, to a genetic algorithm [3]. These methods generate the walking pattern similar to human's one, but they make the movement of biped robot unnatural by ruling out the dynamics of the robot.

In order to resolve this problem, 3D-LIPM for the mechanical model of the biped robot is generally used. Three-dimensional linear inverted pendulum mode (3D-LIPM) is derived from a general three-dimensional inverted pendulum whose motion is constrained onto an arbitrarily defined plane [4]. This method is able for the biped robot to stably walk on the constrained surface of the movement of the inverted pendulum. Sugihara proposed the method that controls COM of the whole humanoid body system in real-time through ZMP manipulation [5]. The way to manipulate ZMP significantly reduces the complexity of the problem and thus allows real-time implementation [6,7]. However, since the walking pattern generated by the simplified model contains the internal error, it causes the difference error of the robot movement according to the mechanism of biped robots.

Biped robots having multiple joints may fall in unstable state when using ZMP based balance control technique due to an approximation of the robot dynamics and the time delay included in the process of error compensation. This paper proposes a new method of adaptively compensating the states of 3D-LIPM using the Kalman filter to solve the first problem, and a new method of determining ZMP using the state of 3D-LIPM predicted by the Kalman filter to solve the second problem. The performance of the proposed algorithm is evaluated using the biped robot with 24 joints which was designed for the experiments. The stability of the robot walking is tested using the 3D simulator with an assumption that COM calculated from ZMP includes a white Gaussian noise.

The paper is organised as follows. In section 2, we provide the basic concept of the 3D linear inverted pendulum and the algorithm for generating the walking pattern. In section 3, the method of estimating the ZMP position is described. In section 4, the simulation result for evaluating the proposed algorithm is provided.

2 Kinematics of a Biped Robot and Inverted Pendulum Movement

2.1 Structure of a Biped Robot

For a manipulator to approach with a specific pose to an object in 3D space, its degree of freedom should be at least 6. With the same reason, for a biped robot to maintain a stable pose while walking in 3D space, each foot should have at least 6 degrees of freedom. Human beings have the arms and legs with 7 degrees of freedom and thus they can generate various poses at one position while maintaining the balance. However, in order for a robot to provide the arms and legs with 7 degrees of freedom like a human being, it costs too much for the hardware and also for the control software since there exist too many solutions for the poses of the arms and legs. For these reasons, a robot with the arms and legs with at most 6 degrees of freedom is constructed and used in general. In this paper, a robot is constructed with two arms having 3 degrees of freedom and two legs having 6 degrees of freedom. Its waist and head have 3 degrees of freedom respectively so that the robot has the total 21 degrees of freedom. The legs are assigned 6 degrees of freedom to minimally simulate the human walking. Figure 1a shows the structure of the robot to be used in this paper and the coordinate systems assigned to the individual joints of the legs. For simplifying the analysis of the kinematics, the coordinates systems of individual joints of a leg are aligned to reduce the number of Denavit-Hartenberg (DH) parameters. The size of the feet is determined to be smaller than the projected area of the robot body with considering the fact that the robot may fall easily in unstable states if it is too small, and that the walking pattern becomes unnatural and the robot wastes unnecessary extra energy if it is too large

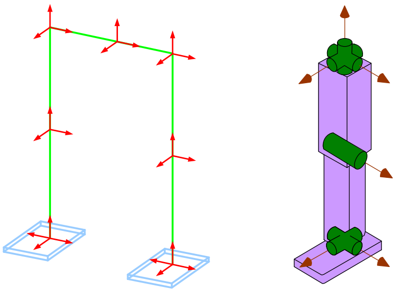


Figure 1: The structure of the leg and the coordinate systems assigned at individual joints.

2.2 3D Linear Inverted Pendulum Mode

It is very difficult to establish the perfect model of human walking since it has an extremely high degree of freedom against irregular ground conditions and external forces. Thus, a human walking is restrictively simplified using the inverted pendulum mode where the human body is a moving mass inversely pendent on the centre of a foot. Figure 2 shows the 3D-LIPM (3D linear inverted pendulum mode) which maps a human walking to the inverted pendulum whose motion is restricted to the known curved surface. As shown in figure 2a, during the dynamic motion of a robot, the centre of the pelvis region corresponding to the COM (centre of mass) does not deviate from its original position. In other words, a walking motion of the robot can be interpreted as the motion of COM keeping the constant height (zc) on the plane whose normal is $(k_x, k_y, -1)$, as shown in figure 2b. Thus, the surface on which the COM moves can be expressed by the following equation.

$$z = k_x x + k_y y + z_c \tag{1}$$

If a motion of COM is restricted on a plane ($k_x = k_y = 0$ in (1)), then the dynamics of the inverted pendulum can be expressed by the following:

$$\ddot{x} = \frac{g}{z_c} x - \frac{1}{m z_c} \tau_y \tag{2}$$

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} \tau_x \tag{3}$$

where m is the weight of the pendulum M, g is the gravity acceleration, and τ_x and τ_y represent the torques of X and Y axes respectively.

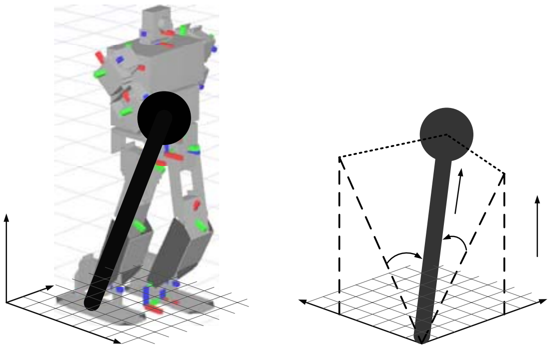


Figure 2: Interpretation of a robot walking using the inverted pendulum mode.

Since the above equations are dependent on z_c, the horizontal motion is not affected by variation of the normal of the plane. This observation enables us to derive the position of the ZMP as follows:

$$p_x = -\frac{\tau_y}{mg} \tag{4}$$

$$p_y = -\frac{\tau_x}{mg} \tag{5}$$

where (p_x, p_y) is the position of ZMP in the X-Y plane. By applying these values to (2) and (3), the dynamics of the inverted pendulum can be rewritten as follows:

$$-\frac{z_c}{g}\ddot{x} + x = p_x \tag{6}$$

$$-\frac{z_c}{g}\ddot{y} + y = p_y \tag{7}$$

2.3 Generation of a Walking Trajectory using 3D-LIPM

The centre of mass (COM) of a robot is in general extracted from the mass centres of links. The centre of mass of a link is determined using the mass diagram which illustrates the distribution of mass in the link [11]. Once the COM of a robot is obtained, then the walking motion of the robot can be simplified as the motion of an inverted pendulum from which the walking pattern of a robot can be extracted. A new method of generating the stable walking pattern of a biped robot based on the trajectory of ZMP is illustrated here.

The trajectory of ZMP while a robot is walking is consisted of the cyclic repetition of the trajectory of ZMP constructed during one step motion [12,13]. Since the legs are left and right symmetric about the pelvis, the trajectory of ZMP of the robot becomes symmetric and it can be approximated as a sine wave form, as shown in figure 3.

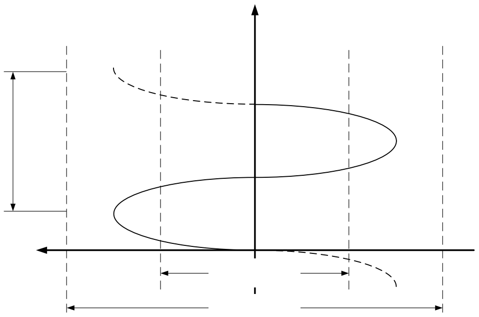


Figure 3: Trajectory of ZMP during one step movement.

When the robot moves along the X axis, the range of the trajectory of ZMP can be divided by three areas: Range A-B is the LSS (left single support area), range C-D is the RSS (right single support area) and range B-C is the DS (double support area). The width of SS (single supporting) area equals to the width of the foot of the robot and the DS becomes the gap between two legs of the robot.

Once the trajectory of ZMP which determines the walking pattern of the biped robot is obtained as illustrated above, the positions of COM corresponding to those of ZMP are extracted using (6) and (7). Since (6) and (7) are the non-homogeneous 2nd order differential equation where (p_x , p_y) which is the ZMP on the sine wave are the force function, the steady state solution of theses equations will be (x,y), the coordinates of COM. To determine x point of the COM first, the trajectory of p_x having the angular velocity ω is represented as follows:

$$p_x = A\sin\omega t \tag{8}$$

By applying (8), (6) is represented as the canonical non-homogeneous 2nd order differential equation as follows.

$$\ddot{x} - \frac{g}{z_c}x = -\frac{g}{z_c}A\sin\omega t \tag{9}$$

Since the force function p_x in the above equation is a sine wave, the steady state solution x will be also represented as a function of the size waves whose angular velocity is ω.

$$x = A_1\cos\omega t + B_1\sin\omega t \tag{10}$$

By applying this function to (9) and solving the equation with respect to x, (11) is obtained as the steady state solution.

$$x = \frac{g}{z_c\omega^2 + g}\cdot p_x \tag{11}$$

In the same way, the y position of COM can be determined as follows:

$$y = \frac{g}{z_c\omega^2 + g}\cdot p_y \tag{12}$$

Equations (11) and (12) show that the coordinates of COM is linearly proportional to the coordinates of ZMP when the motion of COM of the biped robot is restricted to a plane where z = zc and the trajectory of ZMP is approximated as a sine wave.

3 Estimation of the ZMP State using Kalman Estimator

As illustrated in the previous section, for stable walking of a biped robot the ZMP should remain inside the safety region. When modelling the trajectory of ZMP of a biped robot in practice, it is very difficult to completely consider all the environments that the robot should meet. That's why an approximation is inevitable in the modelling of the trajectory of ZMP and the error included in the approximation renders the robot fallen into unstable state. The time delay occurring when compensating the error included in the measured ZMP with reference to the goal ZMP is also a major cause of failure in controlling the balance of a biped robot.

To solve these problem, this paper uses the Kalman filter which compensates adaptively the states of 3D-LIPM to solve the problems caused by an approximated modelling of the trajectory of ZMP, and determines the ZMP based on the new states of 3D-LIPM estimated by the Kalman filter to solve the time delay problem in the compensation of measurement error in the ZMP.

Figure 4 shows the configuration of the balance control technique proposed in this paper. For balance control of an inverted pendulum, the error (e(k)) included in the estimated ZMP (Px) with reference to the goal ZMP (*Px) calculated a priori is compensated. Here, the estimated ZMP (Px) is obtained by inserting the predicted state in the correction stage of the Kalman filter into (6) and (7). The servo control module which controls the actual motion of the robot calculates the trajectory of COM using (11) and (12) with the errors in the measured ZMP. From the dynamic equations of the robot, the velocities of individual joints to control the robot can be derived using the corrected ZMP.

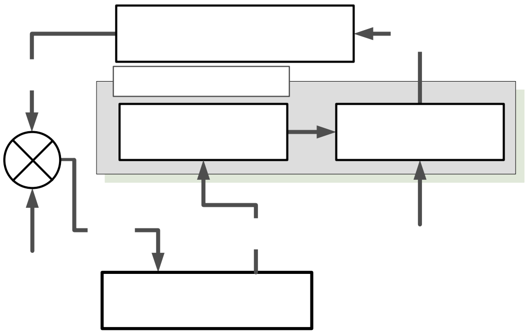


Figure 4: The configuration of the proposed balance control technique.

The state and measurement equations of the Kalman filter for estimating the ZMP state are given as follows.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_k \tag{13}$$

$$z_k = Hx_k + v_k \tag{14}$$

where $x_k = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$ is the present state which is consisted of the position, velocity, and acceleration of the COM and A is the transition matrix which has the following form:

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

In (14), $z_k = \begin{bmatrix} x_{zmp} & y_{zmp} \end{bmatrix}$ is the measured position of the ZMP state, wk and vk are the state error and measurement error which are modelled as zero-mean white Gaussians, and the related matrix H has the following form.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & -z_c/g & 0 \\ 0 & 1 & 0 & 0 & 0 & -z_c/g \end{bmatrix} \tag{16}$$

Based on these equations, the estimated state of ZMP in the next step is determined by the following equation.

$$x_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \\ \ddot{x}_{k-1} \\ \ddot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} xu_{k-1} \\ yu_{k-1} \end{bmatrix} + \begin{bmatrix} wx_k \\ wy_k \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

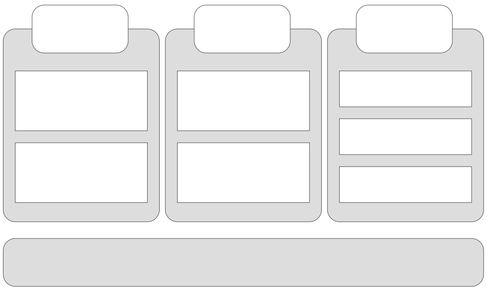
In (17), x_k is composed of three terms. The first one calculates the predicted state of ZMP for the next step from the present state. The second one which is the control input to be added to the first term takes into account the errors included in the COM the measurement as the present state. The third one is added to consider the errors that may be included in the state transition process. The measurement process is performed by the following equation.

$$z_k = \begin{bmatrix} 1 & 0 & 0 & 0 & -z_c/g & 0 \\ 0 & 1 & 0 & 0 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} x_{COM} \\ y_{COM} \\ \dot{x}_{COM} \\ \dot{y}_{COM} \\ \ddot{x}_{COM} \\ \ddot{y}_{COM} \end{bmatrix} + \begin{bmatrix} vx_k \\ vx_k \end{bmatrix} \tag{18}$$

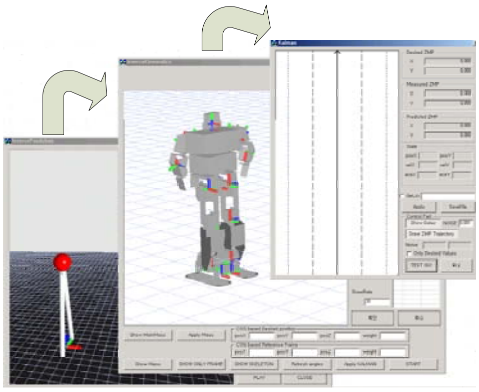
In (18), z_k is composed of three terms. The first one transforms the measured COM to ZMP. The second term is added to consider the errors that may b included in the measurement process.

4 Experiments and Discussion

The performance of the proposed scheme for stable balance control of a biped robot has been tested with the biped robot, constructed in this laboratory. To visualise the process of the proposed algorithm, the GUI is implemented which is composed of three modules. The first one is to show the generation of a walking trajectory, that is, the trajectory of COM of the biped robot using the 3D-LIPM, the second one is to show how to predict the ZMP state in the next step from the measured states of individual joints which are assumed to include white Gaussian noise, and the third one is to show how similarly the actual robot trajectory follows the estimated one. This GUI includes the graphic module which constructs the robot if provided with the kinematics information. Figure 5a shows the three modules are connected by the shared memory and figure 5b shows graphically the process of the proposed algorithm.



(a) Block diagram of the proposed algorithm.



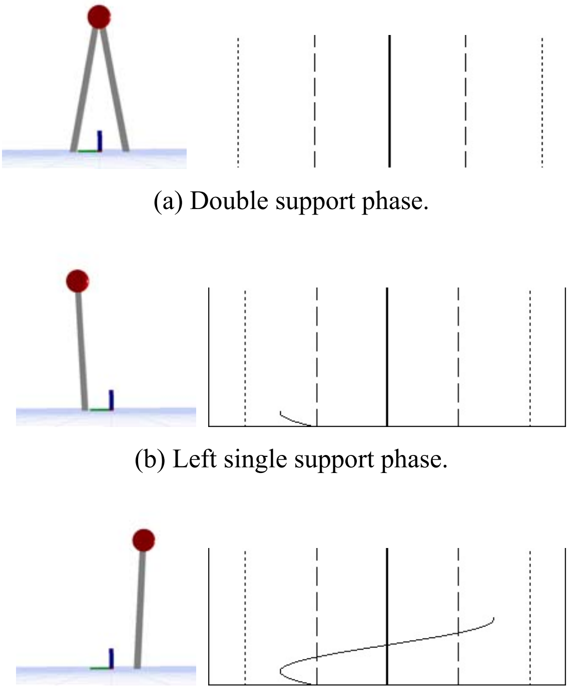
(b) GUI showing process of proposed algorithm.

Figure 5: GUI for visualising experimental process.

The proposed algorithm has been tested with respective to three viewpoints: 1) with no noise, 2) with limited measurement noise, 3) with impulsive measurement noise.

The first case that biped walking robot generally walks is that the measurement noise is not exist. While the module to generate walking pattern is running, the desired trajectory of ZMP is generated by the inverted pendulum mode, and then robot is received the desired trajectory of COM converted by ZMP's one in real-time. Fig 6 shows experimental result that the trajectory of ZMP is generated by inverted pendulum mode.

The desired ZMP trajectory can be transformed into the desired trajectory of COM. Then, each angle of joint is able to be calculated by inverse kinematics. When the displacement of joint at each joint is transferred to the robot, it starts walking according to the movement of inverted pendulum. The experimental result is shown in figure 7.

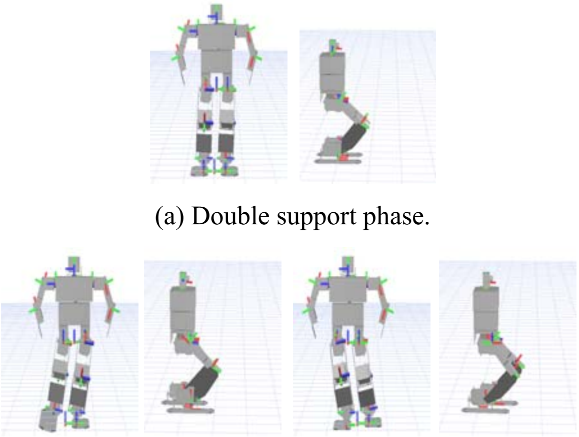


(a) Double support phase.

(b) Left single support phase.

(c) Right single support phase.

Figure 6: Generating the desired ZMP trajectory of the biped robot.



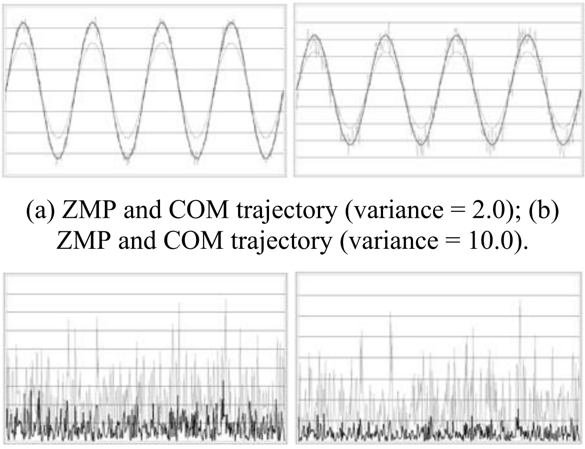
(a) Double support phase.

(b) Left single support phase; (c) right single support phase.

Figure 7: Modelling the robot's walking

Second, the experiments are conducted in the case that the variance of the measurement noise is 2.0 and 10.0.

The trajectory of ZMP as well as the transformed COM with difference variances is shown in figures 8a and 8b. The thick line shows the desired trajectory of ZMP, the thin line shows the measured trajectory of ZMP, and finally dot line shows the trajectory of COM. The displacement between desired value and measured value of ZMP is shown in figures 8c and 8d. The experimental result using Kalman filter is shown by the dark colour, but the experimental result not using the Kalman filter is shown by the light colour. From the experimental results, the proposed algorithm works in poor environment robustly.



(a) ZMP and COM trajectory (variance = 2.0); (b) ZMP and COM trajectory (variance = 10.0).

(c) Desired value and measurement error (var. = 2.0); (d) desired value and measurement (var. = 10.0)

Figure 8: Experimental results with limited noise.

Finally, the experiment of the impulse measurement noise with variance (50) is conducted. Figure 9 shows the displacements of each joint angle when impulse measurement noise exists.

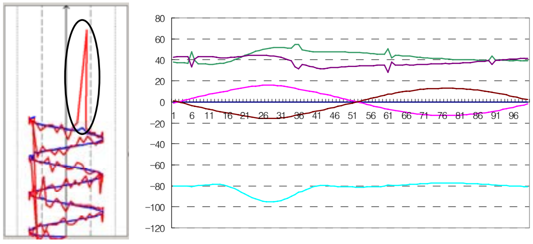


Figure 9: Displacement of joint angle in impulse measurement noise.

The performance of the proposed algorithm has been evaluated by all the previous experiments. Even through the experiment is conducted by the simulation, al1 the experiments have been conducted in conditions similar to real environment. That is, the stability of the robot walking is tested with an assumption that the COM calculated from the ZMP includes a white Gaussian noise. Considering experiments, the proposed method allows the biped robot to be stable in the poor environment.

5 Conclusion

Biped robots with multiple joints may fall in unstable state when using the ZMP based balance control technique due to an approximation of the robot dynamics and the time delay included in the process of error compensation. This paper solved the first problem, caused by approximated modelling, by adaptively compensating the states of 3D-LIPM using the Kalman filter. To solve the second problem caused by the time delay, this paper determined the ZMP using the state of 3D-LIPM predicted by the Kalman filter. The performance of the proposed algorithm has been evaluated using the biped robot with 24 joints which was designed for the experiments. The stability of the robot walking was first tested using the 3D simulator with an assumption that the ZMP calculated from the COM includes a white Gaussian noise. The experimental results have shown that the proposed algorithm improves the stability of robot walking

6 References

[1] M. Vukobratovic and B. Borovac, "Zero-moment point: thirty five years of its life", *International Journal of Humanoid Robotics*, 1(1), pp 157-173 (2004).

[2] K. Erbatur, A. Okazaki, K. Obiya, T. Takahashi and A. Kawamura, "A study on the zero moment point measurement for biped walking robots", *International Workshop on Advanced Motion Control*, 7, pp 431-436 (2002).

[3] T. Arakawa and T. Fukuda, "Natural motion trajectory generation of biped locomotion robot using genetic algorithm through energy optimization", *Proceedings 1996 IEEE International Conference on Man and Cybernetics*, Beijing, pp 1495-1500 (1996).

[4] Napoleon, S. Nakaura and M. Sampei, "Performance limitation on control systems: an analysis to ZMP control of biped," *Proceedings IECI Japan Workshop*, Tokyo, (2003).

[5] T. Sugihara, Y. Nakamura and H. Inoue, "Realtime humanoid motion generation through ZMP manipulation based on inverted pendulum control," *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, Washington, D.C., pp 1404-1409 (2002).

[6] Q. Huang, S. Kajita, N. Koyachi, K. Kaneko, K. Yokoi, H. Arai, K. Komoriya and K. Tanie, "A high stability, smooth walking pattern for a biped robot," Proceedings IEEE International Conference on Robotics and Automation (ICRA), Detroit, pp 65-71 (1999).

[7] Q. Huang, G. Wang and K. Li, "Sensory reflex control for humanoid walking", *Proceedings 4th World Congress on Intelligent Control and Automation*, Rome, pp 3256-3260 (2002).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Mobile Robot Self-Localisation Using Adaptive Data Fusion Structure

Pooyan Khajehpour, Behzad Moshiri and Babak N. Araabi
Control and Intelligent Processing Centre of Excellence
School of Electrical and Computer Engineering,
University of Tehran, Tehran, Iran
p.khajehpour@ece.ut.ac.ir, moshiri@ece.ut.ac.ir, araabi@ece.ut.ac.ir

Abstract

In this paper, the mobile robot self-localisation problem is selected as a challenging test-bed for a novel adaptive data fusion structure. In the presented structure, the fusion process is adapted to the sensory inputs, in order to obtain an accurate perception from a highly noisy environment. This adaptation emerges in two major steps; firstly, in a sensor selection procedure and then gradually in a model adaptation process. Actually, the more incorrect data are removed, the more reliable localisation results are gained. In addition, as more precise sensor models are employed, less amount of error appears in the final consequences. The simulation results prove effectiveness of the proposed structure, especially in comparison to the conventional localisation methods which work based on the information theory.

Keywords: sensor selection, model adaptation, adaptive data fusion, and self-localisation

1 Introduction

In order to make rational decisions, an intelligent system must gather information about its situation in environment from the various sensors, as human beings wield their different senses. Therefore dealing with the sensory data is a necessity and one of the most challenging problems in the AI topics. Actually, raw sensory information is not applicable, so an especial process is needed to merge them and extract the true state of the system in the environment. This process is called Data Fusion, which combines the sensory data to make an integrated perception.

There are different data fusion methods which are mostly quite old. Due to their strong mathematical foundation, practical applying of these methods needs current high-speed processors. This is the main reason why data fusion has become a relatively new field. Despite all efforts to find a generally efficient data fusion method, the previously proposed methods have not been so effectual in face of noise, so far. This is due to the presumptions about the noise model and lack of sufficient adaptation during a fusion process. In this paper, a novel adaptive data fusion structure, ADFS, is introduced as a general solution, to facilitate any data fusion system to be capable to handle noisy input data.

Initially, the literature of the data fusion methods is reviewed in the next section, and the reasons why the previous methods are not completely successful in noisy environments, are explained. Then in section 3, Sensor Selection methods are explicated, as the first step to have an efficient data fusion method, and the relative works are studied. In section 4, a novel adaptive data fusion structure (ADFS) is presented in details. The self-localisation of a mobile robot in a highly noisy sensor networks has been chosen as a challenging problem, to demonstrate the effectiveness of this structure. The way how one can exert ADFS in this problem and the similar ones, is explained in Section 5. In section 6, the results of the simulation are demonstrated separately for each part of ADFS. In addition, in that section, the results of the famous information theory based fusion methods are compared to prove the relative effectiveness. Finally, the last section is the summary and conclusion of this paper, in addition to the suggested future works.

2 Data Fusion: A Brief Review

Different approaches to data fusion have been proposed in the literature. The main approaches can be separated as stated below:

- Bayesian fusion and probabilistic methods
- Dempster-Shafer rule of combination and the methods based on the theory of evidence
- fuzzy fusion methods
- artificial neural networks for data fusion
- biological and cognitive models for data fusion

The oldest approach to data fusion, and the one with the strongest foundation, is the Bayesian theory. The applications using this theory, deal with the probabilities of events happening, which can be achieved by statistics. Due to its strong mathematical foundation, this theory has been enormously around in the literature of data fusion [1], [2], [3]. In contrast to the probability theory, which assumes something is either true or not, the Dempster-Shafer theory embraces the uncertainty [4]. Many fusion methods have been proposed in this framework, for example Yager, Iangaki, Zhang and Dubois rules of combination [5]. Relatively, the fusion methods based on fuzzy logic [6], [7], or neural networks [8] are not widely represented in the recent literature. Finally, there are many proposed fusion models having either biological or cognitive foundations, for example SFX [9], active memory model [10] and democratic integration [11]. Moreover, various theories have been presented, namely Mark's unity of the senses [12], and Bower's taxonomy of fusion modes [13].

In spite of all efforts to propose a suitable fusion method, there are many criticisms to their functionality [14]. Though the methods employing mathematical theories theoretically present more efficiency, which is because of their strong mathematical support, in a real environment their ability to deal with noise are inferior [15]. The main reason of this weakness is lack of sufficient adaptation. In contrast the biological and cognitive models are relatively more adaptive, because they are inspired from the brain, which is known as the most adaptive system. But due to lack of generalisation in matching with different problems, also these methods have not been so successful.

Usually, sensor models are presumed in data fusion methods, although assuming a suitable model with precisely constant parameters is impossible. Lack of adaptation in treating with the presumed models, is the first reason why the previous methods are not completely successful. There are some relatively recent works in the literature which concern on this issue [16]. To address this issue, in this paper, the Model Adaptation technique is proposed, which can be added to the end of the fusion process to learn the sensor models and their parameters simultaneously during the process.

The second reason why fusion methods are not effective in noisy environments is related to the fact that in most of them, all data are assumed to be useful and always should be used in the fusion process. In the next section, as a general solution, Sensor Selection methods are suggested, which can be added to any desired fusion system as the first step, to detect the incorrect sensory data and eliminate them. Therefore, after this phase, the aforementioned assumption about the correctness of data will be proper and the system can use all of its sensory data.

3 Sensor Selection Literature

There are different methods proposed for sensor selection in the data fusion literature, which follow various approaches. These approaches can be roughly divided into some groups as stated below:

- Reducing the cost of communication [17]
- Removing redundancies in the feature selection process [18], namely PCA [19]
- Selecting more effective data [20]
- Removing incorrect information, which is the first step in ADFS

In fact, there are various points of view, for example, using the possibility measure [8] or geometrical solutions [17], [21], although in comparison to other methods, it is very usual to use the information theory as a measure of importance, correctness or information gain [22], which is used frequently in the tracking problems. In [23] the measure of mutual information between estimated probability density of target location and the likelihood functions of sensor observations is used to select the most informative sensors. In addition, two suitable sensor models, for acoustic-amplitude distance sensor and direction-of-arrival sensor are presented. In [24] several measures are introduced and compared for sensor selection, for example the nearest neighbour method, Mahalanobis distance, entropy measure, or relative entropy. In [3] the mutual information measure is used again in the array of acoustic sensors. Then by assuming Gaussian noise model, the result of the selection is estimated by calculating the expected value of the mutual information. Finally, in [25] a fast numerical heuristic method is proposed to estimate the expected values of mutual information functions.

4 Adaptive Data Fusion Structure

Bower [13] decomposed sensor fusion into a four-level taxonomy: I. Complete sensor unity; II. Unity with the possibility of recalibration; III. Unity with tendency toward suppression; and IV. No unity at all.

ADFS is designed based on this taxonomy. Firstly, the sensor selection procedure removes the incorrect sensory data, which is included in level III. If one assumes more than one category for the input data, the system will work in level IV, and clusters the united data together, for instance in the multiple objects tracking problem. Then the remainders of data are combined together in the normal mode of operation belonging to level I. Finally, during the model adaptation process, the sensor models used in the fusion process are adapted, which exhibits the recalibration in level II. In this way, ADFS is constructed to provide the required adaptation and robustness for any data fusion method (figure 1).

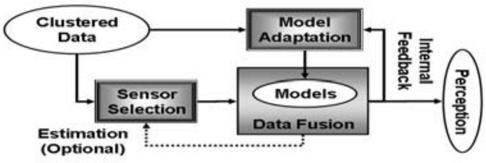


Figure 1: ADFS configuration.

4.1 Sensor Selection Method

In this subsection, two different approaches are introduced as the sensor selection scheme. In the first approach, each sensory data value is compared to the cue resulted from the fusion of all remained sensory data, including this sensory data value. In contrast in the second approach, the difference of each sensory data value, with the cue resulted from the fusion of other remained sensory data, excluding this one, is considered as a measure. An essential assumption is required: at each instance, the concentration of the cue around the true answer must be denser than the concentration around any other point. If sensors are agreed on a misleading answer, there will be no way to show its incorrectness, however, another idea which can be beneficial in such a situation, will be proposed later in this subsection. The proposed algorithm for both approaches is shown below:

```
S = $s_1 \ldots s_N$; Flag = True;
While Flag And (|S| > Min_Need)
  Switch Approach
  Case 1:
    A = Fusion(S + Est);
    If Max Error($s_i$|A) > Thr * Min Error($s_i$|A) Then
      S = S – Arg Max Error($s_i$|A);
    Else Flag = False;
  Case 2:
    For each $s_i$ in S:  $A_i$ = Fusion(S – $s_i$ + Est);
    If Max Error($s_i$|$A_i$) > Thr * Min Error($s_i$|$A_i$) Then
      S = S – Arg Max Error($s_i$|$A_i$);
    Else Flag = False;
  End Switch
End While
```

In this algorithm, Fusion(S) returns the fusion result for the elements in S, and Error(s|A) shows the error of a sensory data value s, when A is assumed as the true answer. Initially, the selected set S is initiated with all of N sensory data value $s_1 \ldots s_N$. Then at each step, one detected incorrect sensory datum is removed from S. This process continues till the selected sensory data concentrate on one answer (based on the value of Thr), or the number of the remaining sensors becomes equal to the minimum required number of sensory data needed to find an exact answer (Min_Need). Some data fusion methods use estimation, namely Bayesian fusion. This estimation (Est), if it is available, is used to bias the answer, which weakens the erroneous sensory data to help the algorithm finding the correct answer; especially when the essential assumption, which is mentioned at the beginning of this section, has not been satisfied.

4.2 Model Adaptation Process

As mentioned in section 2, in many data fusion methods, presumed mathematical functions are used as the sensor models. In this section, the Model Adaptation method is proposed to adapt these models to the actual ones. In fact, the system learns the models during the process, by supposing the fusion results (internal feedbacks) are mostly correct. The proposed algorithm is very simple; at first, an initial guess is needed for the models; then at each time step, the selected data are fused to obtain an answer, which is considered as the true answer, and finally this answer, in addition to all of the sensory data is used, as supervised data samples, in a machine learning process to find an actual model for each sensor. To find the parameters used in the mathematical function, any machine learning method can be employed here, namely maximum likelihood, MAP, or expectation maximisation [19]; even the whole sensor model can be gained by artificial neural networks or numerical methods like Parzen window and nearest neighbour estimation. Details are expressed in the next section.

5 Using ADFS in Self-Localisation

The self-localisation problem has been chosen, as it is in RoboCup soccer middle-size robot league: each robot has an Omni-directional camera to observe the environment. There are some known landmarks in the competition field, thus the robot can calculate its location by measuring its distance to these landmarks. The observed distances between the robot and the landmarks are considered as the sensory data values.

Each robot has a central processing unit to gather the sensory data and combine them with the beliefs of the robot to find its exact location. This is a very common problem in the field of mobile robot navigation, although it is not so simple. Actually, using vision to collecting sensory information has many difficulties, which is due to the unknown objects appearing in the images and the high amount of noise.

The observation of the landmark i at time t, is the distance of R_i^t when the real distance is assumed as d_i^t. In the simulation, this observation is modelled by a Gaussian PDF $P_i(R_i^t \big| d_i^t) = N(R_i^t, d_i^t + \Delta_i, \alpha_i d_i^t + \beta_i)$ where α_i and β_i show the relative and constant error factors respectively and Δ_i is the error offset. A similar model is assumed for the encoder data.

In (2), the Bayesian fusion method is applied to combine the sensory data, with the encoder odometry data, which is used for estimation as stated in (4). By assuming the independencies between likelihood functions and Markov evolution in (2) can be solved by (3), and finally the expected robot location $\hat{x}^t$ can be calculated from (1).

$$\begin{aligned} &\mathit{Expected\ Location} \;=\; \hat{x}^t = E\left\{P\left(x^t \middle| \overline{R^t}\right)\right\} \\ &\mathit{where}\;\; \overline{R^t} = \left\{R^1, R^2, \ldots, R^t\right\} \end{aligned} \tag{1}$$

$$P\left(x^t \middle| \overline{R^t}\right) = P\left(R^t \middle| x^t\right).P\left(x^t \middle| \overline{R^{t-1}}\right) \div P\left(R^t \middle| \overline{R^{t-1}}\right) = \frac{Likelihood \cdot Estimation}{Normalization} \tag{2}$$

$$P\left(R^t \middle| x^t\right) = \prod_{s_i \in S} P_i\left(R_i^t \middle| x^t\right) = \prod_{s_i \in S} P_i\left(R_i^t \middle| d_i^t\right) \quad where \;\; d_i^t = \left\| x^t - X_i \right\| \tag{3}$$

$$P\left(x^t \middle| \overline{R^{t-1}}\right) = \int P\left(x^t \middle| x^{t-1}\right) \cdot P\left(x^{t-1} \middle| \overline{R^{t-1}}\right) \cdot dx^{t-1} = \int Odometry \cdot Old\ Distribution \tag{4}$$

In (3), X_i is the position of the sensor i, and R^t is the set of the selected observations, at time t. All distributions are modelled by discrete arrays. The higher precision is needed in quantisation, the more complexity should be handled in the calculations, thus many heuristic algorithms and numerical techniques for approximation are used in this simulation to make it as fast as possible.

5.1 Sensor Selection Technique

To use the sensor selection method with both aforementioned approaches, the two functions of Fusion and Error are expressed in (5) and (6) respectively. In addition, the value of Thr is set to 1.5 and Min_Need = 3, because at least three measured distances are needed for triangulation. By supposing R as the real distance, $\sigma_i(R)$ is the standard deviation for the i-th sensor model, which is learned during the Model Adaptation process. The higher amount for $\sigma_i(R)$, the less the i-th sensor should be able to affect the final result. Actually, more accuracy is expected from a sensor with littler $\sigma_i(R)$, thus in (4), the amount of difference between a sensory data value and the integration of the others, is divided by its related $\sigma_i(R)$ to satisfy this expectation.

$$Fusion(S) = \frac{E\left\{ P\left(x^t \middle| \overline{R^{t-1}}\right) \cdot \prod_{s_i \in S} P_i\left(R_i^t \middle| x^t\right) \right\}}{Normalization\ \ Factor} \tag{5}$$

$$Error\left(R_i^t \middle| x\right) = Abs\left(R_i^t - \left\| x - X_i \right\|\right) \div \sigma_i\left(R_i^t\right) \tag{6}$$

5.2 Model Adaptation Technique

In this phase the initial functions used for sensor models are adapted to the actual models by means of a learning method. The main idea used in this method, is inspired from the Parzen Window method. Due to the small number of data samples, a softening windows is used for interpolation and compensating the lack of data samples. The proposed algorithm has an updating loop for each of the N sensory input, which is shown in the following:

```
For i = 1 to N
  $\forall r, d$ : Tmp(r,d) = Models(r,d,i) * SampleCnt(d,i)
  Tmp(.,.) = Tmp(.,.) + SoftWin($s_i$,$d_i$)   // updating
  For each plausible real distance d
      SampleCnt(d,i) = sum(Tmp(.,d));
      Models(.,d,i)= Tmp(.,d) / SampleCnt(d,i);
      $\sigma_i(d)$ = Std(Models(.,d,i));
  End For
End For
```

For each sensor i, with the observation R_i from the true distance d_i, the value of the cell Models(R_i,d_i,i) in the three-dimensional array Model(.,.,i) represents the probability of $P_i(R_i|d_i)$ to model the observations. Also, the two-dimensional array SampleCnt(d,i) is considered, to count the number of the samples which have been used to learn the distribution $P_i(.|d)$. For initialisation, this array is filled with the constant number Init_Sample, and Model(.,.,.) is initiated with an arbitrary distribution. The larger Init_Sample value, the slower learning rate, although the learning process would be more stable during the process. In this algorithm Models(.,.,i) is a continuous function; thus instead of adding one supervised data sample per one internal feedback data value, a softening window (SoftWin) is created around the main data sample and added to the past samples in its neighbourhood. Finally, Std(P) returns the standard deviation of distribution P to use further in the Error function.

6 Results of Simulation

The environment is a 10m x 6m square with 6 known landmarks all around it. Sensor models are assumed to work with different parameters (table 1) by the mentioned Gaussian model in section 5. To show the effectiveness of the proposed structure, high amount of noise is considered in simulation. Also, for odometry, the encoder models are assumed similar to the other sensor models, with $\alpha = 0.1, \beta = 0.02, \Delta = 0$.

Table 1: Parameters of the observation models.

i	1	2	3	4	5	6
x	-6	-6	+6	+6	0	0
y	-4	+4	-4	+4	-4	+4
α	0.01	0.01	0.01	0.01	0.01	0.1
β	0.05	0.05	0.05	0.05	0.5	0.05
Δ	0	0	0	1	0	0

6.1 Sensor Selection Results

A simple path with 100 set-points is considered as the main trajectory for the robot. At each set-point, the robot observes its environment and detects the landmarks to collect the sensory data. The resulted errors of six different methods are shown in figure 2. The error of *odometry* system is accumulative, which is expected. The amount of noise can be seen in *Simple Bayesian*, which proves the malfunction of this method to deal with noise.

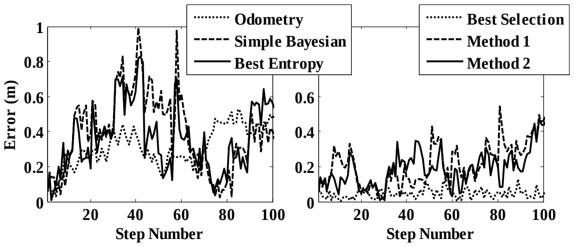


Figure 2: Results of the sensor selection procedure for different methods.

Moreover, *Best Entropy*, which has been obtained by selecting the sensors to achieve the best entropy, shows its deficiency in face of noise. The results of the proposed methods, which are very close to *Best Selection*, show their relative effectiveness. *Best Selection* has been obtained by selecting the sensors to achieve the best error when the location of the robot is known. Obviously, such accuracy is not expected from the proposed methods.

6.2 Model Adaptation Results

To apply the model adaptation, the distribution functions with parameters $\alpha = 0.01, \beta = 0.05, \Delta = 0$ are considered as the sensor models. In addition, Init_Sample is assumed 20. The initial models and the results of adaptation are shown in figure 3. Referring to the results, Models(.,6.2,4) has a second peak in 7.2 with the same distribution as its initial guess. Similarly Models(.,3.7,6) has a dominant peak in 3.7 with a wider distribution than its initial distribution. Both of these features could be expected from table 1, and show the proposed model adaptation process detects the error offsets and forces models array to converge to the true models.

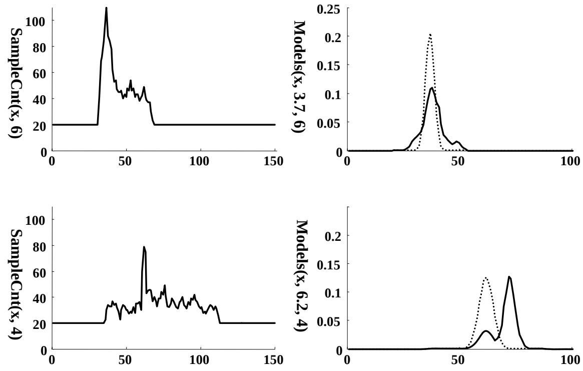


Figure 3: Two results of the model adaptation; initial distribution are plotted with dots.

6.3 Final Results of Self-Localisation

Bayesian data fusion without sensor selection or with Entropy based selection neither is able to handle the error offsets, nor tolerates high error magnitudes in the sensory data. This is the main reason why highly noisy sensors have not been considered in this phase of the simulation. In addition, these methods need to know the initial position of the robot in the field. In contrast, both of the proposed methods in ADFS dramatically demonstrate their independency to the magnitude of the noise and also they are capable to estimate the initial position of the robot.

The final results are shown in figure 4. The average errors of the proposed methods are better than the other methods, though in some instances the results of the other methods are slightly better. By comparing the results of the two proposed approaches for sensor selection, the second one demonstrates the better average result, though the first one has less complexity, and thus it is faster.

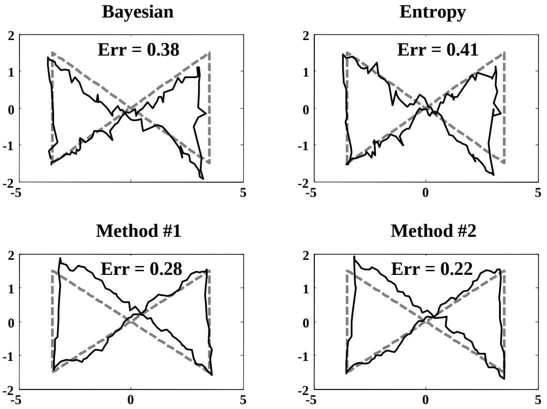


Figure 4: Results of the localisation algorithms and the amount of their average errors in metres.

7 Conclusion

In order to compensate the weaknesses of the conventional data fusion methods in face of noise, adaptive data fusion structure (ADFS) was proposed as a novel solution. In this solution, two different sensor selection methods were introduced, which are able to support any data fusion method, to detect the incorrect sensory data and eliminate them, even if the sensory data values belong to more than one data categories. Moreover, a new model adaptation process was presented, in which sensor models are learned during the fusion process. The self-localisation problem was selected as a challenging test-bed for the proposed structure, and by simulation, effectiveness of ADFS and its relative superiority against the conventional methods using information theory, was clearly demonstrated.

Since the inputs are the temporal signals, considering the continuity along the time axis might be helpful to find the better answers in the sensor selection procedure, for the future works. Also, ADFS can be applied to the other aspects of fusion, namely decision fusion or action fusion, and various test-beds can be used to show the generality of ADFS.

8 References

[1] B. Moshiri, M.R. Asharif and R. HoseinNezhad, “Pseudo information measure: a new concept for extension of Bayesian fusion in robotic map building”, *Information Fusion*, 3, pp 51-68 (2002).

[2] S. Thrun, “Probabilistic algorithms in robotics”, *AI Magazine*, 21(4), pp 93-106 (2000).

[3] E. Ertin, J.W. Fisher and L.C. Potter, “Maximum mutual information principle for dynamic Sensor query problems”, *Lecture Notes in Computer Science*, 2634, pp 405-416 (2003).

[4] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton (1976).

[5] K. Sentz and S. Ferson, *Combination of Evidence in Dempster-Shafer Theory*, SANDIA Report, SAND2002-0835, Springfield (2002).

[6] R.R. Yager, “Intelligent decision making and information fusion”, *Intelligent Systems*, 4, pp 1-4 (2004).

[7] M.A. Simard, E. Lefebvre and C. Helleur, “Multi-source information fusion applied to ship identification for the recognized maritime picture”, *Sensor Fusion: Architectures, Algorithms and Applications*, 4, pp 67-78 (2000).

[8] F. Kobayashi, F. Arai and F. Fukuda, “Sensor selection by reliability based on possibility measure”, *Robotics and Automation*, pp 2614-2619 (1999).

[9] R.R. Murphy, “Biological and cognitive foundations of intelligent sensor fusion”, *IEEE Transactions on Systems, Man and Cybernetics - Part A*, 26(1), pp 42-51 (1996).

[10] S. Wrede, M. Hanheide, C. Bauckhage and G. Sagerer, “An active memory as a model for information fusion”, *Proceedings International Conference on Information Fusion*, Stockholm, 1, pp 198-205 (2004).

[11] J. Triesch and C. von der Malsburg, “Democratic integration: self-organized integration of adaptive cues”, *Neural Computation*, 13, pp 2049–2074 (2001).

[12] L.E. Marks, *The Unity of the Senses: Interrelations among the Modalities*, Academic Press, New York (1978).

[13] T.G.R. Bower, “The evolution of sensory systems”, in R.B. MacLeod and H.L. Pick Jr. (eds), *Perception: Essays in Honor of James J. Gibson*, Cornell University Press, Ithaca, pp 141-153 (1974).

[14] D. Hall and A. Garga, “Pitfalls in data fusion (and how to avoid them)”, *Proceedings 2nd International Conference on Information Fusion*, 1, Sunnyvale, pp 429–436 (1999).

[15] D. Koks and S. Challa, *An Introduction to Bayesian and Dempster-Shafer Data Fusion*, DSTO-TR-1436, Department of Defence, Australia, Edinburgh (2005).

[16] M. Spengler and B. Schiele, “Toward robust multi-cue integration for visual tracking”, *Machine Vision and Application*, 14, pp 50-58 (2003).

[17] V. Isler and R. Bajcsy, “The sensor selection problem for bounded uncertainty sensing models”, *Proceedings 4th IEEE International Symposium on Information Processing in Sensor Networks*, Los Angeles, pp 151-158 (2005).

[18] D. Koller and M. Sahami, “Toward optimal feature selection”, *Proceedings 13th International Conference on Machine Learning (ICML'96)*, San Francisco, pp 284-292 (1996).

[19] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification* (2nd edition), Wiley Press, New York (2001).

[20] S.C.A. Thomopoulos, “Sensor selectivity and intelligent data fusion”, *Proceedings IEEE International Conference on Multi-sensor Fusion and Integration for Intelligent Systems*, Las Vegas, pp 529-537 (1994).

[21] C. Giraud and B. Jouvencel, “Sensor selection: a geometrical approach”, *Proceedings IEEE International Conference on Intelligent Robots and Systems*, Pittsburgh, pp 555-560 (1995).

[22] K.J. Hintz, “A measure to the information gain attributable to cueing”, *IEEE Transactions on Systems, Man and Cybernetics*, 21(2), pp 434-442 (1991).

[23] J. Liu, J. Reich and F. Zhao, “Collaborative in-network processing for target tracking”, *Journal on Applied Signal Processing*, 4, pp 378-791 (2003).

[24] F. Zhao, J. Shin and J. Reich, “Information-driven dynamic sensor collaboration for tracking applications”, *Signal Processing Magazine*, 19(2), pp 61-72 (2002).

[25] H. Wang, K. Yao, G. Pottie and D. Estrin, “Entropy based sensor selection heuristic for target localization”, *Proceedings 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, Berkeley, pp 36-45 (2004).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Real-time Visual Vehicle Tracking for Autonomous Convoy Control

Hans de Ruiter and Beno Benhabib
Department of Mechanical and Industrial Engineering,
University of Toronto, Toronto, Ontario, Canada.
deruiter@mie.utoronto.ca and beno@mie.utoronto.ca

Abstract

Driving vehicles autonomously in convoys could potentially reduce traffic congestion by safely reducing the separation distance between vehicles. This paper presents a novel 6-DOF visual object-tracking system for fully autonomous convoy following. The trajectory of a lead vehicle is obtained using the object tracker. This trajectory data is processed to extract control inputs that are used by a controller to adjust both the velocity and steering direction of the following vehicle. Simulations of a two-vehicle convoy confirmed the viability of our method. One difficulty with using visual object trackers for control is obtaining accurate information at high enough rates (i.e., in real-time). In the simulations, the tracking algorithm obtained high accuracies whilst operating at 33 to 35 Hz. The tracking algorithm itself has also been tested using real objects under various lighting conditions.

Keywords: Computer vision, object tracking, 3D pose tracking, autonomous vehicle control

1 Introduction

Studies by Hochstadter and Cremer [1], Wu et al. [2], and others, have shown that autonomous vehicles driving in convoys have the potential to reduce traffic congestion. This is due to convoy control systems having a faster response time than would human drivers, allowing vehicles to safely travel closer together. Such convoys could also allow single-driver freight transport both on the road, and in industrial settings. A human driver would control the lead vehicle whilst the autonomous vehicles follow in a convoy.

A key problem for fully autonomous convoying (also called platooning) is tracking the vehicle in front (i.e., the lead vehicle). Possible techniques include radar [3], linear (1D) cameras [4] and computer vision (using 2D cameras) [5]. Computer vision could potentially provide more information than radar or linear sensors, without requiring special templates mounted on all vehicles. However, vision algorithms are generally computationally intensive, making it difficult to extract the required data from image sequences both accurately and in real-time.

In order to achieve real-time operation, past vision algorithms have restricted measurements to 2D position only ([6, 7]) or relied on known object size and symmetry [8]. Such restrictions limit the usefulness of vision algorithms. More recent algorithms use feature-based methods ([9]-[13]). These methods match features in an image, such as points or lines, to corresponding features in a model. Such techniques have been used successfully for both robotics [14] and vehicle tracking [15] applications. Feature-based tracking is prone to feature detection failure. It also limits the system to use a small sub-set of the available visible data (e.g., corners or line-segments). Moreover, complex objects and patterns cannot be modelled effectively.

Marapane et al. [16] describe a convoy system that uses stereo cameras to obtain 3D position at 15 Hz. However, the core of their technique is a 2D tracking algorithm and no orientation is obtained. Orientation information is useful for a convoy control system as it provides the lead vehicle's direction of travel without resorting to differentiation, which is sensitive to noise. Benhimane et al. [5] use a homography-based tracker that also extracts orientation. Their technique requires a template that must be planar (due to being homography based).

This paper presents a novel, vision-based object-tracking method for convoy control. The proposed object-tracker tracks the full six DOF pose (3D position and orientation) of the lead vehicle, in real-time (at over 30 Hz), using a single camera. This facilitates full autonomous control of both speed and steering, without the limitations discussed above. Whilst only 2D information (three DOF pose) may be required, the additional three DOF could be used to extract the terrain contour along the desired trajectory.

Unlike the feature-based tracking methods mentioned above, the proposed object tracker's model is composed of a tesselated surface and texture-maps containing surface features (such as coloured markings). Hence, all available visible surface features of the target object are used for tracking, rather than a set of features. The key advantages over Benhimane et al.'s method [5] are: that no template needs to be mounted on the vehicle, the whole vehicle is used for tracking, and, it follows the lead vehicle's path rather than heading straight for the lead vehicle. Following the lead vehicle's path avoids cutting corners and hitting objects/pedestrians.

2 Proposed Methodology

Our proposed methodology uses a six DOF visual object tracker to follow a lead vehicle. The tracker provides the lead-vehicle's trajectory as a series of poses for each time-step. The control system uses this data to follow the desired trajectory whilst maintaining constant separation distance.

2.1 The Visual Pose Tracker

Tracking is achieved using a novel visual 3D model-based approach [17, 18]. It performs the following steps:

- Predict the pose of the target at the current time-step,
- Project the target object model onto the image plane(s) at its current estimated pose using OpenGL [19],
- Find an improved estimate by using optical-flow to calculate the motion between the virtual, projected image and the actual input image, and
- Advance to the next time-step and repeat the procedure.

3D pose is defined herein by

$$\mathbf{P}(t) = \begin{bmatrix} \mathbf{R}(t) & \boldsymbol{p}(t) \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \qquad (1)$$

where $\mathbf{R}(t)$ is a (3x3) rotation matrix representing the target's (i.e., lead vehicle's) orientation at time t and $\boldsymbol{p}(t)$ is its position.

2.1.1 Camera Projection

Depth information is lost as part of the projection from 3D world to 2D image planes. This projection is defined by

$$\boldsymbol{p}' = \mathbf{M}_{int}\mathbf{M}_{ext}\boldsymbol{p}, \;\; \boldsymbol{p}' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix}, \text{ and } \boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}, \quad (2)$$

where $\boldsymbol{p}'$ is a (3×1) vector denoting the 2D position of the projection of point $\boldsymbol{p}$ in homogeneous coordinates[1]. The (3×4) matrix $\mathbf{M}_{int}$ is based on the camera's internal parameters (focal length, scaling factors, etc.). $\mathbf{M}_{ext}$ is a (4×4) matrix that transforms 3D coordinates between the world frame and the camera's coordinate frame.

The projection is performed using OpenGL as this relieves the main processor from this task, thus, significantly increasing processing speed. OpenGL also provides a depth-map and image mask that are exploited by the tracking algorithm, figure 1.

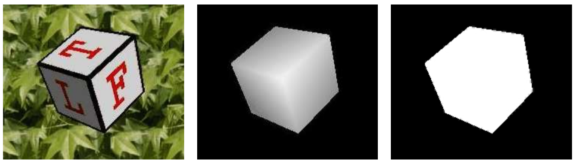


Figure 1: OpenGL is used to generate (a) a virtual-image, (b) depth information in the z-buffer, and (c) a mask in the stencil buffer.

Modern graphics hardware is capable of rendering complex models containing millions of vertices [20]. For example, a Radeon X800 is capable of rendering 700 mega-vertices per second and texture-mapping 6 giga-pixels per second. Moreover, the processing time for tracking is dominated by the optical-flow calculations. Hence, a twofold increase in model complexity should only increase processing time a little.

2.1.2 Optical Flow for 3D Motion

The equation for optical-flow is

$$\begin{bmatrix} \nabla_x I(x,y,t) & \nabla_y I(x,y,t) & \nabla_t I(x,y,t) \end{bmatrix} \begin{bmatrix} v'_x \\ v'_y \\ 1 \end{bmatrix} = 0, \quad (3)$$

where ∇_x and ∇_y are the x and y derivatives respectively and ∇_t is the time derivative. $I(x,y,t)$ is the intensity of pixel $[x\ y]^T$ at time t. Solving (3) for a set of pixels yields the mean 2D velocity for that set of pixels $[v'_x\ v'_y]^T$.

2D on-screen velocity corresponding to 3D motion of a target object can be found by differentiating the projection equation (equation (2)) with respect to time

$$\boldsymbol{v}' = \frac{d\boldsymbol{p}'}{dt} = \frac{1}{p'_z}\mathbf{M_{int}}\mathbf{M_{ext}}(\boldsymbol{v} + \omega \times \boldsymbol{p}), \qquad (4)$$

where $\boldsymbol{v}'$ is the projected 2D velocity corresponding to the linear and angular 3D velocities $\boldsymbol{v}$ and ω, respectively. substituting (4) into (3) yields

$$\begin{bmatrix} \nabla_x I & \nabla_y I & \nabla_t I \end{bmatrix} \left(\frac{\mathbf{M}_{int}\mathbf{M}_{ext}(\boldsymbol{v} + \omega \times \boldsymbol{p})}{p'_z} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) = 0. \quad (5)$$

[1] Homogeneous coordinates are used in projective geometry and are defined up to a scale-factor. Thus, a 2D point is given in homogeneous coordinates as a three element vector in which the third element is set to one.

A problem with the optical-flow equation (equation (5)) is that the angular velocity is calculated about the centre of the world reference frame. Preferably, the angular velocity should be about the target's own origin. To achieve this, one more transformation is required:

$$\boldsymbol{p}_{rel} = \boldsymbol{p} - \boldsymbol{p}_{obj}, \tag{6}$$

where $\boldsymbol{p}_{obj}$ is the target's origin with-respect-to the world reference frame. Replacing $\boldsymbol{p}$ with $\boldsymbol{p}_{rel}$ in (5) yields

$$\begin{bmatrix}\nabla_x I & \nabla_y I & \nabla_t I\end{bmatrix}\left(\frac{\mathbf{M}_{int}\mathbf{M}_{ext}(\boldsymbol{v}+\omega\times\boldsymbol{p}_{rel})}{p'_z}+\begin{bmatrix}0\\0\\1\end{bmatrix}\right) = 0. \tag{7}$$

Generally, equations (5) and (7) cannot be solved since p'_z – the z-axis distance to the object at each pixel – is not known. In our case, however, the value of p'_z is calculated by OpenGL (figure 1b) during model projection. Equations (5) and (7) can, therefore, be solved. The above equations are solved using least-squares for only those pixels that belong to the lead vehicle as indicated by the target-object mask (figure 1c). This novel combination of model projection with optical flow facilitates full 3D pose tracking, even if only one camera is available.

2.2 Autonomous Vehicle Control

The following vehicle's controller takes the pose trajectory given by the visual tracker and outputs linear acceleration and angular velocity commands. A simple discrete-time state-space controller was used herein as the prime focus of this paper is the use of our vision algorithm for the purpose of autonomous convoy control. In practice, a more sophisticated controller could be utilised to achieve efficient and effective performance.

Given a following vehicle (Vehicle i) and a lead vehicle (Vehicle $i-1$), the control law is as follows:

$$\begin{bmatrix}a_i\\ \omega'_i\end{bmatrix} = \begin{bmatrix}c_1(d_i - v_i) + c_2(v_{i-1} - v_i)\\ c_3(\phi_i - \omega_i) + c_4(\psi_i - \omega_i)\end{bmatrix}, \tag{8}$$

where a_i and ω'_i are the linear acceleration and angular velocity commands for Vehicle i, respectively. D_i is the distance between Vehicle i and Vehicle i–1, along the trajectory followed by Vehicle $i-1$. V_i and ω_i are the linear and angular velocities of Vehicle i, respectively. ϕ_i is the rotation between the current direction of travel and the next desired position. ψ denotes the angle between the current 2D orientation (θ_i) and the desired orientation in the next time-step.

2.2.1 Extracting Control Inputs from Tracking Data

The object tracker outputs an array of poses corresponding to the trajectory followed by Vehicle $i-1$ over time. This does not correspond to the control inputs required by the controller.

Let A_i be the set of all N poses of Vehicle $i-1$ between the current positions of Vehicles i and $i-1$:

$$A_i = \left\{\bigcup_{n=1}^{N}\mathbf{P}_{n,i-1}\right\}, \tag{9}$$

where $\mathbf{P}_{n,i-1}$ is the pose of Vehicle $i-1$, n steps away from Vehicle i. The *following distance*, d_i, is calculated using the following equation:

$$d_i = \sum_{n=1}^{N-1}\left\|\boldsymbol{p}_{n,i-1} - \boldsymbol{p}_{n+1,i-1}\right\| + \left\|\boldsymbol{p}_{1,i-1} - \boldsymbol{p}_i\right\|. \tag{10}$$

Above, the 3D positions $\boldsymbol{p}_{n,i}$ and $\boldsymbol{p}_i$ are obtained from the pose matrix (equation (1)). The variable $\boldsymbol{p}_i$ contains the current position of Vehicle i.

The value of ϕ_i is calculated to steer towards a point $\boldsymbol{p}_{j,i-1}$ along the lead vehicle's trajectory that is a set distance ahead. Variable j is a non-negative integer. Thus, ϕ_i is

$$\phi_{=}\angle(\boldsymbol{p}_{j,i-1} - \boldsymbol{p}_i, \boldsymbol{v}_i), \tag{11}$$

where $\boldsymbol{v}_i$ is the current traveling direction of Vehicle i. Likewise, the next desired orientation (θ_d) is taken to be the orientation obtained from $\mathbf{P}_{1,i-1}$. Thus,

$$\psi_i = \theta_d - \theta_i, \tag{12}$$

where θ_i is the 2D orientation of Vehicle i. Once all of the variables above have been calculated, they are passed on to the controller.

2.3 Overview of Complete Convoy Control Algorithm

The overall convoy-control algorithm for the following vehicle (Vehicle i) is as follows:

- Obtain the lead-vehicle's (Vehicle $i-1$) current pose using the visual object tracker,
- Add this pose to the array of lead vehicle poses, A_i,
- Calculate the input values to the controller and execute the control command, and
- Remove all poses from A_i that are now behind the following vehicle (Vehicle i).

Repeat the above procedure for each time-step.

3 Simulation

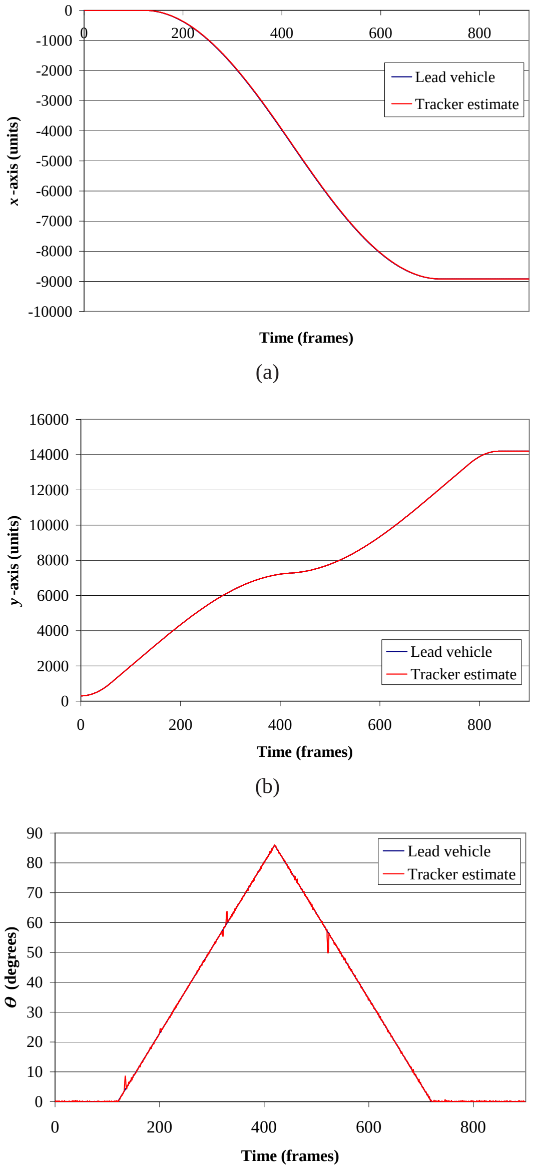
Use of our visual object tracker for convoy control was verified by simulating a two-vehicle convoy. The lead vehicle follows a predefined trajectory including both acceleration and deceleration whilst the following vehicle maintains a required distance using the vision-based control system. The object tracking algorithm itself has been tested using a real camera and target object, and can track objects under various lighting conditions. These results are not included for brevity.

The simulated vehicles were 55×55×55 units in size. The lead vehicle accelerated to 720 units/s in 2 seconds. Following this, it executed a left-hand turn followed immediately by a right hand turn. Finally, the lead vehicle decelerated to a stand-still in 2 seconds. The desired separation distance was set to 300 units between the lead vehicle's centre and the centre of the following vehicle's camera. Synthetic images of the lead-vehicle were generated using OpenGL.

In a real convoy, neither the lead, nor the following vehicle would follow a perfect trajectory. Jitter in these trajectories would manifest itself as noise in the captured images. To simulate this, Gaussian noise was added to the 3D position of the lead vehicle relative to the camera (on the following vehicle) when capturing images. The added noise was set such that 99.73% percent (the $\pm 3\sigma$ limits) of the resulting 2D on-screen errors are within ±4 pixels: this results in frame-to-frame motion of up to ±8 pixels, a large amount of noise for the object-tracker to cope with.

The *x*- and *y*-axis trajectories for the lead vehicle and the proposed tracker's estimates of these are given in figures 2a and 2b, respectively. Figure 2c shows the orientational trajectories. As can be noted, the tracker's estimates follow the lead-vehicle's actual trajectory closely. Figure 3 shows the tracking errors for these trajectories over time. RMS errors for the *x*- and *y*-axis were 0.31 and 0.35 units, respectively; RMS orientational errors were 0.51°. All maximum errors were within acceptable limits. It is noteworthy to remember that the proposed algorithm is also tracking the remaining three spatial DOF. However, these are not shown here as they are not used in this paper to control the following vehicle.

The following vehicle's camera has a resolution of 512×256. This resolution was used since graphics-cards are optimised for processing images with power-of-two dimensions. The vehicle is approximately 200×200 pixels on-screen. Figure 4 shows a sample (synthetic) image of the lead vehicle. The simulated camera in the following vehicle was set to a capture-rate of 30 Hz, corresponding to the standard NTSC video rate. However, the algorithm itself operated at 33 to 35 Hz (much faster than the minimum 10 Hz required). In practise, therefore, the algorithm would be capable of operating at standard video rates. This was achieved using a 2.0 GHz AMD Athlon 64 3000+ with Windows XP and a Radeon X800 Platinum Edition GPU.



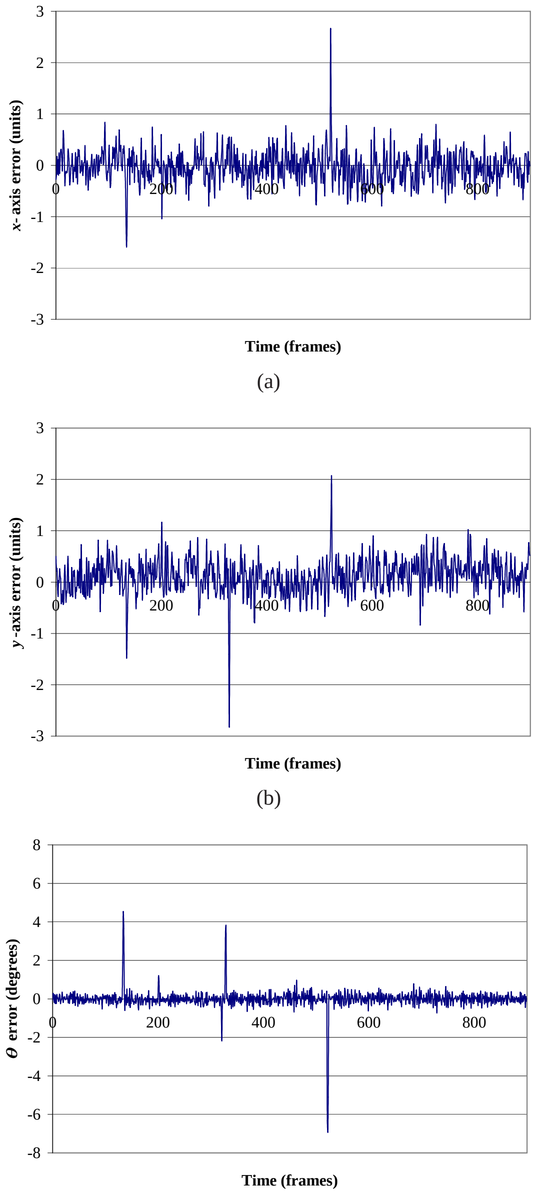
(a)

(b)

(c)

Figure 2: The lead vehicle's (a) *x*, (b) *y*, and (c) θ trajectories.

The many simulations run, one of which is given in this paper, successfully maintained a separation of 300 units to within ±5 units, figure 5. This corresponds to a separation of about 0.4 seconds. Additionally, the perpendicular distance (error) between the following- and lead-vehicle trajectories was less than 3 units, figure 6. This demonstrates the potential of using the proposed visual object-tracking based convoy-control methodology: It can autonomously maintain safe *following distance* and follow the correct trajectory when traveling with a separation of just 0.4 seconds between vehicles.



(a)

(b)

(c)

Figure 3: The (a) x, (b) y, and (c) θ tracking errors.

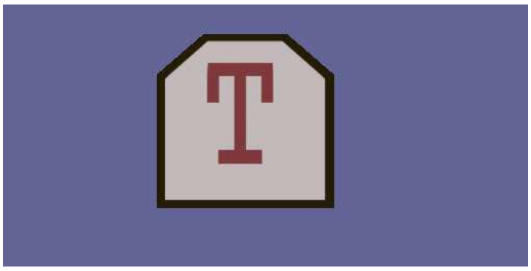


Figure 4: A lead vehicle image as captured by the following-vehicle's camera.

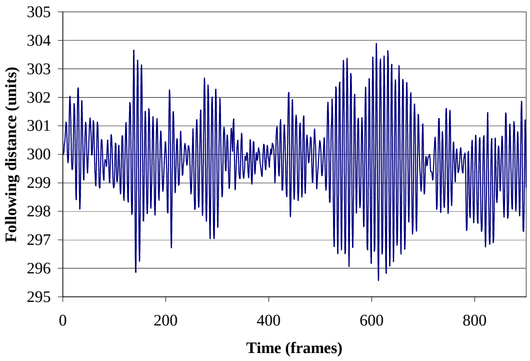


Figure 5: The *following distance* as a function of time.

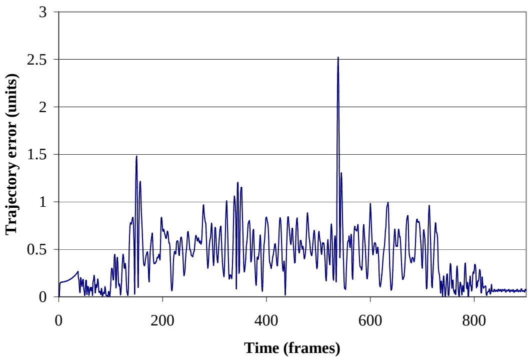


Figure 6: The perpendicular trajectory error.

4 Conclusions

This paper has presented a novel, fully autonomous, convoy-following methodology based on a six DOF visual object-tracking system. It follows a lead vehicle's path whilst maintaining constant following distance. The trajectory of a lead vehicle is obtained using the object tracker. This trajectory data is processed to extract control inputs that are used by a controller to adjust both the velocity and steering direction of the following vehicle. The potential of the proposed methodology was demonstrated by simulating a two-vehicle convoy. The object tracker has also been tested using a real camera and target object. It can track objects under various lighting conditions.

One difficulty with using visual object trackers for control is obtaining accurate information at high enough rates (i.e., in real-time). In the simulations, the lead vehicle's trajectory was estimated with errors of 0.35 units RMS along the x and y-axes, and, 0.51° RMS for the orientation, θ. The algorithm obtained these high tracking accuracies whilst operating at 33 to 35 Hz. Furthermore, the following vehicle matched the lead-vehicle's trajectory to within 3 units perpendicular error whilst maintaining a separation of about 0.4 seconds (300 ± 5 units distance when traveling at 720 units/s). Thus, our visual object-tracking algorithm provides high accuracy and speed, enabling successful control of an autonomous vehicle in a convoy.

5 Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

6 References

[1] A. Hochstadter and M. Cremer, “Investigating the potential of convoy driving for congestion dispersion”, *Proc. IEEE Conf. on Intelligent Transportation Systems*, Boston, MA, pp 735–740, (1997).

[2] J. Wu, M. McDonald, and M. Brackstone, “Effects of convoy driving on motorway flow stability and capacity”, *Proc. Road Transport Information and Control*, London, United Kingdom, pp 91–95, (2000).

[3] C. Cavello, B. Baertlein, and J. Young, “Radar retro-reflective patch for vehicle convoying applications”, *Proc. IEEE Conf. on Intelligent Transportation System*, Boston, MA, pp 667–671, (1997).

[4] P. Daviet and M. Parent, “Longitudinal and lateral servoing of vehicles in a platoon”, *Proc. IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, pp 41–46, (1996).

[5] S. Benhimane, E. Malis, and P. Rives, “Vision-based control for car platooning using homography decomposition”, *Proc. International Conf. on Robotics and Automation*, Barcelona, Spain, pp 2161–2166, (2005).

[6] A. Cretual, F. Chaumette, and P. Bouthemy, “Complex object tracking by visual servoing based on 2d image motion”, *Proc. International Conf. on Pattern Recognition*, Brisbane, Australia, pp 1251–1254, (1998).

[7] L. Hsu and F. Lizarralde, “Robust adaptive visual tracking control: analysis and experiments”, *Proc. IEEE International Conference on Control Applications*, Anchorage, AK, pp 25–27, (2000).

[8] T. Zielke, M. Brauchkmann, and W. von Seelen, “Cartrack: computer vision-based car following”, *Proc. IEEE Workshop on Applications of Computer Vision*, Detroit, MI, pp 156–163, (1992).

[9] F. Jurie and M. Dhome, “Real time robust template matching”, *Proc. 13th British Machine Vision Conf.*, Cardiff, Wales, pp 123–132, (2002).

[10] A. Comport, É. Marchand, and F. Chaumette, “A real-time tracker for markerless augmented reality”, *Proc. IEEE and ACM Int. Symposium on Mixed and Augmented Reality*, Tokyo, Japan, pp 36–45, (2003).

[11] S. Kim and I. Kweon, “Robust model-based 3d object recognition by combining feature matching with tracking”, *Proc. Int. Conf. on Robotics and Automation*, 2, Taipei, Taiwan, pp 2123–2128, (2003).

[12] H. Tang and G. Welch, “Model-based 3D object tracking using an extended-extended Kalman filter and graphics rendered measurements”, *Proc. Computer Vision for Interactive and Intelligent Environment*, Lexington, KY, pp 85–96, (2005).

[13] V. Kyrki and D. Kragic, “Integration of model-based and model-free cues for visual object tracking in 3d”, *Proc. Int. Conf. on Robotics and Automation*, Barcelona, Spain, pp 1554–1560, (2005).

[14] M. Vincze, M. Schlemmer, P. Gemeiner, and M. Ayromlou, “Vision for robotics: a tool for model-based object tracking”, *IEEE Robotics and Automation Magazine*, 12(4), pp 53–64, (2005).

[15] J. Lou, T. Tan, W. Hu, H. Yang, and S. Maybank, “3-d model-based vehicle tracking”, *IEEE Trans. on Image Processing*, 14(10), pp 1561–1569, (2005).

[16] S. Marapane, M. Trivedi, and M. Holder, “Motion control of cooperative robotic teams through visual observation and fuzzy logic control”, *Proc. International Conf. on Robotics and Automation*, Minneapolis, MN, pp 1738–1743, (1996).

[17] H. de Ruiter and B. Benhabib, “Tracking of rigid bodies for autonomous surveillance”, *Proc. IEEE Int. Conf. on Mechatronics and Automation*, Niagara Falls, Canada, 2, pp 928–933, (2005).

[18] H. de Ruiter and B. Benhabib, “Colour-gradient redundancy for real-time spatial pose tracking in autonomous robot navigation”, *Proc. Canadian Conf. on Computer and Robotic Vision*, Québec City, Canada, pp 20–28, (2006).

[19] OpenGL.org, “OpenGL - the industry standard for high performance graphics”, http://www.opengl.org, visited on 10/11/2004.

[20] ATI, “Radeon X800 graphics technology”, http://www.ati.com/products/radeonx800/index.html, visited on 18/07/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Measuring Horizon Angle on a Small Unmanned Air Vehicle using Digital Video Camera and an FPGA

T.D. Cornall, A. Price and G.K. Egan
Department of Electrical and Computer Systems Engineering,
Monash University, Melbourne, Australia
terry.cornall@eng.monash.edu.au

Abstract

Describes a method to extract aircraft pose from visible-spectrum video of the horizon. Presents an implementation based on a Field Programmable Gate Array (FPGA). Discusses some methods of numerically calculating measures of confidence for the results. Relates the results of real-flight tests comparing performance to an alternative system that uses infrared light.

Keywords: UAV, unmanned aircraft, horizon detection, horizon angle, aircraft attitude, atmosphere, sky, ground, image processing

1 Introduction

Vision processing techniques promise to lend themselves well to many autonomous navigation and control tasks, but the limited payload capacity of many Unmanned Aircraft applications is at odds with the high amount of processing that is often needed. Technological advances and increased sales volumes continue to shrink the size, weight and cost of such processors, but still the electrical power and weight constraints militate against complex onboard vision processing systems. The risk of loss and damage to the equipment that comes with the nature of the missions that UAVs may be called upon to perform also requires a low-cost approach, (especially for PhD students). Work carried out by others in this field has often made use of bidirectional radio links to do the video processing on the ground and send back control signals. [1, 2]. At least one other research group has developed video horizon measurement equipment that is simple enough that it could be implemented in a way that could be airborne, [3], which uses a thermal imaging camera or scanned linear array. Similar devices sensing in the infra-red spectrum, using a small number of discrete infra-red sensors, are used in UAV and aerospace applications, [4] [5] [6] for stabilising aircraft and satellites. There are also devices such as mechanical, solid-state and optical rate gyros that are used with great success in inertial guidance systems in many aircraft. Many of these devices are too large, heavy and require too much power to be useful in a UAV context, but some low-cost solid-state inertial measurements systems are well suited to UAV applications and are being used for such. [7].

This article discusses a method that is different from the previously discussed methods in that it uses visible light on a platform light enough to be carried by a small UAV. It has been shown to work with reasonable accuracy in simulation and in open-loop flight. It is small, light and requires low computational power and has been through a number of implementations. In 2004 at ICARA version based on a CMUCAM [8] and a PIC microcontroller was presented. [9] The subject of this article is an improved version based on an FPGA coupled with a digital video camera.

Other researchers have also developed light-weight, specialised visible-light sensors for use with UAVs. For example [10], discusses a VLSI optic flow sensor for Micro UAVs (MAVS) and many, including [11, 12, 13] discuss systems inspired by insect vision for use in flight stabilisation, obstacle avoidance and terrain following.

This article briefly discusses a method to measure the horizon angle using the visible light spectrum and a method that has been developed that calculates a measure of confidence in the result. More detail on the background to these issues can be found in previous articles by the authors [14], [15], [9] and the Australian patent document [16], also by one of the authors, Cornall. The method has been proven in simulation, [14], [15] and more recently in real-flight and the real-flight results will be discussed in depth here.

2 Method

For an image of the horizon in a circular viewport that has been segmented into sky and ground classes the line joining the centroids of the sky and ground classes will bisect the horizon at a right angle. This is true, regardless of the roll angle and of the pitch angle, as long as the horizon makes a straight line in the view,

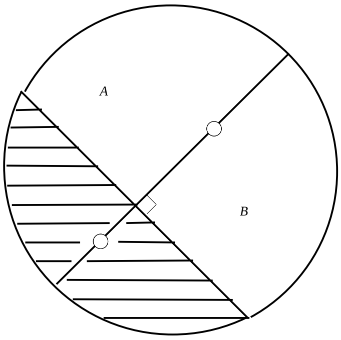


Figure 1: The horizon is perpendicular to the line joining the sky and ground centroids

approximating a chord of the circular viewport. The proof and development of this idea is detailed in [14], [15], [9] and [16]. Figure 1 illustrates the idea, showing the centroids of the classes as small circles joined by a line, the bisector. Because of the circular viewport, the average line of the horizon will cut this bisector at a right angle.

This means that we can find the angle of the horizon simply by measuring the average coordinates of the sky and ground classes to find the centroids. The method has no dependence on the position of the horizon within the view, only on its angle. The measured angle will not change as the horizon moves with perturbations of the camera platform that do not cause a change in the relative angle between the camera horizontal axis and the horizon angle. In other words, disturbances to the pitch and yaw that are not so extreme as to move the horizon out of the view do not have to be explicitly compensated for in the measurement. This simplifies the implementation of the method considerably.

The measurement task imposes a relatively low computational burden on the vision processing system and, most importantly, does not require a frame buffer as all the operations of classifying the pixels and accumulating the average coordinates are local operations. Even the task of applying a circular mask to the image can be done with no frame buffer. The advantage of this is that a relatively simple vision architecture can achieve the task, and it can be done at a fast rate using algorithms of O(N) where N is the number of pixels in the image.

3 Segmentation and Validity

It must be emphasised that *any* method that assumes that the pixel values can be classified into two classes may produce false results when applied to an image that does in fact not contain a horizon. Also, there will be situations where even though a horizon is present, other circumstances, such as dark clouds, or bright ground cover, will produce false results. Whatever segmentation method is used, a measure that can be used to reliably detect unreliable segmentation is needed.

3.1 Otsu's Histogram Analysis

Otsu's algorithm [17] works on a one-dimensional histogram to produce a threshold value that segments the histogram values into two classes in a manner that can be shown to be optimal (in a mean squared error sense) for class separation. For each 'bin' in the histogram it calculates the value of a metric σ_b^2 (equation 1) that would be generated if that level was used as a threshold to segment the image into two classes.

$$\sigma_b^2 = (u_2 - u_1)^2 (\omega_1 \omega_2) \tag{1}$$

Selecting the threshold value that gives the best value of σ_b^2 is a means of segmenting the image into two classes. For image segmentation, this method can be applied to a value that is dependent upon the pixel values of the image in order to segment them into two classes. In the case of this thesis, segmentation into sky and ground classes is desired. In equation 1 u_1 and u_2 are the mean values of the measured variable for each class where ω_1 and ω_2 are the number of pixels in or the probability of belonging to the respective class. The first term in the righthand side of 1 maximises class separation and the second term tends to equalise the size of the classes because is a maximum when $\omega_1 = \omega_2$. The metric σ_b^2 is a measure of the variance between the classes [17]. Over all, maximising σ_b^2 maximises the difference between the values of the pixels belonging to the different classes [17]. The threshold that generates the greatest value of σ_b^2 is the level that creates classes with the largest interclass variance [17]. In general, the higher the value of σ_b^2, the better the segmentation is for the purposes of discerning sky and ground. (This is not an absolute rule however, as false horizons can also generate a high value.) By observing the segmented video and corresponding values of σ_b^2, the author has determined that a low value of σ_b^2 is a good indicator of unreliable segmentation. [14]

Unfortunately, as observed during testing, and as discussed in [14], there are cases of failed classification that are not well indicated by the σ_b^2. This indicates that the σ_b^2 metric by itself is not an adequate indicator of reliability. This is not surprising, as the metric does not take the spatial grouping of the pixels into account at all, merely their blue value, so in marginal cases misclassification will occur undetected.

3.2 Metric2.5

A new metric has been developed by one of the authors (Cornall) to measure the success of a segmentation of an image into sky and ground. Metric2.5 (or M2p5) is measured using the statistics of the classified pixels in a manner similar to Otsu's σ_b^2 but operating on their spatial coordinates, not their pixel values. It combines the mean *spatial* coordinates u_1 and u_2 of the two classes (rather than the mean histogram value as per Otsu) with the populations n_1 and n_2 of each class and the radius R of the circular viewport. The formula for metric2.5 is very similar to that used by Otsu's algorithm and the method could be seen as an extension of that work. The major differences are that it is applied to a two-dimensional variable, the spatial coordinate, and that the product of population terms is de-emphasised:

$$m2p5 = \frac{(u_1 - u_2)^2 (n_1 n_2)^{1/3}}{3R^3} \tag{2}$$

In Eqn. 2 the exponent of $1/3$ applied to the product of the populations is to reduce its weight compared to the separation of the classes given by $(u_1 - u_2)^2$. The $3R^3$ in the denominator is a normalising factor to bring the value down to near 1. The effect of the population product is to increase the metric in favour of classes that have similar sized populations as the product $n_1 n_2$ is a maximum when the populations are the same, because $n_1 + n_2$ is a constant that depends on the radius of the viewport. The term $(u_1 - u_2)^2$ containing the class average coordinates, or centroids, u_1 and u_2, increases as the class separation increases, which favours segmentations that have the sky and ground classes well separated in space, which is clearly desirable because any segmentation wherein the sky contains ground is likely to be suspect. This term could favor segmentations where there is one large class with a centrally located average coordinate and one small class with its average coordinate right on the circumference, but the term $n_1 n_2$ containing the populations discourages this by decreasing as the class sizes become dissimilar. For a constant viewport size, the $3R^3$ is a constant. Again, it is possible to find frames where misclassified pixels are not indicated by a low value for metric2.5. It can be that the segmented classes are distinct and similarly sized so metric2.5 has a relatively high value, even though the pixels classified as ground are in fact due to dark clouds, for example. Note however that in this case σ_b^2 could have a relatively low value, although this is not guaranteed. The combined use of these two metrics will be better than either alone.

3.3 Other Measures

Both metrics, σ_b^2 and M2p5, can inform about the quality of the partition into sky and ground and the validity of the straight-line horizon model. Thresholds can be set that define acceptable limits for the metrics which trials have shown are adequate for a wide range of lighting conditions.

Another two measurements that can be of use in deciding if an image or part of an image is suitable for the purpose of detecting the horizon. These are the average brightness of the image, and the average contrast of the image. It would also be possible to use the average brightness of the ground class and the average brightness of the sky class, as well as contrast measures applied to these parts of the image. In a way, the σ_b^2 and M2p5 metrics do take these aspects into account, but the author sees merit in considering overall image brightness and contrast explicitly as a means of being able to accept or reject the results of the image segmentation.

The difficulty arises when trying to decide what the thresholds for acceptability should be. An absolute threshold can be reasonably applied to σ_b^2 and to M2p5 on the basis that below a certain absolute level these measures indicate that the image is unsuitable for segmentation into two distinct classes based on Otsu's histogram analysis method, or that the result of that partitioning does not produce a binary image that conforms to the straight-line model of the horizon.

Brightness, on the other hand, depend strongly on the lighting conditions and the camera settings and orientation that prevail at the time the image is captured. A level of brightness measured directly from the image will be affected by the camera auto-gain and auto-exposure settings, for example, if they are enabled. Disabling these adaptive aspects of the camera is not feasible, because it restricts the ability of the camera to cope with the conditions of lighting that will routinely be met, resulting in the image being too dark, or too over-exposed at times. This means that we need a measure of brightness that takes the camera settings into account. The brightness metric is measured by reading the camera's Automatic Exposure Control (AEC) and Automatic Gain Control (AGC) settings. It was found that a useable measure of brightness of the image can be formulated by multiplying the average brightness $3B^2/(R+G+B)$ as measured from the image by $1/(AGC*2+AEC+1)$.

If this measure of brightness is very low, it indicates that the amount of ambient light is insufficient for the image to be useful. Furthermore, by deciding appropriate ranges of brightness, a crude determination of whether the ground only is in view, or whether the image is bright enough to contain at least part of the sky. This is very useful in being able to reject image partitions that can produce false horizons from features on the ground. The brightness measure may also be suitable to determine if the camera is pointing only at the sky, but given the propensity for the high clear sky to drop in brightness it may be difficult to tell the difference between ground and sky in some cases.

4 Horizon Sensor Hardware

An ALTERA Cyclone I EP1C6T144C6 field programmable gate array (FPGA) was used to implement a 'soft' embedded NIOS II microcontroller from Altera and an interface to the digital camera. See figure 2 for a photo of the circuit-boards developed by the authors. (The side shown carries the FPGA and the other side carries a video I/O circuit). The ‘firmware’ design is a mixture of VHDL and C software. The size and weight of the FPGA version is compatible with small UAVs, as required. (Weight of 100*g* including camera, size of 100*x*50*x*30*mm*.) The horizon angle measurements are carried out in software written for the microcontroller (CPU) in the FPGA. The CPU passes the results of the measurements via a serial port to telemetry systems in flight or to software running on a PC during simulations. During closed-loop simulation tests software running on the PC can also implement a PID controller to send control adjustments to a flight simulator. The reason for using the FPGA to interface to the camera rather than the CMUCAM was primarily to make it possible to access the processed video so that it can be transmitted during real flight by a video telemetry link, which was something that the CMUCAM based system couldn’t easily be modified to do at a fast enough frame rate. This feature is deemed imperative by the authors as a means to monitor the success or otherwise of the algorithm. The output video can be displayed as a binary image showing the ground and sky classes, or the raw image can be left and the determined horizon angle indicated. Even real-time advice can be given in the form of directional arrows showing the pilot the direction to move the control sticks in to restore horizontal flight. (See figure 4 for example.) The output video makes it possible to tell when the classification method is not successful and to provide evidence for why such failures occurred and what effect parameter adjustments have on the success rate. Statistics and frame number can also be superimposed on the video, making synchronisation of data to video simple. The use of the FPGA also enables us to increase the frame transfer and processing rate and give access to raw image data rather than relying on the built-in functionality of the CMUCAM to do the image processing. The use of the FPGA meant that the camera interface and image masking can be done in hardware implemented outside the CPU (but still in the FPGA) and so doesn’t have any negative impact on the processing. Also the frame transfer to the CPU memory space is implemented using DMA techniques in real-time and so there is shorter delay before the frame processing takes place. Eventually the time taken by the CPU for frame processing was reduced to 200 ms giving a rate of five frames per second. This is still not completely satisfactory and the authors have plans to improve it to the point where the processing can run at 20 ms per frame, which is as fast as the camera can deliver them. This will be done by doing the image processing using a pipelined series of hardware processing stages described using VHDL and implemented on a larger FPGA, allowing image processing stages to run concurrently in hardware instead of sequentially in software. The current version of the FPGA used does not have capacity enough to implement both the CPU and the pipelined image processing hardware, which is why this faster implementation has not already been carried out. (90% of the Cyclone FPGA logic resources and 70% of its memory elements were used in implementing the CPU plus camera interface.) Use of a CPU in conjunction with the image processing pipeline is still considered the most flexible way to act on the results of the image processing, as well as for supervisory and configurational tasks, such as setting up the digital video camera, communicating results or control signals to other parts of the system and so on. Work described in [18] explains how the image processing pipeline can be implemented.

The system was also used to compare the video horizon sensor measurements to those of an infrared sensor similar to that described in [6], which has been used successfully by its developer, Professor Greg Egan, for autonomous flight of UAVs at Monash Uni. The results of these measurements are discussed in the section 5.

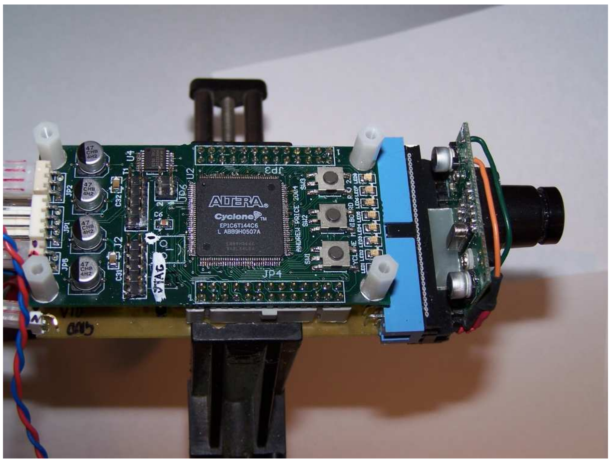


Figure 2: Horizon sensor based on digital video module and FPGA.

5 Flight Measurements

Simulation results have been carried out, and reported in [15], showing that this system is capable of relatively accurate measurements (3% error) of the roll and pitch angle. More recently, tests were conducted during real flight to compare the horizon sensor with a commercially available infrared sensor [4]. Note that both the horizon sensor and the infrared sensor were calculating the angles onboard during flight. The results are not from post-processing or processing of video at the ground-station. The results of one such test are shown in figure 3. It was late in a winter afternoon, with an overcast day, just before a storm, so conditions were not ideal. Note however the high correlation of 0.9 between the two sets of data. Note also that there is substantial disagreement in places, such as near sample number 629 where the infrared sensor result disagrees with the $-46°$ measurement of the horizon sensor. The data shown are only for when the horizon segmentation metrics indicate a high confidence in the result so the discrepancy is not due to poor view of the horizon.

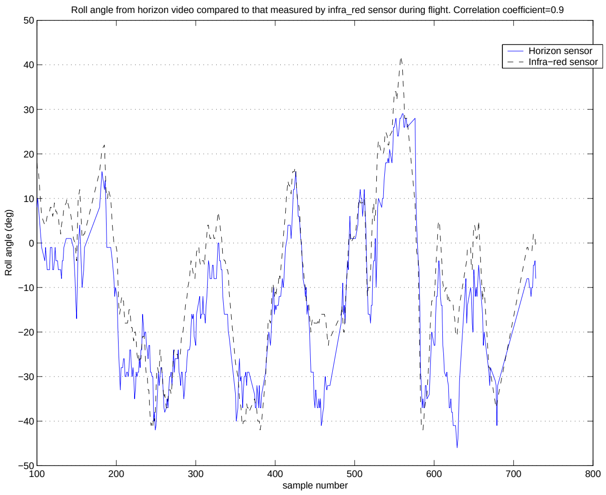


Figure 3: Horizon sensor v infrared sensor for roll angle.

Investigation of the video telemetry, one frame of which is shown in figure 4 showed that the horizon angle does in fact reach $-46°$ at sample number 629 which indicates that the horizon sensor is in fact substantially correct for that section. It appears that the infrared sensor is giving a low roll reading at that point in the flight. It is thought that some environmental feature such as a dark cloud might have caused the anomalous infrared reading. Anecdotal reports from other users of the infrared sensor indicate that this sort of effect has been noticed before.

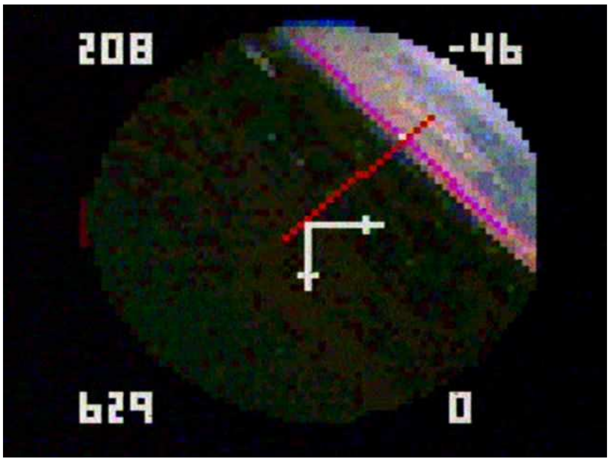


Figure 4: Snapshot of frame 629 showing correct horizon sensor reading.

Another observation came from the flight tests that is encouraging, and that is that the M2P5 confidence metric worked well to eliminate false readings due to objects on the ground that were 'sky-coloured'. In particular, there was a medium-sized freshwater dam in the view at a number of times, and it was reflecting the sky. During those frames, M2P5 was below the acceptable threshold of 200 because the segmentation resulted in a ground class with a large area of 'sky' embedded within it and these frames were consistently rejected.

Yet another observation, not so encouraging, was that in the launch or landing approaches during some trials when the horizon was out of view and only ground was visible, 'false positive' results were given when horizons were erroneously detected in ground features. The efforts, using M2P5 and σ_b^2 and brightness and contrast measures to defeat this problem were not completely sufficient. Although this did only happen in 2 out of the approximately 1000 frames of the trial that produced figure 5 for example, nonetheless, control actions based on these erroneous results would probably have resulted in crashes. This emphasises the notion that the output of the horizon sensor must be further processed to check it for conformity with previous measurements before action is taken.

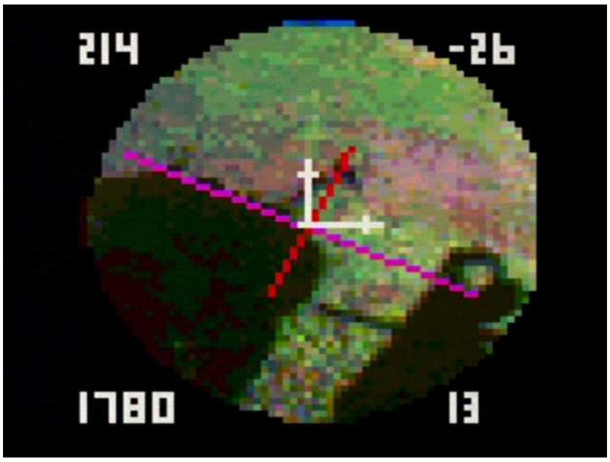


Figure 5: False positive on launch.

The discussion has focussed on roll measurements, partly because of space concerns, but also because during tests it became noticeable that the pitch angle varies considerably faster than the roll angle. It also seems that the 200ms sampling rate is too slow for the pitch variation as in many cases there are only as few as five measurements between peaks in the pitch. A rule of thumb mentioned in [19] is that the sampling frequency for a digital control system should be 30 times faster or more than the highest frequency present in the output of the controlled plant. It is clear that 200ms sampling rate is not adequate for the longitudinal (pitch) characteristics of this aircraft. Nonetheless, measurements of the pitch angle correlates very well in more slowly varying tests done on the ground and in simulation and it is expected that a 20ms frame rate will enable adequate performance in both the the pitch and roll control of future closed-loop experiments.

6 Conclusions and Future Work

It is the authors' contention that it will be impossible to reject all false horizons on the basis of the visual appearance of the image, because it will always be possible to get images that look like horizons even when they are not, no matter what segmentation method is used. It is clear, however, from images such as 5 that the relatively simple method of segmentation used in this work can be improved, and in fact more complex methods have been reported that appear to work very well. Nonetheless, there is a place for simple methods that work adequately, especially if there are also measurable values such as the M2P5 and σ_b^2 metrics that can be used to qualify the results. Future directions, as mentioned earlier, will be to increase the frame rate to the maximum possible and to use temporal filtering to reject those false measured angles due to the few false horizons that do get past the confidence metrics.

7 References

[1] S Ettinger, P. Ifju and M. C. Nechyba, “Vision-guided flight for micro air vehicles”, http://mil.ufl.edu/ nechyba/mav/index.html#vision1, visited on 27 October 2006.

[2] S. Ettinger, M. Nechyba, P. Ifju and M. Waszak, “Vision-guided flight stability and control for micro air vehicles”, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3, pp 2134–2140 (2002).

[3] L. Fety, M. Terre and X. Noreve, “Image processing for the detection of the horizon and device for the implemention thereof”, United States Patent 5214720, Thompson TRT Defense (1991).

[4] FMA Direct website, http://www.fmadirect.com, visited on 27 October 2006.

[5] B. Taylor, C. Bil, S. Watkins and G.K. Egan, “Horizon sensing attitude stabilisation: A VMC autopilot”, *Presented at the 18th International UAV Systems Conference*, Bristol, UK (2003).

[6] J.A. Gozdecki, “Aircraft attitude sensor and feedback control system”, United States Patent 6181989 (2001).

[7] T. Hudson, “Autopilot: UAV command and control”, http://sourceforge.net/projects/autopilot, visited on 27 October 2006.

[8] Carnegie Mellon University Robotics Institute, “Cmucam vision system”, http://www-2.cs.cmu.edu/ cmucam, visited on 27 October 2006.

[9] T.D. Cornall and G.K. Egan, “Measuring horizon angle from video on a small unmanned air vehicle”, *Presented at the 2nd International Conference on Autonomous Robots and Agents (ICARA 04)*, Palmerston North, NZ (2004).

[10] G. Barrows, PhD Thesis, “Mixed-mode vlsi optic flow sensors for micro air vehicles”, ftp://ftp.centeye.com/pub/Dissertation.pdf, visited on 27 October 2006.

[11] G. Barrows, J.S. Chahl and M.V. Srinivasan, “Biomimetic visual sensing and flight control”, *Presented at the 17th International UAV Systems Conference*, Bristol, UK (2002).

[12] T. Netter and N. Franceschini, “A robotic aircraft that follows terrain using a neuromorphic eye”, *Proceedings IEEE/RSJ Int Conference on Robots and Systems* 1, pp 129–134 (2002).

[13] G. Stange, S. Stowe, J.S. Chahl and A. Massaro, “Anisotropic imaging in the dragonfly median ocellus: a matched filter for horizon detection”, *Journal of Comparitive Physiology A*, 188, pp 455–467 (2002).

[14] T.D. Cornall and G.K. Egan, “Heaven and earth and how to tell the difference”, *Presented at the 11th Australian International Aerospace Congress*, Melbourne, Australia (2005).

[15] T.D. Cornall, G.K. Egan and A. Price, “Aircraft attitude estimation from horizon video”, *IEE Electronics Letters*, 42(13), pp 744–745 (2006).

[16] T.D. Cornall, “Method and apparatus for determining horizon angle and displacement”, Standard Complete Australian Patent Number 2004202924 (2004).

[17] N. Otsu, “A threshold selection method from gray level histograms”, *IEEE Trans. Systems, Man and Cybernetics*, 9, pp 62–66 (1979).

[18] A. Price, J. Pyke, D. Ashiri and T.D. Cornall, “Real time object detection for an unmanned aerial vehicle using an FPGA based vision system”, *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL, pp 2854–2859 (2006).

[19] B. Etkin, *Dynamics of Atmospheric Flight*, John Wiley and Sons (1972).

Experiences Using EMId and EKF for UAV Online Identification

Abhijit G. Kallapur, Shaaban S. Ali, Sreenatha G. Anavatti
School of Aerospace, Civil and Mechanical Engineering,
UNSW@ADFA, Canberra, Australia
a.kallapur@adfa.edu.au, s.salman@adfa.edu.au, agsrenat@adfa.edu.au

Abstract

A comparative study of findings for UAV identification using the methods of error mapping identification (EMId) and extended Kalman filter (EKF) are presented. The UAV attitude dynamics considered is highly coupled and nonlinear. It is important to maintain consistency in estimation in spite of real-time implementation restrictions imposed by hardware limitations. The efficiency of numerical as well as hardware-in-the-loop (HIL) simulations is addressed in the context of hardware limitations.

Keywords: Nonlinear, coupled, identification, EKF, EMId, UAV.

1 Introduction

Small unmanned aerial vehicles (UAVs) have unmatched quality which makes them the only solution for special tasks where human intervention is not necessary, dangerous or expensive. Controlling UAVs during such missions demand design and development of robust and adaptive controllers for which it becomes imperative to identify the system. Unlike the normal aircraft, the dynamics of UAV is not well documented. They fly at very low speeds and Reynolds numbers, have nonlinear coupling, and tend to exhibit time varying characteristics. Progress in terms of nonlinear system identification is not as substantial as compared to the case of linear systems [1]. The issues of identification relating nonlinear multi-input multi-output vehicles are a challenging area of research. Error mapping identification (EMId) has proved to be a robust technique in this field for a nominal nonlinear model [1]. Similarly state and parameter estimations, an integral part of system identification have been carried out on various flight data and simulations as evident from literature [2-4]. Though methods such as Least Squares [2], Gauss-Newton [4] and Kalman filters [5] have been used successfully for flight parameter estimation, their performance is known to deteriorate with increase in system nonlinearities. Constant gain filter methods have been used to estimate system coefficients showing satisfactory results for moderately nonlinear systems [6], but time varying filter methods become essential for systems with higher degrees of nonlinearity [4]. The use of Extended Kalman Filter (EKF) in such cases has proved to be a good technique for identification [4-6]. In this paper, a comparison of numerical and Hardware-In-the-Loop (HIL) simulation is presented for identification of UAV attitude dynamics using the two identification techniques, error mapping identification [1, 7] (EMId) and the extended Kalman filter (EKF) [4-6].

Section 2 describes the UAV attitude dynamics. Sections 3 and 4 introduce the concept of EMId and EKF respectively. Section 5 documents the results. Section 6 presents concluding remarks.

2 UAV Attitude Dynamics

The flight data used for identification was acquired from a 3 degrees-of-freedom (DOF) inertial measurement unit (IMU) designed and built at UNSW@ADFA. This data was highly noisy as can be seen from figures 1(a), 1(b), 2(a) and 2(b) and nonlinear. In order to cope with these characteristics, the UAV attitude dynamics considered in this paper is mapped by a set of three highly coupled nonlinear differential equations [7]:

$$\left.\begin{aligned}
\dot{p} &= \frac{1}{I_x I_z - I_{xz}^2}\left\{\begin{matrix} I_z[L+(I_y-I_z)qr]+ \\ I_{xz}[N+(I_x-I_y\ +I_z)pq- \\ I_{xz}qr] \end{matrix}\right\} \\
\dot{q} &= \frac{1}{I_y}[M+pr(I_z-I_x)+(r^2-p^2)I_{xz}] \\
\dot{r} &= \frac{1}{I_x I_z - I_{xz}^2}\left\{\begin{matrix} I_x[N+(I_x-I_y)pq]+ \\ I_{xz}[L+(I_y-I_x-I_z)qr+ \\ I_{xz}pq] \end{matrix}\right\}
\end{aligned}\right] \tag{1}$$

where p, q, r; L, M, N are the angular rates and aerodynamic moments of roll, pitch and yaw respectively; I_x, I_y, I_z, I_{xz} are moments of inertia; $\delta_e, \delta_r, \delta_a$, and δ_{th} are the elevator, rudder, aileron and throttle servo deflections respectively;. The aerodynamic moments are modelled as [7]:

$$\begin{aligned} L &= L(p, r, \delta_a, \delta_r, \delta_{th}) \\ &= L_p p + L_r r + L_{\delta_a}\delta_a + L_{\delta_r}\delta_r + L_{\delta_{th}}\delta_{th} \end{aligned} \tag{2}$$

$$\begin{aligned} M &= M(q,\delta_e,\delta_{th}) \\ &= M_q q + M_{\delta_e}\delta_e + M_{\delta_{th}}\delta_{th} \end{aligned} \tag{3}$$

$$\begin{aligned} N &= N(p,r,\delta_a,\delta_r,\delta_{th}) \\ &= N_p p + N_r r + N_{\delta a}\delta_a + N_{\delta r}\delta_r + N_{\delta th}\delta_{th} \end{aligned} \tag{4}$$

From (1) - (4) we can define the system dynamics in the form:

$$\dot{x}(t) = A(t)\, f(x,u) \tag{5}$$

where *x(t)* represents the state vector, *u(t)* the input vector and *A(t)* the coefficient matrix.

$$x(t) = [\,p\,q\,r\,]^T \tag{6}$$

$$u(t) = \begin{bmatrix} \delta_e & \delta_r & \delta_a & \delta_{th} \end{bmatrix}^T \tag{7}$$

$$f(x,u) = [\,pq \;\; qr \;\; pr \;\; p^2 \;\; r^2 \;\; p \;\; q \;\; r \;\; \delta_e \;\; \delta_r \;\; \delta_a \;\; \delta_{th}\,]^T \tag{8}$$

The matrix coefficients of (9) are unknown and need to be identified either directly or indirectly.

$$A(t) = \begin{bmatrix} A_{11} & 0 & A_{31} \\ A_{12} & 0 & A_{32} \\ 0 & A_{23} & 0 \\ 0 & A_{24} & 0 \\ 0 & A_{25} & 0 \\ A_{16} & 0 & A_{36} \\ 0 & A_{27} & 0 \\ A_{18} & 0 & A_{38} \\ 0 & A_{29} & 0 \\ A_{110} & 0 & A_{310} \\ A_{111} & 0 & A_{311} \\ A_{112} & A_{212} & A_{312} \end{bmatrix}^T \tag{9}$$

where

$$\left.\begin{aligned} &A_{11} = \frac{I_{xz}(I_x - I_y + I_z)}{I_x I_z - I_{xz}^2} \qquad A_{12} = \frac{I_z(I_y - I_z) - I_{xz}^2}{I_x I_z - I_{xz}^2}, \\ &A_{16} = \frac{I_z L_p + I_{xz} N_p}{I_x I_z - I_{xz}^2} \qquad A_{18} = \frac{I_z L_r + I_{xz} N_r}{I_x I_z - I_{xz}^2}, \\ &A_{110} = \frac{I_z L_{\delta_r} + I_{xz} N_{\delta_r}}{I_x I_z - I_{xz}^2} \quad A_{111} = \frac{I_z L_{\delta_a} + I_{xz} N_{\delta_a}}{I_x I_z - I_{xz}^2}, \\ &A_{112} = \frac{I_z L_{\delta_{th}} + I_{xz} N_{\delta_{th}}}{I_x I_z - I_{xz}^2} \end{aligned}\right\} \tag{10}$$

$$\left.\begin{aligned} &A_{23} = \frac{I_z - I_x}{I_y} \; A_{24} = \frac{-I_{xz}}{I_y} \quad A_{25} = \frac{I_{xz}}{I_y} \\ &A_{27} = \frac{M_q}{I_y} \quad A_{29} = \frac{M_{\delta_e}}{I_y} \quad A_{212} = \frac{M_{\delta_{th}}}{I_y} \end{aligned}\right\} \tag{11}$$

$$\left.\begin{aligned} &A_{31} = \frac{I_x(I_x - I_y) + I_{xz}^2}{I_x I_z - I_{xz}^2} \qquad A_{32} = \frac{I_{xz}(I_y - I_x - I_z)}{I_x I_z - I_{xz}^2}, \\ &A_{36} = \frac{I_x N_p + I_{xz} L_p}{I_x I_z - I_{xz}^2} \qquad A_{38} = \frac{I_x N_r + I_{xz} L_r}{I_x I_z - I_{xz}^2}, \\ &A_{310} = \frac{I_x N_{\delta_r} + I_{xz} L_{\delta_r}}{I_x I_z - I_{xz}^2} \quad A_{311} = \frac{I_x N_{\delta_a} + I_{xz} L_{\delta_a}}{I_x I_z - I_{xz}^2}, \\ &A_{312} = \frac{I_x N_{\delta_{th}} + I_{xz} L_{\delta_{th}}}{I_x I_z - I_{xz}^2} \end{aligned}\right\} \tag{12}$$

3 Error Mapping Identification (EMId)

The nonlinear EMId method is a robust, nonlinear, direct estimation technique based on state-space identification [1, 7-10]. EMId considers the system in the form,

$$\dot{x}(t) = F(x,u), \quad t \ge 0, \quad x(t_0) = x_0 \tag{13}$$

System (13) can be written in the matrix state space form as [1, 7]:

$$\dot{x}(t) = A(t)\, f(x,u), \quad t \ge 0, \quad x(t_0) = x_0 \tag{14}$$

where $A(t) \in R^{cxn}$ is the real matrix, $f(x,u)$ denotes a given real analytic function.

The identified state space model is subscripted as '*m*' and defined as:

$$\dot{x}_m(t) = A_m(t)\, f(x_m,u) \tag{15}$$

for: $t \ge 0$ and $x_m(t_0) = x_{m0}$

The matrix coefficients, $A_m(t)$, are to be identified from flight data.

The normalised parameter error matrix $\Delta A(t) \in R^{cxn}$ defined as the difference between the actual and identified matrix coefficients is given by:

$$\Delta A(t) = A(t) - A_m(t)\,. \tag{16}$$

The error vector is then defined as:

$$e(t) = \Delta\dot{x}(t) - A_m(t)\Delta f(x, x_m, u) = \Delta A(t)\, f(x,u) \tag{17}$$

For an identification algorithm to converge it must satisfy,

$$\lim_{t \to +\infty} \left\| \Delta A(t) \right\| = 0 \Rightarrow \lim_{t \to +\infty} \left\| e(t) \right\| = 0 \tag{18}$$

Using the differential equation for the normalised parameter error matrix as in [9], one has:

$$\begin{aligned} &\Delta\dot{A}(t) = -e(t)\, f(x,u)^T K \\ &where\, K \in R^{nxn}, \Delta A(t_0) = \Delta A_0 \end{aligned} \tag{19}$$

where '*K*' is the weighting matrix the selection of which affects the convergence of the identified parameters to their real values. It is chosen by the designer to guarantee the convergence and to attain the desired convergence rate.

From (16) and (17), we get:

$$\dot{A}_m(t) = \dot{A}(t) + e(t)\, f(x,u)^T K \qquad \text{with}\, A_m(t_0) = A_{m_0} \tag{20}$$

At a given flight condition, UAV dynamics can be assumed as time invariant, i.e., system parameters are constant. Hence system (14) becomes time invariant, so $\dot{A}(t)=0$ .Then, we have the following nonlinear equation:

$$\dot{A}_m(t) = [\Delta\dot{x}(t) - A_m \Delta f(x, x_m, u)]\, f(x,u)^T K \qquad \text{with}\, A_m(t_0) = A_{m_0} \tag{21}$$

The unknown parameters are found by solving the nonlinear differential equation as in (21) [7].

The weighting matrix '*K*' is chosen so that there is a quick convergence of the identified parameters to the actual values. In order to ensure that each coefficient of matrix A_m (as example A_{m11}, A_{m12} and ...etc) is affected only by the corresponding multiplying terms (as example *pq, qr* etc), the '*K*' matrix in this paper is chosen to be diagonal, given by:

$$K = diag\,[k_{ij}] \in R^{12x12}, \text{ with } k_{ij} = 0.001 \tag{22}$$

4 Extended Kalman Filter (EKF)

The central idea of nonlinear filtering using EKF is to reduce the mean-square error of the estimates using a blending factor or variable gain called the Kalman gain, K. The state estimation problem can be further extended to state and parameter estimation by including the unknown parameters with the states to construct an augmented state vector [11]. The combined state and parameter estimation can then be modelled as a single-pass procedure using the EKF [4].

The EKF used in this paper is based on the nonlinear state-space process and observation models described by:

$$x_{k+1} = f(x_k, u_k, w_k) \qquad z_k = h(x_k, v_k) \tag{23}$$

where u_k, w_k and v_k denote the control input, process noise and measurement noise respectively at time step *k*. The nonlinear system described by (23) is approximated to a first-order linear form about the estimated values by computing Jacobians as in (24). These estimates need to be as close as possible to the actual value for the linearisation to be accurate and the estimate to converge [12].

$$F_k = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k^+} \qquad H_k = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_k^-} \tag{24}$$

The superscripts (^), (-) and (+) in (24) denote an estimate, a priori value and posteriori value respectively. This representation will be used for all further EKF-related equations in this paper.

Considering I to be an identity matrix and Δt the sampling time duration the linearised, continuous form mentioned above can be discretised by truncating the exponential function to the second order as:

$$F = \exp(F_k\, \Delta t) \approx I + (F_k\, \Delta t) + \frac{(F_k\, \Delta t)^2}{2!} \tag{25}$$

The fact that the measurements are the states themselves in our case, renders the measurement sensitivity matrix H to be a constant and hence the evaluation of H_k is no more necessary. This improves computation efficiency of the filter on the whole. With [p, q, r]^T as measurements and 0_{axb} an a-by-b zero matrix,

$$H = [I_{3x3} \,|\, 0_{3x13}] \tag{26}$$

The EKF approach to identification in this paper deals with estimating 3 states and 13 parameters that constitute equations (1) - (4). The augmented state vector is given by:

$$x = [\,p\ q\ r\ \mathrm{L_p}\ \mathrm{L_r}\ \mathrm{L}_{\delta\mathrm{r}}\ \mathrm{L}_{\delta\mathrm{a}}\ \mathrm{N_p}\ \mathrm{N_r}\ \mathrm{N}_{\delta\mathrm{r}}\ \mathrm{N}_{\delta\mathrm{a}}\ \mathrm{M_q}\ \mathrm{M}_{\delta e}\ \mathrm{M}_{\delta th}\ \mathrm{L}_{\delta th}\ \mathrm{N}_{\delta th}\,]^{\mathrm{T}} \tag{27}$$

These parameters provide definitive, physical significance in terms of aerodynamic derivatives and can be used directly to study aerodynamic stability of the flight data for a certain UAV.

Initial estimates of various states, parameters and covariance are necessary for the estimation procedure using EKF. Here the initial estimates x_0 of state variables as well as the augmented parameters are set to a nominal value based on engineering judgement. The diagonal elements of initial covariance matrix P_0 are set to reflect the confidence in corresponding initial estimates [4]. For the three states the covariance values are set to the corresponding measurement error variances [11] while those for the augmented parameters are set to a relatively high value [4].

The process of estimation using EKF consists of two iterative steps namely; *propagation* and *correction*. The linearised, discrete model arising from (23) - (26)

is used to propagate states and covariance to the next time-step based on the present knowledge of the model [3] as:

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, t_{k-1}) \tag{28}$$

$$P_k^- = F\, P_k^+\, F^T + Q_k \tag{29}$$

Continuous propagation leads to accumulation of model-induced errors and hence the need for a *correction* step. Corrections are generally made as and when new measurements are available as [3]:

$$K_k = P_k^-\ H^T\ (H\ \ P_k^-\ H^T + R_k)^{-1} \tag{30}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\,(z_k - H\,\hat{x}_k^-) \tag{31}$$

$$P_k^+ = (I - K_k H\ \ )P_k^- \tag{32}$$

Though EKF simultaneously provides state as well as parameter estimation for nonlinear, time-variant and unstable systems [3] it comes with its own cost. The filter performance and divergence are highly sensitive to initial estimates. Moreover output precision can be severely hampered in case of highly nonlinear systems due to first order linearisation described in (24). As can be seen from various equations mentioned in this section the computational requirements of EKF are pretty high which can be controlled by reducing the frequency of correction but as a trade-off for estimation efficiency.

5 Results

Numerical as well as HIL simulations were carried out for UAV identification using techniques of EMId and EKF. Numerical simulations are a good platform to check for the operation of the algorithm which in our case was executed on MATLAB © Simulink. The HIL simulations on the other hand provide near-realistic implementation results in the sense that only the actual UAV is replaced by a computer simulation while the autopilot is used as it would have been onboard the UAV. The autopilot in our case is built around the PC-104 architecture. This PC-104 unit accepts data from sensors, processes it and sends the processed output to MATLAB© which then plots the data as needed [7]. HIL simulations were carried out using MATLAB© XPC Target. Owing to restrictions imposed by the processing power of the PC-104 unit the sampling rate as well as correction frequency for the EKF was reduced in case of HIL.

EMId estimated states (p, q, r) and matrix coefficients of (9) at every time step as was described in section 3, whereas EKF estimated the three states directly and thirteen other parameters,

$$\begin{gathered}\mathrm{L_p; L_r; L_{\delta r}; L_{\delta a}; N_p; N_r; N_{\delta r};}\\ \mathrm{N_{\delta a}; M_q; M_{\delta e}; M_{\delta th}; L_{\delta th}; N_{\delta th}}\end{gathered} \tag{33}$$

which were then substituted into (10), (11) and (12) to obtain various matrix coefficients. The root mean square error (RMSE) values are used to compare these two techniques.

Figures 1a and 1b provide results for p, q, r estimation using EMId and EKF respectively during numerical simulations. Figures 2a and 2b provide results using HIL simulations. Table 1 provides a comparison of RMSE values obtained from these state estimations for both numerical as well as HIL simulations.

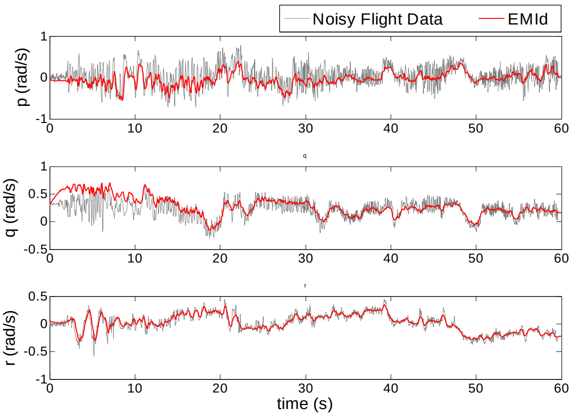


Figure 1a: EMId state estimation results - numerical simulation.

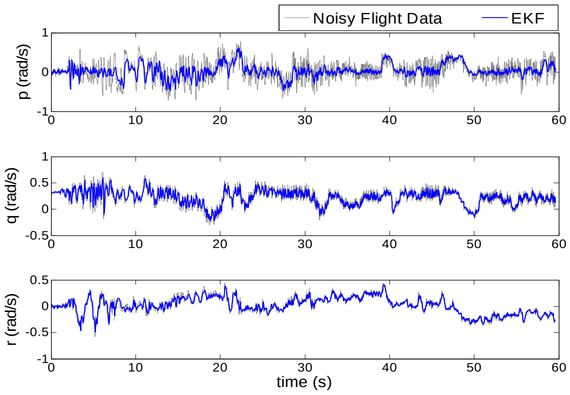


Figure 1b: EKF state estimation results - numerical simulation.

It is clear from table 1 that the EKF estimates were better than EMId estimates for numerical simulations whereas EMId performed better than EKF in case of HIL simulations (apart from q estimates). This variation in performance can be attributed to the fact that EKF was run at a higher sampling rate and the states and parameters were updated every single time step in case of the numerical simulation, whereas in case of the HIL simulation they were updated once every ten time steps and the filter (EKF) was operated at a lower sampling rate. This was seen as a processor related restriction imposed due to high computational requirement of the EKF. EMId on the other hand had no change in its computational method but was operated at corresponding sampling rates as the EKF was for both numerical as well as HIL simulations.

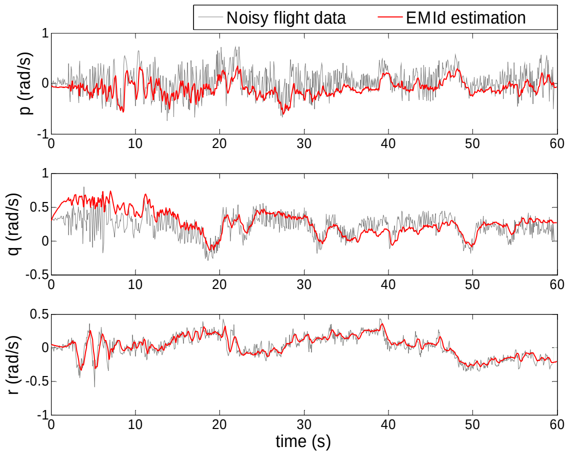


Figure 2a: EMId state estimation results - HIL.

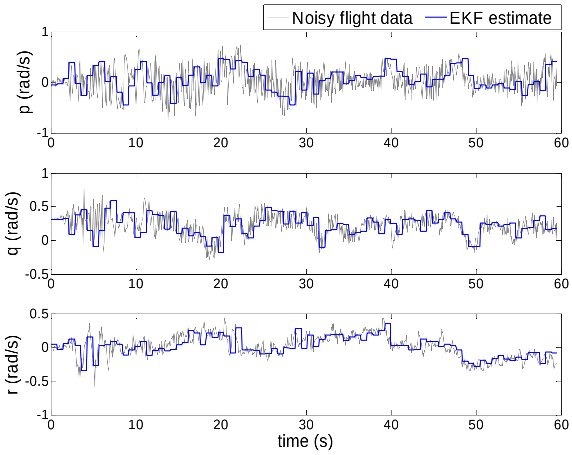


Figure 2a: EKF state estimation results - HIL.

From previous figures it was clear that EKF estimation averaged noise effects more efficiently than EMId due to its inherent filter properties and hence could be used efficiently with noisy data as presented here.

Table 1: RMSE comparison for states estimated by EMId and EKF for numerical as well as HIL simulations

		p_{RMSE}	q_{RMSE}	r_{RMSE}
Numerical Simulation	**EMId**	0.2120	0.1445	0.0741
	EKF	0.1580	0.0477	0.0311
HIL	**EMId**	0.2837	0.1758	0.0864
	EKF	0.2925	0.1565	0.1129

It was clear from equations (9) - (12) that 7 matrix coefficients out of the total 20 would remain constant since they were functions of time-invariant inertial moments and hence not discussed here. The remaining 13 coefficients were functions of inertial moments and a combination of time-varying parameters as listed in (33). Figures 3 and 4 compare eight of these matrix coefficients as estimated by EKF and EMId for numerical and HIL simulations respectively. These coefficients were selected in order to cover the entire range of parameters depicted in (33). The remaining matrix coefficients presented similar trends.

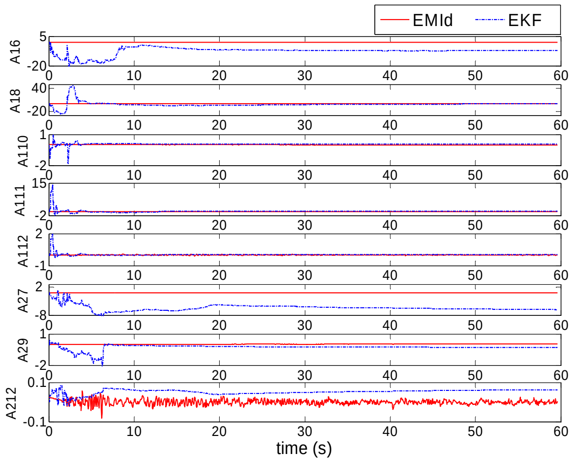


Figure 3: Comparison of eight matrix coefficient values as estimated by EMId and EKF during numerical simulations.

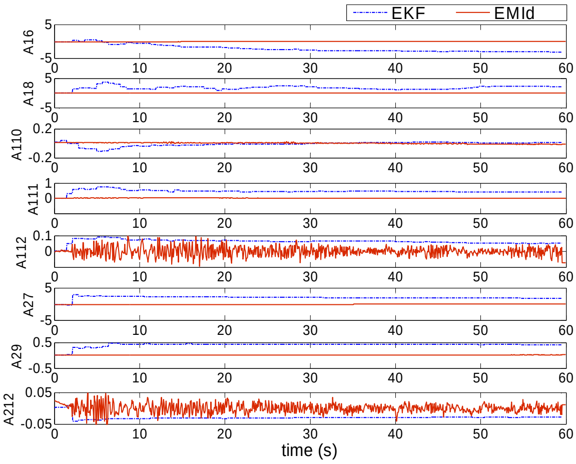


Figure 4: Comparison of eight matrix coefficient values as estimated by EMId and EKF during HIL simulations.

6 Concluding Remarks

The results indicate that both EMId and EKF estimation methods do a good job in identifying coupled dynamics of a small UAV. EMId is a more recent technique and can be used for identification with a nominal processor as the computational requirements are not great. On the other hand the more conventional EKF method presents large computational requirement and hence better processors are needed in order to run the method to gain maximum advantage. However if the loss of precision is not of significance, EKF can be used at reduced sampling rates with slight modification in implementation methodology as presented in this paper. Moreover for highly noisy input data EKF can be a better option due to its inherent filtering properties.

As mentioned in section 5, EMId computes the matrix coefficients directly whereas EKF computes the aerodynamic coefficients and then substitutes them into equations (10) - (12) to obtain corresponding matrix coefficients. Tracking these coefficients can give a better physical insight into the flight dynamics of the UAV, in a conventional sense.

Both these methods have their own advantages and depending upon the system requirements and available resources one can choose either method for identification.

Results can be further enhanced by making use of better processors so that EKF can be operated at higher sampling rates with more regular corrections whereas EMId can perform better if the input data is pre-treated for noise.

7 Acknowledgement

We would like to acknowledge the inputs provided by Prof Ian Petersen, ITEE, UNSW@ADFA, towards this published work.

8 Acknowledgement

[1] S.E. Lyshevski, "State-space identification of nonlinear flight dynamics", *Proceedings IEEE International Conference on Control Applications*, Hartford, CT, pp 496-498 (1997).

[2] K.W. Iliff, "Parameter estimation for flight vehicles", *Journal of Guidance, Control and Dynamics*, 12, pp 609-622 (1989).

[3] M.E. Campbell and S. Brunke, "Nonlinear estimation of aircraft models for on-line control customization", *Proceedings IEEE Aerospace Conference*, Big Sky, MT, 2, pp 621-628 (2001).

[4] R.V. Jategaonkar and E. Plaetschke, "Algorithms for aircraft parameter estimation accounting for process and measurement noise", *Journal of Aircraft*, 26(4), pp 360-372 (1989).

[5] R.W. Johnson, S. Jayaram, L. Sun and J. Zalewski, "Distributed processing Kalman filter for automated vehicle parameter estimation - a case study", *Proceedings IASTED Int'l Conf. on Applied Simulation and Modeling*, Banff, Canada, pp 270-276 (2000).

[6] R.V. Jategaonkar and E. Plaetschke, "Identification of moderately nonlinear flight mechanics systems with additive process and measurement noise", *Journal of Guidance, Control and Dynamics*, 13(2), pp 277-285 (1990).

[7] S.A. Salman, V.R. Puttige and A.G. Sreenatha, "Real-time validation and comparison of fuzzy identification and state-space identification for a UAV platform", *Proceedings Joint Conferences IEEE CCA, IEEE CACSD and IEEE ISIC*, Munich (in press) (2006).

[8] C. Lyshevski and Y. Chen, "Nonlinear identification of aircraft", *Proceedings IEEE International Conference on Control Applications*, Dearborn, MI, pp 327-331 (1996).

[9] S.E. Lyshevski, "Identification of nonlinear flight dynamics: theory and practice", *IEEE Transactions on Aerospace and Electronic Systems*, 36(2), pp 383-392 (2000).

[10] V. Pappano, S.E. Lyshevski and B. Friedland, "Nonlinear identification of induction motor parameters", *Proceedings American Control Conference*, San Diego, pp 3569-3573 (1999).

[11] M. Curvo, "Estimation of aircraft aerodynamic derivatives using extended Kalman filter", *Journal of Brazilian Society Mech. Sci.*, 22(2), pp 133-148 (2000).

[12] J. Farrell and M. Barth, *The Global Positioning System and Inertial Navigation*, McGraw-Hill, San Fransisco (1998).

A Novel CMOS Sampled-Data Centre-of-Mass Tracker Circuit for Robotic Visual Feedback Object Tracking

S.M. Rezaul Hasan and Johan Potgieter
Centre for Research in Analogue and VLSI Microsystem Design (CRAVE)
Institute of Engineering and Technology, Massey University, Auckland, New Zealand
hasanmic@massey.ac.nz

Abstract

Object position tracking is a well-known problem in robotic vision system design. This paper proposes a new centre-of-mass (COM) object position tracker integrated circuit design using standard low-cost digital CMOS process technology without the need for a floating gate poly2 layer. The proposed COM tracker is designed using an IBM 130nm CMOS process with a 1V supply voltage and can be used for a single-chip robotic visual feedback object tracking application. It uses an analogue sampled-data technique and is operated by a two-phase clocking system. The clock frequency can be varied to suit the real-time throughput requirements of the specific robotic object tracking tasks. A 6-input COM detector (easily extendable to 12-input or higher) was simulated and the results for various COM positions were found to be correctly tracked by the circuit. In addition, Monte Carlo simulations were carried out to prove the robustness of the technique.

Keywords: robotic vision, analogue CMOS VLSI, centre-of-mass, robotic object tracking

1 Introduction

Centre-of-mass (COM), the first moment of an object's intensity distribution, represents the position of an object. Consequently it has wide applications in robotic vision for visual feedback object tracking [1]. The COM of an object has to be determined in real time with minimal delay for robotic vision image processing. Compared to digital image processing techniques, analogue processing of image signals provides faster COM response time since no A/D conversion of image signals is required. The authors of [2] proposed a COM tracker circuit based on implementation by Neuron MOS technology [4]. Although neuron MOS technology simplifies implementation of many robotic and neural network circuits [3], it requires extra CMOS processing steps (e.g. POLY2) which are not usually available in many low-cost CMOS process technologies. In this paper a novel COM detector circuit is proposed which replaces the need for floating gate neuron MOSFETs with the use of a capacitive sampled-data technique following [5]. The proposed circuit uses a 1V supply voltage and is quite robust to process related threshold voltage variations. The circuit can be extended to any number of analogue image signal inputs and produces a binary (thresholding) COM decision which can be stored as a digital signal in a micro-controller status register.

2 Proposed COM Circuit Architecture

Figure 1 shows the basic algorithm to be implemented in each axis for a 2-D separable image intensity distribution of an object (e.g. a creature object) for the determination of its COM. Each axis thus requires a separate block of the proposed COM detector.

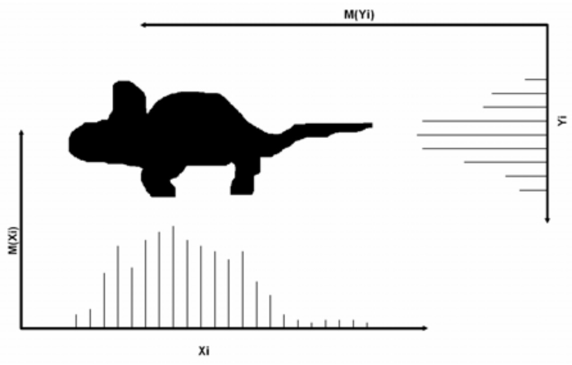


Figure 1: Illustration of the centre-of-mass (COM) detection algorithm. 2D separable image intensity distribution projections in X and Y directions are separately obtained and processed.

Figure 2 shows the circuit architecture of the proposed sampled-data centre-of-mass tracker circuit. It essentially consists of (n+1) capacitive moment-of-intensity circuits and n sampled-data inverter comparators for an n-position COM circuit. The location (position) of the moment-of-intensity for a set of analogue sampled intensities [v_1, v_2, v_3,…..,v_n] using capacitive position-scalars [C, 2C, 3C,……,nC] is given by

$$C_x = \frac{\sum_{r=1}^{n} rCv_r}{\sum_{r=1}^{n} v_r} \tag{1}$$

where C_x is the capacitive COM position-scalar.

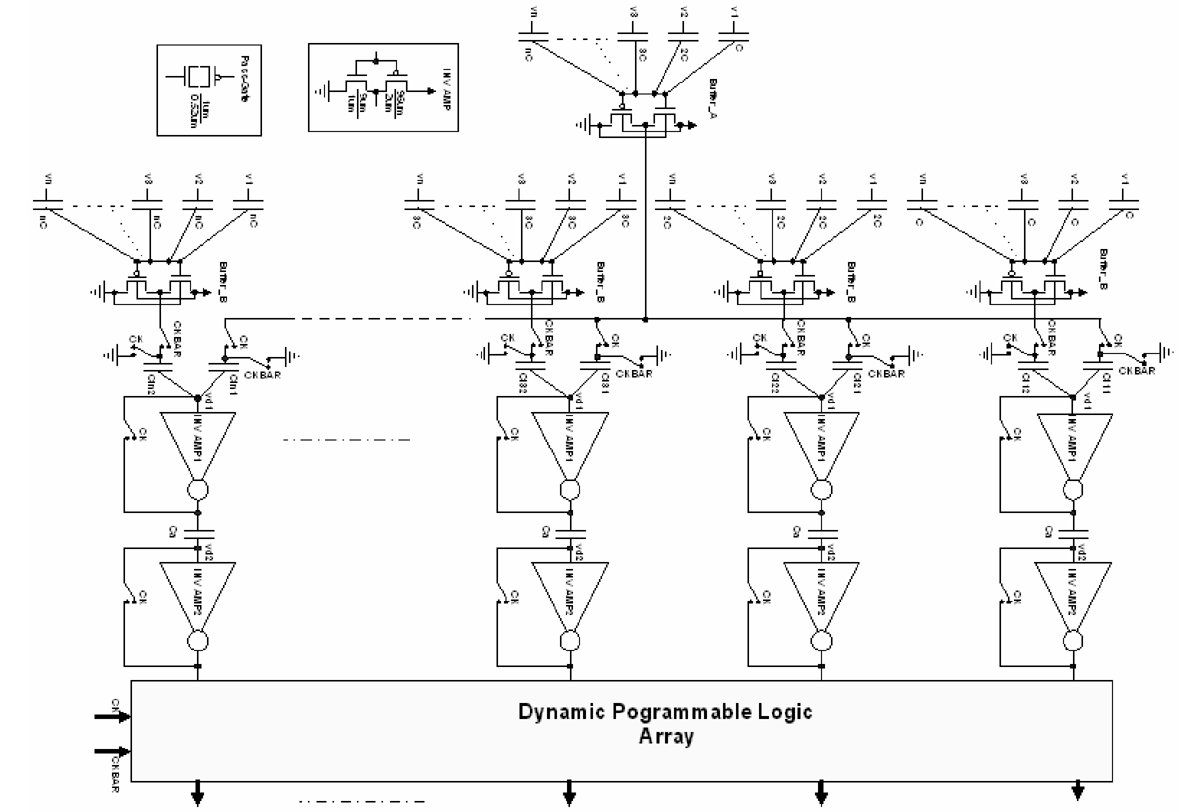


Figure 2: Architecture of the proposed sampled-data real-time centre-of-mass detector circuit for each axis of the 2-D separable image.

Next, equation (1) can be rearranged as

$$C_x \sum_{r=1}^{n} v_r = \sum_{r=1}^{n} rCv_r \ . \qquad (2)$$

Now, introducing a proportionality constant, k, equation (2) can be written as

$$\left[\frac{C_x \sum_{r=1}^{n} v_r}{nC_x}\right] * k * nC_x = \left[\frac{\sum_{r=1}^{n} rCv_r}{\sum_{r=1}^{n} rC}\right] * k * \sum_{r=1}^{n} rC \cdot \qquad (3)$$

Next, dividing both sides by

$k * nC_x + k * \sum_{r=1}^{n} rC$ and changing sides, we have

$$\frac{\left[\frac{C_x \sum_{r=1}^{n} v_r}{nC_x}\right] * k * nC_x - \left[\frac{\sum_{r=1}^{n} rCv_r}{\sum_{r=1}^{n} rC}\right] * k * \sum_{r=1}^{n} rC}{k * nC_x + k * \sum_{r=1}^{n} rC} = 0 \cdot \qquad (4)$$

In the proposed COM circuit, the closest value $C_x = rC$ is sought so that equation (4) holds. So making this replacement in equation (4), we have

$$\frac{\left[\frac{rC \sum_{r=1}^{n} v_r}{nrC}\right] * k * nrC - \left[\frac{\sum_{r=1}^{n} rCv_r}{C * \frac{n(n+1)}{2}}\right] * k * C * \frac{n(n+1)}{2}}{k * nrC + k * C * \frac{n(n+1)}{2}} = M \qquad (5)$$

where the specific value r is sought for which the difference M changes sign. The proposed COM tracker is an implementation of equation (5) with the value r (for which M changes sign) being the COM position detected by the circuit. In the implementation architecture of figure 2, n sampled data comparison paths are shown (for an n position COM tracker) with the top path (position) for r = 1 and the bottom path (position) for r = n. For all the paths,

Ci11=Ci21=Ci31=Cir1….=Cin1= $k * C * \frac{n(n+1)}{2}$ (6)

and for any path r,

Cir2 = knrC (7)

Ca is an appropriately chosen value for sampled-data input to the second inverter-amplifier. The rth path (position) source-follower Buffer_B produces the output voltage $\frac{rC \sum_{r=1}^{n} v_r}{nrC}$ = Vr (say). Also, the source-follower Buffer_A produces the output voltage $\frac{\sum_{r=1}^{n} rCv_r}{\sum_{r=1}^{n} rC}$ = Vc (say). The two inputs Vc and Vr along the rth path are sampled during setup phase, PH1 (CK) and the comparison phase PH2 (CKBAR) intervals respectively of a 50% duty cycle global clock signal. The perturbation signal at the input of the inverter-amplifier 1 (V_{d1}) in the rth path during PH2 interval is then given by

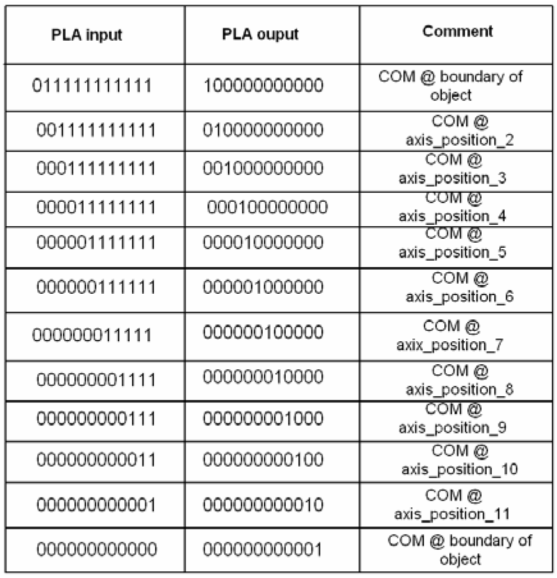
$$V_{d1} = \frac{(Vc * Cir1 - Vr * Cir2)}{(Cir1 + Cir2 + C_{ip})} \qquad (8)$$

where C_{ip} is the input parasitic capacitance of the inverter-amplifier. Due to electrostatic coupling of only the weighted difference signal, (Vc*Cir1-Vr*Cir2), both the inputs Vc and Vr can individually vary over a wide dynamic range. Both the inverter-amplifiers are self-biased at the common mode voltage (V_{cm}) and are disabled during the setup phase (i.e. during the PH1 interval). During the comparison phase (i.e. during the PH2 interval) the perturbation input voltage V_{d1} (given by equation (8)) at the input of the inverter-amplifier 1 is amplified by its gain A1. As a result the perturbation input V_{d2} at the input of the inverter-amplifier 2 in the rth path is given by

$$v_{d2} = [\frac{(Vc * Cir1 - Vr * Cir2)}{(Cir1 + Cir2 + C_{ip})} * A1] * \frac{Ca}{Ca + C_{ip}} \qquad (9)$$

Finally, this perturbation V_{d2} is amplified by the gain A2 of the inverter-amplifier 2. The rth path (position) starting at which the output of the inverter-amplifier 2 switches to opposite polarity compared to the (r-1)th path (position) is the tracked COM position for the input image intensity distribution. A PLA or simplified logic can be used to flag the COM position using the truth table in table 1 and the logic implementation shown in figure 3. The problem of charge injection and charge absorption along the signal path is minimised by using dummy pass gates. The optimum size of the capacitors was determined based on consideration of response time, parasitic input capacitance of the inverter amplifiers and charge leakage. The comparator output is available during the comparison phase (PH2) for the input read cycle of any back-end logic/instrumentation. Figure 4 shows a small signal (perturbation signal) equivalent circuit of the COM detector's analogue sampled data signal path during the comparison phase (PH2).

Table 1: PLA truth table of the COM tracker output logic.



PLA input	PLA ouput	Comment
011111111111	100000000000	COM @ boundary of object
001111111111	010000000000	COM @ axis_position_2
000111111111	001000000000	COM @ axis_position_3
000011111111	000100000000	COM @ axis_position_4
000001111111	000010000000	COM @ axis_position_5
000000111111	000001000000	COM @ axis_position_6
000000011111	000000100000	COM @ axix_position_7
000000001111	000000010000	COM @ axis_position_8
000000000111	000000001000	COM @ axis_position_9
000000000011	000000000100	COM @ axis_position_10
000000000001	000000000010	COM @ axis_position_11
000000000000	000000000001	COM @ boundary of object

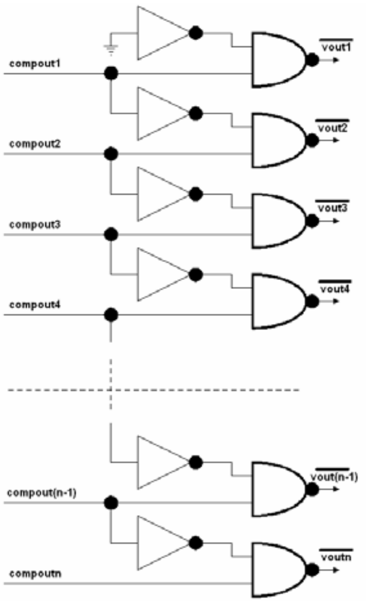


Figure 3: Simple CMOS circuit for the implementation of the PLA truth table of the COM tracker output logic.

3 Simulation and Analysis Results

In order to verify the operation of the proposed novel COM tracker circuit extensive simulations were carried out using the 8M1P 0.13μm IBM CMOS process technology parameters using Tanner T-SPICE v.12. The supply voltage was 1.0V and an input unit capacitance position-scalar of 0.1pF was used. A six position COM tracker was simulated so that n=6. Ca was set at 1pF. k was set to 1, and, the maximum value of capacitance used was 3.6pF. The circuit design of the pass-gates and the inverter-amplifiers is shown in figure 2. The tracked COM output is indicated with an active-low signal during PH2 in the COM output logic. Figure 5 shows the output of the 6-input COM detector circuit with discrete positions of 0.1 through 0.6, for an exact COM capacitive position-scalar value of 0.46.

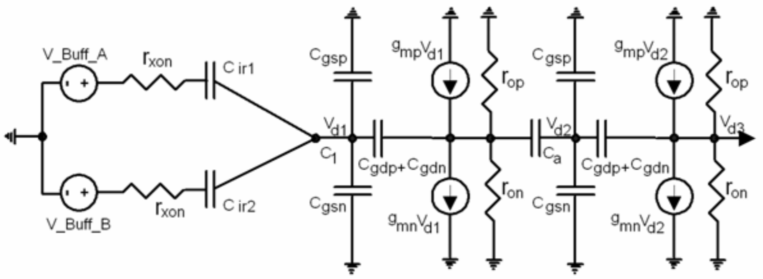
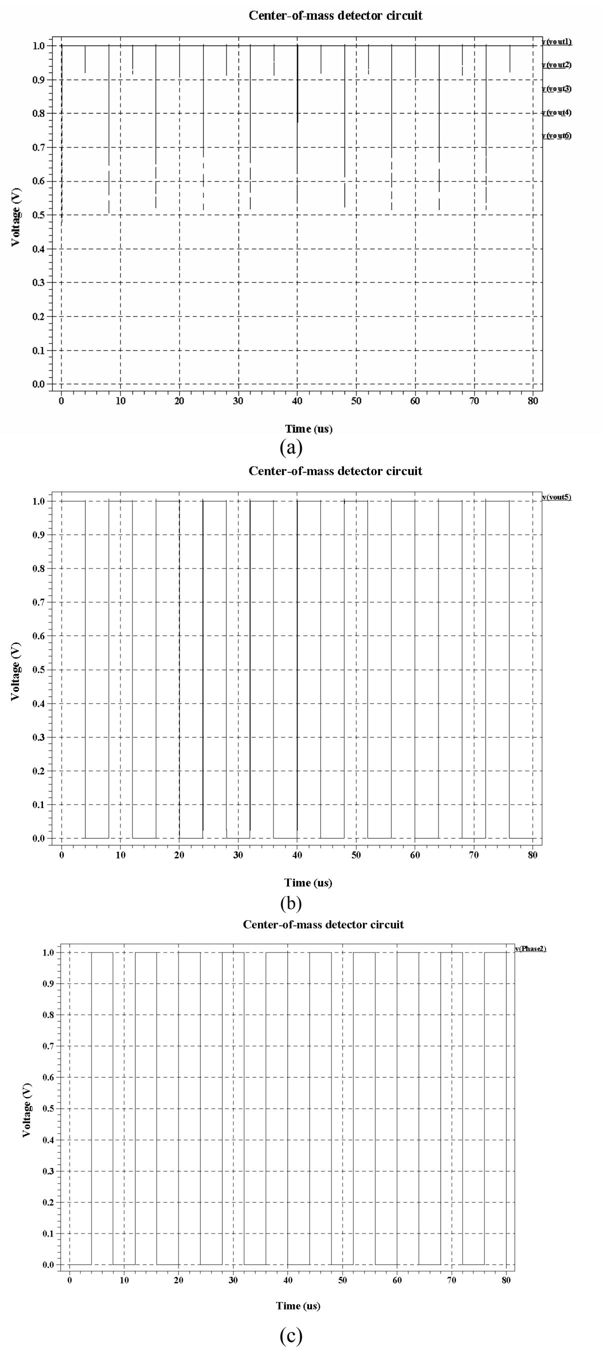


Figure 4: Equivalent circuit of the COM detector's analogue sampled-data signal path for response time estimation.

Figure 5(a) shows the outputs for positions 1, 2, 3, 4 and 6 (all high) while figure 5(b) shows the output for position 5 (low during PH2) and figure 5 (c) shows the clock PH2. So the nearest COM position 5 (corresponding to the capacitive position-scalar value of 0.5) is tracked by the circuit corresponding to the exact value of 0.46. Similarly, COM position is also correctly tracked for an exact COM of 0.28, as indicated in figure 6, resulting in a tracked COM position of 3. Next, 100 iterations of Monte Carlo simulations are carried out for process-related threshold voltage variation of 25% using a Gaussian distribution function. An exact COM position of 0.38 was applied at the input. The COM was correctly tracked to position 4, as shown in figure 7. This proves that the proposed architecture is quite robust under process variations in any low-cost digital CMOS technology. The static power dissipated by the COM tracker from a 1V supply voltage was under 100μW, making it extremely suitable for mobile robotic systems. The response time constraint of the COM tracker circuit can be determined by identifying the major time-constant nodes of the circuit. For this purpose, figure 4, showing the small signal (perturbation signal) equivalent circuit for the COM detector's signal path during the comparison phase, can be utilised. Here r_{xon} is the finite on-resistance of the transmission-gate switches (which ideally should be very small). All other symbols and notations have their usual meaning for a standard small signal MOSFET model. The dominant time constants are at the outputs of the inverter-amplifier 1 and the inverter-amplifier 2. The time constant (τ_1) at the output of inverter-amplifier 1 is given by (including the miller capacitance due to the C_{gd} of inverter-amplifier 2) $\tau_1 \approx \left[\frac{r_{op} r_{on}}{r_{op}+r_{on}}\right]*$

$$\left[\begin{array}{l}(C_{gdp}+C_{gdn})+ \\ \dfrac{Ca*(C_{gsp}+C_{gsn}+g_{mp}r_{op}C_{gdp}+g_{mn}r_{on}C_{gdn})}{Ca+(C_{gsp}+C_{gsn}+g_{mp}r_{op}C_{gdp}+g_{mn}r_{on}C_{gdn})}\end{array}\right] \quad (10)$$

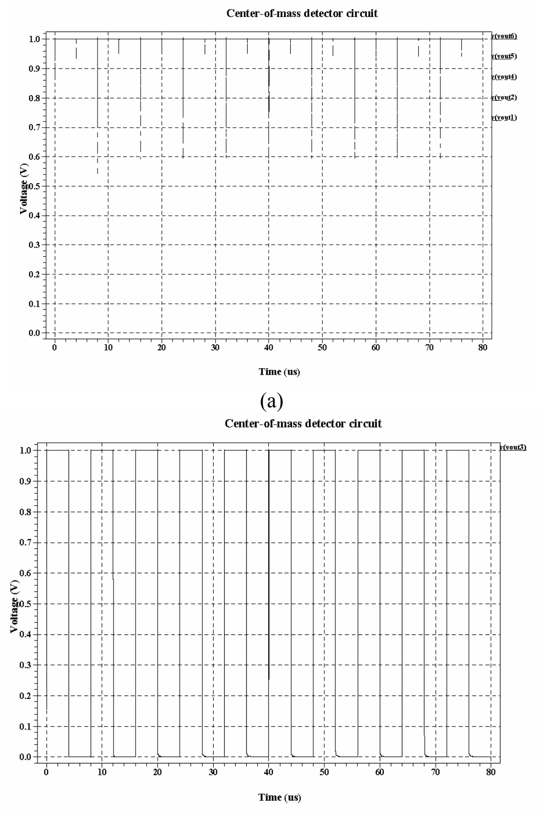


(a)

(b)

(c)

Figure 5: Final output of the 6-input COM detector circuit for a COM of 0.46, (a) outputs for positions 1, 2, 3, 4 and 6, (b) output for position 5, and, (c) the clock phase 2.



(a)

(b)

Figure 6: Final output of the 6-input COM detector circuit for a COM of 0.28: (a) outputs for positions 1, 2, 4, 5 and 6; (b) output for position 3.

Next, the time constant (τ_2) at the output of the inverter-amplifier 2 is given by

$$\tau_2 \approx \frac{r_{op} r_{on}}{(r_{op} + r_{on})} * (C_{gsnL} + C_{gspL} + C_{gdn} + C_{gdp}) \quad (11)$$

where C_{gsnL} and C_{gspL} are the logic gate input capacitances. The overall -3dB clock bandwidth for the COM tracker is then given by

$$\omega_{3\,dBClock} = \frac{1}{\tau_1 + \tau_2} \quad (12)$$

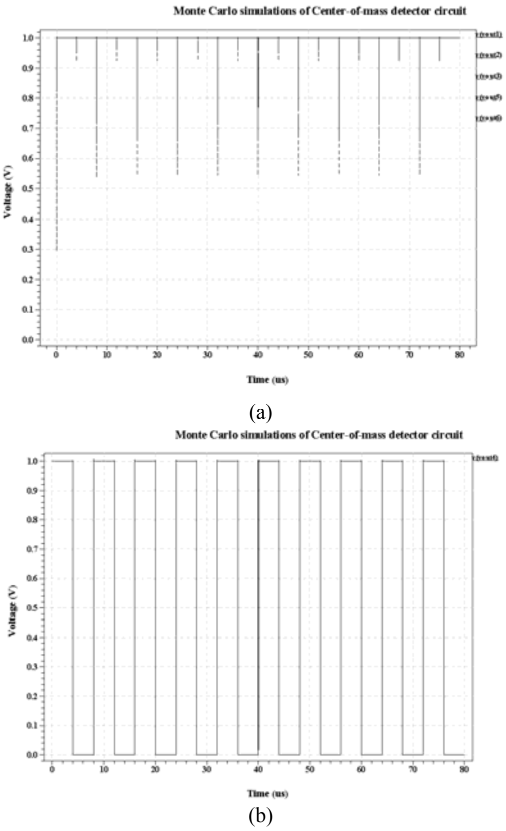
Inspecting equations (10) and (11), it is clear that the response is dominated by the time constant τ_1 at the output of the inverter-amplifier 1. The -3dB clock bandwidth of the comparator can then be approximated (assuming that Ca is large compared to the parasitic device capacitances and the miller multiplied capacitances by an order of magnitude) by

$$\omega_{3\,dBClock} \approx \frac{1}{\frac{r_{op} r_{on}}{(r_{op} + r_{on})} * C_{tot}} \quad (13)$$

where

$$C_{tot} = (C_{gdp} + C_{gdn} + C_{gsp} + C_{gsn} + g_{mp} r_{op} C_{gdp} + g_{mn} r_{on} C_{gdn}) \quad (14)$$

The -3dB clock bandwidth thus depends on the size ($W*L$) of the transistors and the bias drain current (I_D). As the intrinsic inverter-amplifier gain ($g_m r_o$) is $\propto \frac{1}{\sqrt{I_D}}$ and output impedance r_o is $\propto \frac{1}{I_D}$, the response time will trade with the output signal swing driving the logic gates and hence the noise margins.



(a)

(b)

Figure 7: Final output of the 6-input COM detector circuit for a COM of 0.38: (a) outputs for positions 1, 2, 3, 5 and 6; (b) output for position 4, considering threshold voltage variation by over 25% after 100 iterations of Monte Carlo simulations.

4 Conclusion

A low-voltage centre-of-mass tracker circuit design using only 1V supply voltage has been achieved. The significance of this circuit compared to other recent COM tracker circuits is that it does not require floating-gate POLY2 process layer and can be conveniently fabricated in any standard low-cost digital CMOS process technology. Also, although a 6-position COM tracker is presented in this paper, the novel architecture is very modular and can be easily extended to any number of COM positions. Thus, the precision achievable by the COM tracker circuit at such low supply voltage makes it favourable for applications in mobile robot visual object tracking systems.

5 References

[1] I. Ishii, Y. Nakabo and M. Ishikawa, "Target tracking algorithms for 1ms visual feedback system using massively parallel processing", *Proceedings IEEE Int. Conf. Robotics and Automation*, Minneapolis, 3, pp 2309-2314 (1996).

[2] N.M. Yu, T. Shibata and T. Ohmi, "A real-time center-of-mass tracker circuit implemented by neuron MOS technology", *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing,* 45(4), pp 495-503 (1998).

[3] K. Nakada, T. Asai, and Y. Amemiya, "Analog CMOS implementation of a CNN-based locomotion controller with floating gate devices", *IEEE Trans. Circuits and Systems-I: Regular Papers,* 52(6), pp 1095-1103 (2005).

[4] W. Weber, S.J. Prange, R. Thewes, E. Wohlrab and A. Luck, "On the application of the neuron MOS transistor principle for modern VLSI design", *IEEE Trans. on Electron Devices,* 43(10), pp 1700-1708 (1996).

[5] X. Li, Y. Yang, and Z. Zhu, "A CMOS 8-bit two-step A/D converter with low power consumption ", *Proceedings IEEE Int. Workshop VLSI Design and Video Tech.,* Suzhou, China, pp 44-47 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Autonomous Traversal of Rough Terrain Using Behavioural Cloning

M. Waleed Kadous, Claude Sammut, Raymond Ka-Man Sheh
ARC Centre of Excellence for Autonomous Systems
Computer Science and Engineering, University of New South Wales, Sydney, Australia
waleed@cse.unsw.edu.au, rsheh@cse.unsw.edu.au, claude@cse.unsw.edu.au

Abstract

Behavioural cloning is a method for acquiring skills by building generalised behaviours based on the observations of human performance. In this work, we examine the application of behavioural cloning to autonomous navigation of a robot in an unstructured environment. In particular, we examine the traversal of the random stepfields, introduced in the Robocup Rescue Robot League competition, by a tracked vehicle. One critical issue in behavioural cloning is representing the state of the environment in a manner amenable to machine learning. Our representation and behavioural cloning technique was evaluated by implementing it on our CASTER advanced mobility robot and training it to traverse a sequence of two NIST-specification stepfields. Despite being derived from only ten training runs, the cloned behaviour was able to successfully traverse the stepfield unaided 40% of the time and required only minor intervention the remaining 60% of the time.

Keywords: behavioural cloning, rescue robotics, modelling terrain

1 Introduction

One commonly cited example where robots can be of benefit to humanity is urban search and rescue (USAR). Robots are deployed at a disaster site and autonomously search the area, co-ordinate with each other, deliver assistance to those in need and assist in rescuing survivors. While distant, research towards this goal continues.

The robot rescue environment is highly unstructured. Because of this, approaches to control the robot have so far focused on tele-operation. However, in real rescue situations, radio communication is unreliable, tethered robots have limited range and operators have limited attention. Clearly, therefore, some degree of autonomy is extremely desirable. But how can such autonomy be achieved?

One promising technique is *behavioural cloning*, a process by which the actions of a human operator are recorded whilst the robot is being controlled. This behavioural trace is pre-processed and then input to a machine learning program that outputs control rules capable of driving the robot. Behavioural cloning has already been demonstrated in tasks such as piloting aircraft [1] and operating a container crane [2]. However, in those domains, the information required to control the systems was relatively easy to obtain. This is not the case for robots in an unstructured environment.

In this paper, we propose methods for characterising the environment in a way that makes behavioural cloning possible. We evaluate the proposed approach on simulated rubble: a NIST specification stepfield. Our results show that even with minimal training examples (traversing approximately 20 metres of rubble), the learnt clone can perform well – completing autonomous runs over difficult terrain 40 per cent of the time and requiring simple interventions in the remaining 60 per cent. In one case it outperformed the human operator who trained it.

2 The Rescue Robot Environment

The annual Robocup Rescue Robot League (RRL) competition encourages developments in mobility, sensing, autonomy, mapping and human-robot interfaces. An arena is created in which competing teams must demonstrate the above skills.

One particularly challenging element introduced in the 2005 competition was the random stepfield, which consists of 121 wooden blocks of varying heights (between 4.5cm and 36cm) arranged in a grid. This is intended as an experimentally reproducible representation of the type of rubble one would find at a disaster site. The blocks are not securely fastened, so the terrain can change based on the actions of the robot. Clearly, analytical approaches for elements such as stepfield are likely to be very complex.

Instead, our approach is to record the actions performed by a human operator who controls the robot as it traverses stepfields and to create a “clone” of their behaviour. The intended result is a control policy that can take sensor input and autonomously control the robot to traverse stepfields. As with other branches of machine learning, developing a suitable representation is vital for effective performance. We propose a representation that facilitates the application of standard supervised classification techniques.

3 Background

To understand the proposed approach, behavioural cloning will first be presented followed by a brief survey of existing approaches to terrain representation. The hardware platforms under consideration will also be discussed.

Experts are people who know what they are doing but don't necessarily know what they are talking about. By that, we mean that by the time a task becomes second nature to a person, most decision making is subconscious and therefore not accessible to introspection.

However, it is possible to construct a model of an expert's skill by recording the actions of the expert, along with the corresponding state of the system at the time of each action. This can be processed to create a controller that mimics the behaviour of the expert. The controller is thus a "clone" of the expert, and can be constructed by using machine learning to predict the action based on the state.

This approach of *behavioural cloning* is a form of learning by demonstration [3] that tries to model the behaviour of a human operator rather than modelling the system. Consistently with behaviour-based robotics [4], it eschews development of an explicit model of the world: we simply model the users' actions based on the sensor input. Distinctively however, the behaviours are learnt by observation rather than by being programmed.

This approach was first done by [5]. The method was to record human operators as they controlled a simulated pole-balancer. Subsequent work was done by [1] to apply behavioural cloning to learning to fly an aircraft in a flight simulator and by [2] to control a container crane. In these cases, it was relatively easy to define the inputs to learning. For example, to learn to pilot an aircraft, we recorded each time the pilot moved the joystick or adjusted throttle or flaps.

Representing the state of the world, when piloting an aircraft, is relatively easy since the instruments in the cockpit provide sufficient information to control the system. Controlling a ground vehicle in an unstructured environment is much more complex since it must negotiate uneven terrain with different surface textures. The challenge in modelling the operator's behaviour in the case of our rescue robot has little to do with the learning algorithm. The greatest difficulty is in finding a representation of the environment that provides the appropriate information for learning. By appropriate, we mean that the information must be sufficient to be able to learn control rules but it must also be concise enough not to swamp the learning algorithm with too much data.

Before moving on to describe our representation, it is worthwhile contrasting our approach to behavioural cloning with *apprentice learning* [6]. The key difference between behavioural cloning and apprentice learning is that in the latter, we learn a model of the system, while in the former, we learn a model of the operator. The reasoning is that since we have data of a human controlling the system and the human has already learned a good control policy, why not learn directly from the human?

Several groups have addressed the issue of 3D terrain representation in the context of robot control, although most have done so for the application of autonomous road vehicles.

Representations such as those used in [7] process pointclouds sensed at a particular instant in time and do not make use of an ongoing map. Obstacles are segmented based on their deviation from the driving surface, which need not be flat but is considered to be easily traversable.

Other approaches that attempt to segment out interesting features from raw pointclouds include [8] where a statistical approach was used to classify sections of pointclouds taken from an autonomous vehicle in a forest environment and used to detect such obstacles as trees and wires.

Alternative representations can, instead, consist of occupancy grid based techniques, with some relying on map generation. [9] used a 3D scanning laser to detect solid obstacles hidden amongst sparse vegetation by maintaining an occupancy grid.

[10] also use an occupancy grid based technique, with stereo vision on a planetary rover used to build an occupancy grid map of the terrain around the robot. This is then analysed for terrain "goodness" by observing the roll, pitch and roughness of patches of ground comparable in size to the robot.

4 Approach

4.1 Platforms

For these experiments, we used the robot CASTER, which was the basis of our entry in the RRL Competition in 2005, where we came third out of 26 teams. One of the distinguishing features of our entry was the ability to generate 3D maps of the arena. CASTER is built on a Yujin Robotics Robhaz DT3 base [11]. The robot has two pairs of differentially driven rubber tracks for locomotion. The robot is articulated in the center, allowing it to follow terrain such as stairs. The DT3 robot base can move in highly unpredictable ways on unstructured terrain due to its length, suspension and skid-steering properties. Two main concerns are getting stuck on an obstacle (e.g. an object lodging under the body of the robot) and flipping or rolling the robot when it attempts to climb an obstacle that is too high.

The core of CASTER's mapping and 3D sensing capabilities lies in the range imager [12] and a web camera. Rather than providing colour values for each pixel, the range imager provides distances. The addition of an accelerometer to measure tilt and roll enables the production of textured, level 3D reconstructions.

Initially the goal of our work is to develop a system that will be able to safely and autonomously drive in a straight line over obstacles such as stepfields. Subsequent work will then examine more complex issues, particularly turning on the stepfield. Note that while simplified, driving in a straight line over a stepfield is a highly non-trivial process as the terrain will cause the robot to veer from side to side and at times it may be necessary to aim off-center to avoid obstacles or to ensure that the robot's tracks hook onto desired parts of the terrain.

Our approach is to use the simplest form of behavioural cloning, known as "situation-action" behavioural cloning. This consists of three stages: the gathering of training data, building of the clone and execution of the clone.

During training, at each step the situation is viewed by the operator. The operator selects an appropriate action which is then executed. The operator is presented with a new situation for the next step. This continues until the task is complete. During this phase, the situation, in the form of an image from the range imager, and the roll and pitch of the robot, are recorded along with the operator's action taken in response to that situation.

The clone is built by first extracting features from the recorded situations, in this case the range images, to form a vector that numerically represents the situation, amenable to machine learning. The corresponding actions become class labels and the problem treated as a supervised classification task. Standard machine learning algorithms can then be applied to produce a classifier that can determine the class – in this case an action – corresponding to the feature vector from an unseen situation.

Finally, the clone is executed by sensing the situation, extracting the representative feature vector from the sensed data, again the range image, applying the classifier built in the previous stage to determine the appropriate action, then performing the action which is then executed. This process repeats for each time step until the task is complete.

4.2 Representation of the Problem

We represent the actions CASTER can perform as one of those typical of a vehicle: forward left, forward, forward right, spin left, spin right reverse left, reverse and reverse right. In practice, this combination of actions was found to be sufficient for a human operator to effectively control the robot.

Representing the situation consists of two components: representing the state of the robot and representing the terrain around the robot. The state of the robot in this case involves the roll and pitch.

To determine the appropriate representation of the terrain, we spoke to human drivers about how they drove the DT3. Strategies for human control tended to focus on two types of features of the stepfield directly in front of the robot, to a distance of about one robot length. The general layout of the terrain, such as flat, hill or valley would determine overall strategy. Particular obstacles such as blocks or holes that deviate from this general shape would then determine particular actions.

The terrain is captured using the range imager, which is pointed at the terrain immediately in front of the robot. After processing, this generates a cloud of points that lie ahead of the robot as shown in figure 1.

The pointcloud obtained from the range imager is converted into a height map, where the vertical height above ground level z is a dependant variable in x and y. To represent the terrain, a plane is fitted to the points using multivariate linear regression. This gives an equation of the form $z = ax + by + c$. Thus, a indicates the tilt of the terrain around the robot's roll axis and b the tilt around the robot's pitch axis. This gives a general characterisation of the terrain and the way in which it is leaning. The values a, b and c become features in the situation description.

Once this plane is fitted, holes and protrusions can be detected. The space in front of the robot is divided into a 3×3 grid. Each of the grids is then divided into a 10×10 subgrid. Within each subgrid square, the average z value is computed amongst all points bounded by the x and y limits of that subgrid square. The subgrid square which has the maximum absolute difference between the

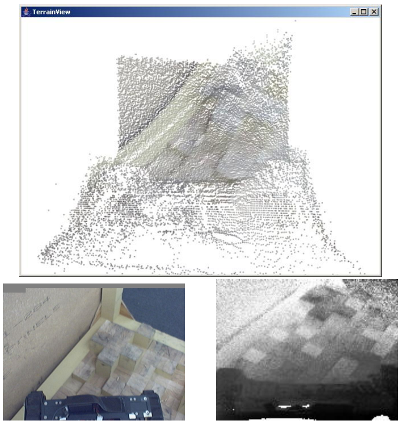


Figure 1: A pointcloud generated by the range imager (top) and the corresponding colour and range images (bottom).

average z value and the z value predicted by the plane of best fit for the centre of the subgrid square is found. In a sense, this tries to find the biggest obstacle (hole or protrusion) in that grid square. The x and y indices of the subgrid square with the maximum absolute difference become features for each grid square. This captures information about where the biggest obstacle within each grid square is. For example, if there is a large protrusion on the right hand side of grid square $(0, 1)$, the x index would be 9, whereas if the obstacle was on the outer edge of the stepfield, the x index of the biggest obstacle would be 0. In addition, the sign and magnitude of the obstacle is used as a feature. Thus, for each of the 9 grid squares, 3 features are generated.

Situation-action clones are essentially Markovian or reactive: they make the decision as to their actions based solely on the current state. This means that the robot has no concept of "getting stuck", and it will continually repeat the same action in the same situation, no matter how many times this has occurred. In order to prevent this, features were added that attempted to capture some information about change over the last interval. The technique is based on comparing consecutive frames from the range imager and measuring an average change per pixel. This is more robust to changes in lighting and slight movement of the robot than the visual camera, since fine texture changes will produce large average differences between pixels. This difference was also thresholded at approximately 1.5cm/pixel and a count generated of the last time this threshold was exceeded – this count was used as an indication of the number of frames during which the robot had been "stuck" for. This allows a distinction in the situation space between the actions taken by the operator when the robot first becomes stuck and the actions taken to extricate the robot, even if the situation has not otherwise changed, while maintaining the simplicity of the situation-action approach.

Thus, in total, the situation is represented by 3 features representing the general situation of the terrain, 27 features representing obstacles in different grids in front of the robot, 2 features representing the tilt of the robot and 2 features representing how long and how much the robot has moved since the last action. In total, 34 features are used.

5 Evaluation

5.1 Experimental Setup

To evaluate the approach, CASTER was modified for the purposes of the experiment. Rather than the camera head being mounted on a pan-tilt unit as usual, it was mounted on the end of a robot arm and pointed downwards to maximise resolution of the sensors over the area of interest. A photo of the configuration is shown in figure 2. The view from the robot arm captures the front part of the robot as well as the area immediate to the sides and in front of the robot as shown in figure 1. The cameras sit at an angle of approximately 15 degrees from straight down and is approximately 1 metre above the ground.

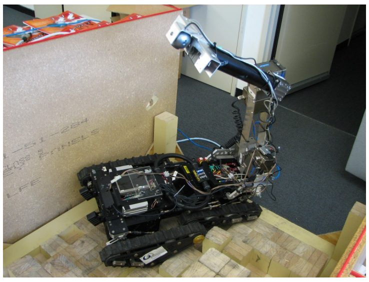


Figure 2: CASTER traversing the stepfield. Note the position of the arm.

To test the techniques, a sequence of two stepfields of official NIST construction were prepared. They were configured to be similar in difficulty to the Robocup 2005 competition stepfields.

A user interface was developed that allowed the operator to look at the scene from the top-down perspective. The operator was instructed to drive the robot over the stepfield as quickly as possible. We chose an experienced operator for these experiments who had approximately 50 hours of driving experience.

Training runs, during which a human operated the robot, were started with the robot approximately lined up with the center of the stepfield. The operator, based only on the scene observed through the range imager and colour camera, chose the direction to travel in. The robot would proceed along this path for one second. The operator would then have to make a choice again. This constituted a single step of the operation. The experiment would terminate when the operator reported that only saw carpet on the screen, which would occur when the robot had completed traversing the two stepfields.

For each step, the images seen by both the camera and the range imager, the pitch and the roll of the robot and the action performed by the operator were recorded.

The stepfield was traversed 8 times in this manner to gather sufficient data for training (one run was not completed due to technical difficulties midway through, but the data up until the point was recorded). A typical run, including set up and run time, would take approximately 15 minutes. Two additional datasets were also collected that involved the robot being placed in a number of unusual and potentially dangerous situations (such as very close to an obstacle or at an unusual angle) and the operator manoeuvring the robot back to a safe state. Other research in behavioural cloning has shown that such "difficult problem" training can be an advantage. Since a clone may not be identical in its behaviour to the expert, it may end up in situations that were not similar to the training data were it collected from only "normal" runs. Actions of the operator in less frequent but still important "difficult situations" can therefore improve performance significantly in such tasks.

Once the data was collected, feature extraction was conducted as discussed. This was then processed using J48 from Weka [13] to produce a decision tree. In order to increase the amount of data for training, the data was mirrored along the front-to-rear axis of the robot. For each training example, the terrain data, the roll and the action were mirrored. If the action had no direction element (e.g. forward), it was set to the same value, while, for example, a forward-left became a forward-right. This served to increase the number of examples, while simultaneously ensuring that the robot had no conceptual bias to veer to either the left or the right.

The robot, as for the training cases, was started approximately in line with the center of the stepfield. The robot was then allowed to continue operating until an observer decided it was irrevocably stuck, or it reached the other side of the sequence of stepfields. The operator was not involved in the operation of the robot except to start the robot and, when necessary, determine that the robot was stuck and stop the robot. When the robot became stuck, an "intervention" was recorded and the robot moved the minimal amount required such that it could proceed safely.

Table 1: Performance Operator (Op), Behavioural Clone and Always Forward (AF)

Approach	Runs	Steps	Intervns	Success
Op	7	75.7	0.14	0.86
BC	5	71.8	0.6	0.40
AF	5	45.5	2.0	0.00

In order to provide a baseline for comparison, this method was also compared with a default "always forward" action. This is because empirically it was observed that 67 percent of actions taken by the operator were to drive forward.

5.2 Results

A total of 707 actions over the eight normal and two "difficult" runs were collected. 67 per cent of these actions were instructions for the robot to drive forward (and hence our baseline "always forward" action seems to be well-founded). In comparison to the massive situation space (34 dimensions, many of them continuous) and the complexity of the task, this is a very small training set.

Table 1 shows a comparisons of the different techniques. The obvious measure of success is the number of times that the stepfield was successfully traversed. As can be seen below, the human operator was able to traverse the stepfield 86 per cent of the time – the task is sufficiently difficult that even the human operator got the robot irrevocably stuck at one point. The clone was able to complete the traversal 40 per cent of the time. It required only three interventions over the 5 runs and each of these interventions only required minor repositioning of the robot.

The "always forward" controller failed to complete any traversals, and required constant interventions. It required an average of 2.0 interventions per run and many of these were due to the robot becoming dangerously close to rolling as it collided with the sidewalls of the stepfield. Clearly this approach is impractical, and significantly worse that behavioural cloning.

The clone appears to have acquired some of the characteristics of the human operator.

5.3 Empirical Observations

Several interesting and anomalous behaviours were observed and noted. Firstly, the best run (in terms of having zero interventions and minimal steps) was actually performed by one of the clones – it completed the stepfield in 48 steps with zero interventions, while the best human did so in 53 steps. The operator's comments on seeing it perform was that “it drove like I would have!” Secondly, the robot would also do things that to a human would appear “strange.” One of such strange actions is that the robot would sometimes drive forward, then drive backward, then drive forward again. It must be remembered that the clones being built here are purely reactive or Markovian. While the SR differencing technique helped prevent it from repeating actions when stuck, it did not help the robot getting stuck in a loop between two actions.

6 Conclusions and Future work

The experiments reported here show that behavioural cloning is a promising approach to building a controller for locomotion over rough terrain. For simplicity, these initial experiments adopted the simple situation-action model of behavioural cloning.

Previous application of behavioural cloning in piloting aircraft have demonstrated that generalisation can be improved by decomposing the controller into two components. The first determines the appropriate values for “goal variables” and the second determines the actions required to achieve those goal values. Future experiments with locomotion will investigate methods for problem decomposition that will achieve similar improvements in robustness.

Clearly one possibility is to study the efficacy of different learning algorithms such as support vector machines and naive Bayes as possible clones. It is also worth noting that having to use a propositional learner because of the noisy and numerical data that exists in this problem domain imposes some difficult limitations, such as the fixed 3 × 3 grid representation of the world. Obviously driving in this domain would benefit from using a relational representation of the obstacles and their spatial relations to one another, and thus relational learning could be useful. Unfortunately, relational learning techniques are not usually capable of handling noise and numerical data of this kind.

Finally, it is not clear whether cloning for one particular stepfield generalises to other terrains and stepfields. We therefore plan to collect training data from multiple stepfields and then test on new unseen stepfields.

7 References

[1] C. Sammut, S. Hurst, D. Kedzier, and D. Michie, “Learning to fly”, *Proceedings 9th International Conference on Machine Learning*, Aberdeen pp 385-393 (1992).

[2] T. Urbancic, I. Bratko, “Reconstructing human skill with machine learning”, *Proceedings of the 11th European Conference on Artificial Intelligence*, Amsterdam pp 498-502 (1994).

[3] C.G. Atkeson, S. Schaal, “Robot learning from demonstration”, *Proceedings Fourteenth International Conference on Machine Learning*, Nashville, pp 12-20 (1997).

[4] R. Brooks, “A robust layered control system for a mobile robot”, *IEEE Journal of Robotics and Automation*, 2(1) pp 14-23 (1986).

[5] D. Michie, M. Bain, J.E. Hayes-Michie, “Cognitive models from subcognitive skills”, in M. Grimble, S. McGhee, and P. Mowforth eds, *Knowledge-base Systems in Industrial Control*, Peter Peregrinus pp 71-99 (1990).

[6] A.Y. Ng, H.J. Kim, M.I. Jordan, S. Sastry, “Autonomous helicopter flight via reinforcement learning”, *Advances in Neural Information Processing Systems 16*, Vancouver (2003).

[7] A. Talukder, R. Manduchi, A. Rankin, L. Matthies, “Fast and reliable obstacle detection and segmentation for cross-country navigation”, *Proceedings IEEE Intelligent Vehicles*, Versailles pp 610-618 (2002).

[8] N. Vandapel, D. Huber, A. Kapuria, M. Hebert, “Natural terrain classification using 3D ladar data”, in *Proceedings IEEE International Conference on Robotics and Automation 2004*, New Orleans pp 5117-5122 (2004).

[9] C. Wellington, A. Stentz, “Online adaptive rough-terrain navigation in vegetation”, In *Proceedings IEEE International Conference on Robotics and Automation 2004*, Versailles pp 96-101 (2004).

[10] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, K. Schwehr, “Recent progress in local and global traversability for planetary rovers”, In *Proceedings IEEE International Conference on Robotics and Automation 2000*, San Francisco, pp 1194-1200 (2000).

[11] Yujin Robotics, Robhaz DT-3 Robot, http://www.robhaz.com/about_dt3_main.asp, Visited 18/08/2005.

[12] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, N. Blanc, “An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger)”, *Optical Design and Engineering. Proceedings of the SPIE*, 5249, pp 534-545 (2004).

[13] I.H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann (1999).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Developing an Emotion-based Architecture for Autonomous Agents

Chris S. Tingley and Will N. Browne
Cybernetics
University of Reading, Reading, Berkshire, UK
w.n.browne@reading.ac.uk

Abstract

The importance of emotions in control mechanisms for autonomous agents is demonstrated using real and virtual robotic platforms. A novel agent architecture was developed to provide a foundation for emotion-based control. An appropriate test platform was created allowing real and virtual agents to coexist and allowed production of a number of emotional rules. The emotion-based architecture is shown to provide a number of benefits over conventional approaches, which include simpler behavioural programming and improved performance on complex exploration tasks.

Keywords: autonomous agent architecture, emotion-based reasoning

1 Introduction

Emotions are known to be linked into how humans make decisions and help guide the way our species lives in both work and play [1]. They are also known to be important in controlling and regulating ways that humans communicate and how humans express themselves in a verbal and no-verbal sense [2]. The difficulty comes when attempting to classify emotions and to what they actually relate. To this extent, many different definitions for emotions are often used in varying contexts; including neuro-endocrine, physiological, behavioural, cognitive, communicative and social settings.

It is considered here that biological emotions are simply a result of neuro-mechanisms linked to actions and consequences. Rolls, [3] defines emotions as “the states elicited by reward and punishers, including changes in those rewards and punishments”, by which “a reward is anything for which an animal will work, i.e. a Positive Reinforcer” and “a punishment is anything that an animal will work to escape or avoid, i.e. a Negative Reinforcer”. Emotions are based on experiences and help guide future actions.

“The question is not whether intelligent machines can have any emotions, but whether machines can be intelligent without emotions” [4], and to facilitate this; Sloman [5] suggests that “[Reactive mechanisms] make possible alarm-driven primary emotions” while “[deliberative mechanisms] make possible secondary emotions using global alarm mechanisms linked to deliberative [processes].”

This project aims to develop an emotional-based architecture for implementation in autonomous agents in both real-world and virtual environments. Once an architecture has been established, the effectiveness of the emotional agents will be tested using a custom built development platform. It is hypothesised that an emotional-based architecture, that is inherently non-linear, will produce the following advantages over a traditional deterministic rule-based architecture due to emergent properties:

- Provide a framework allowing simpler creation and development of generalised agent behaviours.
- Yield faster and more complete abilities e.g. environment exploration.
- Reduce bandwidth requirements between agents and/or multi-agent controllers.

A further point of interest will be in observing the emergent behaviours (if any) of the multi-agent system and how heterogeneous emotions within the system affect its overall performance.

2 Uses of Emotions in Autonomous Agents

Any agent (humans included) operating in a dynamic and unpredictable environment suffers from the problem of selecting between conflicting short and long term goals. In the biological world, emotions are one of many mechanisms employed to deal with such situations and their associated problems. To address similar problems in autonomous agents, biological emotions[1] can be examined for inspiration, other research exploring the uses of emotions within autonomous agents can be found in [6] and [7].

Classical Artificial Intelligence (AI) within autonomous agents started with purely reactive and/or deliberative systems [8]. These are effective at dealing with immediate situations (i.e. reacting on instinct) by matching situations to actions through a linear

[1] When referring to autonomous agents, the word “emotion” is used to provide a simple description of a biologically inspired emotional control mechanism for inclusion in an autonomous agent.

deterministic process. The subsumption architecture [9] allows complex interactions and behaviours to emerge and has been widely adopted as a basis for agent architecture design [10], albeit with minor modifications between incarnations. It is anticipated that a multi-layered emotional-based architecture will add flexibility to the process of action selection and produce generalised responses in similar situations. The desired operation should work akin to fuzzy logic where rule boundaries are not crisp and are non-linear in their processing.

A wider view of the benefits of emotions within autonomous agents are given in [11]. Scheutz describes 12 roles of emotions within emotional agents: Action selection - and the ability to decide what to do next based on current emotional state(s) or mood. Adaption - of behavioural traits or goals either in short-term or long-term due to emotional state(s). Social regulation - communicating or exchanging information with other agents via emotional expressions. Sensory integration - and abstraction by emotional filtering of data or blocking of specific sensory integration dependent on a situation. Alarm mechanisms - fast, reflex-like reactions in critical situations that interrupt other processes. Motivation - creating motives as part of an emotional coping mechanism. Goal management - and the creation of new goals or reprioritisation of existing ones. Learning - the use of emotional evaluations e.g. as Q-values in reinforcement learning. Attentional focus - the selection of data to be processed based on emotional evaluation Strategic processing - the selection of different search strategies based on overall emotional state. Self model - emotions as representations of "what a situation is like for the agent", e.g. self evaluation and categorisation.

Importantly, these uses can be seen to have parallels with the characteristics of multi-agent systems. Sycara [12] characterises multi-agent systems as:

1. Each agent having incomplete information or capabilities to solve the [given] problem and, thus has a limited viewpoint;

2. There is no global system control;

3. Data is decentralised and

4. Computation is asynchronous.

In order to improve the facilitation of points three and four, it is hypothesised that certain emotional outcomes can be communicated between agents rather than being used to directly control the behaviour of the agent. This would alleviate the problem of communicating large quantities of sensory or unprocessed information between agents by communicating a smaller set of abstracted emotional values. This fits with the hypothesis that bandwidth between agents and controllers can be reduced by using an emotion-based architecture.

3 The Architecture

The architecture developed consists of three layers, see figure 1, each varying in their level of complexity. The lowest layer allows for reactive (reflex like) behaviours, the middle layer controls goal management, whilst the highest (and most abstract) layer governs the overall emotional state of the agent. Lower levels, which are the focus of this work, are able to subsume emotions generated by higher levels and connection to a multiagent system controller is possible by the communication of non-observable emotions.

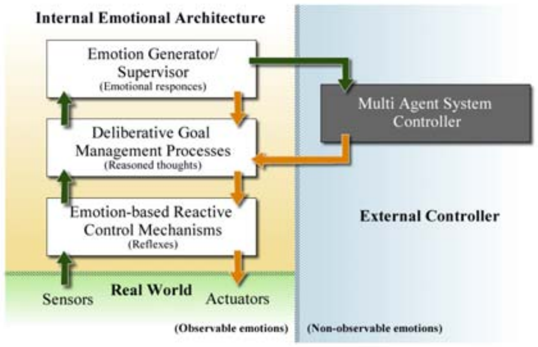

Internal Emotional Architecture
Emotion Generator/ Supervisor
(Emotional responces)
Deliberative Goal Management Processes
(Reasoned thoughts)
Emotion-based Reactive Control Mechanisms
(Reflexes)
Multi Agent System Controller
External Controller
Sensors
Real World
Actuators
(Observable emotions)
(Non-observable emotions)

Figure 1: The emotion-based agent architecture.

3.1 The Developed Emotional Controllers

Each input into the system is abstracted into one or more emotions and dependent on the relevance and magnitude of the input, a corresponding emotional "value" is attached. On each execution of the control loop, each emotional value is combined resulting in complex, non-linear behaviours. See figure 2.

Rolls describes ten functions of emotions [3] which can be seen to control lower order functionality such as fight or flight responses, or higher order processes such as direction of behaviour (goal management). These are similar to the ideas proposed by Scheutz in [11] who describes the use of emotions in controlling autonomous agents. Five emotions were chosen providing a number of instantaneous (curiosity, motivation, satisfaction) and time-dependent (anger, tiredness) processes and are considered a mix of low and high order. Their creation (generation) and effects (transformation) are simple to understand from a human perspective:

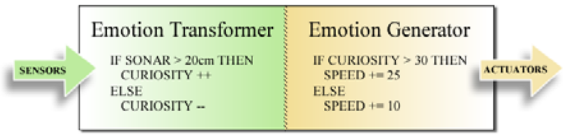


Figure 2: An example of an emotion transformer and associated transformer.

Curiosity is determined by what the agent is able to see in its neighbourhood. If the agent is in a large open space, it becomes curious and develops the desire to explore the space until it encounters another object. The level of curiosity directly affects the speed in which the robot moves within the environment and can be seen to have similarities to animal behaviour.

Satisfaction is based upon the agents' goals. The agents have a very simple task to perform, which is to flock towards the nearest other agent. The level of satisfaction is calculated from the average distance between the agent and all the other agents in the environment, but this fitness function can easily be replaced by any other measurable goal.

Anger is elicited when the agent becomes stuck in its environment, i.e. when the control system is instructing the agent to travel in one direction, but this is overridden by the reactive control mechanism that stops the agent from bumping into an object. This emotion is time-dependent: as long as the agent is stuck with conflicting interests, the level of anger will continue to rise. Once this level is sufficiently high, a secondary control mechanism will take over and temporarily overrides the current goal, thus allowing the agent to move out of the restricted area. As the agent reverts back to its original goal, the anger level will decay, however it will leave the agent more prone to reaching its anger threshold and its behaviour is slightly less deterministic than normal until the anger level has completely subsided. This simple emotional process can be seen to facilitate both action selection and adaptation. Simplistic goal management could emerge when the robot ignores its primary strategy.

Motivation is another time-dependent, cumulative emotion and is affected by other emotions in order to generate a compounded (complex) emotion. As the agents move around the environment, they will experience differing levels of curiosity. As their task is to flock together, if no other agents are in the vicinity, it is in the best interest for the agent to explore an area further away from its current position. Based upon this reasoning, as the agent actively explores its surroundings, its level of motivation will increase as it is considered that the task is being performed adequately. The level of motivation is only adversely affected by the onset of anger, which dramatically reduces the motivation level. This causes the agent to "want" to avoid situations that increase its anger level and which subsequently decrease its motivation.

Tiredness is determined from two inputs, battery level and the length of time the current task has been running. The level of tiredness is used when calculating the speed of the agent. Tiredness may also be considered a simple self-model, although possibly not a basic emotion [13].

4 Results and Discussion

The proposed architecture has been implemented into scalable, completely object-orientated multi-agent system controller applied to three physical Miabots[2] and a multitude amount of virtual agents, see figures 3 and 4. Miabots have eight pairs of evenly spaced ultrasonic sensors as input linked to two motors (0.04mm encoder resolution) as output. The benchmark system links inputs directly to outputs, e.g., if sonar > 20cm, then speed = 30m/s. The resultant system can be seen to exhibit three important features, all of which can be seen in some form in the list of important roles proposed by Scheutz [11].

Consolidation of inputs refers to the ability to easily consolidate a high number of external inputs, into fewer "abstracted emotional inputs". These algorithms are known as "emotion generators" and are located in the "Emotion Generator/Supervisor" level of the architecture. The developed architecture allows this to be achieved by a more natural, pragmatic and ultimately real-life approach than current architectures allow, thus making much simpler control mechanisms to be developed from a more natural viewpoint, e.g., when curious, move faster and so on.

After emotional abstraction has taken place, fewer inputs mean that the communication channels between individual agents and the multi-agent controller benefit from requiring less bandwidth as emotional values can be communicated rather than complete sensor readings. Turning the generated emotional levels into actions is accomplished using "emotion transformers" that are the algorithms that combine the emotional levels and convert them into movement. These algorithms are located within the reflex level of the architecture.

The control system functions on the virtual agents just as well as on the physical agents, suggesting that the architecture is feasible for virtual and physical agents alike.

Observers of the system considered that the agents with the emotional behaviours exhibited more natural and realistic responses to stimuli. This can be beneficial for any agent system in the general consumer market and may be even more desirable for virtual agents, where it is known to be difficult for people to relate or interact with virtual agents; as there is no tangible connection between the agent and the user.

In addition to being easier to develop, the emotional-based behaviours also requires less lines of code (LOC)[3] in order to produce them. The experiments detailed below use two separate architectures, the newly developed emotional architecture and a

[2] The Miabots are manufactured and supplied by Merlin System Corp and provide a flexible robotic platform. An accurate localisation system was developed to enhance the functionality of the Miabots with a location beacon affixed to the top of each Miabot.

[3] LOC is a simplistic method used to describe the complexity of a programmed function.

classical rule-based architecture. The reactive control for the rule-based system is written in approximately 100 LOC whereas the newly developed architecture requires under 40 LOC to produce the same (if not better) behavioural responses.

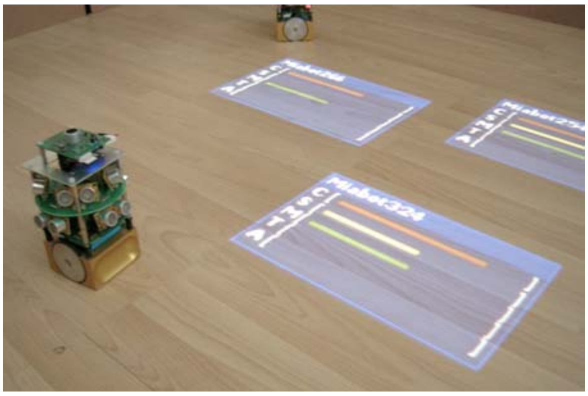


Figure 3: The mixed reality software overlaying information on an agents' emotional state.

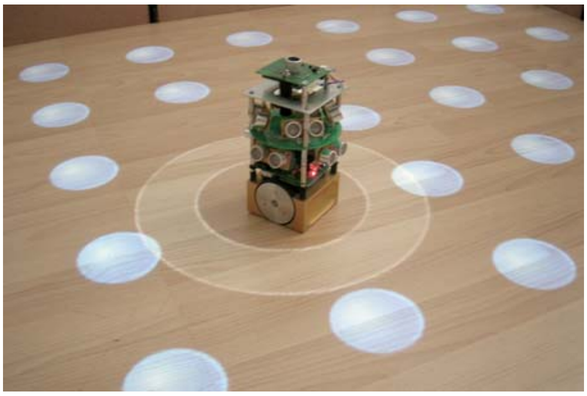


Figure 4: The location system tracking the Miabot within the virtual environment.

Although LOC is a good measure for a programmer to determine of the quantity of code needed to produce the behaviours, it suffers from an inherent problem that a better programmer may be able to refactor any given source code into fewer lines. The supporting classes and methods behind the algorithm must also be taken into consideration as they may affect the LOC required for any specific task.

4.1 Environment Exploration

In the first simple environment there were no objects to negotiate, it was simply an open space with a walled perimeter. The second (complex) environment had the same perimeter but also contained three objects located within the environment; creating non-perpendicular angles, corridors and open areas. This was to simulate unknown and irregular environments where standard path planning algorithms are unsuitable. In all cases a non-deliberative reactive control system was used to "wander" around the environment in order to set a benchmark for comparison.

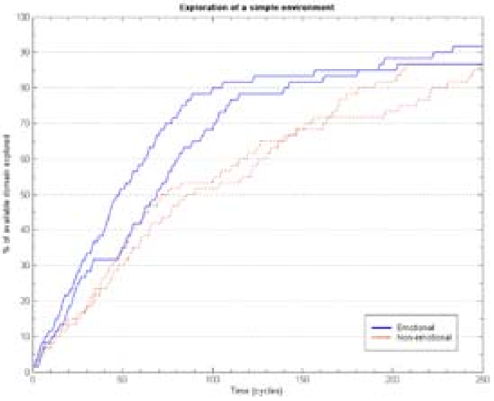


Figure 5: Comparison of an emotional based control architecture versus a non-emotional architecture when exploring a simple domain.

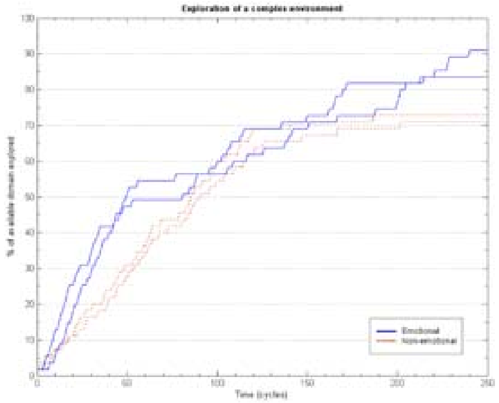


Figure 6: Comparison of an emotional based control architecture versus a non-emotional architecture when exploring a complex domain.

Figure 5 shows two runs of an emotional and non-emotional architecture in a simple domain. The graph is a plot of the percentage of the domain explored after each location cycle (approximately one second). In this graph, it can be seen that the emotional control system explores the environment[4] faster than the non-emotional control system reaching 80% exploration by 125 cycles compared to 200 cycles, making the emotional control system 1.6 times faster. At 250 cycles both architectures plateau at around 90% which was the entirety of the navigable domain.

The actual runs within the simple domain are shown superimposed in figures 7 and 8. With the non-emotional architecture the two plots are seen to be similar, even when the starting points are different, which underlines the deterministic nature of the standard architecture. The emotional architecture demonstrates a more random exploration path, noticeably covering more of the environment.

[4] The term 'environment' is used to describe the area covered by the localisation system whereas 'domain' is used to describe the area that the agent is able to explore.

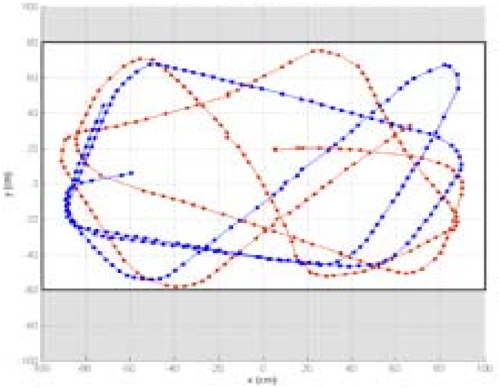


Figure 7: Non-emotional agent architecture exploring a simple domain for three minutes.

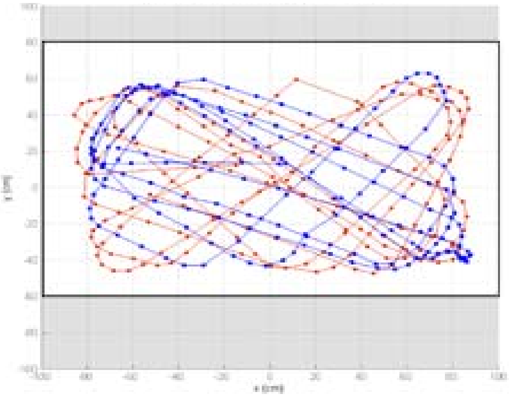


Figure 8: An emotional-based agent architecture exploring a simple domain for three minutes.

In the complex environment, the emotional-based architecture is also seen to perform better. Figure 6 has three interesting features: 1) from 0 to 50% of the environment explored, the emotional architecture is able to reach 50% exploration approximately twice as fast as the standard architecture. This leads to 2) where 50 to 70% of the environment has been explored and both architectures are seen to perform equally as well and finally 3) 70% and beyond where the non-emotional architecture is seen to stop exploring as opposed to the emotional architecture which carries on and reaches in excess of 90% of the total environment explored.

Again, superimposed runs are shown in figures 9 and 10. The non-emotional architecture produces almost identical paths when starting from the same position, which is common in deterministic systems. Introducing randomness to the controller was ineffective as the robot makes little forward progress. However, when considering the paths generated by the emotional system, it is obvious that both runs were completely different, but effective. This shows the non-deterministic nature of the emotional controller and explains the generality of responses generated by the emotional control system. The top-left area within the domain produces interesting results from the differing architectures; with the non-emotional architecture spending a long time navigating this area as opposed to the emotional architecture, which is able to navigate this area with little difficulty.

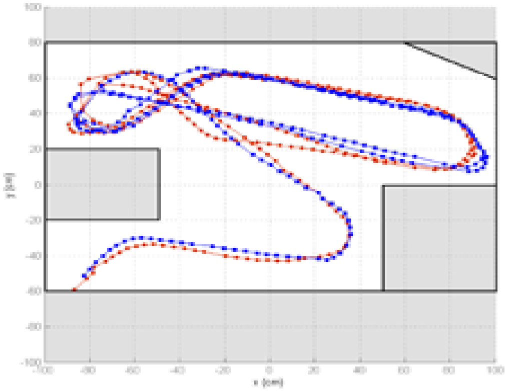


Figure 9: A non-emotional agent architecture exploring a complex domain for three minutes.

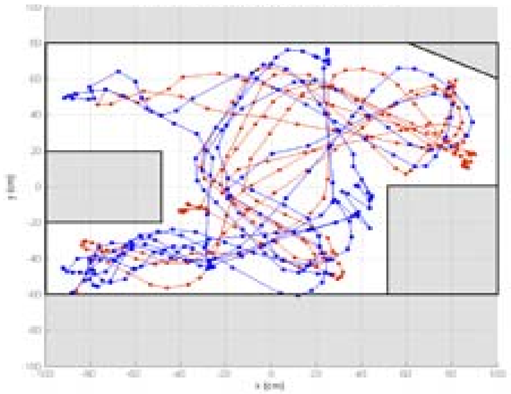


Figure 10: An emotional-based agent architecture exploring a complex domain for three minutes. Note the non-deterministic nature of the two runs.

4.2 Emergent Behaviours

A number of emergent behaviours were observed alongside the behaviours that were intentionally programmed as part of the emotion-based reactive control mechanisms. The most interesting behaviour of these gave the agent the ability to move itself out dead-ends and situations where the rudimentary control systems do not provide the intelligence to calculate a path around obstructions (as there is no deliberative control). As the agent approaches an obstacle, its curiosity level falls resulting in the agent moving more slowly. When the agent gets even closer (as it is unable to turn left or right due to object on the left and right hand sides) it will back away from the object. Because there are objects seen on both sides, the agent does not try and turn and so continues to approach and back away from the object directly in front of the agent. As this oscillation in movement continues it causes the agent to suffer from a prolonged period of diminished curiosity leading to increased anger levels. When such time as the anger emotion out-weighs the other emotion-based mechanisms this results in the robot backing away quickly and thus moving completely away from the obstructions. This behavioural trait had not been programmed into the agent, but gave it the ability to navigate its way out of awkward situations and demonstrates how complex interactions between the different abstracted emotions can take place and how they can generate desired behaviours. Importantly, a non-emotional architecture would have been indefinitely stuck in this loop, but the emotional nature of the developed architecture allows flexibility in responses and avoids infinite loops.

Unlike many layered robot control architectures, the interaction between reflexes and higher order control was not fixed directly, i.e. both the reflex trigger and effect changed with emotional level. A robot with a high level curiosity would approach a potential collision with higher speed, less gap and with less prioritised reflexes, than a standard robot. This led to the robot adapting its behaviours to its environment.

Not every emergent behaviour was useful and on occasion, the “emotion generators and transformers” had to be manually adjusted in order to prevent undesired behaviour. Adjusting the thresholds controlling how external states generate emotions and how those emotions are then transformed into movement is a simple process. It can be argued however, that these adjustments are much easier than developing rule based behaviours as it is often the case that an emotional value needs changing as opposed to the adoption of a completely new rule. It is also easier to design the behaviours from a human perspective as it is asked: “What sort of emotional responses would a human feel from a given situation?”, a sub-set of those responses can then be ported to the agent. This processes changes the way in which behaviours for autonomous agents are designed from: “What are the steps I perform in this unique situation?” to “How would I react to this situation?” which includes some degree of generalisation.

When comparing the rule-based system to the emotional-based system, it was seen that the emotional architecture performed better in both exploring the environment faster and to a greater extent. This was due to rule-based architectures having a low level of generalisation when compared with an emotional architecture. Due to the lower level of generalisation, the rule set tended to hinder the ability to explore complex areas where conflicting rules will generally lead the agent to avoid certain areas of the environment. The rule-based control system could be augmented to include speed adjustment, but this would require more rules, more LOCs and would lead to a more complex control system.

5 Conclusion

Three key benefits of an emotional-based autonomous agent architecture compared with classical, deterministic and linear rule-based architectures have been identified. Firstly, this framework allows simple creation of intuitive, non-linear and generalised agent behaviour. Secondly, it has been shown that an emotional-based architecture can yield faster environment exploration abilities. Finally, it is anticipated that if the developed architecture were implemented on-board the agents, bandwidth requirements between the agents and the controller would be significantly lower.

6 References

[1] M. Bond, “Science of happyness, part 1: 10 keys to a contented life”, *New Scientist*, pp 40-47 (2003).

[2] S. Planalp, *Communicating Emotion*, Cambridge University Press, London (1999).

[3] E.T. Rolls, *The Brain and Emotion*, Oxford University Press, Oxford (1999).

[4] M. Minsky, *The Society of Mind*, Simon & Schuster, New York (1988).

[5] A. Sloman and B. Logan, “Cognition and affect: architectures and tools”, *Proceedings of the 2nd International Conference on Autonomous Agents*, New York, pp 471-472 (1998).

[6] D. Canamero, “Issues in the design of emotional agents”, *Emotional and Intelligent: The Tangled Knot of Cognition: Papers from the 1998 Fall Symposium*, pp 23-27 (1998).

[7] K.R. Scherer, “Criteria for emotion-antecedent appraisal: a review”, in V. Hamilton, G. Bower and N. Frijda (eds), *Cognitive perspective on Emotion and Motivation*, Kluwer Academic, pp 89-126 (1988).

[8] H.P. Moravec, “The Stanford cart and the CMU rover”, *Proc. IEEE*, 71(7), pp 872-884 (1983).

[9] R.A. Brooks, *A Robust Layered Control System for a Mobile Robot*, Technical Report, Massachusetts Institute of Technology, Cambridge, MA, (1985).

[10] E. Gat, “On three-layer architectures”, in D. Kortenkamp, R.P. Bonasso and R Murphy (eds), *Artificial Intelligence and Mobile Robots*, MIT/AAAI Press, Cambridge, MA (1997).

[11] M. Scheutz, “Useful roles of emotions in artificial agents: a case study from artificial life”, *American Association for Artificial Intelligence AAAI 2004*, pp 42-48 (2004).

[12] K. Sycara, “Multiagent systems”, *AI Magazine*, 10(2), pp 79- 93 (1998).

[13] P. Ekman, “Basic emotions”, in T. Dalgleish and T. Power (eds), *Handbook of Cognition and Emotion*, John Wiley & Sons, Sussex, pp 45-60 (1999).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Texture-Based Segmentation of 3D Probabilistic Occupancy Maps for Robot Navigation

Bassel Abou Merhy, Pierre Payeur and Emil M. Petriu
School of Information Technology and Engineering
University of Ottawa, Ottawa, Canada
bassel@site.uottawa.ca, ppayeur@site.uottawa.ca, petriu@site.uottawa.ca

Abstract

In previous work, an unsupervised texture-based segmentation algorithm was introduced to process 2D probabilistic occupancy maps used to navigate mobile robots. Given that numerous robotic applications are carried out in 3D environments, this paper presents the extension of the approach to cover the case of 3D probabilistic occupancy maps. The proposed unsupervised segmentation technique relies on the analysis of texture defined by a double distribution of the "local binary pattern" (LBP) and the "contrast" (C) operators in order to subdivide the space into regions characterised by a uniform occupancy state. The algorithm is able to identify in separate segments the different objects present in an environment through the analysis of their proximity. In order to keep the method tractable in 3D applications, a redefinition of the texture unit and an adaptation of the subdivision process are presented that provide very satisfactory results while significantly reducing the size of the texture distribution histograms that need to be computed and iteratively compared to achieve the segmentation in uniform regions.

Keywords: 3D segmentation, probabilistic maps, local binary pattern, contrast, texture.

1 Introduction

Probabilistic occupancy grids offer a powerful mode of representation of occupancy maps computed from the fusion of data collected from multiple points of view. But the use of range sensors characterised by a limited angular resolution plays a major role in introducing uncertainty on the topological distribution of objects in the environment. This leads to occupancy maps that do not exhibit sharp transitions on objects' boundaries as would be desirable for the extraction of the occupancy information in autonomous robotic applications where mobile robots or manipulator arms need to detect safe navigation areas, or objects on which they need to perform an action. Taking into account all of these factors, this work aims at extending a segmentation algorithm that was previously developed in the context of 2D occupancy maps [1], to generalise it up to a scheme capable of subdividing a probabilistic grid of a 3D environment into regions characterised by uniform and deterministic occupancy states.

The characteristics of the probabilistic representation of occupancy grids and the application considered in autonomous robotics impose some constraints on the design of an appropriate segmentation algorithm. Unlike supervised schemes that assume a preliminary knowledge of the processed model, an unsupervised segmentation is necessary here as autonomous robots typically operate in a priori unknown environments.

Considering that the workspace is progressively scanned by a laser range finder or a sonar only along specific directions, the resulting map of explored space reveals the content of the workspace only along a series of rays corresponding to the beams emitted by the active range sensor. Because of this property, a region-based segmentation approach is favoured over a boundary-based one which tends to confuse the rays with the edges of the objects. Approaches based on classical methods such as split and merge [2], pyramid node linking [3], as well as quadtrees [4] were the first to provide unsupervised region-based segmentation. Recent unsupervised segmentation methods explore, either multi-resolution filtering, using Gabor filters [5] or the wavelets [6], or statistics with hidden Markov fields [7].

Given the specific characteristics of probabilistic occupancy maps used in robot guidance, an approach that combines contrast and texture properties to identify regions of uniform occupancy state reveals to be a more appropriate strategy for differentiating between segments exhibiting different occupancy properties. Ojala et al. [8, 9, 10] proposed such a segmentation technique based on "Local Binary Pattern" and "Contrast" (*LBP/C*) operators to subdivide images with sharp patterns in the context of texture classification. But unlike the images that they considered, the transitions between free and occupied spaces in probabilistic maps do not define clear boundaries and spread out according to the uncertainty level introduced by the sensor and data fusion. Refinement to the original *LBP/C* segmentation mechanism has been initially proposed in [1] to handle smooth transitions in complex images while achieving accurate contours definition corresponding to segments with deterministic occupancy states found in the context of robot guidance: free, unknown and occupied regions.

The quality of the results achieved in the 2D case and applied to mobile robots navigation motivated the extension of the 2D segmentation scheme to the 3D case in order to address the problem of manipulator robots collision-free path planning. The development of the extended scheme took into account the minimisation of the impact of the models' size increase on the execution time. This issue is handled by a clever redefinition of the texture unit which is adapted to the 3D representation without degrading the overall performance of the segmentation.

2 Texture Representation

Ojala and Pietikainen [8] defined the principle of texture analysis using the distribution of the "Local Binary Pattern" (*LBP*). This operator describes the local texture characterising the entourage of each pixel. Although it constitutes an important source of information about the local texture characteristics, the *LBP* alone cannot represent the contrast between the values of neighbour pixels. Thereafter, Ojala et al. extended their scheme by adding a "Contrast" operator, *C*, for each local texture unit. Thus, the texture in a given area of the model is to be characterised by a double distribution of *LBP* and *C* measures. The proposed segmentation algorithm exploits this multi-variable distribution to achieve classification of uniform regions.

2.1 Bidimensional Texture Representation

In the original scheme [8], the *LBP* and *C* values are calculated for every pixels of a given region in the original image, except for those located on the external boundaries. For each processed pixel, a block of size [3 x 3] pixels immediately surrounding the pixel of interest is considered (figure 1a). A discretisation process, which leads to a binary representation, is applied to this texture unit. In this process, the central pixel is used as a threshold, and all neighbouring pixels with a value higher or equal to the threshold are set to one, the others are set to zero (figure 1b). Next, the binary values are multiplied by binomial weights (figure 1c) and the results (figure 1d) are added, excluding the value of the central pixel, in order to obtain the *LBP* value of the texture unit.

The *LBP* parameter is combined with a simple measure of contrast that is equal to the difference between the average of the pixels' original values having a unity binary representation (i.e. 1 after application of the threshold) and the average of the pixels' original values having a null binary representation.

The resulting *LBP/C* distribution is encoded in a bidimensional histogram of size [256 x *b*]. The first dimension size comes from the fact that a central pixel has 8 neighbour cells for a texture unit of size [3x3] pixels. Each of them being represented by a binary variable (figure 1b), a total of 2^8 possible *LBP* values can be computed. The second dimension size, *b*, corresponds to the number of levels of discretisation of contrast, *C*. This decimal value belongs to the interval [-254; 255] for an image encoded on 8 bits. For this reason, a process of discretisation is necessary to determine the correspondence between a given value and its position in the *LBP/C* distribution histogram. The selection of a proper number of bins, *b*, for the *C* values remains a compromise between precision and performance.

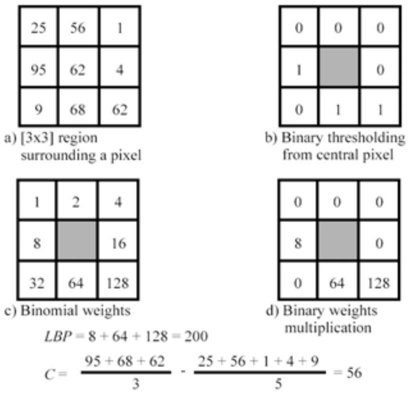

25 56 1
95 62 4
9 68 62
a) [3x3] region surrounding a pixel
0 0 0
1 0
0 1 1
b) Binary thresholding from central pixel
1 2 4
8 16
32 64 128
c) Binomial weights
0 0 0
8 0
0 64 128
d) Binary weights multiplication
LBP = 8 + 64 + 128 = 200
C = (95 + 68 + 62)/3 - (25 + 56 + 1 + 4 + 9)/5 = 56

Figure 1: *LBP* and *C* values estimation for a 2D texture unit.

2.2 Adapted Tridimensional Texture Representation

A segmentation scheme based on the analysis of a double distribution of the *LBP* and *C* operators directly depends on the definition of the texture unit. As introduced previously, the dimensions of the *LBP/C* histogram are defined by the range of the *LBP* possible values and by the number of levels of discretisation, *b*, selected to represent the contrast. The 2D texture unit of size [3x3] implies eight neighbour cells for each central pixel, which leads to 2^8 = 256 possible values for the *LBP*. Under this framework, if a supplementary dimension was introduced, the size of the distribution histogram would double for each additional neighbouring cell. As a result, fully considering a cubic texture unit of size [3x3x3] voxels involves 26 neighbours for each central voxel, as shown in figure 2.

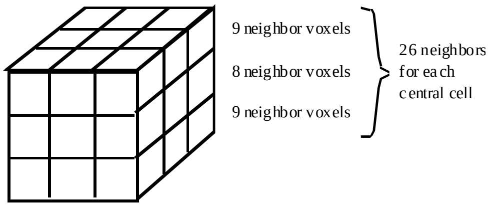


Figure 2: 3D texture unit of size [3x3x3] with its 26 neighbours.

The proposed redefinition of the texture unit keeps the histogram dimensionality low by taking into account only the neighbour voxels that share a face with the central one. Edge neighbours and vertex neighbours are eliminated. The resulting texture unit is symmetrical relative to the three Cartesian planes (x, y), (x, z) and (y, z) respectively, as illustrated in figure 3. The symmetry implies that the proposed segmentation algorithm does not discriminate between any of the three dimensions of the space during the analysis of the textures which characterise the environment occupancy. The adapted texture unit results in distribution histograms of size [2^6 x *b*]. In spite of the reduction of the histogram size by a factor four compared to the 2D case, the results of the 3D segmentation algorithm remain coherent and stable, as will be shown in the following sections.

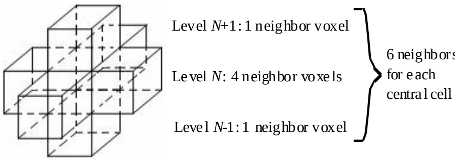

Level N+1: 1 neighbor voxel
Level N: 4 neighbor voxels
Level N-1: 1 neighbor voxel
6 neighbors for each central cell

Figure 3: 3D representation of the adapted texture unit.

In a similar way to the 2D case, for a given 3D region, the *LBP* and *C* values are computed for each voxel, except for those located on the external edges of the map. For each considered voxel, the six neighbours with which the latter shares a face are considered. Figure 4 presents a flattened representation of a tridimensional texture unit. The central voxel as well as its right, left, bottom and top neighbours are located at level *N*, while the neighbour voxels at the lower and upper levels are located respectively at levels *N-1* and *N+1*. The texture unit (figure 4a) first undergoes a discretisation process which leads to a binary representation. The occupancy probability of the central voxel is used as a threshold and all neighbours with a higher or equal value are set to one, the others are set to zero (figure 4b). The binary values obtained are multiplied by binomial weights (figure 4c) and the results (figure 4d) are added, excluding the central voxel value, in order to obtain the *LBP* value of the tridimensional texture unit of interest (here, *LBP* = 2+8+16+32 = 58).

The calculation of contrast follows the model introduced by Ojala and Pietikainen [8]. Thus, *C* corresponds to the difference between the average of the neighbouring voxels' intensity having a unity binary representation (after thresholding) and the average of the neighbouring voxels' intensity having a null binary representation. For examples, the contrast value of the texture unit presented in figure 4 is computed as follows: *C* = [(8+7+7+6)/4]-[(2+1)/2] = 5.5. For the 3D case, the resulting *LBP/C* distribution is represented in a two-dimensional histogram of size [64x*b*]. Based on [8] and on our previous 2D results presented in [1], a *b* value of 8 or 16 leads to similar results. However, according to the tests we carried out, an attempt to further decrease the number of discretisation levels, *b*, to 4 reduces considerably the sensitivity of the segmentation algorithm to the differences in contrast between occupied and free regions. Consequently, a value of 8 discretisation levels on the contrast was selected, resulting in a *LBP/C* histogram of size [64x8] in the 3D case.

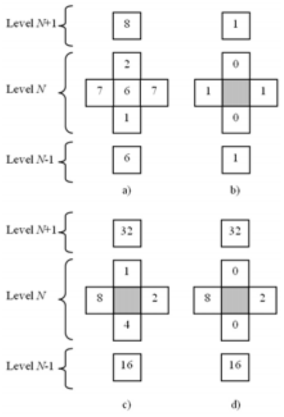


Figure 4: Calculation of the 3D texture unit's *LBP/C* characteristics.

2.3 Texture Comparison

The logarithmic likelihood ratio, the G-statistic [11], is used to compare histograms of *LBP/C* distributions, in both 2D and 3D cases. It provides robust means to classify uniform map segments. The value of *G* indicates the level of similarity between two frequency distributions. It estimates the likelihood that two compared regions have similar texture and contrast distributions. This measurement of similarity is calculated as follows:

$$
\begin{aligned}
G = {} & 2\left[\sum_{s,m}\sum_{i=1}^{n} f_i \times \ln\left(f_i\right)\right] - 2\left[\sum_{s,m}\sum_{i=1}^{n} f_i \times \ln\left(\sum_{i=1}^{n} f_i\right)\right] \\
& - 2\left[\sum_{i=1}^{n}\left(\sum_{s,m} f_i\right) \times \ln\left(\sum_{s,m} f_i\right)\right] \\
& + 2\left[\left(\sum_{s,m}\sum_{i=1}^{n} f_i\right) \times \ln\left(\sum_{s,m}\sum_{i=1}^{n} f_i\right)\right]
\end{aligned}
\qquad (1)
$$

where f_i corresponds to the number of texture units characterised by a pair of (*LBP*; *C*) values in bin *i*. *s* and *m* represent the two distributions to compare, and *n* equals [64x*b*] which is the number of bins in each of the analysed histograms.

3 3D Segmentation Algorithm

The structure of the tridimensional segmentation algorithm is inspired by a split and merge approach and is divided in three successive phases. The hierarchical division subdivides iteratively the probabilistic model in cubic blocks of uniform texture. Next, the segments creation phase merges adjacent regions with a similar occupancy state. Finally, the refinement phase improves the contours' localisation between neighbouring segments. The adjustments that were made to adapt the initial 2D segmentation technique to take into account the third dimension are detailed in this section.

3.1 Hierarchical Division

The objective of the 3D hierarchical division consists of subdividing the occupancy map in cubic blocks of variable sizes and relatively uniform texture. This phase first subdivides the probabilistic model in regions of size [S_{max}xS_{max}xS_{max}] voxels, S_{max} being equal to 64. A modified version of the uniformity test proposed by Ojala and Pietikainen [8] has been developed in order to determine if a “parent” subdivision contains heterogeneous textures and must be subdivided in eight sub-blocks of equal size or not. After having identified the eight subdivisions for each block, the *LBP/C* distribution histogram is computed in each subdivision. The eight resulting histograms are used to calculate the 28 logarithmic likelihood ratios (1) between each of the 28 possible pairs of subdivisions.

The largest and the smallest G-statistic values, denoted respectively by G_{max} and G_{min}, are identified. The parent block is considered non-uniform and thus subdivided if the ratio between G_{max} and G_{min} is higher than a threshold value, *X*. A proper value has been estimated empirically as *X*=1.2, as it provides a good discrimination between regions of different textures.

$$R = \frac{G_{\max}}{G_{\min}} > X \qquad (2)$$

If a block is recursively subdivided, its eight sub-regions undergo the same test. The iterative process is applied on the subdivided blocks until the subdivision size reaches [S_{min}xS_{min}xS_{min}], S_{min} being equal to 8 voxels. In a similar way to what was observed in the 2D case [1], a value of S_{min} equal to 8 rather than 16 provides more stable segmentation results as it leads to three recursive subdivision levels.

3.2 Segments Creation

The goal of the 3D segments creation phase consists of merging adjacent subdivisions that are characterised by a similar occupancy state until a convergence criterion is met. The segments creation starts by identifying for each of the subdivisions obtained at the end of the first phase, its neighbours from the top, the right and the lower level. This process guarantees that, at the end of the subdivisions scan, all possible pairs of adjacent regions will have been taken into consideration, and merged if they share similar texture characteristics.

The neighbouring blocks characterised by a mean occupancy probability, *MOP*, in the same interval are merged in a single region. This parameter is defined as the average cells’ occupancy probability in a given region, R$_i$, and is calculated as follows:

$$MOP_i = \frac{1}{M \cdot N \cdot O} \sum_{l=0}^{O-1} \sum_{k=0}^{N-1} \sum_{j=0}^{M-1} OP(r_{j,k,l}) \Big| r_{j,k,l} \in R_i \qquad (3)$$

where *M*, *N* and *O* are the dimensions of the region R_i, and $OP(r_{j,\ k,\ l})$ is the confidence level on the occupancy state of a voxel *r* with coordinates (*j*, *k*, *l*).

The motivations behind the choice of the *MOP* value as a criterion to evaluate the similarity between adjacent regions are related to the nature of the probabilistic occupancy map. After normalising the cells’ occupancy probability values over the range [0;1], three possible occupancy states are considered: *i*) if a region is totally unknown, it is characterised by an occupancy probability in the interval [0.498; 0.502] and is tagged as *unknown*; *ii*) if the occupancy probability belongs to the interval [0; 0.498[, this region of space is mostly free and is tagged as *free*; *iii*) finally, an occupancy probability in the interval]0.502; 1] shows a mostly occupied region of space that is tagged as *occupied*. These deterministic states, $S(R_i)$ = {*free*, *unknown*, *occupied*}, can indicate whether or not a region is safe for robot navigation.

3.3 Refinement

The completion of the first two phases leads to an approximate segmentation given that the boundaries between the segments do not correspond perfectly to the edges between adjacent regions that have a different occupancy state [1]. Consequently, the last phase is dedicated to perfect the contours localisation of the segments by reallocating the voxels on the boundaries between two or several adjacent regions. The refinement process can be divided in two parts: the first one deals with the boundaries between the *free – unknown* segments, while the second handles boundaries between the *occupied – free* and *free – occupied* segments.

The first refinement step is based on the fact that the range of occupancy probability values leading to an unknown segment classification is relatively narrow, being contained in the interval [0.498; 0.502]. Even if a segment overlaps between an unknown region and a free one by a limited number of voxels, it will still be considered as free by the segments creation phase. Therefore, the space whose occupancy is free always juts out into the unknown one, which is unsafe for robot navigation and collision avoidance. Thus, the first refinement step consists of expanding the unknown segments to the detriment of their free neighbours. At the implementation level, this process consists of scanning the tridimensional model along the six possible directions: right – left, left – right, bottom – top, top – bottom, below – above and above – below. Along the direction of each scan, the free cells, located on the boundary between a free and an unknown segments, with occupancy probability equal to 0.5 are reassigned to the unknown segment. This process continues until a cell with a different value is met. The six directions scanning procedure ensures the coverage of all possible boundary shapes.

The second refinement step consists, in a similar way to the first one, of scanning twice the probabilistic map in the six possible directions. During each of these scans, boundaries are considered respectively between the *free – occupied* and *occupied – free* segments. Along the direction of a scan, when a *free – occupied* (*occupied – free*) boundary is reached, the three following voxels in the adjacent occupied (free) segment are examined.

The choice to consider several voxels beyond the boundary between two segments of known occupancy state comes from the existence of unknown regions between the adjacent probing directions of an active range sensor. Any misclassification in these unknown areas can create discontinuities in the segments, and thus distort the object detection process. According to our investigation, the choice of three voxels provides the best safety/performance ratio.

Depending on the value of the three considered cells, four reclassification cases are identified: *i*) if the three cells have a normalised occupancy probability of 0.5, no reclassification is applied given that these pixels might fall into the unknown region between adjacent sensor measures; *ii*) if at least one of the three voxels has a normalised occupancy probability strictly higher (lower) than 0.5, and the other voxels have an occupancy probability equal to 0.5, no reclassification is applied; *iii*) if at least one of the voxels has an occupancy probability strictly lower (higher) than 0.5, and the other voxels have an occupancy probability equal to 0.5, then the three voxels are reassigned to the free (occupied) segment; and *iv*) if at least one of the three voxels has an occupancy probability strictly lower (higher) than 0.5, and at least another one is strictly higher than 0.5, then the voxels with value lower (higher) or equal to 0.5 are reassigned to the free (occupied) segment.

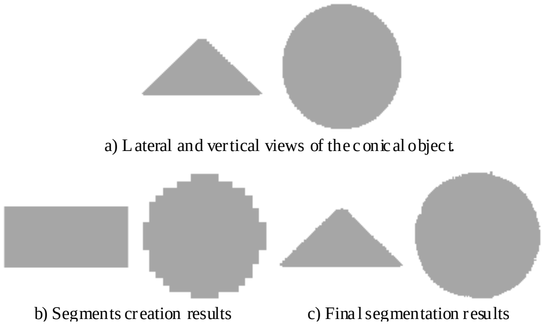
4 Experimental Results

In this section, segmentation results on three 3D probabilistic models each of size [320x320x128] are presented. These tridimensional probabilistic maps of an environment are created using a stack of planar images of the same size separated by a constant gap to create volume. These images are obtained using a laser range finder simulator for planar surface mapping that was developed in previous work. The width, the height and the depth of these environments correspond respectively to the dimensions of the images ([320x320]) and the number (128) of images that were piled up.

The occupied space shape as well as the number and position of the range sensor's points of view differ from a model to another. Gray areas in figures 5a, 6a and 7a represent the occupied regions by the objects in the models. These objects have respectively a conical, cubical and cylindrical shape.

The range sensor scans, used in the probabilistic models construction, are collected assuming a Gaussian error with standard deviation of 4 cm on the range measurements. The step angle between two adjacent sensor's rays which defines the angular resolution is fixed to 0.5 degree over all the measurements. The parameters used in this implementation of the segmentation technique are the same as the ones described in the preceding sections and are not modified when dealing with objects of different shape. In the hierarchical division phase, the size of the first level of subdivided blocks, S_{max} is [64x64x64] voxels, and three subdivision levels are conducted, leading to a S_{min} of [8x8x8] voxels.

Figures 5b, 6b and 7b present the segmented maps after the segments creation phase. The segments obtained at the end of the second step approximate well the shape of the regions present in the probabilistic map of the environment for all cases. Segmentation results obtained after the refinement phase are shown in figures 5c, 6c and 7c respectively for the three probabilistic maps. Important improvement on contours definition is achieved. The rendering of these 3D objects uses a slicing approach to allow easy navigation inside the map for accurate evaluation of the segmentation performance. The white regions observed inside the cubical and cylindrical objects in figures 6 and 7 respectively, correspond to the interior volume of the objects that was occluded from the sensor and therefore was never scanned. Beyond the external shape of the 3D objects, unknown regions are also correctly identified by the proposed segmentation scheme.

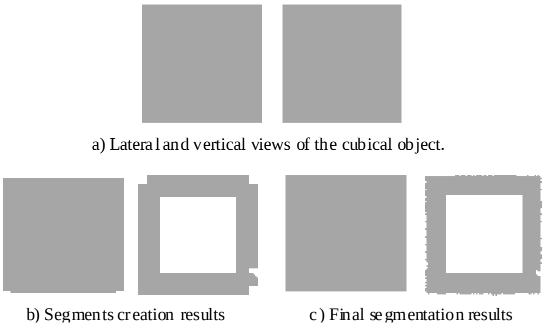


a) Lateral and vertical views of the conical object.

b) Segments creation results

c) Final segmentation results

Figure 5: Segmentation results of a 3D environment containing a conical object.



a) Lateral and vertical views of the cubical object.

b) Segments creation results

c) Final segmentation results

Figure 6: Segmentation results of a 3D environment containing a cubical object.

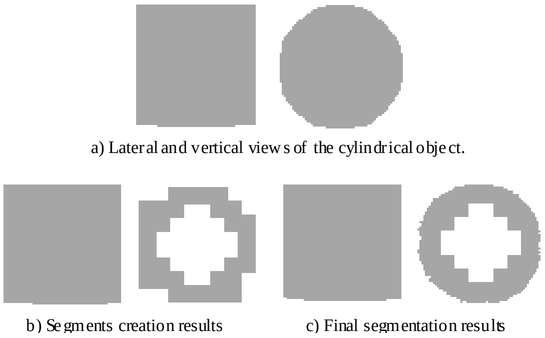

a) Lateral and vertical views of the cylindrical object.
b) Segments creation results
c) Final segmentation results

Figure 7: Segmentation results of a 3D environment containing a cylindrical object.

From a qualitative point of view, these experimental results demonstrate that the segmentation technique previously developed for bidimensional maps [1] can readily be extended to process tridimensional probabilistic maps of an environment. The proposed extension opens the door to a multitude of applications involving interaction between autonomous manipulator robots and their environment in exploration or manufacturing tasks.

From a quantitative point of view, the proposed tridimensional segmentation algorithm remains computationally efficient for a static model of the environment when global path planning is considered. This was made possible by the redefinition of the texture unit for 3D voxels that compresses the intermediate histogram representation by a factor of 1048576 [= (2^{26} x 8) / (2^{6} x 8)]. On average, the segmentation of a probabilistic model of size [320x320x128] voxels, as presented here, requires less than 11 minutes, which represents about 50 µsec per voxel. The segmentation time is minor compared to what is required to generate tridimensional probabilistic maps (data acquisition, registration, and range data fusion). Therefore it offers a realistic solution considering the accuracy of the segmented maps that can be achieved.

5 Conclusion

This paper proposes and evaluates an extended version of a texture-based segmentation scheme to operate on 3D probabilistic maps as obtained from limited resolution active range sensors. The redefinition of an optimal “Local Binary Pattern” and “Contrast” texture metrics successfully limits the increase in processing workload associated with the additional dimension introduced in the model which is known to drive numerous algorithms into a computational explosion.

The proposed texture unit definition is designed to preserve the robustness of the *LBP/C* texture mapping, thus leading to accurate segmentation results of tridimensional objects. Such segmented 3D maps can readily be used by manipulator arms classical path planning approaches to navigate without collision in cluttered environments, or to perform tasks on specific objects. It can also operate as a first processing stage in pattern recognition and object classification in 3D space.

6 References

[1] B. Abou Merhy, P. Payeur and E.M. Petriu, “Application of segmented 2D probabilistic occupancy maps for mobile robot path planning”, *Proceedings International Instrumentation and Measurement Technology Conference (IMTC2006)*, Sorrento, pp 2342-2347 (2006).

[2] X. Wu, “Adaptive split-and-merge segmentation based on piecewise least-square approximation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8), pp 808-815 (1993).

[3] F. Arman and J.A. Pearce, “Unsupervised classification of cell images using pyramid node linking”, *IEEE Transactions on Biomedical Engineering*, 37(6), pp 647-650 (1990).

[4] M. Spann and R. Wilson, “A quad-tree approach to image segmentation which combines statistical and spatial information”, *Pattern Recognition*, 18, pp 257-269 (1985).

[5] D.P. Mital, “Texture segmentation using Gabor filters”, *Proceedings International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies,* Brighton, UK, 1, pp 109-112 (2000).

[6] M. Unser, “Texture classification and segmentation using wavelet frames”, *IEEE Transactions on Image Processing*, 4, pp 1549-1560 (1995).

[7] H. Choi and R.G. Baraniuk, “Multiscale image segmentation using wavelet-domain hidden Markov models”, *IEEE Transactions on Image Processing*, 10, pp 1309-1321 (2001).

[8] T. Ojala and M. Pietikainen, “Unsupervised texture segmentation using feature distributions”, *Pattern Recognition*, 32, pp 477-486 (1998).

[9] T. Mäenpää, T. Ojala, M. Pietikäinen and M. Soriano, “Robust texture classification by subsets of local binary patterns”, *Proceedings International Conference on Pattern Recognition*, 3, pp 935–938 (2000).

[10] T. Ojala, M. Pietikäinen and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, pp 971-987 (2002).

[11] R.R. Sokal and F.J. Rohlf, *Biometry: The Principles and Practice of Statistics in Biological Research* (2nd edition), W.H. Freeman and Co., San Francisco (1981).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Span Indexed Greedy Algorithm for Task Planning in Multi Agent Systems

Sayyid Hasan Riyaz and Otman Basir
Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo, Ontario, Canada
shriyaz@uwaterloo.ca, obasir@uwaterloo.ca

Abstract

Multi Agent Systems (MAS) are often employed in applications having dynamic and uncertain characteristics. This dynamic and uncertain nature emanates from a variety of sources such as variations in the environment, changes in agent attributes and evolving job dynamics, etc. The plan development for these agents in any MAS therefore demands a mechanism that is responsive and adaptive to the application and system dynamics. Traditional planning techniques may prove infeasible quite often and require replanning whenever the system undergoes variations. Approximate dynamic solutions are sought for to address the issues of planning, scheduling and task allocations for agents in a timely manner. A popular choice is the use of the greedy algorithm due to its simplicity and effective solutions. This paper proposes a formal task planning framework for MAS. The framework provides models for the agents, the jobs and the operational environment. Furthermore, it describes the criterion function and the communication methods between the agents. A novel Span Indexed Greedy Algorithm (SIGA) is developed for scheduling and allocation of dynamically introduced jobs in the system using a market based mechanism. The efficiency and efficacy of the proposed algorithm is demonstrated through simulation, comparative studies and analysis.

Keywords: Multi Agent Systems, task planning, greedy algorithm, utility maximisation, autonomous systems.

1 Introduction

Multi Agent Systems (MASs) are particularly viable in applications that are hazardous, laborious, expensive and difficult or impossible for humans to perform. Reconnaissance missions, de-mining fields, transportation of hazardous materials, search and rescue operations, manufacturing systems, remote exploration of celestial bodies such as Mars exploration and even entertainment like robotic soccer are some of their applications. The operational environment of these applications is often inherently dynamic and uncertain, which emanates from a variety of sources such as variations in the capabilities and types of agents, job dynamics, etc. Jobs may be dynamically introduced in the MAS either internally or exogenously. The jobs may be elementary that may be handled by a single agent or abstract requiring collaborative efforts for their completion. Uncertainty in these jobs may also be compounded by virtue of precedence such as temporal or logical inter relationships within the jobs. Uncertainty in the agents may be due to their dynamic availability and varying performance efficiencies which may result from their commitments, resource consumptions and breakdowns.

The dynamics and uncertainties within a MAS domain require a dynamic, adaptive and responsive MAS planning framework. Planning in multi agent systems is an active area of research and is considered critical for the collaboration and coordination of agents in accomplishing system objectives. Planning and management techniques have been largely centralised in nature. These centralised planning techniques maintain an omniscient database of the jobs and the agents within the dynamic environment of the MAS. For instance the Central Mission Planner (CMP) of GRAMMPS [1] and the Central Task Planning and Scheduling (CTPS) [2] maintained constant communication among the agents and the central planner for planning in the MAS. Greedy algorithm is employed to provide effective solutions in a timely manner. These mechanisms although manage the online job planning and allocation effectively but pose maintainability and scalability issues. Further to the computing and communication overheads of such mechanisms, these are also susceptible to a single point of failure.

The distributed nature of any MAS facilitates decentralised approaches in planning and job allocation that provides flexibility in the system [3]. Agents perform the planning process in mutual coordination with other agents. A well known coordination mechanism is the Contract Net Protocol (CNP) [4], which provides planning and allocation in a simple yet effective manner. CNP is based on a Negotiation Based Market Mechanism (NBMM) where each agent evaluates a payoff (utility) for performing a certain job and an agent with the maximum payoff is assigned the job using a greedy approach. Jobs are sequentially considered and one agent acts as a coordinator for the job; receiving, evaluating and then assigning the job. A variant of CNP, M+, allows each agent to evaluate the payoffs for every job and communicates these payoffs to all other agents [5]. Upon receiving payoffs from all other agents, an agent considers itself a winner if it has the highest payoff for the job. A one step anticipation method allows agents to negotiate for two jobs at a time, however, the second job is allowed to be reassigned before its execution. MURDOCH, another variant of CNP, allows agents to subscribe/publish to receive information about the available jobs and to place bids based on the payoffs for executing the jobs [6]. Jobs are then assigned on a one job per agent basis and a job is lost if none of the agents bid for it. One agent acts as an auctioneer and receives bids from all other agents, thus reducing the communication overheads of M+. The Traderbots also uses an online job allocation similar to M+, but allows job reallocation whenever requested by an agent [7]. The abstract jobs are decomposed offline and job trees are developed before presenting the jobs in the system. The modified Traderbots allows multiple round auction mechanism where agents may decompose the jobs and put up a part of the job for allocation to other agents [8].

This paper proposes an intelligent planning mechanism where agents act as distributed solution providers and provide solutions with efficiency and efficacy. The proposed framework provides a structure for agents to compete for the jobs as they arrive in the system and to arrive at an execution plan in order to efficiently achieve the overall goals of the system. Models of agents, jobs and environment are developed, and a mechanism is presented where agents evaluate their strategies based on the available resources and the dynamic arrival of the jobs. A novel Span Indexed Greedy Algorithm (SIGA) is proposed for assigning dynamically introduced jobs to agents in a market based auction mechanism. SIGA is evaluated by extensive simulation to prove its effectiveness within the planning framework. A comparative study with the optimal and the Greedy Algorithm solutions is presented. The proposed framework and algorithm provide a structured planning approach for MAS and improves the solution quality of the existing approaches. This research is expected to advance and facilitate effective and efficient dynamic MA planning and execution.

2 Dynamic Planning Framework

The proposed planning framework models the three basic components of any MAS; the operating agents in the system, the jobs in the system and the system's operational environment. The communication methods and the optimality criterion are then presented. This MAS planning mechanism provides a distributed planning solution that is responsive, flexible, scalable and robust to the varying dynamics of the real world systems. The framework employs the NBMM which provides for an agent to set its individual goal in relation to the simple or complex MAS problem. An agent collaborates and coordinates with other agents to fulfill its part in achieving the system goals.

2.1 Agent Model

An agent is an autonomous entity operating in a dynamic environment, possessing some degrees of intelligence in achieving its objective [9]. A finite set of agents $A = \{A_1, A_2, ..., A_n\}$ is considered to be operating in the system. An agent A_i is modelled as a set of its resources, associated costs, initial state and final state. The strategy of an agent is formulated for transition from its initial to final state depending on the available jobs and resources. A utility value provides a numeric value for resource consumptions and cost accumulations. Formal definition of an Agent is provided as:

$$A_i = \{R_i(t), C_i, X_0, X_f, S_{A_i}, u_{A_i}\} \tag{1}$$

where

$R_i(t)$ represents the resources of the agent at time t

C_i is the associated cost of resources consumption

X_0 represents the current state of the agent

X_f is the final state of the agent

$S_{A_i} \to R_i(t) \times C_i \times X_0 \times X_f \times J(t)$ is the strategy of the agent A_i based on the set of available jobs at any particular instant. $J(t)$ is explained later in the Jobs model.

$u_{A_i} : S_{A_i} \to \Re$ is a set representing the mapping of the strategy set S_{A_i} to a set of real numbers. This gives the reward or the utility factor for an agent A_i to perform a job or a set of jobs $J(t)$.

$A_i \subset A$ represents the presence of an agent A_i in the operational environment.

The team formation of agents is modelled as follows:

$$\hat{A}_{i,j,...} :\to A_i \times A_j \times \cdots \tag{2}$$

where $\hat{A}_{i,j}$ represents a team called in by the agent A_i which also acts as the coordinator or manager of the team and therefore $\hat{A}_{i,j} \neq \hat{A}_{j,i}$. This $\hat{A}_{i,j,...}$ will give us a joint strategy $S_{\hat{A}_{i,j,...}}$ and payoffs $u_{\hat{A}_{i,j,...}}$ for the team member agents $A_i, A_j, \ldots$. The strategy is developed while considering the initial state of the agent and its final state when the strategy is executed. It is worth mentioning here that the team formation also represents the team structure and is important in getting the strategies and payoffs for the teams. In other words, $u_{\hat{A}_{i,j}}$ may not be equal to $u_{\hat{A}_{j,i}}$. If during the execution of a job there is a change in the structure of the team, the team may be reconstructed with the removal or addition of its members.

2.2 Environment Model

The environment defines the action space for agents. The dynamic nature of the environment is due to the varying availability of the number and types of agents. The available jobs and agents in the environment govern the individual goals of agents and the global goal of the system. The agents in the environment receive announcements of the jobs and participate in developing cooperative relationships with other agents for job planning and execution.

The environment model includes the structural constraints within which the agents operate. A number of agents may operate in one environment or an agent may have its own environment. When a job requires more than one agent to accomplish a common objective, two or more agents will work in a common environment. The environment for an agent or a group of agents is defined as

$$E = \{G, A, O_A, J(t)\} \tag{3}$$

where

G is the structural constraints of system operational environment.
A is the set of agents operating in the environment.
O_A is the set of group or individual goals.
$J(t)$ is a set of jobs or tasks currently present in the system.

Changes in any of the above mentioned entities of environment may trigger the re-planning process in the MAS.

2.3 Jobs Model

Jobs in any MAS may not be predetermined and may evolve over time. These jobs may be elementary or abstract. Elementary jobs can be handled by a single agent whereas collaborative efforts are required for an abstract job. The jobs may have temporal dependencies like the earliest start time or the latest completion time. Precedence constraints between jobs must be taken into consideration during the planning and execution stages, i.e., certain jobs need to be completed before the execution of other jobs etc. A set of jobs at time instant t is defined as:

$$J(t) = \{J_1, J_2, \ldots, J_i\} \tag{4}$$

Each job J_i is defined as a tuple

$$J_i = \langle \overline{J}_{job}, \overline{J}_{dep}, \overline{J}_t, \overline{J}_{req} \rangle \tag{5}$$

$\overline{J}_{job}$ can be an abstract job or a primitive job, $\overline{J}_{job} = \overline{J}_{ejob} \bigcup \overline{J}_{ajob}$ where $\overline{J}_{ejob}$ is the set of elementary jobs and $\overline{J}_{ajob}$ is the set of abstract jobs that require team formation and/or refinement before execution. $\overline{J}_{dep}$ represents the set of relations between the jobs like precedence, concurrence etc. This is particularly important during planning process when agents formulate their strategies and evaluate utilities for committing to these jobs. $\overline{J}_t$ represents the temporal dependencies of the set of jobs. Utility value for executing a job may also be a function of time. $\overline{J}_{req}$ provides a set of requirements and descriptors for the job $\overline{J}_{job}$ given as $\overline{J}_{req} = \{r_1, r_2, \ldots, r_m\}$. The elements r_i represent values or descriptors for each resource requirement of the job. The elements may have a numeric value, a descriptor or a limitation depending on the type of requirement the element is representing.

2.4 Communication and Coordination

Contract Net Protocol (CNP) [4] provides a communication and coordination protocol for agents using a market based mechanism. In CNP one of the agents take up the role of a manager and announces a job in the system. The job is broadcasted to all other available agents in the system. The individual agents then evaluate their utility (payoff) for the job and communicate this information to the managing agent. The managing agent upon receiving all the utilities from the agents establishes the agent with the highest utility and provides a contract for the execution of the job. This process is repeated sequentially for all the jobs available in the system. CNP provides a very simple and efficient mechanism for job allocation in a complex MAS scenario but it has its drawbacks such as an increased computational complexity with the increasing number of agents. MURDOCH [10] used a variant of CNP and allows the agents to use a publish/subscribe mechanism for receiving relevant information about jobs and posting the utility values. This results in reducing the communication and computational overheads and the same procedure is repeated sequentially for all the available jobs in the system.

The proposed framework also uses a protocol similar to MURDOCH but allows multiple jobs to be evaluated at the same time. Agents use a publish/subscribe method and coordinating agent(s) presents a set of jobs in the environment. The role of a coordinating agent is not fixed and any agent is allowed to take up this role and may publish new set of jobs in the environment. This helps in removing the single point of failure associated with a dedicated coordinator agent. It further allows multi processing of jobs and reduces overloading of the coordinator agents. The coordinator agent may also participate in the jobs acquisition process. If there are more than one coordinator agents then these agents coordinate their activities through communication.

2.5 Optimality Criterion

Considering the dynamic nature of jobs and agents in any MAS, a utility maximisation criterion function is considered where each agent evaluates its utility for the job while coordinator agent maximises overall utility of the system. Approximate solution using greedy approach has been used in M+, MURDOCH, Traderbots etc. Whenever a set of jobs becomes available, one or more agents take up the role of a coordinator and broadcasts the available jobs to the subscribing agents. Depending on the temporal properties of the jobs, the coordinating agent may decide on a cutoff time for the scheduling and planning process. Each agent evaluates the job and transmits its utility value to the coordinator(s). Coordinator agent evaluates these values as bids (strategies and corresponding utilities) in an auction mechanism through an approximation algorithm that is described in detail in the next section.

At any given instant t, M jobs are presented to N available agents that are to be satisfied over a time interval $T = [t_i, t_f]$. It is important to note that the planning time is different than the job time. A job may pass through certain states before being completed. The initial state is the state when the job is started and the final state describes the state when the job is finished. These states form a trajectory set for the jobs and may be defined in the trajectory space X with some topology as given below:

$$X : x(t), t_i \leq t \leq t_f \tag{6}$$

Each job, J_I, can be completed either by a single agent or a team of agents as represented by the set A_I. Depending on the set of participating agents A_I, jobs may follow different trajectories X_I towards their completion. Each member of the set A_I has an associated utility value also represented as a set U_I which provides a fitness value for the corresponding agents in case of accepting the job. The criterion function for each job can then be defined as:

$$C_I : A_I \times U_I \times X_I \rightarrow \Re^+, \qquad I = 1, 2, ..N. \tag{7}$$

The objective is to maximise the overall utility value of the system, i.e. to come up with a set of agents and their utility values that will optimise the criterion function given in Equation 7. This problem is formulated similar to a multi criterion optimisation problem [11]. The difficulty in this problem is that there does not exist a single or unique solution that optimise the perspectives of all the operating agents. A favourable solution to one agent (or a group of agents) may lead to an undesirable solution to the other. Due to this a Pareto optimal solution is to be obtained. A solution is said to be Pareto optimal if:

$$\begin{aligned} C_I(A_I^*, U_I^*, X_I) \geq C_I(A_I, U_I, X_I) \quad \forall i \leq 1, 2, ...N \\ \texttt{and} \;\; C_I(A_I^*, U_I^*, X_I) > C_I(A_I, U_I, X_I) \end{aligned} \tag{8}$$

for at least one $i \leq 1, 2, ...N$ i.e. a unilateral deviation from this solution by any agent or a team of agents will leave at least one agent (or a team of agents) in a situation worse than before.

For N agents and M jobs, the optimality criterion is given in Equation 9

$$\max_U \sum_M \sum_N C_{IK}(A_{IK}, U_{IK}, X_{IK}) \tag{9}$$

It is to be noted here that the value of N varies for each job, but the main objective here is to maximise the utility value for the system by completing as many jobs as possible in the most efficient manner.

3 Span Indexed Greedy Algorithm

The Greedy Algorithm as used in simultaneous job allocation of M+ requires each agent to find a job for which it has the maximum utility. Communication overhead is large as all agents transmit and receive utility values for all the available jobs from all the participating agents. The agents then evaluate to check if they have the highest utility for any job. The sequential job allocations of MURDOCH and Traderbots, which also employ the greedy approach, require the coordinator agent to sort the utility values for each job and to assign the job to the agent with the maximum utility. The solution quality in this case depends heavily on the sequence the jobs are presented for allocation.

The proposed framework provides a modified greedy algorithm which may be implemented on M+ and MURDOCH mechanisms for improved solutions. We consider simultaneous jobs but reorder these jobs based on the span of the maximum and the minimum utilities of the participating agents. We consider M jobs given in a set of jobs $J(t)$. For simplified notation J will be written in place of $J(t)$ form this point forward. Also only elementary jobs are considered at this stage. Let the set of agents interested in performing these jobs to be A, where $|A| = N$. The criterion function for a single agent single job scenario can be written as an integer programming problem as given in Equation 10:

$$\max_U \; \sum_N \sum_M C_{IK}(A_{IK}, U_{IK}, X_{IK}) y_{IK} \tag{10}$$

where $\quad y_{IK} \in \{0, 1\}$
and $\quad \sum_I y_{IK} = 1, \; \forall I; \;\; \sum_K y_{IK} = 1, \; \forall K$

The value C_{IK} effectively represents the utility values U_{IK} with the associated weights for the job J_I if an agent A_K executes it.

Formally, the Span Indexed Greedy Algorithm (SIGA) is written as follows:

Input to the algorithm is an ordered set of jobs J_I, $I = 1, \cdots, M$ and the criterion function values from each agent for each job C_{IK}, $I = 1, \cdots, M$, and $K = 1, \cdots, N$. If an agent K is not interested in job I then the criterion function value $C_{IK} = 0$.

The SIGA algorithm is described by the following steps:

1. Calculate for each job J_I the span of maximum and minimum criterion function values for all the agents.

   $\forall I : Span(I) = (maxC_{IK} - minC_{IK})$
   where $K = 1, \cdots, N$

2. Sort $Span$ such that $Span(1) \geq Span(2) \geq \cdots \geq Span(M)$

3. Index jobs based on $Span$ where $\hat{J}_1$ now corresponds to $Span(1)$, Repeat for all jobs J_I, $I = 1, \cdots, M$

4. Apply the greedy algorithm

```
for $I = 1 \cdots M$ :
find max $C_{IK}$ $\forall$ $K = 1 \cdots N$ :
set $C^*_{IK} := C_{IK}$
Assign $\hat{J}_I$ to $A_K$
Remove $A_K$ form all other $\hat{J}_I$
end
```

4 Simulation Results

A test case scenario of the transportation problem is considered where a set of jobs arrive in the system to be transported from an initial location to a final destination. Agents act as the transport service providers and coordinate with each other to execute the jobs on a single agent single job basis. Individual agents evaluate the jobs and present their utility values to the coordinator agent which in turn assigns the jobs to the agents using the proposed SIGA algorithm. The solution is compared with the Greedy Algorithm and the optimal solution of the problem. As the dimension of the optimal solution space increases exponentially we have restricted the maximum number of agents and jobs to be not greater than 10. A uniform utility value calculation function is used for all the agents depending on the initial location of the agent, pick up point of the consignment and its destination. The problem is posed as a maximisation problem.

Simulation studies are performed with a number of cases by varying number of agents and jobs and each case is simulated for 1000 runs. All the cases provide a common trend of the solution and to validate our strategy, results form one of the cases is presented. Figure 1 shows when four jobs are presented in an environment of MAS having 4 agents. The performance of the proposed SIGA is far better than the Greedy Algorithm (GA). Figure 2 shows the comparison of SIGA and GA with the optimal solution in a proportional representation. SIGA is nearer to the optimal solution in almost all of the cases compared to the Greedy Algorithm.

An overwhelming number of runs show that over 99% of the times SIGA provided much better solution than the GA. These preliminary results show a very promising improvement in the Greedy Algorithm without much computational complexity.

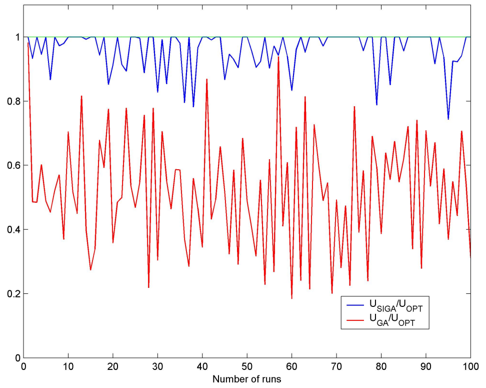


Figure 1: Ratio of payoffs for SIGA and GA with optimal solution.

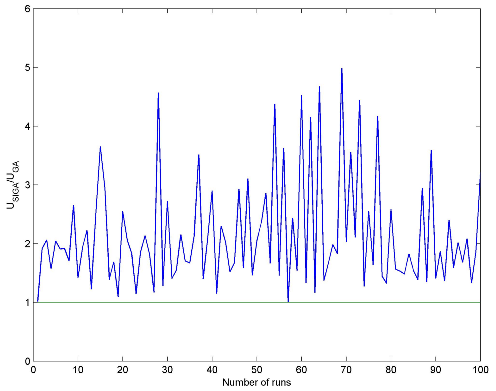


Figure 2: Comparison of payoffs for SIGA and GA.

Table 1 shows a comparison of the utility values (payoffs) obtained through the SIGA and GA with the optimal solutions for different agents/jobs configurations. The values are averaged per 1000 runs of each configuration. The results demonstrate that the SIGA solutions provide payoffs that are about double the values of the greedy algorithm solutions. It is also observed that the solution quality of SIGA is close to 95% of the optimal solution value with a less than one percent chance of an inferior solution then the GA solution.

Table 1: Comparison of SIGA with GA and optimal solution.

Agents/ Jobs	U_{SIGA}/U_{GA}	U_{SIGA}/U_{OPT}	SIGA out-performs GA (%)
2	1.68	1.00	100
3	1.99	0.97	99.0
4	2.13	0.96	99.1
5	2.24	0.95	99.7
6	2.35	0.94	99.7
7	2.36	0.94	100
8	2.34	0.94	100

5 Conclusion

The uncertainty and dynamic nature of the agent's capabilities as well as the job arrival in a Multi Agent System requires planning and management to be in an online decentralised fashion. To address these needs this work provides a framework for planning in Multi Agent Systems operating in dynamic uncertain environments. The framework is developed through the development of models for agents, jobs and operational environment that are flexible, adaptive and conducive to various operational modes. A novel span indexed greedy algorithm SIGA is developed for the case of a single agent single task planning. The effectiveness of SIGA is demonstrated through extensive simulation analysis and comparisons with the optimal and greedy algorithm solutions.

This research framework is instrumental in developing planning strategies for agents operating in uncertain environments. It allows various team formations and job structures in the operational environmental for effective planning. The proposed algorithm, SIGA, provides an improved solution to the greedy approach. In future we want to work on evolving this framework so it can be employed for the autonomous and intelligent operation of MAS in complex scenarios. We intend to expand the SIGA algorithm to include multi job multi agent modes of operations. Analytical analysis and mathematical study of the proposed SIGA algorithm will be conducted.

Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and Dr Otman Basir's Ontario Premier's Research Excellence Award (PREA) for providing support to this work.

6 References

[1] B. Brumitt and A. Stentz, "GRAMMPS: A generalized mission planner for multiple mobile robots", *Proceedings IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp 1564-1571 (1998).

[2] P. Caloud, W. Choi, J. Latombe, C. LePape and M. Yim, "Indoor automation with many mobile robots", *Proceedings IEEE International Workshop on Intelligent Robotics and Systems '90 (IROS '90)*, Ibaraki, Japan, pp 67-72 (1990).

[3] S. Paquet, N. Bernier and B. Chaib-draa, "Comparison of different coordination strategies for the RoboCupRescue simulation", *Proceedings 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)*, Ottawa, Canada, pp 987-996 (2004).

[4] R. Smith, "The contract net protocol: high-level communication and control in a distributed problem solver", *IEEE Transactions on Computers*, C-29(12), pp 1104-1113 (1980).

[5] S. Botelho and R. Alami, "M+: A scheme for multi robot cooperation through negotiated task allocation and achievement", *Proceedings IEEE International Conference on Robotics and Automation*, Detroit, USA, pp 1234-1239 (1999).

[6] B. Gerky and M. Matarić, "Sold!: Auction methods for multi robot control", *IEEE Transactions on Robotics and Automation*, 18(5), pp 758-768 (2002).

[7] M. Dias and A. Stentz, "A free market architecture for distributed control of a multi robot system", *Proceedings 6th International Conference on Intelligent Autonomous Systems (IAS-6)*, Venice, Italy, pp 115-122 (2000).

[8] R. Zlot and A. Stentz, "Complex task allocation for multiple robots", *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, pp 1515-1522 (2005).

[9] M. Wooldridge, *An Introduction to Multi-Agent Systems*, John Wiley & Sons, Chichester, England (2002).

[10] B. Gerky and M. Matarić, "A formal analysis and taxonomy of task allocation in multi robot systems", *The International Journal of Robotics Research*, 23(9), pp 939-954 (2004).

[11] O. Basir and H. Shen, "Goal-driven task assignment and sensor control in multi-sensor systems: a multicriteria approach", *Proceedings IEEE International Conference on Robotics and Automation*, Atlanta, USA, pp 559-566 (1993).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

An Agent Based Approach for Project Performance Monitoring and Control Assistance

Nora Houari and Behrouz H. Far
Department of Electrical and Computer Engineering,
University of Calgary, Calgary, Alberta, Canada
nhouari@ucalgary.ca, far@ucalgary.ca

Abstract

Distributed software development is pervasive in today’s enterprise environment; getting a clear understanding of he software development evolution, whether for performance monitoring, prediction or decisionmaking, is a non-trivial task. This paper presents an agent-based assistance system that supports access to distributed information sources and uses concept-relational model of the domain. This model is augmented with multi-strategy learning techniques that aggregate background knowledge and contextual information. There are three main objectives: (1) developing a framework for the software project performance characteristics; (2) a establishing a multiagent architecture endowed with multi-strategy learning techniques, that allows integrating historical learned models with contextual information of the current situation; (3) using a large data set to evaluate our system’s performance in a case study. The promise of our research is that integrating concept relational model of the domain characteristics and cooperative software agents that proactively monitor, and use learning techniques, significantly improves human decisionmaking. More broadly, this work makes important strides for software development organisations to improve monitoring and control over project success.

Keywords: multiagent software systems, machine learning, decision support systems.

1 Introduction

Rapid development in computing combined with globalisation has created software industry characterised by shorter product lifecycles, faster delivery, multiplicity of development technologies and distributed processes across multiple partners. This produces highly distributed systems with increased complexity and an urgent need for collaboration among organisations. To remain competitive, software enterprise needs to manage its knowledge assets and balance on the one hand the demand of information and knowledge that are distributed across people, objects, tools and environments; participants on the other hand need to develop a shared understanding of the current situation to make fact-based and timely decisions.

Software agents have been recognised as a new paradigm to build complex software systems by simplifying the complexity of distributed systems and by overcoming the limitations of existing computing approaches. Like a human agent, a software agent can carry out a task and has its own characteristics, such as autonomy, social ability, reactivity, proactiveness, cooperative, learnable and adaptable. Therefore an agent is considered a natural abstraction of the real world, since it can model the real world with its own goals and interacts with other agents to achieve mutual benefits. In the last two decades several systems and distributed collaborative multiagent architectures have been proposed, offering capabilities to support network structures and decentralised problem solving [1, 2, 15].

This paper describes the *intelligent data orchestration decision support environment* (IDODSE), a strand of our current ISS-OKM laboratory project [2] at the University of Calgary. Its main objective is to provide a decision support environment based on multiagent technology endowed with machine learning techniques to provide software project performance monitoring and control.

State-of-the-art software development systems show that many project characteristics influence project success. If certain project characteristics lead to certain project outcomes, it becomes important to be able to identify similar projects, since they would likely lead to similar project outcomes. Some project characteristics can be measured objectively, while others are subjective. The same is true for project success variables. This paper focuses on objective measures, primarily due to the availability of data in the case study presented below.

Nowadays when many software systems evolve over time, it becomes even more crucial to learn from past projects to increase the likelihood for success in future projects. To this end, software organisations must understand which project characteristics are most important for ensuring successful projects. Success indicators include timeliness of delivery, quality of software and more long-term properties such as maintenance and evolution. To predict project success reliably, it is first necessary to find a method to identify similar projects. Projects are similar if they show similar project characteristics and/or similar project outcomes, e.g. project delivery is determined by different factors: fulfilling a project plan often requires a combination of several variables. Due to the distributed and the multi-stage nature of software development, several projects every year [7] are delayed or cancelled because of short planning, longer development time, changed requirements and a variety of unexpected events during the project lifecycle, so monitoring each project manually is not feasible. The agent assistance mechanisms described in this paper are designed to effectively solve some of these problems by providing proactive monitoring and prediction throughout the project lifecycle. In addition to keeping track of current events (e.g. task milestones delayed), the agents can learn from experience to provide users with advance warning that a problem is likely to occur.

We make the following assumptions in this paper: (1) basic service agents are associated with the corresponding databases of their knowledge base and cooperate with other agents during project development; (2) agents share the same ontology of the organisation, that is, domain terminology with a clear meaning; (3) not least importantly, agents are collaborative - they all agree to use the same protocol for propagating information among themselves.

The IDODSE approach offers the following: (1) a domain concept model of the software engineering project lifecycle, providing taxonomy for the project context characteristics; this is considered a declarative representation of agent knowledge; (2) a multiagent system that helps improve communication between software participants; agents are endowed with learning algorithms used to integrate historical knowledge (captured from historical cases) with context related information on the current project evolution; (3) to assist the participants in distributed and ubiquitous software development environment.

2 Related Work

Prediction models for software engineering projects have used a wide variety of statistical methods and approaches, including regression analysis, multivariable statistics and machine learning techniques [16, 17, 18]. Some of these methods require substantial amounts of data for model building. To address this problem, researchers have looked at prediction methods with few data points, e.g. using different decision-making methods such as the analytic hierarchy process, applied to effort estimation in [19]. Case-based reasoning has also been used as a means for prediction [20]. The research in this paper uses a multi-strategy learning approach related to meta-learning. Various meta-learning strategies have been proposed [21, 22, 23]. *Voting* is one of the most used and the simplest method of combining predictions from multiple classifiers. It has two forms, the standard and the weighted form; for both, the collective prediction is determined by the majority of the votes. Weighted voting differs in the way weighting is given to the classifiers, which is determined by its performance on the validation set. The voting method integrates models in a linear fashion, which is not always adequate in some applications. In this work we adopted a customised *stacking* [8] strategy, which was successfully used in several applications such as logistics performance prediction [15] and information extraction [24].

3 The Proposed Methodology

The main goal of the Intelligent Software Systems (ISS) Laboratory of the University of Calgary in recent years has been to advance technologies that are required to overcome the current challenges of knowledge management in software engineering. The objective is to integrate concepts and approaches from software engineering techniques, ontology engineering and computational intelligence, to provide optimal and unified solutions towards this goal. It has produced innovative research results in this area [2, 5, 25].

This section highlights the ISS laboratory's most recent project, the *intelligent data orchestration decision support environment* (IDODSE), focused on improving and automating software engineering project assistance for performance monitoring and control. It presents an overview of its methodology, based on the concept relational knowledge base, then delineates the multi-strategy learning model.

Software systems are usually multi-team and multi stage development. They can belong to one organisation or be shared among partners. Due to the wide range of distributed data, information and knowledge, it is impossible to effectively locate, share or learn from these projects, raising an urgent need for data integration and orchestration. Addressing these issues requires learning from previous experience, monitoring and finding similarities between entities across heterogeneous sources. This is despite some promising work such as: (a) *machine learning*, where a set of features is created then one or more learning algorithms are applied to build the classifier; this standard approach has many problems because it requires expertise for the feature design process, which must be repeated if new data sources become available. In addition, the rationale behind the set of features is typically not captured in any formal way. (b) *case-based reasoning*, a method that addresses uses of a set of cases (a set of data structured in form of attributes) and uses a similarity measure to extract the most similar case; case similarity leaves a large gap, as users may not know exactly what they want, so adaptation to the current situation is not obvious.

All these efforts, while valuable, are insufficient as a foundation for intelligent software project performance assistance, as it lacks an integrated view of monitoring, learning from previous experience and finding similarities between entities across heterogeneous data sources. Automated creation of such contextual information and historical mapping is still in its infancy because equivalences and differences manifest themselves at all levels, from individual data values through metadata to the explanation surrounding the data collection as a whole. We take a different approach in this work: the key underlying idea is using multiple agents that identify the most informative sensed data then match built background knowledge models that share the situation. This can significantly reduce the human effort involved in sensing the multiple distributed data that needs to be combined with previously captured historical knowledge to use in decision support.

In this methodology, contextual information is captured based on recently received data and is automatically transformed to design the features for the meta learning system. We believe that if the contextual knowledge is made explicit, it can be incorporated into the learning algorithms to be integrated as the parameter values of the fitted predictive model. This would enable faster learning and provide higher system performance than can be obtained with either case-based reasoning systems or machine learning classifiers.

Our methodology integrates the following steps:

1. Development of the concept-relational model of the domain.
2. Building a multiagent and multi-strategy learning model, where agents collaborate among themselves to take into account the historical built knowledge models and continuously sense the environment. Agents in this meta learning process follow a customised version of the stacking strategy of [8].

These are explained in the following sub-sections.

3.1 Developing the Concept Relational Model

To build a unified tool for software engineering project assistance support, we first addressed the issue of knowledge representation and knowledge management, which requires the development of ontology. Since concepts are the most basic units of thoughts, it is not surprising that they became the important building blocks of our ontology. They appear widely in knowledge representation (e.g. in semantic networks, conceptual graphs, description logics), but also as instances in machine learning (e.g. conceptual clustering, concept learning). The IDODSE ontology is a lightweight conceptual domain model able to (a) provide a shared terminology for the software enterprise where agents can both understand and use; (b) define the meaning of each term in a precisely and unambiguously; (c) implement the semantics that will enable IDODSE to deduce automatically the answer to many “common sense” questions about the software engineering organisation; (d) define a symbol for depicting a term, or the concept of it constructed in a graphical context.

To define the set of concepts of the software engineering project, we reviewed the literature [9, 10, 3] and interviewed two specialists with experience in software development. The taxonomy retained for IDODSE contains five categories [12]:

C1: *Software product characteristics* (5 general classes of characteristics)

C2: *Software project development processes* (5 classes)

C3: *Software project resources* (7 groupings elements)

C4: *Software organisation factors*

C5: *Software environment constraints.*

The last two categories are still under development as most are subjective features. We used Protégé-2000 ontology editor for ontology creation. Protégé is an open source java tool which enables the user to separately define *classes* of information (schema) and store *instances* of these classes [11]. By utilising *slots* in Protégé we could define features of classes or relationships between classes, as well as properties of instances. Classes in Protégé correspond to concepts in our IDODSE ontology, while instances correspond to attribute values.

Once the ontology of the system was developed, we looked at the required agents and the learning method. We use the DIBRA (desire, intention, belief, rapport, adaptation) approach [25] for designing the individual agents. DIBRA integrates (a) agent internal structure (belief, desire, and intention - DBI); (b) inter-agent, inter-environment and agent’s autonomy structure in the rapport element of the agent; and (c) the agent’s adaptation of its beliefs and plans.

3.2 Building a Multi-Agent Multi-Strategy Learning Model

Our research focuses on exploring the full capacity of multiagent systems by endowing them with a hybrid learning approach of historical based models and contextual information, to improve monitoring and control of project performance. The methodology adopted for this combination is built on the meta learning concept. We chose a compelling meta-learning strategy called *stacking* [8], which deals with the task of learning a meta-level classifier. Inputs to the meta-model (the level-1 model) are the predictions of the base models (level-0 models). A level-1 instance has as many attributes as there are in level-0 learners, and the attribute values give the predictions of these learners on the corresponding level-0 instances. When the stacked learner is used for classification, an instance is first fed into level-0 models and each one guesses a class value. These guesses are then fed into the level-1 model, which combines them. The success of *stacking* arises from its ability to exploit the diversity in the predictions of base-level classifiers and thus predicting with higher accuracy at meta-level. Therefore, *stacking* accords effective non-linear combination of base classifiers.

We customise the standard *stacked generalisation* method described in [8] in this way:

- Different algorithms are supported for the background knowledge and contextual information: generative machine learning methods (e.g. Bayesian classifier [14]) are used in the base learner, whereas discriminative machine learning approaches (e.g. support vector machine, SVM) are applied in the meta learner. This is because contextual information learning must be performed based on a considerably smaller training set than historical-based learning. SVM proved an efficient learning method for learning problems based on a small training set [16,15].
- Only considerably pertinent attributes are taken into account as project attributes in the meta set, based on the performed mutual information analysis. Ignoring less pertinent attributes in fact improves learning performance in our experiments.

The multi-strategy learning approach works as follows:

1. Construct the historical learning models using one of the generative learning methods; in this paper we use the Bayesian classifiers.
2. Specify the contextual set, including the situated information to be considered. In this case study, we take a recent data set (data from the last three software metrics projects in a course offered by one of the authors) to represent the context set.
3. Create a meta set with the context set and the prediction of the historical models.
4. Learn a meta model on the meta set using the SVM algorithm.
5. Apply the meta model to make predictions on the prediction set.

4 System Architecture and Application Scenario

The application scenario considered in this paper is on methods for intelligent agent software engineering project performance monitoring and control, in particular software delivery time. The case study data comes from two sources:

1. The International Software Benchmarking Standards Group (ISBSG) [4], a publicly available data repository. The 2005 release of ISBSG contains information on 100 data fields for 3024 projects. The ISBSG repository contains data originating from organisations across the world with projects from different industries, using different methodologies, phases and techniques.
2. The 52 projects developed in one of the authors' course of software metrics during the period of 2002-2004 [6].

Figure 1 shows the system architecture. To carry out a project, different data come from different teams; monitoring the project delivery rate requires combination of means from the different participants. The agent support mechanisms described here are geared towards effectively providing proactive monitoring and control throughout the project development process. As well as keeping track of current events (e.g. a given milestone reached with delay), agents can learn from experience to provide users with advance warning that the problem is likely to occur. Considering the dynamic nature of software engineering project development, we propose a multi strategy learning approach so that agents can sense the current context and aggregate this contextual information with previously acquired background knowledge. Multi-agent architecture offers several advantages; agents solve a complex task by dividing it among them in a distributed manner. The IDODSE system structure is presented in figure 2. It has five layers: the user, knowledge management, concept management, service and data layers. Specialised agents reside at the various layers and perform well-defined functions. Different agents represent different sources of events in the project fulfilment process (planning, specification, design, implementation, testing, and implementation) as well as the adaptive assistance functions (history model learners, meta learners). Based on interaction protocols, agents collaborate. (This collaboration is beyond the scope of this paper that focuses on decision support aspects).

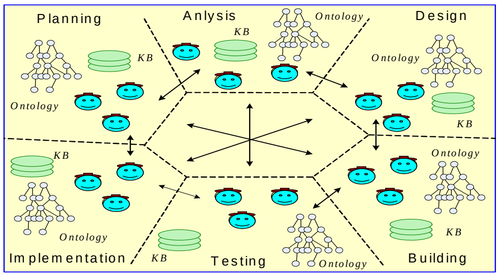


Figure 1: The multi-agent based system architecture.

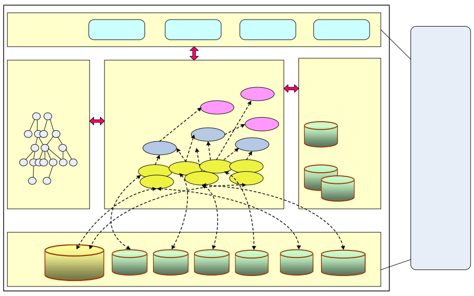
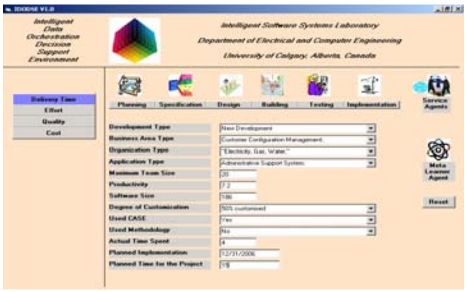


Figure 2: System structure**.**

Based on the methodology and system architecture described, we have developed a prototype of a client/server agent-based monitoring and control tool for project performance (here we cosider software delivery time, but it is for effort, defects, quality, etc).


IDODSE V1.0
Intelligent Data Orchestration Decision Support Environment
Intelligent Software Systems Laboratory
Department of Electrical and Computer Engineering
University of Calgary, Alberta, Canada
Delivery Time
Effort
Quality
Cost
Planning
Specification
Design
Building
Testing
Implementation
Service Agents
Meta Learner Agent
Reset

Development Type: New Development
Business Area Type: Customer Configuration Management
Organization Type: "Electricity, Gas, Water"
Application Type: Administrative Support System
Maximum Team Size: 20
Productivity: 7.2
Software Size: 106
Degree of Customization: 50% customised
Used CASE: Yes
Used Methodology: No
Actual Time Spent: 4
Planned Implementation: 12/31/2006
Planned Time for the Project: 18

Figure 3: Screenshot of SE project performance monitoring and control.

Figure 3 is a screenshot of the system, showing a list of the attributes (features) characterising the project currently to be predicted. Selecting *meta learner* displays the project prediction page. During project execution, events regarding completion of milestones are received via dedicated tracking agents (for each phase of the project). With this information and the available historical models, delay probabilities for individual project are updated. The user can select details of the project and inspect details of the project execution status, triggering actions such as notifying other users of delays, or re-assigning activities.

The platform chosen for implementation of the system is Java Agent Development framework (JADE) [13]. JADE is a software development framework, fully implemented in Java, for development of multi-agent systems and applications that comply with FIPA standard.

5 Experiments Setting Results and Discussion

We evaluate our approach using data obtained from two sources: (1) the 2005 ISBSG repository release containing information on 3024 projects; (2) the 52 projects developed in a software metrics course, 2002 to 2004. For the historical model we used projects collected over eight years (1995-2002), each a single project including maximum software size, CASE used, methodology used, productivity, development type, degree of customisation, actual time spent, planned implementation, planned time for the project.

Table 1: The data set of the experiments.

Data Set	No. of Projects
Historical models	
1995-1996	232
1997-1998	452
1999-2000	821
2001-2002	553
Context set Three consecutive years of student projects in software metrics course [6]	52
Evaluation set 2003-2004	125

Table 1 describes the data set used for our multi-strategy learning approach, the size of the training sets used for the historical knowledge models and the size of the context evaluation sets. Based on these data sets, we performed the following experiments on the evaluation set:

Exp_ 1 to 4: Predictions are made based on the Bayesian classifier history models from 1995 to 2002 respectively.

Exp_ 5: Predictions are made using one big single model containing the whole data set from 1995 to 2002 as the training set.

Exp_6: Predictions are made based only on the situated model derived from the context set (three years of student software metrics projects [6]); here history data was not taken into account.

Exp_7: Predictions are carried out based on the agent meta learning approach described in section 3.2, where combined background knowledge comes from the eight historical years and contextual information data comes from the set of student projects.

We evaluate the learning method of our approach using the two well known measures of information retrieval: *recall (R)* which defines the number of relevant projects retrieved as a fraction of all relevant projects, and *precision (P)* that specifies the number of relevant projects as a fraction of all projects retrieved by the system, as defined in (1) and (2):

$$P = \frac{P_D}{P_R} \qquad (1)$$

$$R = \frac{P_D}{P_G} \qquad (2)$$

where P_D is the number of correctly predicted delayed projects, P_G is the total of relevant projects both retrieved and not retrieved and P_R the total retrieved projects both relevant and not relevant.

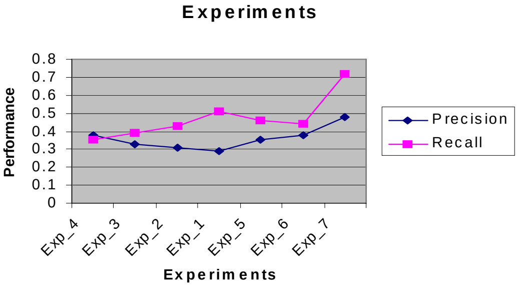


Figure 4: Experimental results.

The apparent result of our experiments is that the learning performance of the meta learning (Exp 7) based on historical knowledge and contextual information outperforms all other methods, meaning there are recurring patterns in software project development characteristics that lead to projects being delayed. Second, single model learning is less effective in a dynamic project development. Third, the experiment reveals that meta learner applied the best knowledge of the background models by sensing the context set. Looking at the meta set, we find that different history models often come up with different indications on the prediction of the project instance, owing to the dynamic and time-dependent nature of project development; this also indicates that meta learning is beneficial in this application domain. Other things to be considered are subjective features like team experience, which were not taken into account here since they were not available from the ISBSG repository. Finally, further extension of the work is to consider exploiting other generative algorithms for the historical knowledge and other discriminative algorithms for the meta learner.

6 Conclusion and Future Work

This paper described an agent-based system founded on combining historical knowledge models and contextual information to enhance agent monitoring and control of project performance. We are testing our approach for project delay, effort and quality management. Future extension will include more indicators such as cost. The learning approach used here employs Bayesian network as the base learn of the stacking strategy and the SVM for the meta learning; using different algorithms for the base level and meta learner will be the next step. The current version of the tool works on client/server environment for Windows. A web-based cross platform version for intranet will be developed in the near future; such a tool would help teams share the knowledge object over a corporate intranet/internet. Similarly, the current version of the tool only supports standard database information structure. To support some of the advanced content we will be looking at OWF, RDF and XML representation.

7 References

[1] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* (1st edition), MIT Press (2000).

[2] M. Afsharchi and B. Far, "Enterprise knowledge management using knowledge orchestration agency", in *Intelligent Decision Support Systems in Agent-mediated Environments* (2005).

[3] I. Watson, *Applying Knowledge Management Techniques for Building Corporate Memories* (2003).

[4] Software Development: The Benchmark ISBSG, "Home Page", http://www.isbsg.org

[5] N. Houari and B. Far, "A computational framework for autonomous and interactive multiagent systems development", *Recent Advances in Control Syst., Robotics and Automation*, pp 118-126 (2006).

[6] B. Far, "Software Metrics Course", http://www.enel.ucalgary.ca/People/far/Lectures/SENG421/i ndex.html

[7] H. Charette, "Why software fails", *IEEE Spectrum* (2005).

[8] D. Wolpert, "Stacked generalization", *Neural Networks*, 5, pp 241-259 (1992).

[9] R. Kosloski and K. Oliveir, "An experience factory to improve software development effort estimates", *PROFES 2005*, pp 560-573 (2005).

[10] S. Balsamo, D. Marco and A. Inverardi, "Model-based performance prediction in software development: a survey" *IEEE Transactions on Software Engineering*, 30(5), pp 295-310 (2004).

[11] Protégé, "Home Page", http://protege.stanford.edu/

[12] N. Houari, *IDODSE Ontology Management (Technical Report)*, University of Calgary Press, Calgary (2005).

[13] JADE, "Home Page", http://jade.tilab.com/.

[14] P. Langley and K. Thompson, "An analysis of Bayesian classifiers", *Proc. 10th National Conference on Artificial Intelligence*, AAAI Press, pp 399-406 (1992).

[15] Y. Guo, J. Müller and B. Bauer, "A multiagent approach for logistics performance prediction using historical and context information", *Proc. AAMAS 2004*, pp 1164-1171 (2004)

[16] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York (1995).

[17] I. Myrtveit and E. Stensrud, "A controlled experiment to assess the benefits of estimating with analogy and regression models", *IEEE Trans. on Software Engineering*, 25(4), pp 510-525 (1999).

[18] N. Ohlsson, M. Zhao and M. Helander, "Application of multivariate analysis for software fault prediction", *Software Quality Journal*, pp 51-66 (1998).

[19] C. Mair, G. Kadoda, M. Lefley, K. Phalp, C. Schofield, M.Shepperd and S. Webster, "An investigation of machine learning based prediction systems", *Journal of Systems and Software*, 53(1), pp 23-29 (2000).

[20] M. Shepperd and M. Cartwright, "Predicting with sparse data", *IEEE Transactions on Software Engineering*, 27(11), pp 987-998 (2001).

[21] M. Shepperd and G. Kadoda, "Comparing software prediction techniques using simulation", *IEEE Transactions on Software Engineering*, 27(11), pp 1014- 1022 (2001).

[22] T. Dietterich, "Machine-learning research: four current directions", *The AI Magazine*, 18(4), pp 136-160 (1998).

[23] A. Prodromidis and P. Chan, *Meta-Learning in Distributed Data Mining Systems: Issues and Approaches* (2000).

[24] G. Sigletos, G Paliouras, Spyropoulos and C. Hatzopoulos, "Combining information extraction systems using voting and stacked generalization", *Journal of Machine Learning Research* (2005).

[25] N. Houari and B. Far, "DIBRA: a new methodology to autonomous and interactive software agent development", *Computational Intelligence*, pp 176-181 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Modelling and Simulation of the Robotic IC Test Handler for Throughput Calculation

Sheau-Chyi Lee[1], Serge Demidenko[2], Kok-Hua Lee[3]
[1]School of Engineering, Monash University, Malaysia Campus, Malaysia
[2]Institute of Information Sciences & Technology, Massey University Wellington, New Zealand[3]
Freescale Semiconductor, Malaysia
sheauchyi@gmail.com, s.demidenko@massey.ac.nz, r59966@freescale.com

Abstract

Final testing is one of the major processes in semiconductor product manufacturing. The testing is performed to assure the quality of the manufactured parts (integrated circuits) before their shipping to customers. The process of testing is highly automated. Special sophisticated electronic systems called Automatic Test Equipment (ATE or just Testers) and robotic electro-mechanical machinery called IC Test Handlers are employed to provide high quality and throughput. Test handlers act as the transportation mechanisms to move the parts to be tested and to present them to the ATE terminals where the test measurements are performed, as well as to perform the tested parts sorting (so-called binning) depending on the test results. In addition, the handlers also provide the pre-defined temperature "soaking" (heating or chilling) of the circuits before their electrical testing. In a mass-production manufacturing environment a handler could support testing of hundreds and even thousands of parts per hour. Ideally, the performance of a handler and that of a tester have to be matched. This makes it important to have tools that could evaluate and predict handler performance (throughput) for various parameters of the test process. The paper discusses development of the predictive throughput calculator for the industrial robotic IC test handler.

Keywords: electronic testing, robotic IC handler, throughput prediction

1 Introduction

Testing plays an important role in controlling and ensuring the required quality and reliability of manufactured *integrated circuits* (ICs) before supplying them to final users. Several types of testing are performed at different stages of the IC fabrication process.

Figure 1 shows a simplified test flow for assembled integrated circuits. It can be seen that ICs have to undergo several types of extensive test procedures such as pre-burn-in test, monitored burn-in test and final test before their shipping to customers [1].

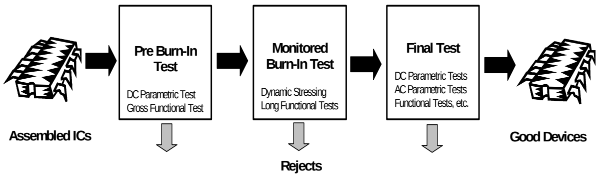


Figure 1: IC test flow.

Pre-burn-in testing provides a limited check to ensure that the ICs are not damaged during assembly, before sending them to a long and expensive monitored burn-in test. During the pre-burn-in test, some limited electrical parameters and functionality of ICs are checked.

The burn-in test is executed inside special burn-in chambers. It subjects chips (that are placed onto load boards) to elevated temperature often combined with functional test application. The burn-in test conditions should trigger IC failures due to defective components without damaging good components in the circuit [2]. A proper burn-in process should stabilise device characteristics, provoke early failures of weak devices and screen out those with excessive parametric drift.

After the ICs have passed burn-in, each of them goes through the electrical *final test* to ensure that the electrical specifications are met. Final testing normally includes a *DC parametric test* (to verify the steady-state electrical parameters of the IC), *AC parametric test* (to verify time-related parameters of the IC as well as to check that it operates over its full frequency range), *functional test* (to verify that the IC performs all its functions as the design intended) and/or *structural test* (defect oriented test to prove that the circuit was manufactured defect free). These tests are performed at room, hot and cold temperatures. Upon passing final testing, ICs are sorted (binned) into various grades according to their test results.

In today's semiconductor industry the process of testing is highly automated. Special sophisticated electronic systems called *automatic test equipment* (ATE or just *testers*) and robotic electro-mechanical machinery called *test handlers* are employed to provide high quality and high throughput.

The test handler acts as the transportation mechanism to move the parts to be tested and to present them to the ATE terminals where the test measurements are performed. It also performs the tested parts binning depending on the test results. In addition, the handlers can provide some pre-defined temperature “soaking” (heating or chilling) of the circuits before their electrical testing.

In a mass-production manufacturing environment a handler could support testing of hundreds or even thousands of parts per hour.

In order to achieve the best efficiency and the highest level of utilisation of the test equipment, the performance of the handler and that of the tester ideally have to be matched. The performance of a tester measured by *units per hour* (UPH) can be found rather easily, although it is not a trivial task for the test handler. This makes it important to develop tools that could evaluate and predict handler performance (throughput) for various parameters of the test process and for various ICs being tested.

The development of the predictive throughput calculator for the industrial robotic IC test handler is the main goal of the presented paper. Achieving the above goal will include detailed study of the equipment, modelling of its operation for various parameters of the manufacturing process as well as for different types of devices being tested, as well as the development of the simulation software that would enable prediction of the throughput provided by the handler for different operating conditions.

2 Test Handlers

On the production floor the IC test handler is physically attached to the automatic tester. The handler collects the device to be tested from the input module, delivers it to the test site and sends the start signal to the tester, which then performs the unit testing. When testing is completed, the handler extracts the *device-under-test* (DUT) from the test site and delivers it to the appropriate output location depending on the test results, thus binning the tested devices [3].

The types and configurations of test handlers are widely varied. However, there are two handler types most commonly used in the industry: *gravity feed* and *pick-and-place* handlers.

The input stage of the gravity feed handlers usually consists of input tracks into which the special plastic IC carriers (tubes) containing the units to be tested are inserted. The units slide down the input track into the test site where they undergo the required electrical testing. After testing is completed, the unit is then transported by the sort shuttle to the appropriate output track (tube) based on whether the unit is good or bad.

Pick-and-place handlers, which are also known as *special electromechanical handlers*, are also performing the same functions as the gravity feed handlers. However, their input and output stages are different from the equipment of the gravity feed type. Here handlers pick the units for testing from a special standard tray and present them to the test site for testing. After testing, the pick-and-place system takes the unit and puts it into the appropriate output tray.

Handlers can be assigned to one or more testers. The equipment in a typical test handler is a loading or input stage, a test site, a sort shuttle, an unloading or output stage, various sensors, and interfaces to the tester [3].

Among the main reasons for the use of test handlers in the industry are that they can prevent human handling errors such as parts mixing, electrostatic discharge damage and IC lead bending. By using test handlers, the DUT positioning at the test station is more accurate since precise CCD video capture with automatic position adjustment is used. The above leads to significant improvements of the test process in terms of both quality and yield. In addition the best part of using test handlers is that the throughput provided by them is high. Often test handlers are multi-site programmed, as up to 8 or even more sites can be programmed simultaneously, thus dramatically increasing throughput measured by UPH.

3 Handler Operation

In essence a robotic IC test handler can be considered as a system comprising three interlinked parts: the input module, the core module and the sort module. This system is relatively universal - it can be used with a wide variety of IC package types and IC carriers (sleeves, trays, coin-stack magazines, tape-and-reel media, etc.).

In order to operate and monitor the modern IC test handler, there is normally a touch screen display attached to it. During normal operation, the information about temperature, sort status and the general state of the handler is presented on the display. In addition, there is a “menu” key on the display enabling users to access the system setup, statistics of operation and temperature calibration, as well as handler diagnostic information. However, the security features of the handler design can also prohibit unauthorised access to selected menus. There is also a system of status light indicators attached to the handler to indicate failure, normal operations and idle modes [4].

In order to configure the handler for different sleeves or trays, the relevant operation data (stored in the tester memory) is called up for use. This information (such as the device and carry media dimensional coordinates) is loaded in and converted into the control codes by using special application software. The *X*, *Y* and *Z* axis stepper motors are controlled by a micro-stepper control board. Encoders are used to ensure a positional accuracy [4].

ICs are picked up by vacuum pick-and-place heads and passed from input module to core module for testing. There is an individual sensor on each of the pickup heads in order to monitor the operation activity.

In the core module ICs are first cycled through the thermal soak chamber in multi-cavity, device-specific storage boats. There are a number of these multi-cavity boats employed in the handler, thus providing a significant device thermal storage capacity. Using a temperature probe, the storage chamber temperature is set and maintained at the level specified by the test conditions. The range of soak temperatures provided by a high-spec handler could be from -60 ºC to 160 ºC or even wider. In addition the thermal soak chamber is normally equipped with moisture control circuitry for cold operation as well as overheat protection for hot temperature "soaking" [4].

4 Essential Parameters

There are five major parameters that have to be taken into account when developing a handler throughput predictive calculator: the motor speed, tray matrix type, test parallelism, test time and storage condition.

4.1 Motor Speed

As mentioned earlier, the robotic IC test handler consists of three different modules: the input module, core module and output/sort module. Different sets and different numbers of step motors are used in each of these modules. For example, the handler employed in the reported research had 7 motors in the *input module* (IM) and 12 in the *core module* (CM), while the number of motors in the *output module* (OM) was 6. The speed for all the motors can be programmed depending on the DUT handling requirements. This is also normally done with the aim of increasing the system performance.

4.2 Tray Matrix Type

The researched handler has been adapted for the standard JEDEC matrix type tray [5]. If the IC size is large, the number of units on the tray is lower. This means the trays in the input and output modules have to be changed more often. Changing the trays requires additional time and as a result the overall handler throughput is decreased.

In the reported research four different JEDEC trays with different IC capacity were employed (3 x 8, 4 x 9, 5 x 12 and 6 x 14). It is important to note that the sequence of picking up ICs from the tray (as well as loading them into the trays at the end of testing) is different for each specific tray type.

4.3 Test Parallelism

The test parallelism parameter specifies how many DUTs are tested simultaneously. Normally the number can be 1, 2, 4 or higher (up to 32 in some cases). In the reported research testing parallelism of 2 and 4 was considered.

The input module passes 4 ICs into the core module each time. With testing parallelism of 2 the number of simultaneously tested ICs is 2. There is an index time required before the other (untested) pair of ICs is inserted into the contactor for testing. After all 4 ICs are tested they are transferred to the output module.

In the case of the testing parallelism of 4, the electrical test is done for all 4 ICs at the same time. Obviously this would lead to a higher throughput.

4.4 Test Time

The test time is the period required by the tester to complete a test programme on the DUTs inserted into the core module. The test time can be varied from fractions of a second to tens of seconds or even more, depending on the complexity of the IC and the customer's requirements on the test fault coverage.

4.5 Storage Condition

In the reported project, the testing conditions can be classified as so-called *storage enabled* (SE) or *storage disabled* (SD).

The SE testing mode means that the DUTs are passed through hot or cold pre-test temperature "soaking". In this mode all the motors in the core module are engaged.

In the SD testing mode the ICs are just inserted into the contactor for running the electrical test without "soaking" in the temperature chamber. Thus, only certain sub-slaves in the core module of the handler are involved in the operation.

5 Operation Sequences Modelling

In order to evaluate and predict the handler performance, special software emulating the system operation has been developed. It is based on the sequence of steps performed by the handler's modules during the full cycle of testing of the IC set.

Figure 2 shows the sequence where the first 4 ICs are picked up from the input JEDEC Tray, placed on the *input staging table* (IST) and delivered to the core module for testing. After the testing is completed, the ICs are placed on the *output staging table* (OST) from where they are finally loaded into the output tray.

Timer1 is used to for the total time of the operations taking place in the input module. It includes the time taken for the motor to pick up 2 ICs from the input tray, place them onto the IST, return back to the input tray to pick up another 2 ICs and place them onto the IST, and finally return back again to the input tray. For this portion of the process the parameters involved are the speed of the motor and the type of the JEDEC tray.

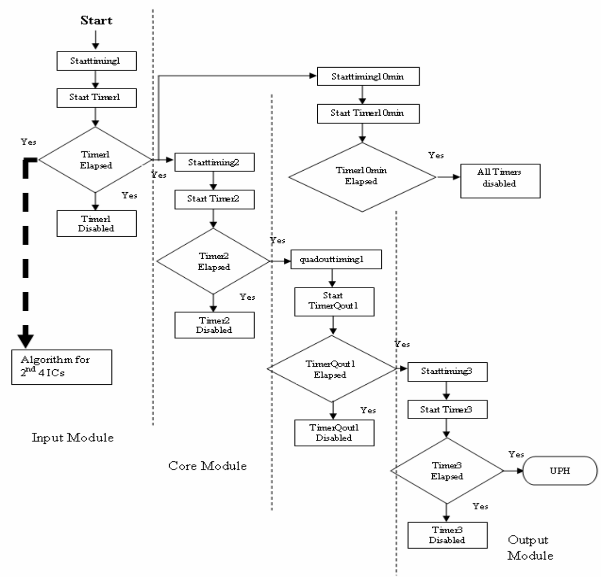


Figure 2: Handling the first ICs of the tray.

After Timer1 has elapsed, Timer2 is activated and the sequence for second group of 4 ICs is started. Timer2 is used for the total time taken in the core module. The parameters involved in the CM are: speed of the motor, test parallelism, test time and storage condition.

After Timer2 is disabled, TimerQout1 is activated. This timer is employed for the time taken by the output quad motor to pass the tested ICs from the core module to the output module. For this portion, only the speed of the motor is concerned.

When Timer3 is activated, it counts the time taken in the output module. There the first pair of tested ICs are picked from the OST and placed on the output tray. Then the handler returns back to the OST to pick up the second pair of tested ICs, places them on the output tray and then returns back to the output staging table. Finally, the UPH counter is incremented by 4.

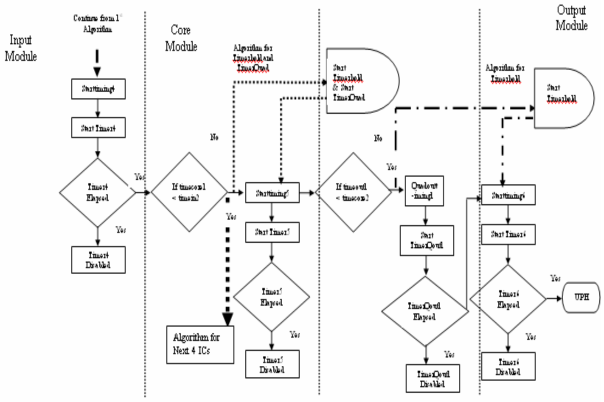


Figure 3: Handling ICs from the middle part of the tray.

Figure 3 shows the algorithm for handling ICs from the middle part of the tray at the input, core and output modules.

When Timer1 at the input module elapses, the comparison is performed between the current time taken in the input module and the previous time taken in the core module. If the previous CM time is greater than the current time taken in the IM, this would mean that there is still a process running at the CM while the process in the IM has been completed. TimerHold and TimerQuad are then activated. However, if the current time taken in the IM is greater than the previous time taken in the CM, this means that the core module is ready to accept ICs from the input module for testing. Thus as soon as the Timer1 in the IM elapses it activates Timer2 in the CM.

A similar algorithm is employed for the sequence of operations between the core module and the output module. After the timer in the CM elapses, it checks whether the previous time taken in the OM is greater than the current time taken in the CM. If the previous time taken in the output module is greater than the current time in the core module, then the OM is not yet ready to accept any tested ICs from the CM. As a result of the comparison TimerHold will be activated. After TimerHold elapses, TimerQout1 is activated. However if the current time in core module is greater than the previous time taken in the output module, TimerQout1 will be activated directly, thus leading to the tested ICs being passed to the output module. After the elapse of TimerQout1, the timer in the output module starts to count the time taken to transfer 4 ICs from the output staging table to the output tray. Finally the UPH counter is incremented by 4.

This algorithm is repeated a number of times (depending on the JEDEC tray type employed) for the IC sets taken from the middle part of the tray.

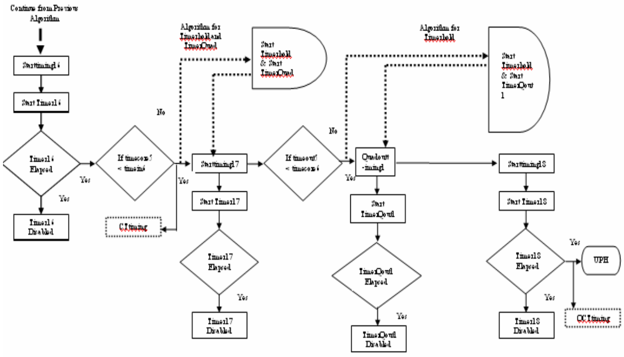


Figure 4: Handling the last ICs of the tray.

Figures 4 and 5 present an algorithm for the last 4 ICs that are processed by the handler before switching to a new tray. The algorithm is very close to the one discussed above for the ICs of the middle part of the tray. The only difference is related to the step sequence after the timer in the input module elapses and comparison between the previous time in core module and the current time in the input module is performed. In this case if the previous time value in the CM is less than the current time in the IM, then TimerCT is activated to count the time taken to change the input tray. A similar sequence would also take place in the output module: after the timer at the output module elapses it activates TimerOCT, which counts the time to change the output tray.

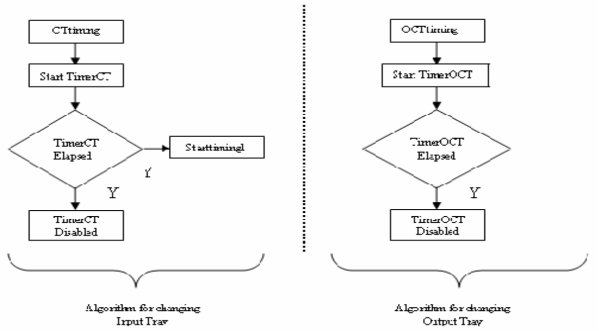


Figure 5: Changing input and output trays.

After completion of the last sequence for a tray, the first algorithm is restarted and loading of ICs from a new input tray begins.

6 Experimental Results

Validation of the developed throughput predictor has been done by calculating UPH values for different process parameters: the motor speed (the speed values have been found experimentally), tray matrix type (four JEDEC tray types were used), test parallelism (two values, 2 and 4, were employed), test time (test times were taken from the production test floor) and storage condition (SE and SD conditions were tried). The UPH values found were then compared with the results obtained by experimental recording of the test handler operation on the production flow and those from the literature [6].

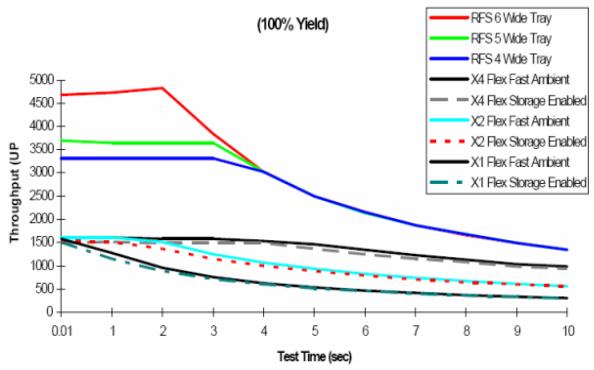


Figure 6: UPH vs. test time by experimental study.

Figure 6 presents the experimental results obtained with a real IC handler on the production floor [6], while figures 7 to 10 show the data obtained with the use of the developed predictive throughput calculator for different sizes of the output tray. It can be observed that the predicted and experimentally obtained throughput values match closely.

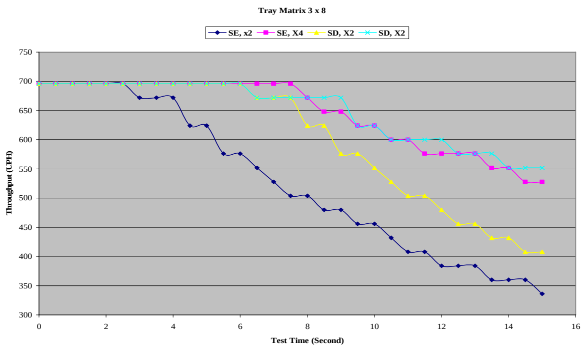


Figure 7: Predicted UPH (tray matrix 3 x 8).

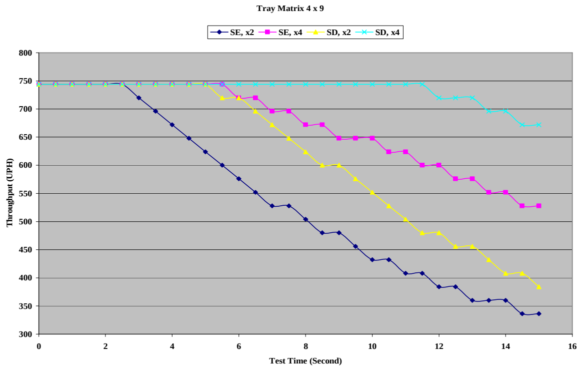


Figure 8: Predicted UPH (tray matrix 4 x 9).

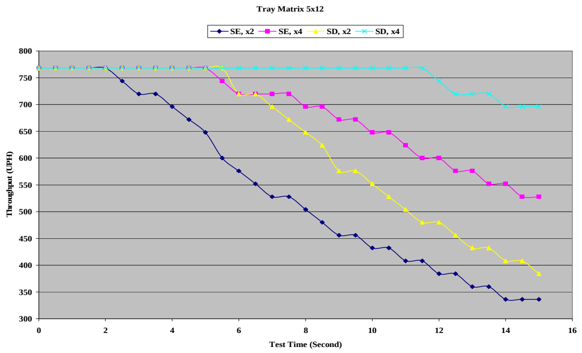


Figure 9: UPH calculation (tray matrix 5 x 12).

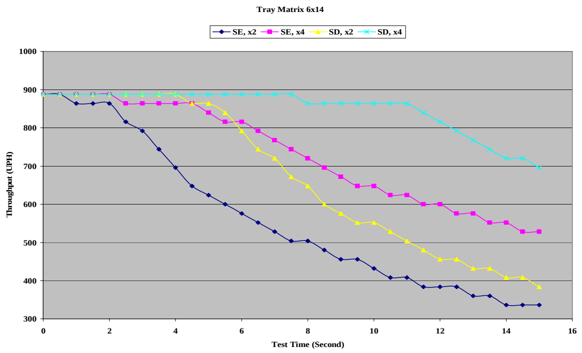


Figure 10: Predicted UPH (tray matrix 5 x 12).

By observing figures 7 to 10 it can be seen that UPH is constant for short test times. This is because in this case the time taken in the core module does not exceed the time taken in the input module and thus there is no delay time between the operations in the input module and those in the core module. This means that the ICs are directly passed to the core module after they are placed on the input staging table.

The time taken by the core module can be shorter than that of the output module. Hence the tested ICs may need to be held for a while until the output module completes sorting the ICs. Then the tested ICs are passed from the core to the output staging table. However, as this delay between the core module and the output module does not exceed the time taken at the input module, this does not affect the overall value of UPH.

It can also be seen that with the growth of the test time (while keeping the other test process parameters intact), UPH starts to decrease. This is due to the delay occurring in the input module and core module. When test time increases, this means that the time taken in the core module is also getting longer. Thus since the time of ICs in the core module is now greater than that in the input module, the ICs in the input module need to be held on the IST until the core module is cleared from the tested chips. Only after this are the ICs on the input staging table transferred to the core module. Since the time taken in the CM is much greater than that in the OM, the output module will be in a waiting mode while the core module is running its processes. Hence, delay between the input and core modules leads to a significant reduction in UPH.

7 Conclusions

The major goal of the project has been successfully achieved: the predictive throughput calculator for the robotic IC test handler has been developed and transferred to the industry (FSM) for application. The calculator is able to deal with 5 operation parameters, such as the motor speed, tray matrix type, test parallelism, test time and storage condition, while calculating the expected throughput.

The modelling, simulation and UPH calculation are done on 100% yield assumption, which means that there are no rejected ICs from the electrical testing in the test handler. Besides, it is assumed that there are no disturbances in the handler operation such as module motor jam, JEDEC tray refill on-the-fly in the input and/or output modules, presence of partially populated trays, etc.

The developed predictive throughput calculator is an executable programme. The user can specify and change entry parameters for the test process and get in real-time data showing whether this would increase or decrease the throughput. This enables the user to select the optimal set of operation parameters that would provide the highest productivity of the handler and apply them to the real equipment on the production floor.

The future development of the system could include expansion of the list of entry parameters such as the size of the step of the motor, which is related to its acceleration and deceleration, different types of JEDEC trays, extended test parallelism values, or additional storage conditions, for example *fast ambient* [6]. In addition, the enhanced system could also take into account different yield levels and different sorting (binning) procedures for the tested ICs.

8 Acknowledgements

The authors would like present their thanks to Freescale Semiconductor Malaysia (formerly Motorola SPS Malaysia) for the research opportunity, resources, equipment and data without which the study would not have been possible.

The reported work was done under the auspices of the ongoing collaboration (MOU) between Freescale Semiconductor Malaysia and Monash University Malaysia.

9 References

[1] M.P-L. Ooi, Z. Abu Kassim and S. Demidenko, “Shortening burn-in test: application of Weibull statistical analysis and HVST”, *Proceedings 22nd IEEE Instrumentation and Measurement Technology Conference (IMTC’05)*, Ottawa, 16-19 May 2005, pp 49-54 (2005).

[2] M.R. Marks, “Effects of burn-in on shear strength of 63Sn-37Pb solder joints on an Au/Ni/Cu substrate”, *Journal of Electronic Materials*, 31(4), pp 265-271 (2002).

[3] Semiconductor Test Equipment, “Test handler”, httrp://www.semiconfareast.com/testeqpt2.htm, visited on 31/10/2006.

[4] C.L. Woodard, *Operation and Service Training Outline*, Delta Design Asia Pte Ltd., Singapore (2005).

[5] Test, Assembly and Packaging Products, “JEDEC matrix trays”, http://www.ictray.com/Default.asp?Page=Jedec, visited on 31/10/2006.

[6] P. Cocharan, A. Dahl, G. Kovar and E. LePore, *Handler Throughput/Efficiency Study for Delta RFS and Delta Flex*, MCTG Manufacturing Engineering Development, Equipment Engineering Section, Austin, Texas (1996)

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Designing Recurrent Neural Network-based Controller for Gyro-Mirror Line-of-Sight Stabilisation System using an Artificial Immune Algorithm

K.C. Tan, C.K. Goh, J.H. Ang and E.J. Teoh
Department of Electrical and Computer Engineering
National University of Singapore, Singapore
eletankc@nus.edu.sg

Abstract

In order to handle the complex dynamics of the gyro-mirror as well as to optimise the various conflicting control objectives, a multi-objective artificial immune system framework which combines the global search ability of evolutionary algorithms and immune learning of artificial immune systems for the design of the gyroscope recurrent neural network controller is proposed in this paper. In addition, a new selection strategy based on the concepts of clonal selection principle is used to maintain the balance between exploration and exploitation of the objective space. Simulation results demonstrate the effectiveness of the proposed approach in handling plant uncertainties and the coupling effects of the cross-axis interactions.

Keywords: Gyro-mirror, recurrent neural network, multi-objective optimisation, artificial immune system

1 Introduction

Gyro-mirror line-of-sight (LOS) stabilisation systems form the basis of many applications for purposes of navigation, targeting and sighting [11], [14]. However, due to the multivariate, highly nonlinear and tightly coupled nature of the system, the control of gyro-mirror LOS stabilisation platforms is a non-trivial problem.

Current control schemes are often restricted to a particular problem or addresses only a subset of possible performance metrics, for it should be noted that a control design problem involves several design tradeoffs which may not be arbitrated explicitly. Intuitively, it may also be more practical to design a controller that satisfies the different requirements and constraints simultaneously, instead of satisfying a specific control scheme [5], [6], [10]. Hence, multi-objective (MO) evolutionary techniques provide a general platform for the simultaneous optimisation of competing control requirements. Using an evolutionary approach for MO, a variety of control schemes capable of addressing practical time and frequency domain performance criteria have been proposed in recent years [2], [11].

The use of artificial neural networks (ANNs) in control has also been receiving increasing attention due to their ability to learn and handle nonlinearities and uncertainties present in the control system [4], [15]. However, a major drawback of the higher complexity neural networks is that gradient-based backpropagation learning algorithms and real-time recurrent learning [16] for NNs are more likely to be trapped in local optima, resulting in sub-optimal solutions. Thus the concept of incorporating evolutionary elements into neural networks has led to the emergence of evolutionary artificial neural networks (EANNs) in recent years. Driven by an evolutionary search process that provides a global perspective to the adaptation process, EANNs have been demonstrated to be less sensitive to initial choices of weights and are also capable of dealing with the problem of premature convergence that is usually associated with conventional gradient-based approaches [7], [17].

With this in mind, artificial immune systems (AISs) [1], [8], [18] are a relatively new paradigm in the domain of computational intelligence that have been inspired by the human immune system’s ability to remember, recognise and eliminate non-self entities. In particular, features such as a predator-prey pattern of response and natural integration with other systems suggest that AIS might be very fruitful for the development of computational hybrid systems. Intuitively, these features are useful in the design of neural networks.

At present in the literature, there is a lack of studies on extending AIS for the adaptation of ANNs, not to mention the construction of an AIS-RNN controller to account for the unpredictable parameter variations, disturbances and nonlinear dynamics of the gyro-mirror LOS stabilisation system. One of the core objectives of this paper is to extend AIS to the multi-objective design of a RNN controller for the gyro-mirror LOS system. Apart from its capability of optimising multiple conflicting control objectives simultaneously, the multi-objective artificial immune system (MOAIS) implementation of the RNN controller provides the means of handling high-order system dynamics. Such an automated design methodology allows the MOAIS to identify controllers that are not only capable of maintaining and tracking the reference position but also remain robust in the presence of parameter uncertainties.

This paper is organised as follows. In section 2, an overview of the gyro-mirror stabilisation system is given. A description of the RNNs architecture is given in section 3. The proposed MOAIS for RNN controller design is discussed in section 4. Section 5 describes the simulation setup while section 6 validates and discusses the performance of the MOAIS control design. The conclusion of the paper is given in section 7.

2 Gyro-Mirror Line of Sight Stabilisation System

The schematic diagram of the gyro-mirror stabilisation system is shown in figure 1. It consists of three main components. The first component consists of the flywheel and its motor and controller. The second component consists of two gimbal structures that are responsible for providing two DOF to the flywheel; the inner and outer gimbal provide movement along the yaw and the pitch axis, respectively. The third component is a mirror that is geared to the gimbals through a 2:1 reduction drive mechanism. Specifically, the inner gimbal provides movement along the yaw axis while the outer gimbal provides movement along the pitch axis. As can be seen, the controllers in the gyroscope drive the torque motor in the gimbals for tracking applications. The flywheel is driven by an AC motor and the angular velocity is kept as constant as possible [11],[14].

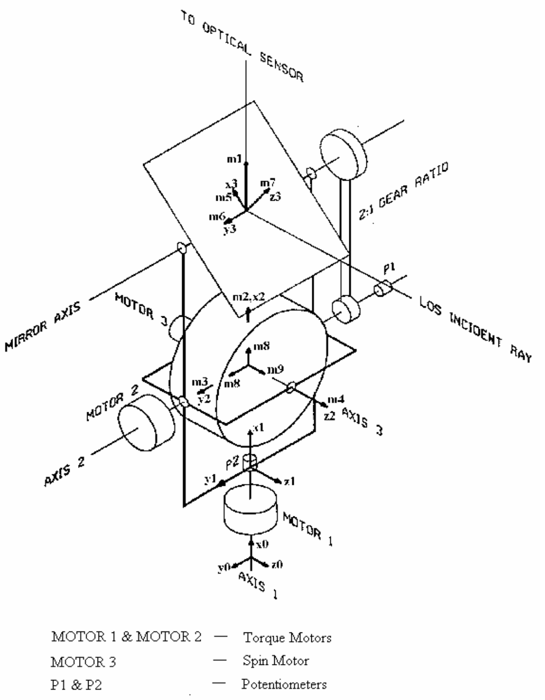


Figure 1: Schematic diagram of gyro-mirror stabilisation system.

3 Recurrent Neural Network Architecture

The RNN structure adopted in this paper is shown in figure 2. This permits state dependencies, allowing previous states to influence the activation of subsequent activations through this recurrency. The recurrent nature of the signal flow in an RNN, however, increases the complexity of the resulting network, as a result complicating the modelling and subsequent training of the free parameters (synaptic weights) of the network.

For RNNs, we have both *structural* as well as *temporal* state dependencies. Structural dependencies arise from the synaptic weights that are present between neurons (this is common in feedforward architectures), whereas temporal dependencies are a consequence of the fact that the present state of a neuron depends on the state of a neuron a few time-steps (say k time-steps) ago. This being said, for k-step state dependencies, the current state depends on the states of the neurons from 1, 2, …, k time-steps ago. For each time-step dependency, a matrix W^k defines the set of synaptic weights leading from the states of the neurons k time-steps ago to the present state. Thus for a system with k-step dependency, we can have up to k weight matrices W. Adopting a mathematical expression, the present state at time t is a function of its k previous states weighted by W^k (neglecting the bias terms b),

$$x(t) = f\left(W^1 x(t-1) + W^2 x(t-2) + \ldots + W^k x(t-k)\right) \cdot$$

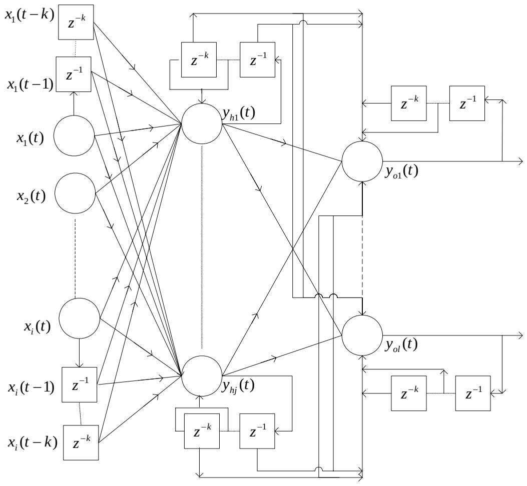


Figure 2: Recurrent neural network structure.

4 Design of MOAIS-RNN for Gyroscope Control System

4.1 Multi-Objective Gyro-Mirror LOS Stabilisation System Design Specifications

The gyroscopic principle has two important attributes, namely precession and directional stability. However, the physical phenomenon of precession is such that the application of a torque to one axis will result in the rotation in the direction of another axis. Consequently, the designed controller must be capable of handling such an effect since movement about one axis will trigger off a coupled movement in the other axis. In addition, the controller must also be able to cope with the multivariate and highly nonlinear nature of the gyro-mirror stabilisation system. Although the mathematical model of the system may be available, it should be noted that uncertainties such as noise, practical imperfection and additional dynamics are often omitted. It is for this reason that the design of a controller should be robust in the face of noise and parameter uncertainties, as under many practical conditions it is almost impossible to obtain a perfectly accurate model. Furthermore, the controller must be capable of maintaining and tracking the desired position despite the hard limited input (of 1 Nm) for the input torque to the plant. The desired design objectives, serving as a guideline for the development of our controller, are listed in table 1.

Table 1: Design objectives.

Design Specifications	**Objectives**
Torque saturation (Max(τ_1),Max(τ_2))	< 1 Nm
Rise-time (T_{rise})	< 2s
Settling time ($T_{settling}$)	< 2s
Steady-state error (E_{SS})	< 0.001
Overshoot (O_{shoot})	< 0.1

4.2 Multi-Objective Artificial Immune System Design for Recurrent Neural Network

The proposed immune system controller design methodology for the gyro-mirror stabilisation platform comprises four different modules: the RNN controller, the plant model, the MOAIS algorithm and the design specification module, as shown in figure 3. Since it is not appropriate to conduct online training of the RNN controller, the response of the control system is simulated based on the plant model. The error feedback of the two channels serves as the input to the RNN controller which generates the appropriate control signals. The candidate controllers identified by the immune system are evaluated based on the desired system response defined in the design specification module. Subsequently, the evaluation results of the controllers are used to guide the optimisation process of the design procedure. This approach utilises the given control specifications in addition to the required constraints in an automated manner, intelligently searching for the controller parameters that are best in a Pareto sense.

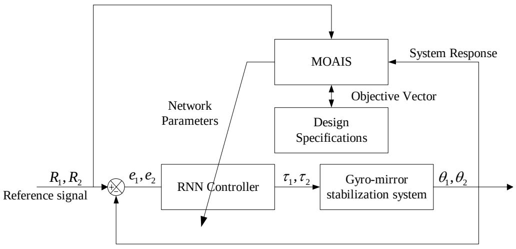


Figure 3: MOAIS automated RNN control design methodology for gyro-mirror stabilisation system.

4.3 Algorithmic Flow of MOAIS

The different AIS algorithms that have been proposed thus far in the literature can typically be distinguished by the methods of affinity measure and selection and the cloning mechanism employed, as well as the presence of memory cells. Despite these differences, the different algorithms follow a standard framework which can be extended to handle multi-objectivity, as shown in figure 4.

In contrast to single-objective optimisation, the solution to an MO optimisation problem exists in the form of alternate tradeoffs known as a Pareto optimal set. Each objective component of any non-dominated solution in the Pareto optimal set can only be improved by degrading at least one of its other objective components [9]. Placed in our context, the objective components of the individuals will correspond to the performance requirements specified in table 1. Therefore, the concept of Pareto optimality features prominently in the proposed algorithm. Furthermore, Pareto-based techniques have been employed successfully in MO problems by many researchers and it is also the most appropriate way to generate an entire Pareto front in a single simulation run [12].

The use of MOAIS has the capability of searching for a set of Pareto optimal solutions for MO optimisations. The MOAIS maintains a population of individuals, each of which is assigned a fitness value based on the different objectives to be satisfied. The individuals then go through a simulated evolution process which is described below.

1. Representation: Similar to the evolutionary optimisation techniques, the AIS processes a set of encoded parameters providing the flexibility to design an appropriate representation of the potential solutions. In order to facilitate the ease of manipulating the embedded information within the individuals, the genotype encodes the weights of the RNN which are represented by real numbers. Chromosome representation could be in any form as long as the architecture of RNN remains after application of variation operators.

2. Archiving: In our algorithm, elitism is implemented in the form of a fixed-size archive. This is done to prevent the loss of good particles due to the stochastic nature of the optimisation process. The size of this archive is usually adjusted based on the desired number of particles distributed along a trade-off curve (known as a Pareto front) in the objective space. The archive is subsequently updated at each cycle. For example, if the candidate solution is not dominated by any members in the archive, it will be added to the archive. In a similar manner, any archive members dominated by this solution will hence be removed from the archive. Thus, in order to maintain a set of uniformly distributed non-dominated particles in the archive, a dynamic niche sharing scheme [13] is employed. When a predetermined archive size is reached, a recurrent truncation process [3] based on a niche count is used to eliminate the most crowded archive member.

3. Clonal selection: Similar to the work presented in [1], the principle of clonal selection forms the core of the MOAIS. In this paper, we define memory cells as an archive of non-dominated solutions to preserve the best antigens found by the immune system and the antibodies as the candidate solutions in the AIS population. At every generation, the population of antibodies is evaluated. Fitness evaluation is based on affinity measurement. The affinity between the memory cells and an antigen is calculated based on their euclidean distance in the objective space. The smaller the euclidean distance between the memory and the antibody, the higher the affinity, i.e. the higher the fitness value. The affinity of an antibody is given in (1).

$$Affinity(k) = 1/\sum_{j=1}^{N}\sum_{i=1}^{M} ||memory_i(j) - antibody_i(k)|| \qquad (1)$$

where $M =$ archive size

$N =$ number of objectives

$k =$ index of individuals

In contrast to existing techniques, the clonal selection of memory cells is based on the degree of its representation in the evolving population.

4. Antibody cloning and variation: In this paper, binary tournament selection is implemented to select the appropriate memory cells for the cloning and hyper-mutation process. The number of individuals selected is based on *clone_rate*. After the selection of the memory cells, these non-dominated antibodies will be cloned such that the number of individuals is equal to the population size. In cloning, exact replication of the cell is done. Subsequently, uniform mutation will be applied to the clones. The mutation strength is adapted as

$$\sigma_w = 0.1 \cdot (uppbd_w - lowbd_w) \qquad (2)$$

where $lowbd_w$, $uppbd_w$ correspond to the minimum and maximum of the associated weights in the population. $uppbd_w$ and $lowbd_w$ may differ in every generation; however, the lower and upper limits of the RNN weights are -1 and 1, respectively.

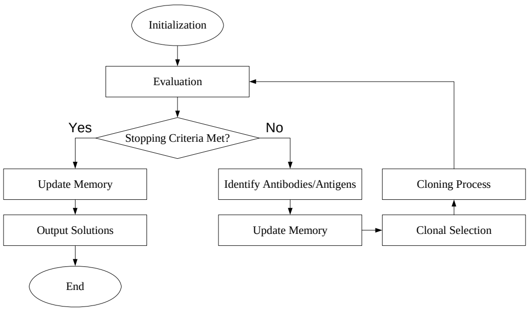


Figure 4: The framework for MO artificial immune systems.

5 Simulation Setup

Thirty independent runs are performed for each of the case studies in section 6 so as to obtain statistical information such as the consistency and robustness of the algorithms.

5.1 RNN Setup

In the proposed architecture, internal recurrency is limited to local signal flows - in other words, only *self-coupling/feedback* is used for both the hidden and output layer of neurons. Recurrency does not occur between nodes of the output and hidden layers, nor does it occur between nodes of the same layer.

The number of hidden layers and hidden neurons used in the simulation is restricted to one and three, respectively. For both internal and external recurrency, the time delay that is afforded is between 1 to 3 time-steps. There are two output neurons, each corresponding to $Torque\,1$ and $Torque\,2$, as shown in figure 1. The initial weights of the RNN are randomly assigned in the range [-1,….,1].

5.2 MOAIS Parameter

The stopping criteria used in this paper is when either all the objectives have been met or the maximum number of generations has been reached. The values of the various parameter settings in the algorithm are tabulated in table 2. During the adaptation of the RNN controller, the reference signals comprise 200 samples as given by (3) and (4).

$$R_1 = 0.2 \quad 0 < t < 20$$
$$R_2 = \begin{cases} 0.0 & 0 < t < 10 \\ 0.2 & 10 \le t < 20 \end{cases} \qquad (3)$$

For the simulation studies, the test signals are given as

$$R_1 = \begin{cases} 0.2 & 0 < t < 10 \\ 0.1 & 10 \le t < 20 \end{cases}$$
$$R_2 = \begin{cases} 0.0 & 0 < t < 5 \\ 0.2 & 5 \le t < 15 \\ 0.1 & 15 \le t < 20 \end{cases} \qquad (4)$$

Table 2: MOAIS parameter settings.

Parameter	**Values**
Population, memory size	20
Mutation method	Normal
Mutation strength	Adaptive
Mutation rate	0.1
Clone rate	0.2
Niche radius	Dynamic
Evaluations	6000

6 Results and Discussions

Studies are carried out to investigate controller robustness in the presence of plant uncertainties. Experiments are conducted at normally distributed noise variation of $\sigma^2 = 30\%$ with zero mean on plant transfer function parameters to simulate the conditions of plant uncertainties. The system response and control signals based on the original plant model are shown in figures 5 and 6, respectively. The system responses for 30% parameter variation are plotted in figure 7 while the control signals are plotted in figure 8.

From figure 5, it is clear that the design RNN controller is capable of satisfying the nominal system specifications defined previously. Furthermore, it can be observed from figure 7 that the RNN controller remains robust despite the variations made to the plant model. By comparing figures 6 and 8, the adaptation of the RNN controller to uncertain plant parameters is evident from the control signals produced by the controller in response to the plant parameter variation. In general, it can be seen that the main effect of model variation is the longer time required to seek the new reference signals as well as greater susceptibility to the coupling effects of the axis. However, it should be noted that the controller is still capable of sustaining good performance in terms of steady state error.

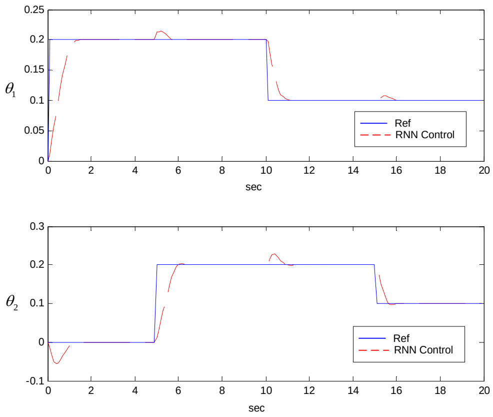


Figure 5: Gyro-mirror stabilisation system response based on original plant model.

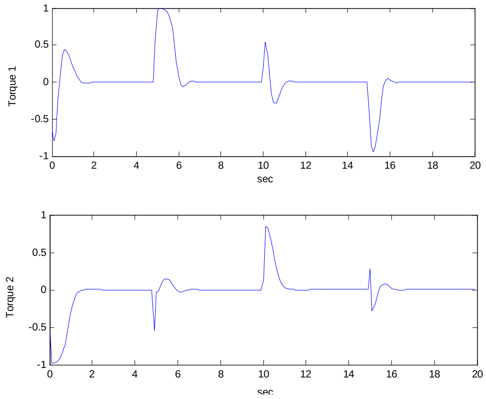


Figure 6: RNN controller output signal based on original plant model.

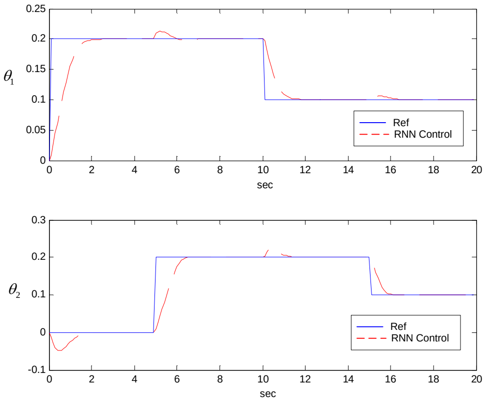


Figure 7: Gyro-mirror stabilisation system response under 30.0% parameter variation.

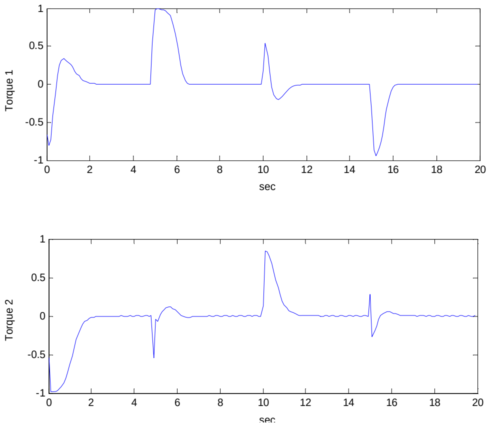


Figure 8: RNN controller output signal under 30.0% parameter variation.

7 Conclusions

In this article, a new algorithm based on an Artificial Immune System (AIS) paradigm to train a recurrent neural network controller applied to a gyro-mirror line-of-sight stabilisation system has been presented. We have demonstrated the viability of AIS in a dynamic setting, as epitomised by the gyro-mirror line-of-sight control problem, and, supported by a number of standard simulation tests, highlighted its performance and robustness. Experimental results obtained from simulations of the evolved controller demonstrate good performance in terms of satisfying the specified control specifications as well as robust behaviour in the presence of uncertainties.

8 References

[1] C.A. Coello Coello and N.C. Cortes, "Solving multiobjective optimization problems using an artificial immune system", *Genetic Programming and Evolvable Machines*, 6(2), pp 163-190 (2005).

[2] C.M. Fonseca and P. J. Fleming, "Multiobjective optimal controller design with genetic algorithms", *Proceedings on IEE Control*, Coventry, UK, pp 745-749 (1994).

[3] E.F. Khor, K.C. Tan, T.H. Lee and C.K. Goh, "A study on distribution preservation mechanism in evolutionary multi-objective optimization", *Artificial Intelligence Review*, 23(1), pp 31-56 (2005).

[4] O. Kuljaca, N. Swamy, F.L. Lewis and C. M. Kwan, "Design and implementation of industrial neural network controller using backstepping", *IEEE Transactions on Industrial Electronics*, 50(1), pp 193-201 (2003).

[5] Y. Li, K.C. Tan, K.C. Ng and D.J. Murray-Smith, "Performance based linear control system design by genetic evolution with simulated annealing", *Proceedings 34th IEEE Conference on Decision and Control*, New Orleans, pp 731-736 (1995).

[6] Y. Li, K.C. Tan and C. Marionneau, "Direct design of linear control systems from plant I/O data using parallel evolutionary algorithms", *Proceedings International Conference on Control '96*, Exeter, UK, pp 680-686 (1996).

[7] Y. Liu, X. Yao and T. Higuchi, "Evolutionary ensembles with negative correlation learning", *IEEE Transactions on Evolutionary Computation*, 4(4), pp 380-387 (2000).

[8] G.C. Luh, C.H. Chueh and W.W. Liu, "MOIA: multi-objective immune algorithm", *Engineering Optimization*, 35(2), pp 143-164 (2003).

[9] N. Srinivas and K. Deb, "Multiobjective optimization using non-dominated sorting in genetic algorithms," *Evolutionary Computation*, 2(3), pp 221-248 (1994).

[10] K.C. Tan and Y. Li, "Performance-based control system design automation via evolutionary optimization", *Engineering Applications of Artificial Intelligence*, 14, pp 473-486 (2001).

[11] K.C. Tan, T.H. Lee, E.F. Khor and D.C. Ang, "Design and real-time implementation of a multivariable gyro-mirror line-of-sight stabilization platform", *Fuzzy Sets and Systems*, 128(1), pp 81-93 (2002).

[12] K.C. Tan, T.H. Lee and E.F. Khor, "Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons", *Artificial Intelligence Review*, 17(4), pp 251-290 (2002).

[13] K.C. Tan, E.F. Khor, T.H. Lee and R. Sathikannan, "An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization", *Journal of Artificial Intelligence Research*, 18, pp 183-215 (2003).

[14] K.K. Tan, A. Mamum, T.H. Lee and C. J. Khoh, "PID-augmented adaptive control of a gyro mirror LOS system", *Proceedings 3rd World Congress on Intelligent Control and Automation*, Heifei, China, pp 3144-3149 (2000).

[15] L. Tian and C. Collins, "A dynamic recurrent neural network-based controller for a rigid-flexible manipulator system", *Mechatronics*, 14, pp 471-490 (2004).

[16] R.J. Williams and D. Zipser, "Experimental analysis of the real-time recurrent learning algorithm", *Connection Science*, 1(1), pp 87-111 (1989).

[17] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks", *IEEE Transactions on Neural Networks*, 8(3), pp 694-713 (1997).

[18] J. Yoo and P. Hajela, "Immune network simulations in multicriterion design", *Structural Optimization*, 18, pp 85-94 (1999).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Robust and Efficient Automatic Adjustment for Optical Axes in Laser Systems Using Binary Search Algorithm for Noisy Environments

Nobuharu Murata[1], Hirokazu Nosato[2], Tatsumi Furuya[1] and Masahiro Murakawa[2]
[1]Faculty of Science, Graduate School of Toho University,
Funabashi, Chiba, Japan
[2]National Institute of Advanced Industrial Science and Technology (AIST),
Tsukuba, Ibaraki, Japan
nobu-murata@aist.go.jp, h.nosato@aist.go.jp, furuya@is.sci.toho-u.ac.jp, m.murakawa@aist.go.jp

Abstract

For laser systems, the adjustment of the optical axes is crucial. However, it is difficult for conventional methods to adjust the optical axes because they require high-precision positioning with μm resolutions and because laser systems have many adjustment points that have an interdependent relationship. We have proposed an automatic adjustment method using genetic algorithms to overcome this problem. However, there are still two problems that need to be solved: (1) long adjustment time, and (2) adjustment precision due to observation noise. In order to solve these problems, we propose a robust and efficient automatic adjustment method for the optical axes of laser systems using binary search algorithm. Adjustment experiments for optical axes with 4-DOF demonstrate that the adjustment time could be reduced to half of conventional adjustment time with GA. Adjustment precision was enhanced by 60%.

Keywords: automatic adjustment, optical axes, genetic algorithm, binary search, noisy environments

1 Introduction

Laser systems are currently essential in various industrial fields. For laser systems, the adjustment for the optical axes is crucial, because the performance of the laser system deteriorates when the optical axes deviate from their specification settings, due to disturbances such as vibrations. However, it is very difficult to adjust the optical axes, because adjustment requires high-precision positioning settings with μm resolutions and because it is necessary to adjust for multi-degrees-of-freedom (DOF) that have an interdependent relationship. Thus, adjustment costs are a major problem due to the huge amount of time required for a skilled engineer to adjust the optical axes. In order to overcome this problem, we have proposed automatic adjustment methods for optical axes using Genetic Algorithms (GA) [1, 2, 3]. For example, our method has been successfully applied to the automatic adjustment of a femto-second laser that has 12-DOF [2]. However, there were two problems with the proposed methods that needed to be solved. First, it has been necessary to reduce the adjustment time to within 10 minutes. Because a laser system should ideally be re-adjusted every time it is used, for practical considerations, adjustment time must be as fast as possible. Secondly, because the adjustment of the optical axes is usually undertaken in very noisy environments, the precision of adjustment can vary widely.

In order to overcome these problems, we propose a novel adjustment method. This method has two characteristics:

1. The method adopts a Binary Search Algorithm (BSA) [4]. The BSA gradually changes from the exploration phase to the exploitation phase. The exploration phase searches a region that has not previously been searched using a binary search tree. The exploitation phase searches a region around good points.
2. The fitness value adopts weighted average of sampled fitness values using a search history.

There are two advantages with the proposed method: 1) adjustment time can be reduced. The method does not search in regions that have previously been searched. In addition, it is not necessary to re-evaluate the fitness function to mitigate the influence of noise; 2) it provides robust automatic adjustment. Instance of premature convergence or falling into local solutions do not occur because the adjustment is less influenced by noise. Accordingly, we can realise robust and efficient automatic adjustment systems for the optical axes within laser system by the proposed method. Conducted experiments involving 4-DOF adjustment with the proposed method demonstrate that 1) adjustment time can be reduced to half of conventional adjustment time; and 2) precision can be enhanced by 60%.

This paper is organised as follows: In section 2, we explain the adjustment system for the optical axes of laser systems and the conventional method of automatic adjustment. Section 3 describes our proposed method, and section 4 outlines the automatic

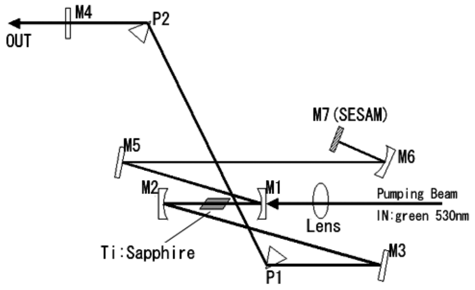


Figure 1: A femto-second laser system.

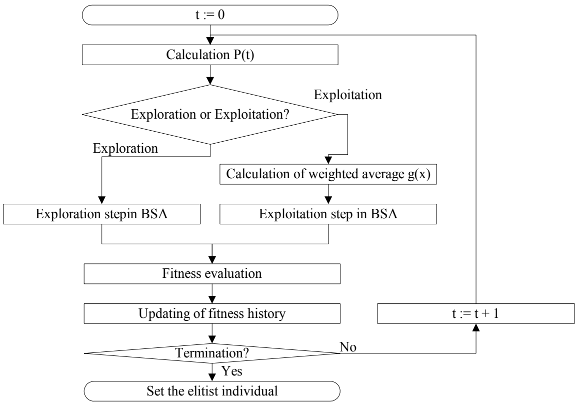


Figure 2: Flowchart for the proposed method.

adjustment system used in the experiments. In section 5, we present the experimental results obtained for the proposed method. Finally, a summary of this study and future investigations are provided in section 6.

2 Adjustment Systems for Optical Axes and Automatic Adjustment Methods

2.1 Adjustment Systems for Optical Axes

For laser systems, the adjustment of the optical axes is crucial because the performance of laser systems deteriorates when the optical axes deviate from their specification settings. Let us explain adjustment systems taking the one in figure 1 as an example. This system is a femto-second laser system that consists of seven mirrors and two prisms. A characteristic of femto-second lasers is that the high-peak power is inversely proportional to the short duration of the laser pulses, so they can generate high power levels, over one megawatt, during femto-second (10^{-15} sec) pulses. This system has 12-DOF to be adjusted with μm resolutions. The optical axes are adjusted by moving stepping motors, so that the output power from the laser system is maximised.

In such an adjustment system, there are two sources of observational noise influencing output evaluation. The first is the noise in the detectors that evaluates the output from the laser system. The second source is the precision of the positioning motors. While the positioning motors are moved according to constant displacements, actual axial displacements are not constant. Therefore, the optical axes can deviate from the desired state, even if motors are moved according to the displacement settings in seeking to adjust the target state. Thus, the adjustment of optical axes must be carried out by considering these sources of noise. In the system shown in figure 1, for example, manual adjustment takes about a week.

2.2 Automatic Adjustment Methods

We have already demonstrated how it is difficult for a hill climbing method to automatically adjust optical axes [2]. There are two reasons for this. The first reason is that the adjustment becomes trapped by local solutions, because the adjustment points of the laser system have an interdependent relationship. The second reason is that adjustment must be executed in noisy environments. In order to overcome these problems, we have proposed an adjustment method using GA [1, 2, 3]. In the proposed method, a chromosome is a set of genes, which represent motor displacements, and fitness is the output from the laser system. We have demonstrated the effectiveness of automatic adjustment using GA for the laser system illustrated in figure 1. However, there were two problems that needed to be solved. (1)Adjustment took a long time, because the method also performed exploration during the final phase. The time for motor movements, which accounts for nearly all of the adjustment time, increases in proportion to the degree of motor displacement. (2)Robust adjustment is difficult, because search is influenced by the two sources of noise explained in subsection 2.1. Consequently, instances of premature convergence occurred or adjustment became trapped to local solutions, so adjustment precision varied widely.

3 Proposed Method

We propose a robust and efficient automatic adjustment for the optical axes of laser system using a binary search algorithm (BSA) [4] for noisy environments. The flowchart of the proposed method is shown in figure 2. This method utilises weighted averaged fitness in the BSA as explained in figure 2. We refer to the proposed method as BSW. The algorithm is explained in more detail below.

3.1 Binary Search Algorithm

The strategy of BSA is to use a binary search tree [4] to divide the search space into empty regions,

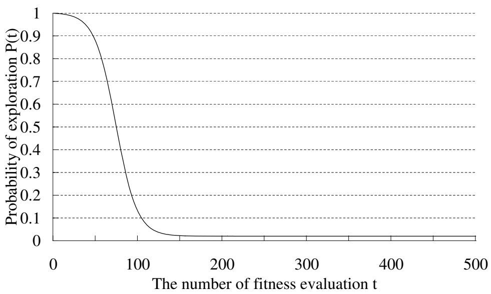


Figure 3: Probability of exploration for C=0.02, K=0.1, σ=0.05 and M=500.

allowing for the largest empty regions to be approximated. The search tree is constructed by generating a point x_t at random within the chosen hypercube, then by dividing the hypercube along the dimension that yields the most `cube-like' subspaces. The basic algorithm for constructing the binary search tree works by repeatedly choosing an exploration or exploitation step:

1. **Exploration:** Next point x_{t+1} is generated within the largest empty region.
2. **Exploitation:** Next point x_{t+1} is generated within the largest empty region that is within a small distance from a `good point'.

The coordinates of the point x_t and the evaluated value f_t at x_t are stored in a search history $F(t)$ represented in equation (1).

$$F(T)=\{(\mathrm{x}_1,f_1),(\mathrm{x}_2,f_2),\ldots,(\mathrm{x}_\mathrm{i},f_i),\ldots,(\mathrm{x}_\mathrm{T},f_T)\} \qquad (1)$$

The decision of whether to perform exploration or exploitation is made based on a probability distribution $P(t)$ that varies with the number of fitness evaluations. $P(t)$ calculated using equation (2) is illustrated graphically in figure 3.

$$P(t)=(C-1)\frac{\tanh(\frac{\mathrm{t/N-K}}{\text{ó}})-\tanh(\frac{-\mathrm{K}}{\text{ó}})}{\tanh(\frac{1-\mathrm{K}}{\text{ó}})-\tanh(\frac{-\mathrm{K}}{\text{ó}})}+1 \qquad (2)$$

C is the minimum probability of performing the exploration step, ó is the rate at which the probability of exploration decays, K is the mid point of the decay and N is the maximum number of trials that are to be performed.

The automatic adjustment method using BSA can efficiently adjust optical axes, because the search phase in BSA is gradually shifts from exploration to exploitation according to $P(t)$ based on the number t of fitness evaluation, as shown in figure 3.

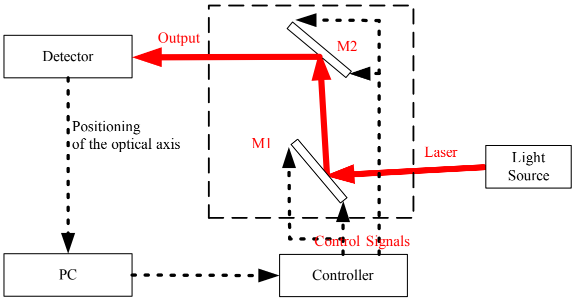


Figure 4: The structure of the experimental system.

3.2 Weighted Averaged Fitness

The conventional method of coping with noisy fitness functions is to evaluate the fitness values several times for each individual and to adopt the average of the sampled values [5, 6]. However, adjustment for laser systems by the conventional method is not practical, because adjustment time increases. For example, if moving the motors and detecting to obtain a fitness value are performed N times, then the adjustment time increases N-fold.

In order to solve this problem, a weighted average value, which is calculated from the search history, is used for the fitness value. For the laser system, we assume the detected value f_t at x_t increases or decreases in proportion to a distance d_t from a certain point x to the point x_t and that noise varies according to a normal distribution. The maximum likelihood estimation of $f(x)$ can be obtained as follows:

$$g(x)=\frac{f(x)+\sum_{t=2}^{T}\frac{1}{1+k\times d_t^2}f_t}{1+\sum_{t=2}^{T}\frac{1}{1+k\times d_t^2}}, \qquad (3)$$

$$d_t=\|x-x_t\|, \qquad (4)$$

where $f(t)$ is evaluated value with the detector, $f_1=f(x)$, $x_1=x$ and k is the proportional value and d_t is the distance from sampled points. The `good point' in exploitation step of the BSA is decided using this weighted averaged fitness $g(x)$.

There are two advantages of this method. The first is that this method can prevent premature convergence during the exploitation phase due to observational noise. The second advantage is that the number of evaluation is just one time for each individual. Thus, this method is capable of adjusting the optical axes robustly and efficiently in noisy environments.

4 An Automatic Adjustment System for Optical Axes

In this paper, we demonstrate the effectiveness of the proposed method using the most basic adjustment system [3]. The structure of this adjustment system is illustrated in figure 4. The adjustment system consists of two mirrors with two stepping motors, an evaluation detector to detect the positioning of the optical axis, a motor controller to control the stepping motor for the mirror and PC to execute the calculations. The mirrors in the system can be adjusted according to 2-DOF to adjust the positioning of the optical axis.

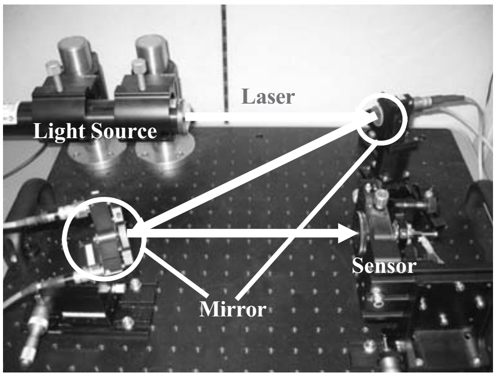


Figure 5: The automatic adjustment system for optical axes with 4-DOF.

A photograph of the experimental system is shown in figure 5. The details of each component in the system listed here.

1. Motor controller

   The motor controller is the stepping motor driver and the mirror holder system with a resolution of 0.075µm/step. With this controller, the time to move the motors in evaluating each individual is at maximum about 4 seconds.

2. Evaluation detector

   The detector consists of two-dimensional position sensitive detectors (S2044) and a signal processing circuit (C9069) produced by HAMAMATSU Photonics KK. This detector, with a resolution of 1.39µm, can detect X-Y coordinates of the optical axes on the detector front.

3. PC

   The flow, which is explained in figure 2, is run on the PC.

4. Light source

   The light source is a He-Ne gas laser (1125P) produced by JDS Unipase. The beam diameter is 2mm.

The optical axes are adjusted automatically based on the flow shown in figure 2. A chromosome in the BSW represents displacement for the four stepping motors, which moves the two mirrors. The displacement of each stepping motor is represented by 8bits. The evaluation value $f(x)$ uses a positioning error, which is the Euclidean distance between the positioning of the optical axes and the target position, calculated from the detected X-Y coordinates.

5 Experiments

5.1 The Details of Experiments

Three experiments were conducted to examine the effectiveness of the proposed method. These were an automatic adjustment experiment using a GA with the Minimal Generation Gap (MGG) model [7], automatic adjustment experiment using BSA and the automatic adjustment experiment using BSW.

The adjustment goal in each case was set to the laser system to its ideal state (i.e., the error in terms of the target positioning of the optical axes is 0). The initial conditions were random state where the positioning was altered within a range of ±5mm. Adjustment started from the initial states. Adjustment terminated when the number of fitness evaluations was 500, and were conducted over 10 trials. After adjustment, the optical axes were set to the best state based on the identified elitist individual.

The parameters in these experiments were as following:

- GA: The population size was 20. The probabilities for crossover and mutation were 0.7 and 0.05, respectively.
- BSA: Parameters of $P(x)$ were C = 0.02, K = 0.15, ó = 0.05 and N = 500.
- BSW: Parameters of $P(x)$ were the same as BSA and k = 100.

5.2 Experimental Results

Table 1 presents the average results for 10 trials for each adjustment. Each trial result is shown in table 2. In these table, `fit' refers to fitness and `reset' is the evaluation value, i.e. positioning error, when the optical axis is reset to the best state based on the adjusted results. The adjusted results obtained with GA and BSA are illustrated in figures 6-8. Figure 6 presents the average of 10 trials. Figure 7 and figure 8 present each trial for both the GA and BSA. Figure 9 presents the results for each trial by BSW which is the proposed method. In these tables and figures, the fitness values for the GA and BSA are actual detected values, while the fitness value of BSW is a calculated weighted average value, which is indicated by * in table 1 and table 2.

Table 1: Results for three adjustments.

	time	**fit-ave**	**fit-σ**	**reset-ave**	**reset-σ**	**positioning error-ave**
GA	26.1min.	17.7	11.2	31.3	16.8	43.5μm
BSA	10.6min.	10.7	16.0	28.9	19.8	40.2μm
BSW	13.1min.	22.7*	12.1*	13.5	10.4	18.8μm

Table 2: Results for each trial of the averaged results for the three adjustments.

	GA		**BSA**		**BSW**	
	fitness	**reset**	**fitness**	**reset**	**fitness**	**reset**
1st trial	14.0	7.6	1.4	10.6	14.0*	4.5
2nd trial	20.0	19.8	7.0	19.0	16.5*	13.0
3rd trial	9.4	40.3	53.6	54.9	13.5*	16.3
4th trial	28.3	51.6	8.5	63.0	11.2*	4.1
5th trial	17.2	22.1	1.0	17.1	46.7*	38.3
6th trial	28.0	24.7	11.4	9.1	23.1*	7.3
7th trial	38.1	49.9	18.0	30.5	21.5*	13.9
8th trial	9.2	56.2	4.1	43.0	26.6*	7.3
9th trial	3.2	23.0	1.0	7.1	40.7*	22.5
10th trial	9.0	17.7	1.4	35.2	13.8*	8.1
average	17.7	31.3	10.7	28.9	22.7*	13.5
standard deviation(σ)	11.2	16.8	16.0	28.9	12.1*	10.4

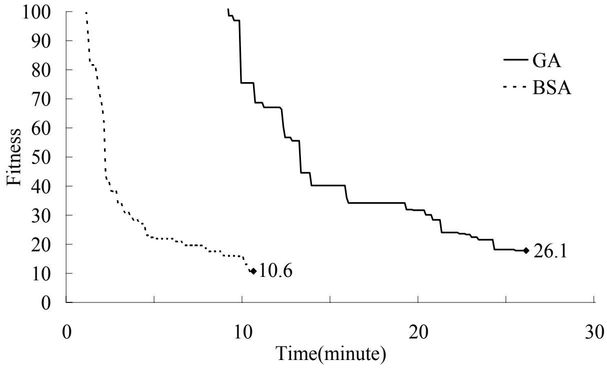


Figure 6: Comparison of BSA and GA.

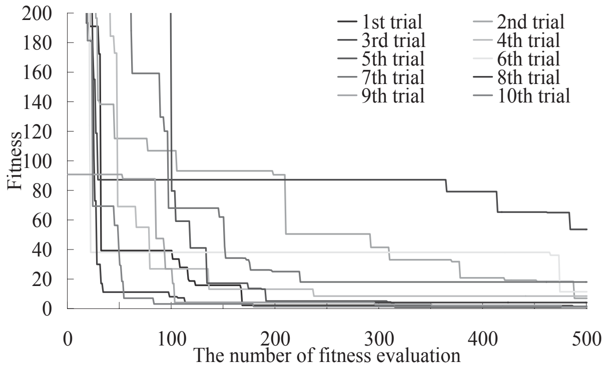


Figure 8: Adjustment results for the BSA.

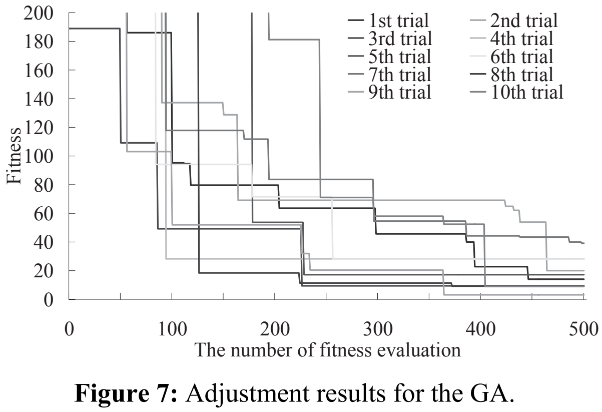


Figure 7: Adjustment results for the GA.

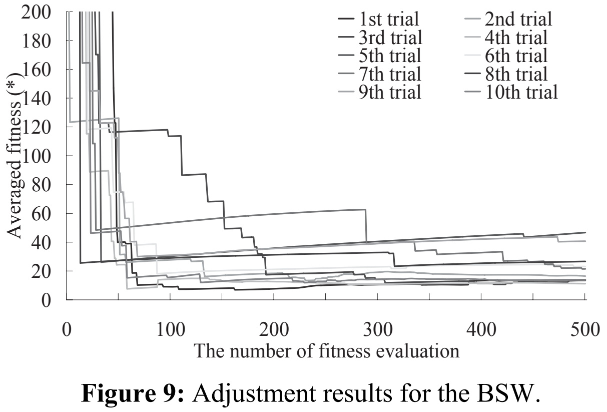


Figure 9: Adjustment results for the BSW.

First, looking at the results for the GA and BSA, clearly both could adjust the optical axes, because the fit-average values were 17.7 (17.7 × 1.39 = 24.6μm) and 10.7 (10.7 × 1.39 = 14.8μm) for the GA and BSA, respectively. Moreover, efficient adjustment was achieved by BSA, because, in terms of adjustment times, the BSA required 10.6 minutes while the GA took 26.1 minutes, as shown in figure 6. The reason for this difference is that BSA does not carry out the exploration step in the final phase. However, the levels of precision with GA and BSA deteriorated after resetting of the optical axes: the precision for the GA dropped to 31.3 (31.3 × 1.39 = 43.5μm) from 17.7 (24.6μm), while the precision for the BSA dropped to 28.9 (28.9 × 1.39 = 40.2μm) from 10.7 (14.8μm). The reason for this is that both the GA and BSA are influenced by observation noise. The large reset-σ values are consistent with this explanation, as shown in table 2. Therefore, both BSA and GA failed to provide robust adjustment.

Second, turning to compare the results for the BSA and BSW, in terms of the average value after the optical axes were reset, the precision with the BSW improves quite remarkably in contrast with that for the BSA, because BSA was 28.9 (40.2μm) while BSW was 13.5 (13.5×1.39 = 18.8μm), as shown in table 1. This result indicates that the precision was enhanced by 60% compared to the results for the GA. Moreover, the BSW was able to achieve a robust adjustment, because reset-σ value for the BSW was 10.4 while reset-σ value for the BSA was 19.8. The BSW was less influenced by noise than the BSA, because it adopted a weighted average value for the fitness value. In terms of adjustment times, the BSW could adjust the optical axes within half time required for the GA, because the BSW required only 13.1 minutes while GA took 26.1 minutes. However, adjustment using BSW took a little more time compared to the BSA, because of the time involved in calculating the weighted average values, although, this can be improved by limiting the extent of search history used.

6 Conclusions

In this paper, we have proposed a method for the robust and efficient automatic adjustment of optical axes using BSW. From the results of adjustments experiments using the proposed method, we have demonstrated that adjustment could be performed with half the time and with enhanced precision compared to the conventional method by 60%. Our proposed method achieved robust and efficient automatic adjustment of the optical axes.

There are three aspects of this method that require future investigation. The first aspect is the application of the proposed method to optical systems involving multiple components with several DOF to verify its effectiveness for more difficult adjustment tasks. The second aspect is the extension of the proposed method in order to adjust the optical axes of laser systems with multi-objective [3], because the adjustment of optical axes needs essentially multi-objective adjustment. The third aspect is to reduce the levels of calculation in the weighted average values, because most of the overhead cost in our method is the additional time to calculate the weighted average values. We are aiming to demonstrate the effectiveness of the proposed method in high-end laser systems, such as femto-second lasers.

7 Acknowledgements

This work was supported in 2004 by Industrial Technology Research Grant Program of the New Energy and Industrial Technology Development Organisation (NEDO) of Japan and Grant-in-Aid for JSPS Fellows in 2005.

8 References

[1] M. Murakawa, T. Itatani, Y. Kasai, H. Yosikawa and T. Higuchi, "An evolvable laser system for generating femtosecond pulses", *Proceedings 2nd Genetic and Evolutionary Computation Conference(GECCO 2000)*, Las Vegas, pp 636-642 (2000).

[2] H. Nosato, Y. Kasai, M. Murakawa, Y. Itatani and T. Higuchi, "Automatic adjustments of a femtosecond-pulses laser using genetic algorithms", *Proceedings 2003 Congress on Evolutionary Computation (CEC 2003)*, Newport Beach, CA, pp 2096-2101 (2003).

[3] N. Murata, H. Nosato, T. Furuya and M. Murakawa, "An automatic multi-objective adjustment system for optical axes using genetic algorithms", *Proceedings 5th International Conference on Intelligent Systems Design and Applications (ISDA 2005)*, Wroclaw, Poland, pp546-551 (2005).

[4] E.J. Hughes, "Multi-objective binary search optimisation", *Proceedings 2nd International Conference on Evolutionary Multi-Criterion Optimisation (EMO 2003)*, Milan, pp 102-117 (2003).

[5] J.M. Fitzpatrick and J.J. Greffenstette, "Genetic algorithms in noisy environments", *Machine Learning*, 3, pp 101-120 (1988).

[6] P. Stagge, "Averaging efficiently in the presence of noise", *Proceedings Parallel Problem Solving from Nature (PPSN V)*, Amsterdam, pp 188-197 (1998).

[7] H. Satoh, M. Yamamura and S. Kobayashi, "Minimal generation gap model for GAs considering both exploration and exploitation", *Proceedings International Conference on Soft Computing*, Iizuka, Japan, pp 494-497 (1997).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Decision Making in Multi-agent Coalitions in a Dynamic Hostile World

Madhu Goyal
Faculty of Information Technology
University of Technology Sydney, Australia
madhu@it.uts.edu.au

Abstract

One of the main underpinning of the multi-agent systems community is how and why autonomous agents should cooperate with one another. Several formal and computational models of cooperative work or coalition are currently developed and used within multi-agent systems research. The coalition facilitates the achievement of cooperation among different agents. In this paper, a mental construct called *attitude* is proposed and its significance in coalition formation in a dynamic fire world is discussed. This paper presents ABCAS (Attitude Based Coalition Agent System) that shows coalitions in multi-agent systems are an effective way of dealing with the complexity of fire world. It shows that coalitions explore the attitudes and behaviours that help agents to achieve goals that cannot be achieved alone or to maximise net group utility.

Keywords: multi-agent, coalition formation, attitudes

1 Introduction

Coalition formation has been addressed in game theory for some time. However, game theoretic approaches are typically centralised and computationally infeasible. MAS researchers [7] [8] [9] [10] [11], using game theory concepts, have developed algorithms for coalition formation in MAS environments. However, many of them suffer from a number of important drawbacks like they are only applicable for small number of agents and not applicable to real world domains. A coalition, is a group of agents who join together to accomplish a task that requires joint task execution which otherwise be unable to perform or will perform poorly. It is becoming increasingly important as it increases the ability of agents to execute tasks and maximise their payoffs. Thus the automation of coalition formation will not only save considerable labour time, but also may be more effective at finding beneficial coalitions than human in complex settings. To allow agents to form coalitions, one should devise a coalition formation mechanism that includes a protocol as well as strategies to be implemented by the agents given the protocol.

This paper will focus on the issues of coalitions in dynamic multi-agent systems: specifically, on issues surrounding the formation of coalitions among possibly among heterogeneous group of agents, and on how coalitions adapt to change in dynamic settings. Traditionally, an agent with complete information can rationalise to form optimal coalitions with its neighbours for problem solving. However, in a noisy and dynamic environment where events occur rapidly, information cannot be relayed among the agent frequently enough, centralised updates and polling are expensive, and the supporting infrastructure may partially fail, agents will be forced to form sub-optimal coalitions. Similarly, in such environments, changes in environmental dynamics may invalidate some of the reasons for the original existence of a coalition. In this case, individual agents may influence the objectives of coalition, encourage new members and reject others and the coalition as a whole adapts as a larger organism. In such settings, agents need to reason, with the primary objective of forming a successful coalition rather than an optimal one, and in influencing the coalition (or forming new coalitions) to suit its changing needs. This paper introduces ABCAS, a novel attitude based coalition agent system in the fire world. The task of fire fighting operations in a highly dynamic and hostile environment is a challenging problem. We suggest a knowledge-based approach to the coalition formation problem for fire fighting missions. Thus the objective of this paper is to design and develop an attitude based approach to the coalition formation for fire fighting problem that would help them to accomplish their tasks during the fire. Owing to the special nature of this domain, developing a protocol that enables agents to negotiate and form coalitions, and provide them with simple heuristics for choosing coalition partners is quite challenging task. The protocol allows the agents to form coalitions, and provide them with simple heuristics that allow the agents to form coalitions in face of time constraints and incomplete information.

2 A Fire Fighting World

We have implemented our formalisation on a simulation of fire world *FFWorld* [5] [6] using a virtual research campus. *FFWorld* is a dynamic, distributed, interactive, simulated fire environment where agents are working together to solve problems, for example, rescuing victims and extinguishing fire. In a world such as this, no agent can have full knowledge of the whole world. Humans and animals in the fire world are modelled as autonomous and heterogeneous agents. While the animals run away from fire instinctively, the fire fighters can tackle and extinguish fire and the victims escape from fire in an intelligent fashion. An agent responds to fire at different levels. At the lower level, the agent burns like any object, such as chair. At the higher level, the agent reacts to fire by quickly performing actions, generating goals and achieving goals through plan execution. This world contains all the significant features of a dynamic environment and thus serves as a suitable domain for collaborating agents. Agents operating in the domain face a high level of uncertainty caused by the fire. Agents in the fire domain do not face the real time constraints as in other domains, where certain tasks have to be finished within the certain time. However, because of the hostile nature of the fire, there is strong motivation for an agent to complete a given goal as soon as possible.

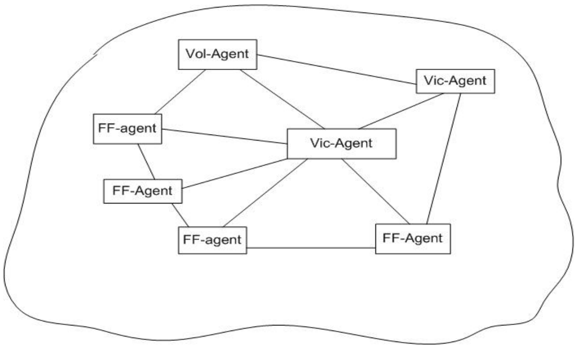


Figure 1: Coalition between fire-fighter, volunteer and victim agent.

There are three main objectives for intelligent agents in the world during the event of fire: self-survival, saving objects including lives of animals and other agents and put-off fire. Because of the hostile settings of the domain, there exist a lot of challenging situations where agents need to do the cooperative activities. Whenever there is fire, there is need of coalition between the fire fighters (FF-agent), volunteers (Vol-agent) and victim agents (Vic-agent) (figure 1). The fire fighters perform all the tasks necessary to control an emergency scene. The problem solving activities of the fire fighters are putting out fire, rescuing victims and saving property. Apart from these primary activities there are a number of sub tasks e.g. run towards the exit, move the objects out of the room, remove obstacles, and to prevent the spread of fire. The first and paramount objective of the victim agents is self-survival. The role of volunteer agents is to try to save objects from the fire and help out other victims who need assistance when they believe their lives are not under threat. To achieve these tasks there is need of coalitions between these agents is necessary. Thus the fire world we consider is sufficiently complex to bring about the challenges involved in to study coalition formation in a typical multi-agent world.

3 Strategic Coalitions in an Agent Based Hostile World

The coalition facilitates the achievement of cooperation among different agents. The cooperation among agents succeeds only when participating agents are enthusiastically unified in pursuit of a common objective rather than individual agendas. We claim that cooperation among agents is achieved only if the agents have a *collective attitude* towards cooperative goal as well as towards cooperative plan. From collective attitudes, agents derive individual attitudes that are then used to guide their behaviours to achieve the coalition activity. The agents in a coalition can have different attitudes depending upon the type of the environment the agent occupies.

3.1 Definition of Attitude

Attitude is a *learned predisposition to respond in a consistently favourable or unfavourable manner with respect to a given object* [4]. In other words, the attitude is a preparation in advance of the actual response, constitutes an important determinant of the ensuing behaviour. However this definition seems too abstract for computational purposes. In AI, the fundamental notions to generate the desirable behaviours of the agents often include goals, beliefs, intentions, and commitments. Goal is a subset of states, and belief is a proposition that is held as true by an agent. Bratman [1] addresses the problem of defining the nature of intentions. Crucial to his argument is the subtle distinction between doing something intentionally and intending to do something. The former case might be phrased as deliberately doing an action, while intending to do something means one may not be performing the action in order to achieve it. Cohen and Levesque [3], on the other hand, developed a logic in which intention is defined. They define the notion of individual commitment as *persistent goal*, and an intention is defined to be a commitment to act in a certain mental state of *believing throughout* what he is doing. Thus to provide a definition of attitude that is concrete enough for computational purposes, we model attitude using goals, beliefs, intentions and commitments. From Fishbein’s [4] definition it is clear that when an attitude is adopted, an agent has to exhibit an appropriate behaviour (predisposition means *behave in a particular way*). The exhibited behaviour is based on a number of factors. The most important factor is goal or several goals associated with the object. During problem solving, an agent in order to exhibit behaviour may have to select from one or several goals depending on the nature of the dynamic world.

In a dynamic multi-agent world, the behaviour is also based on appropriate commitment of the agent to all unexpected situations in the world including state changes, failures, and other agents' mental and physical behaviours. An agent intending to achieve a goal must first commit itself to the goal by assigning the necessary resources, and then carry out the commitment when the appropriate opportune comes. Second, if the agent is committed to executing its action, it needs to know how weak or strong the commitment is. If the commitment is week, the agent may not want to expend too much of its resources in achieving the execution. The agent thus needs to know the degree of its commitment towards the action. This degree of commitment quantifies the agent's *attitude* towards the action execution. For example, if the agent considers the action execution to be *higher importance* (an attitude towards the action), then it may choose to execute the action with greater degree of commitment; otherwise, the agent may drop the action even when it had failed at the first time. Thus, in our formulation, an agent when it performs an activity, since the activity is more likely that it will not succeed in a dynamic world; agents will adopt a definite attitude towards every activity while performing that activity. The adopted attitude will *guide* the agent in responding to failure situations. Also the behaviour must be consistent over the period of time during which the agent is holding the attitude. Thus attitudes, once adopted, must persist for a reasonable period of time so that other agents can use it to predict the behaviour of the agent under consideration. An agent cannot thus afford to change its attitude towards a given object too often, because if it does, its behaviour will become somewhat like a reactive agent, and its attitude may not be useful to other agents. Once an agent chose to adopt an attitude, it strives to maintain this attitude, until it reaches a situation where the agent may choose to drop its current attitude towards the object and adopt a new attitude towards the same object. Thus we define attitude as: *An agent's attitude towards an object is its persistent degree of commitment to one or several goals associated with the object, which give rise to persistent favourable or unfavourable behaviour to do some physical or mental actions.*

3.2 Type of Attitudes

The attitudes of the agents in the world consist of attitudes towards the physical objects, mental objects, processes and other agents. When attitudes are attached to physical objects, the agents are able to evaluate the *liking*, *importance* or *location* etc. of these physical objects. When attitudes are attached to mental objects, agents are able to communicate and reason with those mental objects. For example, agents can actively monitor their plans so those plans can be re-organised or abandoned when the world state changes. If the object denotes a mental object such as a plan, *higher-priority* can be an attitude that the agent may hold towards the plan. In that case, the agent will perform behaviour appropriate to this attitude, which may involve physical, communicative, and mental actions or a combination of these which may lead to behaviour where the agent gives higher preference to the plan compared to the other plans in all possible situations. Agents can also have attitudes towards processes such as execution of actions and plans, the process of achieving goals, etc. For example, if the execution of a plan goes on for too long, appropriate attitude is necessary to define how to handle the situation. Similarly, attitudes are also possible towards process states, such as the execution state of a plan after a crucial action has been executed. Similarly, attitudes towards other agents and their mental states define the agent's behaviour towards these agents.

Behaviours exhibited by an agent in a multi-agent environment can be either individualistic or collective. Accordingly, we can divide attitudes in two broad categories: *individual* attitudes and *collective* attitudes. The *individual* attitudes contribute towards the single agent's view towards an object or person. An agent's attitude toward an object is based on its salient beliefs about that object. The agent's individual attitude toward a fire world, for example, is a function of its beliefs about the fire world. The *collective* attitudes are those attitudes, which are held by multiple agents. The collective attitudes are individual attitudes so strongly interconditioned by collective contact that they become highly standardised and uniform within the group, team or society etc. The agents can collectively exist as *societies*, *groups*, *teams*, *friends*, *foes*, or just as *strangers*, and *collective* attitudes are possible in any one of these classifications. For example, the agents in the collection called *friends*, can all have a collective attitude called *friends*, which is mutually believed by all agents in the collection. A collective attitude can be viewed as an abstract attitude consisting of several component attitudes, and for an individual agent to perform an appropriate behaviour; it must hold its own attitude towards the collective attitude. Thus, for example, if A1 and A2 are *friends*, then they mutually believe they are friends, but also each Ai must have an attitude towards this infinite nesting of beliefs so that it can exhibit a corresponding behaviour. Thus, from A_1's viewpoint, *friends* is an attitude that it is holding towards the collection $\{A_1, A_2\}$ and can be denoted as $friendsA_1$ (A_1, A_2). Similarly, from A_2's view point, its attitude can be denoted as $friendsA_2$ (A_1,A_2). However, in an extreme case, A1 may not be certain about A2's behaviour. That is why; it needs to have an attitude of its own, which generates a behaviour taking all the uncertainties introduced by the dynamic environment into account. Further note that A1 might implicitly expect A2 to perform its role, but it is only an expectation. This is a bottom up view of the friends-relation**,** where the relation is viewed in general differently by each agent depending on the local situations the individual agents face in the world.

3.3 Attitude Based Agents

We adopt a BDA (figure 2 modified BDI architecture) based approach in which agent is comprised of: *beliefs* about itself, others and the environment; set of *desires* representing the states it wants to achieve; and *attitudes* corresponding to the plans adopted in pursuit of the desires.

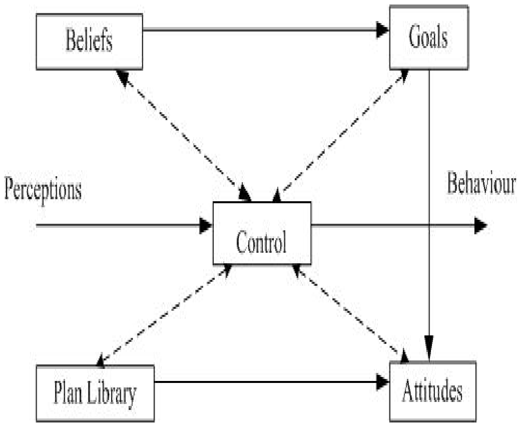


Figure 2: BDA agent architecture.

In comparison to traditional BDI [2] model, we have replaced *intentions* with *attitudes*. We say that *intentions* are primitive forms of *attitudes* without degree of commitment in them. An agent has a set of attitudes, each with a degree of commitment which persists according to the current situation. The attitudes are represented by following attributes:

Name of Attitude: This attribute describes the name of the attitude e.g. *like, hate,* cautious etc.

Description of Object: The description of the object contains the name of the object and a description of the internal organisation in terms of the components of the object.

Basic agent behaviour towards x: This attribute specifies the behaviour that will be performed by the agent with respect to the object *x*.

Evaluation: This attribute specifies whether the attitude is favourable or not.

Concurrent attitudes: This attribute specifies any other attitudes that can coexist with this attitude.

Persistence of Attitude: *This attribute specifies how long the attitude will persist under various situations. For example, it may specify how the attitude itself will change over time; that is, when to drop it and change it to another attitude, when to pick it up and how long to maintain it.*

Type of Attitude: This attribute specifies whether the attitude is individual or collective.

3.4 Attitude Based Coalition Model

We claim that successful coalition is achieved only if the agents have coalition as a collective abstract attitude. From this collective attitude, agents derive individual attitudes that are then used to guide their behaviours to achieve the coalition. Suppose there *n* agents in a coalition i.e. A_1…. A_n. So the collective attitude of the agent A_1…. A_n towards the coalition is represented as $Coal_{A1..An}(A_1, .. ,A_n)$. But from A_1's viewpoint, team is an attitude that it is holding towards the collection $(A_{1,}…,A_n)$ and can be denoted as $Coal_{A1}(A_1, A_2)$. Similarly from A_n's viewpoint, its attitude can be denoted as $Coal_{An}(A_1,.., A_n)$. But the collective attitude $Coal_{A1\ \ An}$ $(A_1,…, A_n)$ is decomposed into the individual attitudes only when all the agents mutually believe that they are in the coalition. The coalition attitude can be represented in the form of individual attitudes towards the various attributes of the coalition i.e. coalition methods, coalition rule base, and coalition responsibility.

The attitudes of an agent existing in a coalition consist of attitude towards coalition as well as attitude towards coalition activity. At any time, an agent may be engaged in one of the basic coalition activities i.e. coalition formation, coalition maintenance, and coalition dissolution. Instead of modelling these basic activities as tasks to be achieved, we have chosen to model them as attitudes.

Coalition (A1,..,A2)

This attitude is invoked when the agents are in a team state. This attitude guides the agents to perform the appropriate coalition behaviours.

Name of attitude: **Coalition**

Description of object: (1) Name of object: set of agents (2) Model of object: {A1,An | Ai is an agent}

Basic agent behaviour: coalition behaviour specified by agent's rule base

Evaluation: favourable

Persistence: This attitude persists as long as the agents are able to maintain it.

Concurrent attitudes: all attitudes towards physical and mental objects in the domain.

Type of attitude: collective.

3.4.1 Coalition Formation

Impetus for attitude and coalition formation may arise from the world and a particular domain or from agent's themselves. Having identified the potential for coalition action with respect to one of its goals, a leader agent will solicit assistance from some group of agents that it believes can achieve the goal. Our agents form coalition, because the inherent nature of the world requires agents to exist and act together. However, a particular situation may force the agents to dissolve the coalition for some time.

In the fire world, the event triggering the coalition formation process is a fire. Whenever there is fire, the security officers call the fire-fighting company to put out the fire. Then the fire fighters arrive at the scene of fire and get the information about when, how and where the fire had started. Suppose there is a medium fire in the campus, which results in the attitudes *medium-fire* and *dangerous-fire* towards the object fire. The attitude *Coal-form* is also generated, which initiates the team formation process. We propose a dynamic team formation model, in which we consider initially the mental state i.e. the beliefs of all the agents is same. The fire-fighting agents recognise appropriateness of the team model for the task at hand; set up the requirements in terms of other fellow agents, role designation, and structure; and develop attitudes towards the team as well as towards the domain.

In order to select a member of the team, our agent will select the fellow agent who has following capabilities:

- Has knowledge about the state of other agents.
- Has attitude towards the coalition formation.
- Can derive roles for other agents based on skills and capabilities.
- Can derive a complete joint plan.
- Can maintain a coalition state.

Our method of forming a coalition is like this; the agents start broadcasting message to other agents “Let us form a coalition”. The agents will form a coalition if two or more than two agents agree by saying, “Yes”. If the agent does not receive the “Yes” message, it will again iterate through the same steps until the coalition is formed. The *coal-form* is maintained as long as the agents are forming the team. Once the team is formed, agents will drop the *coal-form* attitude and form the *coal* attitude, which will guide the agents to produce various team behaviours.

Coal-form (A_1, ..,A_2)

This attitude is invoked when the agents have to form a coalition to solve a complex problem.

Name of Attitude: **Coal-form**

Description of object: (1) Name of object: set of agents. (2) Model of object: {A1,An | Ai is an agent}

Basic agent behaviour: invokes coalition formation rules.

Evaluation: favourable

Persistence: The agent holds this attitude as long as it believes that a coalition formation is possible.

Concurrent attitudes: All attitudes towards physical and mental objects in the domain.

Type of attitude: individual

3.4.2 Coalition Maintenance and Dissolution

While solving a problem (during fire fighting activity) the coalition agents have also to maintain the coalition. During the coalition activity the agents implement the coalition plan to achieve the desired coalition action and sustain the desired consequences. The coalition maintenance behaviour requires what the agent should do so that coalition does not disintegrate. In order to maintain the coalition each agent should ask the other agent periodically or whenever there is a change in the world state, whether he is in the coalition. So the attitudes like *periodic-coalition-maintenance* and *situation-coalition-maintenance* are produced periodically or whenever there is a change in the situation. These attitudes help the agent to exhibit the maintenance behaviours.

When the team task is achieved or team activity has to be stopped due to unavoidable circumstances, the attitude *coal-unform* is generated. This attitude results in the dissolution of the team and further generates attitude *escape*. For example, when the fire becomes very large, the agents have to abandon the team activity and escape. The attitude *coal–unform* is maintained as long as the agents are escaping to a safe place. Once the agents are in the safe place, the attitudes *team-unform* and *escape* are relinquished. In case the fire comes under control, the agents again form a team by going through the steps of team formation.

4 Experimental Results

The key evaluation criterion for the overall performance of the coalition in any environment is that agents must successfully accomplish their tasks within their given environments, both efficiently and accurately. We have done several experiments for fire fighting in *FFWorld* domain to verify our ideas about the coalition model. The motivation behind doing these experiments is to determine the advantages and disadvantages of our attitude based coalition model. The agents in a coalition react to the changes in the world states by generating and achieving new goals. Meanwhile, old goals and plans are constantly being monitored and re-structured if necessary. The attitudes are mainly concerned with how to re-organise plans and goals due to situational changes. Whenever there is a new goal because of the changes in the world, one or more attitudes are usually created along with the goal. The experiments have concentrated on evaluating the performance of attitude based coalition in case of unexpected events. When a problem occurs, the attitude model stipulates a new set of behaviours for the agents. The agents with attitude respond to changes in the world by adopting a set of attitudes towards these changes.

We carried out different types of experiments to investigate the problem solving behaviour of the agents during team action and to analyse their coherence in hostile dynamic environments. These sets of experiments demonstrate the significance of employing attitudes when agents have to deal with individual, group, social and team goals in a changing world. In our experiments, we have tried to measure the average payoff of the coalition under the varying conditions. To provide comparative, as well as quantitative results, three distinctive types of coalition were considered (i) *high* coalition (Coal (high)) (ii) *medium* coalition (Coal (medium)) (iii) *low* coalition (Coal (low)). In the *high* coalition, the commitment of agents is very large (> 80%). In the *medium* coalition, the commitment of agents towards coalition is medium (60%-80%). In the *low* teams (T_{low}), the nature of the task is such that they are not required to work closely together all the time and percentage of attitudes is very less (<60%).

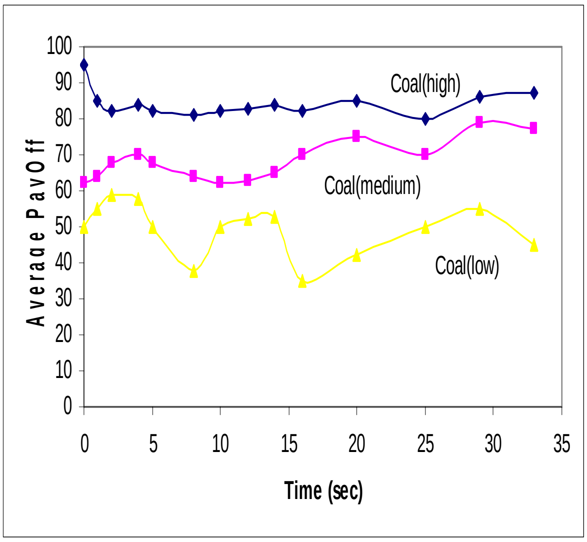


Figure 3: Average payoff of the agents with C_{high}, C_{medium} and C_{low}.

5 Conclusions

This paper has developed a novel framework for managing coalitions in a hostile dynamic world. Coalition is guided by the agent's dynamic assessment of agent's attitudes given the current scenario conditions, with the aim of facilitating the agents in coalitions to complete their tasks as quickly as possible. In particular, it is outlined in this paper that how agents can form and maintain a coalition, and how it can offer certain benefits to cooperation. Our solution provides a means of maximising the utility and predictability of the agents as a whole. Its richness presents numerous possibilities for studying different patterns of collaborative behaviour.

6 References

[1] M.E. Bratman, *Intentions, Plans and Practical Reason*, Harvard University Press, Cambridge, MA (1987).

[2] M.E. Bratman, D. Israel and M. Pollack, Plans and resource bounded practical reasoning", *Computational Intelligence*, 4, pp 349-355 (1988).

[3] P.R. Cohen and H.J. Levesque, "Teamwork", *Special Issue on Cognitive Science and Artificial Intelligence*, 25(4) (1991).

[4] M. Fishbein and I. Ajzen, *Belief, Attitude, Intention and Behaviour: An Introduction to Theory and Research*, Addison-Wesley, Reading, MA (1975).

[5] M. Goyal, *An Attitude Based Teamwork Model for a Hostile Dynamic Multiagent World*, PhD Thesis, University of New South Wales (2001).

[6] M. Goyal, "Collaborative negotiations in a hostile dynamic world", *Proceedings International Conference on Artificial Intelligence (ICAI '04)*, Las Vegas (2004).

[7] N. Griffiths and M. Luck, "Coalition formation through motivation and trust", *Proceedings 2nd International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2003)*, Melbourne, ACM Press, pp 17-24 (2003).

[8] S. Kraus, O. Shehory and G. Taase, "Coalition formation with uncertain heterogeneous information", *Proceedings 2nd International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2003)*, Melbourne (2003).

[9] T. Sandholm, K. Larson, M. Andersson, O. Shehory and F. Tohmé, "Coalition structure generation with worst case guarantees", *Artificial Intelligence*, 111(1-2), pp 209-238 (1999).

[10] O. Shehory and S. Kraus, "Task allocation via coalition formation among autonomous agents", *Proceedings 14th International Joint Conference on Artificial Intelligence*, Montreal, pp 655-661 (1995).

[11] C. Li, S. Chawla, U. Rajan and K. Sycara, *Mechanisms for Coalition Formation and Cost Sharing in an Electronic Marketplace*, Technical Report CMU-RI-TR-03-10, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2003).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Extending Structure Adaptive Self-Organising Map for Mixed Data Classification

Chung-Chian Hsu, Kuo-Min Wang, Yu-Wei Su and Wei-Shen Tai
Department of Information Management,
National Yunlin University of Science and Technology, Taiwan, R.O.C.
hsucc@yuntech.edu.tw

Abstract

In data mining applications, self-organising map (SOM) is regarded as an effective visualised clustering technique for preserving the topological relations in input data. A variant of SOM, SASOM was proposed for clustering as well as classifying multidimensional data. However, SASOM cannot be appropriately applied in categorical or mixed (i.e. categorical and numeric) data due to its model and Euclidean distance function suitable for only numerical data. In this paper, we propose an extended SASOM (ESASOM), integrating with a Generalised SOM (GSOM) which handles mixed-type data, for manipulating both numerical and categorical data in classification applications. ESASOM possesses the ability of both dynamic node-splitting for improving classification performance and measuring the distance between multidimensional mixed data. Experimental results demonstrated that this proposed method can provide better classification and visualised results on mixed-type data than SASOM.

Keywords: data mining, self-organising map (SOM), mixed data, distance hierarchy, classification

1 Introduction

Self-organising map (SOM), proposed by Kohonen, is regarded as an effective data visualisation technique in data mining applications. It can map high-dimensional data into low-dimensional space and preserve the topological relationship between input data through data projection [1]. SOM has been applied to problems in the robotics field, including navigation [2], posture analysis [3] and so on.

Nevertheless, the conventional SOM methods must predefine the map structure prior to training. When the map size is not large enough, it may fail to appropriately reflect the topology of input data. On the contrary, too large map size may cause the similar data to disperse to excess clusters.

Structure adaptive SOM (SASOM) provides a feasible solution for improving the foregoing problem of conventional SOM methods [4]. It adapts dynamically adjusting the SOM map structure for increasing classification accuracy by a node-splitting scheme. However, neither SOM nor SASOM can manipulate both numerical and categorical data due to the Euclidean distance used in their models inappropriate for categorical data. Hsu [5] proposed a Generalised SOM (GSOM) to calculate the distance between categorical data via distance hierarchy. However, when the number of data is tremendous, it may fail to preserve the topological order caused by the predefined fixed map size. Moreover, when used for classification problems, GSOM can not achieve good performance for the same reason.

In this paper, we propose an extended SASOM (ESASOM), integrating SASOM with GSOM, to manipulate mixed data and improve the classification accuracy as well. This paper is structured as follows. In section 2, several related SOM methods are reviewed and compared. In section 3, distance hierarchy and ESASOM are elaborated. In section 4, we present several experimental results of mixed data. Finally, some conclusions are stated at the end of this paper.

2 SOM Methods

To establish background knowledge related to the proposed ESASOM, several SOM methods are elaborated and compared in this section.

2.1 SOM

Due to its projection capability and topology preservation property, SOM has become a popular tool in visualised clustering of multidimensional data. The SOM training algorithm consists of two essential steps: identification of the best matching unit (BMU) to input data, and adjustment of BMU and its neighbourhood [1] to resemble input data. Conventional SOM handles only numeric data since those two training steps rely on a Euclidean distance function. When categorical values are encountered, conventional SOM methods usually resort to data transformation, in particular, transforming into 0-1 binary codes, such that the traditional training algorithm can be applied. However, the approach encounters a serious problem: the similarity of ontology meaning between categorical data cannot be appropriately represented through measuring the distance of binary codes. For example, Pepsi is intuitively more similar to Coke than to coffee. Nevertheless, they possess the same similarity degree in accordance with the computation of geometric distance based on the Euclidean distance function after the transformation [5].

2.2 Growing Variant SOM

Since the fixed network structure may cause a poor result in clustering and classification applications, the incremental growing SOM methods were proposed to conquer the constraint of fixed network structure. Generally speaking, the types of growing SOM can be roughly divided into single layer and multiple layers. The former inserts new neurons in between old ones on the same map. The later applies a hierarchical structure of multiple layers where each layer consists of a number of independent neurons or SOMs.

Single layer growing SOM methods, such as growing grid [6], incremental growing grid [7], growing SOM [8] and growing cell structure [9] can grow on a fixed map and insert neurons according to different schemes. Therefore, they were regarded as a feasible solution for providing more flexible network structure through inserting neurons in the SOM. Nevertheless, they cannot directly represent the hierarchical relationship of data which might be inherent in massive data.

The hierarchical relationship between data can easily be preserved in the hierarchical structure of multiple layers growing SOMs. They possess the ability that traditional SOM owns, i.e. projecting high-dimensional to low-dimensional space. Additionally, they are able to better handle massive data due to their multiple layers structure. For instance, TreeGCS [10], one of popular growing hierarchical SOMs (GHSOMs), applies a dendroid structure to maintain the clustering data in each node. In practical applications, GHSOM has been applied in legal documents and newspaper set [11].

2.3 Structure Adaptive SOM

Conventional SOM methods are usually applied for clustering problems in which class attribute does not participate in the clustering process. Since the fixed network structure of SOM and data with different class labels possibly assigned to the same cluster, SOM could not perform well when used for classification problems. To solve the problem, structure adaptive SOM (SASOM) [4] was proposed to improve the class consistence of each node and classification accuracy via a dynamic node-splitting scheme. However, like other conventional SOM methods, SASOM still process merely numeric data and transformation is needed to convert categorical data prior to training. The same problem mentioned in section 2.1 also occurs to SASOM.

2.4 Generalised SOM

Like encountered in training an SOM, it is regarded as a difficult problem for processing categorical data and appropriately measuring the distance between them. Many researches have proposed various schemes to solve the foregoing problem, such as binary encoding, simple matching and an entropy-based measure [12]. Unfortunately, they are still unable to reasonably represent the ontological distance between categorical data.

Generalised SOM (GSOM) [5] was proposed to manipulate the categorical data for the training of an SOM. It applies distance hierarchy for representing and measuring the distance between categorical data. In this distance hierarchy, the upper nodes represent more general concepts while the lower nodes represent more specific concepts. Through this method, the categorical data can be more effectively and intuitively handled and its structure be better reflected on the trained map.

3 Extended Structure Adaptive SOM

In this paper, an extended structure adaptive SOM (ESASOM) is proposed to not only improve the classification accuracy but also appropriately process mixed data in classification applications.

3.1 Distance Hierarchy of Categorical Data

A distance hierarchy, composed of concept nodes, links and link weights, represents the ontological relationship between concepts [5]. In this distance hierarchical structure, the upper nodes represent more general concepts; on the contrary, the lower nodes represent more specific concepts. For example, the node of Coke and Pepsi belong to carbonated drinks, as shown in figure 1. Juice, coffee and carbonated drinks all belong to “Any”.

To illustrate the difference between distance hierarchy and other popular methods, the distances between Coke, Pepsi and Mocca are measured through distance hierarchy, simple matching and binary encoding, as shown in table 1. In the distance hierarchy (figure 1), the weight of each link is assumed to be a constant, say 1, to represent the distance between a node and its parent node. Neither simple matching nor binary encoding can distinguish the difference between those three drinks. In other words, the three drinks have the same distance/similarity according to the foregoing two methods. By contrast, via the distance hierarchy Coke is measured to be more similar to Pepsi than to Mocca. In fact, distance hierarchy is a general scheme in which both simple matching and binary encoding can be modelled as special cases [5]. Therefore, distance hierarchy is a better approach to represent the distance between categorical data.

A point, say *X*, can be at any position of a distance hierarchy, which is denoted by an anchor (a leaf node) N_x and its positive offset as *X*= (N_x, d_x) where d_x is the distance from the root to *X*. The distance between point *X* and point *Y* can be calculated as follows [5].

$$\delta(X,Y) = d_X + d_Y - 2d_{LCP(X,Y)} \qquad (1)$$

where d_x and d_y are the distance from the root to *X* and *Y*, respectively. $d_{LCP(X,Y)}$ is the distance from the root to the least common point (LCP). For the example of figure 1, assume *X*= (Coke, 2.0) and *Y*= (Mocca, 1.7). The distance δ(*X*,*Y*) = 2.0 + 1.7 – 2*0 = 3.7. LCP(*X*,*Y*) =Any in this case.

Table 1: Distance comparison between different methods.

Node \ Method		Distance hierarchy	Simple matching	Binary encoding
Coke	Mocca	4	1	1.414
Coke	Pepsi	2	1	1.414
Mocca	Pepsi	4	1	1.414

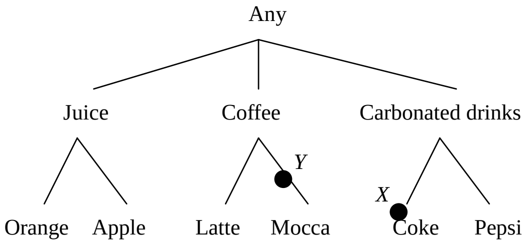


Figure 1: An example of distance hierarchy.

3.2 Processes of GSASOM

The process of training an ESASOM (shown in figure 2), can be divided to data training stage and dynamic node-splitting stage, elaborated as follows.

3.2.1 Data Training

In the data training stage, GSOM training algorithm is applied to process the mixed data. The initial map size is set to 4*4 and random weights are initially assigned to the nodes. During training, nodes which do not satisfy any of the stop conditions are identified for splitting. The stop conditions are (i) the number of data in the node is less than 2% of the total number of the data, (ii) the class consistence of a node has reached user-specified threshold, and (iii) the variance of data in the node is less than user-defined threshold. Class consistence and variance of a node are defined as follows.

$$Class\,Consistence = \max_{c_j} \frac{n_{c_j}}{n_{Total}} \qquad (2)$$

where n_{cj} is the number of data belonging to class C_j and n_{Total} is the number of total data in the node.

$$Variance = \frac{1}{n_{Total}} \sqrt{\sum_{i=1}^{n_{Total}} (x_i - cv)^2} \qquad (3)$$

where x_i is an input data assigned to the node, *cv* is the weight vector of the node and n_{Total} is the number of total data in the node.

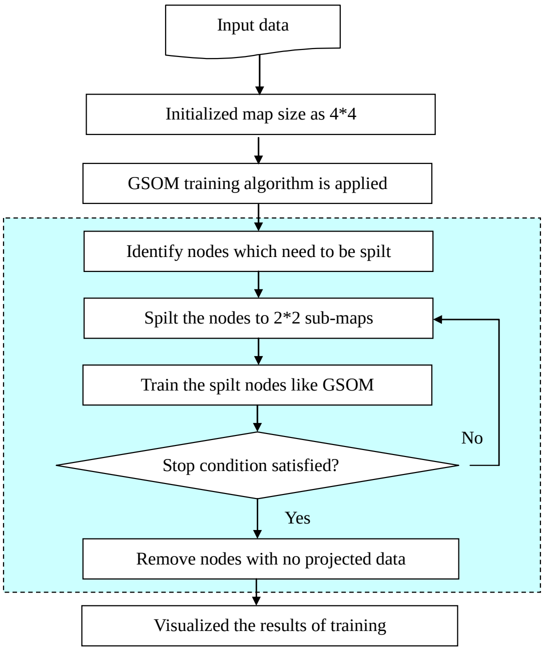

Input data
Initialized map size as 4*4
GSOM training algorithm is applied
Identify nodes which need to be spilt
Spilt the nodes to 2*2 sub-maps
Train the spilt nodes like GSOM
Stop condition satisfied?
No
Yes
Remove nodes with no projected data
Visualized the results of training

Figure 2: Training process of ESASOM.

3.2.2 Dynamic Node-splitting

The node which needs splitting are expanded to 2*2 sub-maps (or child nodes) and trained like GSOM, as shown in figure 3. Before iterative training, the initial weight of each child node is assigned to the mean of the parent node and its neighbourhood nodes. Specifically, for mixed-type training data, the initial weights for the parts of numeric and categorical attributes are assigned separately as follows.

(i) Initial weight for the part of numerical attributes

$$w_c = \frac{w_p * 2 + \sum_{k=1}^{n_c} w_k}{n_c + 2} \qquad (4)$$

where w_c is the initial weight of a child node, w_p is the weight of its parent node, w_k and n_c are the weight and the number of its neighbourhood nodes, respectively. For instance of figure 3, the involving nodes for the weight calculation of the child c_0 include P_0, P_1, and P_4.

(ii) Initial weight for the part of categorical attributes

Regarding categorical attribute values, the mean is represented by the centroid of the points that its parent node and neighbourhood nodes projected onto its distance hierarchy (shown as figure 4). In other words, the centroid p_c is the point that gives the shortest distance from itself to the parent-node point and each neighbourhood-node point. That is,

$$p_c = \arg\min_{p_k} \sqrt{\sum_{i=0}^{n_p} \delta(p_k, p_i)^2} \tag{5}$$

where p_k is a point in the distance hierarchy, and p_i includes the points of the node to be split and neighbourhood nodes (e.g. P_0, P_1, and P_4 in figure 4).

The identification of the centroid can be restricted to the area in between the involving points. For example, the points on the bold lines in figure 4 are the candidates for the centroid. Specifically, we determine the local centroid of each of the involved bold lines and then identify the global centroid from the local centroids. The local centroid of an edge can be determined as follows. The derivation of the formula will be detailed in a full paper.

$$ed_{p_{r,l}} = \frac{\sum_{LCP(p_{r,l},p_i)=p_{r,l}}^{n_s} d_{p_i} - \sum_{LCP(p_{r,l},p_i)\neq p_{r,l}}^{n_d} d_{p_i} - 2d_{LCP(p_{r,l},p_i)}}{n_s + n_d} \tag{6}$$

where $ed_{p_{r,l}}$ is the estimated distance from the root to $p_{r,l}$, $p_{r,l}$ is the local centroid of the l^{th} edge in the r^{th} branch, n_s and n_d are the number of $p_{r,l}$ and p_i in the same and different link, respectively.

Since $ed_{p_{r,l}}$ is the distance of local centroid on the l-th edge to the root, it is expected to be in the range of l-1 to l. The value shall be set to the end values of the edge in case that the calculated result is out of that range. In other words, the offset of local centroid $d_{p_{r,l}}$ is defined as follows.

$$d_{p_{r,l}} = \begin{cases} l & \text{,if } ed_{p_{r,l}} > l \\ l-1 & \text{,if } ed_{p_{r,l}} < l-1 \\ ed_{p_{r,l}} & \text{,otherwise} \end{cases} \tag{7}$$

Then, the centroid of a child node is acquired from the local centroids of the involving edges as follows.

$$p_c = \underset{p_{r,l} \in \text{local } centroids}{\arg\min} \{\sum_{i=0}^{n_p} \delta(p_i, p_{r,l})^2\} \tag{8}$$

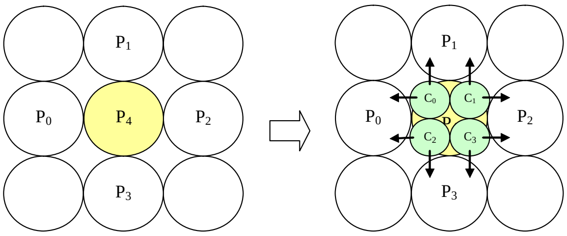


Figure 3: An example of node-splitting.

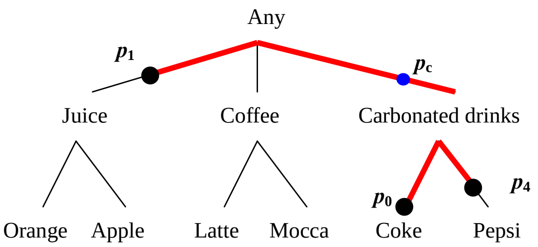


Figure 4: p_c the centroid of p_0, p_1 and p_4, representing the initial weight of a child node.

4 Experiments

We developed a prototype using C++. Two experimental results of a synthetic and a real mixed dataset are presented to show the comparison of SASOM and ESASOM.

4.1 Parameter Setting

The initial map size was 4*4. The number of training iteration was 1,000. The learning rate was a linear function $\alpha(t) = \alpha(0) * (1.0 - t/T)$ with the initial value $\alpha(0) = 0.9$. A Gaussian function was used as the neighbourhood function $r(t) = 1.0 + (r(0) - 1) * (1.0 - t/T)$ with the initial value $r(0)$ set to the length of the map. The class consistence and variance thresholds were set to 0.95 and 0.9, respectively.

4.2 Synthetic Mixed Dataset

The synthetic mixed dataset consists of nine classes with two categorical attributes (Department and Drink) and one numerical attribute (Amount), as shown in table 2. The Amount values were randomly generated according to the specified normal distribution. Each class possesses some characteristics. Two distance hierarchies are built to represent the ontological relationship between categorical data as shown in figure 5. Each link weight is assumed to be 1.

Training results of SASOM and ESASOM are shown in figure 6. Figure 6a shows the data belonging to the same class in table 2 can be projected into the same neuron, which effectively improves the drawback of a neuron containing data with different classes in conventional SOMs. The size of neurons indicates the splitting level of the neuron. For instance, the neuron labelled by 2 is in one level lower than that labelled by 1. A problem shown in figure 6a is that similar classes are not projected nearby, such as class 7, 8 and 9, since SASOM was trained with the categorical values being transformed to 0-1 binary codes which do not consider the similarity embedded in the categorical values.

Figure 6b is the results by the proposed ESASOM method. Like in SASOM, the data in the same class were projected to the same neuron. Two main differences to figure 6a can be identified. First, similar classes were projected nearby, e.g. class 7, 8, and 9 in the upper middle of the map, which better reflects the structure of the data in table 2. Second, we improved the visualisation of ESASOM which help users gain more insights of the training results. Specifically, the size of the dark dots represents the size of the data projected to the neuron. The same colour of a dark dot implies a uniform class of the projected data while a multi-coloured dark dot indicates multi-class in the neuron (see figure 9b). The size of shadow surrounding a dark dot indicates the splitting level of the neural, like that in SASOM.

Table 2: A synthetic mixed dataset "Student".

Class	Dept.	Drink	Amount (μ, σ)	Data Count	Characteristics
1	MIS	Coke	(500, 25)	60	Management with carbonated drinks
2	MBA	Pepsi	(400, 20)	30	
3	MBA	Pepsi	(300, 15)	30	
4	EE	Latte	(500, 25)	60	Engineering with coffee
5	CE	Mocca	(400, 20)	30	
6	CE	Mocca	(300, 15)	30	
7	SD	Apple	(500, 25)	60	Design with juice
8	VC	Orange	(400, 20)	30	
9	VC	Orange	(300, 15)	30	

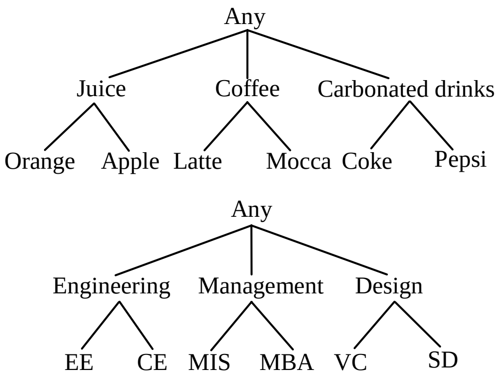
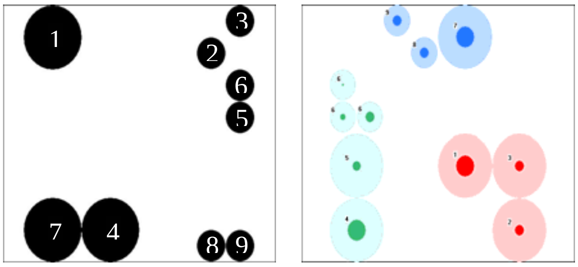


Figure 5: Conceptual distance hierarchies for categorical attributes.



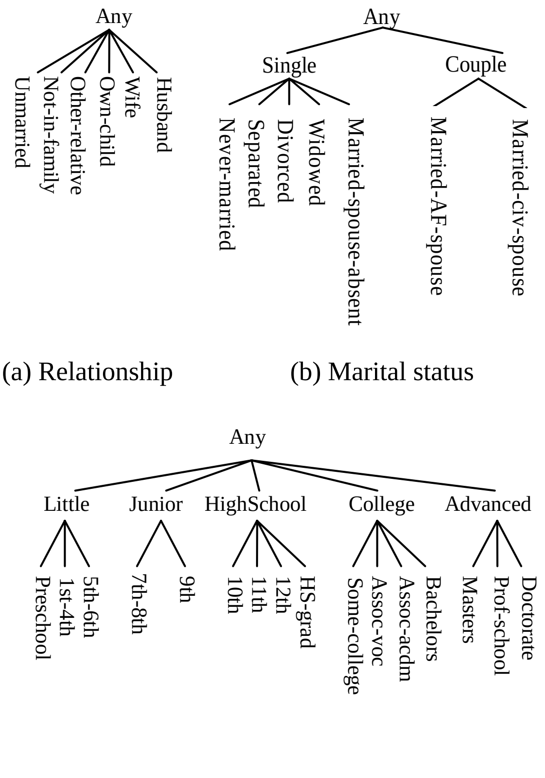
(a) SASOM (b) ESASOM

Figure 6: Visualised training results.

4.3 Real Mixed Dataset

The real mixed dataset Adult from the UCI repository [13] was used, which has 14 feature attributes and one class attribute Salary. An attribute selection technique based on information gain [14] was applied to judge the correlation between feature attributes and class attribute. Seven feature attributes were chosen including three categorical attributes (relationship, marital-status and education) and four numerical attributes (capital-gain, capital-loss, age and hours-per-week). We randomly sampled 10,000 tuples and divided it to two sets of 6,666 and 3,334 tuples as training and testing data, respectively. Three distance hierarchies are built to represent the ontological relationship of categorical data, as shown in figure 7.

As the results shown in figure 8, the stability of classification accuracy for ESASOM is better than that of SASOM in a variety of training iterations. Besides, as shown in figure 9 SASOM does not reflect the cluster structure of the data since the neuron size does not indicate the data size in the neuron. In contrast, ESASOM offers more information on cluster structure. Furthermore, the number and distribution ratio of each class in a neuron can be clearly identified through the pie chart on the map or via a pop window, which help users to acquire more detailed information within a neuron (shown as figure 9b).



(a) Relationship (b) Marital status

(c) Education

Figure 7: Conceptual distance hierarchies for three categorical attributes of Adult dataset.

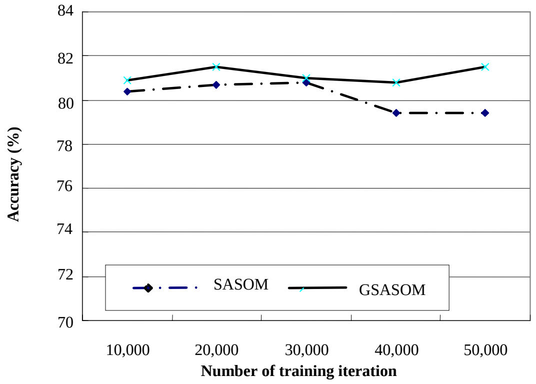
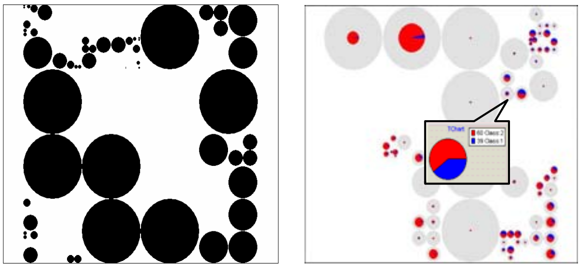


Figure 8: Accuracy in various training iterations.



(a) SASOM (b) ESASOM

Figure 9: Visualised training results of Adult dataset.

5 Conclusions

An extended SASOM integrates the ideas from SASOM and GSOM and improve those conventional models as follows. (i) It can directly cluster and classify mixed categorical and numeric data; (ii) The different extent of similarity between categorical values is considered to better reflect the natural structure in data on the trained map; (iii) It also improves classification accuracy due to the proper handling of categorical data; (iv) The visualisation of a trained ESASOM is improved to help users to get more insights about the projected data. In particular, the size of data projected in a neuron and the level to which a neuron has been split can be represented by dark dots and their shadow surroundings. And the dark dots are represented by pie charts to reveal the diversity degree of the data projected to the neuron. For the future work, we are considering how to automatically cluster the neurons to which the training data are projected on a map.

6 References

[1] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, 78(9), pp 1464-1480 (1990).

[2] A. Saalbach, G. Heidemann and H. Ritter, "Parametrized SOMs for object recognition and pose estimation", *Proceedings International Conference on Artificial Neural Networks* (ICANN 2002), Madrid, pp 902-907 (2002).

[3] K. Ishii, S. Nishida, and T. Ura, "A self-organizing map based navigation system for an underwater robot," *Proceedings IEEE International Conference of Robotics and Automation (ICRA2004)*, New Orleans, pp 4466-4471 (2004).

[4] S.B. Cho, "Structure-adaptive SOM to classify 3-dimensional point light actors' gender," *Proceedings 9th international Conference on Neural Information Processing (ICONIP2002)*, Singapore, 2, pp 949-953 (2002).

[5] C.C. Hsu., "Generalizing self-organizing map for categorical data," *IEEE Transactions on Neural Networks*, 17(2), pp 294-304 (2006).

[6] B. Fritzke, "Growing grid: a self-organizing network with constant neighbourhood range and adaptation strength," *Neural Process. Letters*, 2, pp 9-13 (1995).

[7] J. Blackmore and R. Miikkulainen, "Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map," *Proceedings International Conference Neural Networks (ICNN1993)*, San Francisco, 1, pp 450-455 (1993).

[8] D. Alahakoon, S.K. Halgamuge, and B. Srinivasan, "Dynamic self-organizing maps with controlled growth for knowledge discovery," *IEEE Transactions on Neural Networks*, 11, pp 601-614 (2000).

[9] B. Fritzke, "Unsupervised clustering with growing cell structures," *Proceedings IEEE International Joint Conference on Neural Networks (IJCNN'91)*, Singapore, 2, pp 531-536 (1991).

[10] V.J. Hodge and J. Austin, "Hierarchical growing cell structures: tree GCS," *IEEE Transaction on Knowledge and Data Engineering*, 13(2), pp 207-218 (2001).

[11] A. Rauber, D. Merkl and M. Dittenbach, "The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data," *IEEE Transactions on Neural Networks*, 13(6), pp 1331-1341 (2002).

[12] D. Barbara, J. Couto and Y. Li, "COOLCAT: an entropy-based algorithm for categorical clustering," *Proceedings 11th International Conference on Information and Knowledge Management (CKIM2002)*, McLean, VA, pp 582-589 (2002).

[13] D.J. Newman, S. Hettich, C.L. Blake and C.J. Merz, "UCI repository of machine learning database," http://www.ics.uci.edu/~mlearn/MLRepository.html, visited on 5/20/2006.

[14] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann Press, San Francisco (2001).

A Time-Based Indexing Scheme for Continuous Data Streams

Minsuk Lee and Jinsuk Song
Department of Computer Engineering
Hansung University, Seoul, Korea
mslee@hansung.ac.kr, kopuk@hansung.ac.kr

Abstract

Many embedded autonomous systems generate and store data streams continuously before eventually processing the stored data at a later time. When the data stream is stored in variable bandwidth, additional CPU processing is inevitably involved in locating the exact data positions corresponding to specific times. In this paper, a new indexing scheme for continuous data streams is proposed. This time-based indexing scheme periodically stores the relations between the time and data locations. With this index information, applications can pinpoint the exact data location corresponding to a given time. To prove the usefulness and effectiveness of the proposed indexing scheme in this study, the scheme is embedded in a newly designed file system on a Linux-based DVR and then compared the performance with that of the *ext2* file system.

Keywords: continuous data stream, indexing, file system, embedded systems

1 Introduction

In many manned or unmanned systems such as sensor networks, security cameras, robots in investigations, and even the systems in large-scale space missions, embedded systems generate continuous data streams. While some of these process the generated data streams immediately and then store the results in properly named files, many others, due to several reasons, store all the generated data streams as is or in compressed forms before playing or processing the stored streams later. Some of the reasons are (1) the nature of the application (for any archiving systems to simply store everything online or on-air), (2) keeping the on-board CPU processing requirement and power consumption as low as possible (for systems running with batteries), and (3) not needing to process the data streams on the fly (for systems where the importance of some part of the stored data stream is determined later, as in video surveillance systems, black boxes of airplanes, and message logging systems). Moreover, recent technological advancements in flash memory, optical, and magnetic storage media lead more and more systems to store all the generated data streams.

One typical example of a system handling continuous data streams is the DVR (digital video recorder) which is widely used in video surveillance systems. In peaceful times, a DVR continuously encodes video and/or audio channels and stores the encoded data streams on local or networked storages. Some temporal parts of stored data stream might become very important later. Should the need arise, the user would then examine the stored video by locating the data position corresponding to the time of importance, by playing the video in various speeds and by pushing the “FF” and “REW” buttons to skip frames. To enable the user to perform these operations, the application has to move the stream access position to any frame in the stored video. When the video is compressed in a variable bit rate form, the application would have no way of pinpointing the exact frame starting position that corresponds to a given time. This would prompt the application to jump to a guessed position (possibly with some heuristics) and scan the stream to find a frame delimiter. This procedure inevitably causes some delay in response time and the consumption of more power as well.

In this paper, a new indexing scheme is proposed which is called “the time-based indexing scheme” as it periodically stores the relation between the data position and time. With the scheme, the multimedia application in the previous example can locate a frame start position with *time* as a parameter. This eliminates the scanning and minimises the processing and bandwidth requirements, thereby minimising power consumption in the process.

All the systems processing continuous data streams can utilise the proposed time-based indexing scheme, and the multimedia systems which store compressed multimedia data streams would be the biggest beneficiaries of this scheme. Some of these multimedia systems include media players, VOD servers, non-linear video editors, DVRs, PVRs (personal video recorders), PMPs (portable media players), digital camcorders, and even mobile phones with cameras.

This paper is organised as follows. In section 2, the time-based indexing scheme and the APIs which have been defined are further described. In section 3, the application of the proposed scheme to a Linux-based DVR is explained in which a new file system based on the time-based indexing scheme is developed and then compared the performance with that of the *ext2* file system. section 4 concludes the paper.

2 The Time-Based Indexing Scheme

2.1 Time-Based Indexing

The main idea of the time-based indexing scheme is to maintain separate index information which contains the positions in the data stream that match the periodic times. Figure 1 shows the data stream and the corresponding index information.

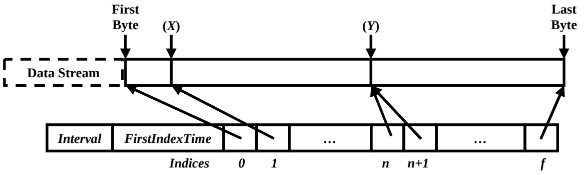


Figure 1: Time-based index information.

As shown in figure 1, the index information has the periodic pointers to the data stream. The index information consists of the *Interval*, the *FirstIndexTime*, and a series of periodic indices. Each index points to a corresponding location in the data stream. In actual implementation, the index information can be maintained in a separate file or as another data stream.

If the indices to 4-byte aligned positions are maintained, any position in a data stream can be pointed to with a total size of 16 GB (*232 * 4* bytes). Sixteen GB can accommodate about two hours of high-definition digital television data stream[1]. For production-level multimedia applications or any other application that stores enormously large data streams, the index with a size of more than 4 bytes can be used depending on the maximum size of the data stream and the maximum size of the storage medium in use.

Applications decide the value of the *Interval* as the temporal interval between any two indices. Each data stream can have its own interval, or all the data streams in a system can share the same interval. For video applications, the *Interval* of *1/framerate* seems to be natural. In MPEG stream, where a GOP (group of pictures) usually contains 5 to 60 frames [1][2], the *Interval* between 165 msec and 1980 msec can be selected. In the DVR system which was used for prototype implementation in section 3, 15 seconds as the indexing interval and 8 bytes as the size of each index were chosen. The 8-byte index is quite enough to specify an absolute location in a data stream on a hard disk with a capacity of several hundreds GB.

The *FirstIndexTime* is the real time when the first index was written. For most of the real-time data streams, the starting time is of importance, but the field has no meaning for some data streams, such as movie files where the *FirstIndexTime* is *NULL*.

In figure 1, index *0* points to the initial position of the data stream, and after the *Interval*, index *1* points to the position *X*. There could be no data to store during several indexing periods[2]. The index *n* and the index *n+1* indicate the situation where the two indices point to the same position *Y*. The last index, index *f*, points to the end of the data stream.

2.2 Applying the Time-Based Indexing to File Systems

The time-based indexing scheme can be applied to any existing file systems. Figure 2 describes the software architecture of the time-based indexing scheme on a UNIX-like operating system. The indexing daemon writes the periodic indices, i.e., the current write positions at the beginning of the indexing intervals into separate index files. As a kernel thread, the indexing daemon can collect current write positions by observing the *write()* system call. If enough buffers are maintained to hold indices, the indexing scheme can be implemented without any real-time operating system support.

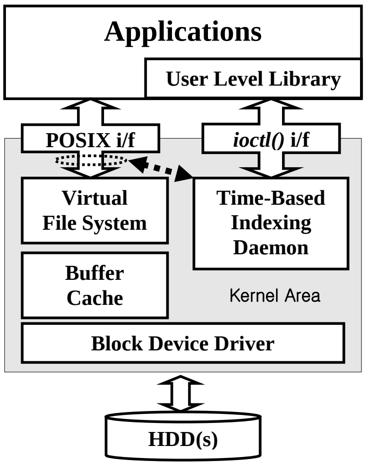


Figure 2: Time-based indexing on a UNIX-like operating system.

The index information itself is also stored as a normal file in the conventional file system[3]. The user-level library over the *ioctl()* interface is the set of API functions utilising the index information and controlling the indexing scheme.

If an MPEG application calls *write()* with a buffer, which is unaligned with an I-frame or a GOP boundary, the write position observed and written to the index file can be useless. Therefore, in using the

[1] According to the ATSC standard[3], the digital television system requires less than or equal to 19.4 Mbps for a channel of HD-quality terrestrial broadcasting, or 38.8 Mbps for high data rate mode. However, some countries have their own regulations which limit the HD bandwidth to less than 17 Mbps.

[2] As in DVR, to save storage medium, nothing is stored while there is no movement detected on video input.

[3] The index file could be hidden to users by changing the file attributes or using a special file name starting with '.' as in UNIX.

time-based indexing scheme, the applications should call *write()* with a meaningfully aligned buffer.

When there are some difficulties in implementing the indexing daemon in an operating system's kernel space, everything can be done in the user level. In extreme cases, the application can build the index information after the entire data stream is stored.

2.3 The APIs

To utilise the time-based indexing scheme, a set of APIs is defined as follows:

The ***start_indexing(int write_fd, int interval)*** triggers the writing index every given interval. If there is no index file corresponding to *write_fd* yet, this function creates a new index information file at the same directory where the data stream file exists, and writes *Interval* and *FirstIndexTime* to the index file. When there is an existing index file for *write_fd*, the *interval* is ignored, and the indices will be appended to the existing index file. If there is a time gap between the last indexing time and the current time, the indices in the gap are filled with the last data stream position.

The ***stop_indexing(int write_fd)*** stops the indexing daemon from writing index for this file.

The ***check_indexing(int fd)*** checks if the index-writing is on for the writing *fd*, or if the index information file exists for the reading *fd*.

The ***stat_indexing(int fd, struct index_info_t *index_info)*** returns the indexing *Interval* and *FirstIndexTime* in the *index_info* structure, which are the first two fields in the index information file.

The ***locate_stream(int read_fd, struct timeval *time)*** moves the file access pointer to the position to which the index at the given *time* is pointing. This API is one of the key functions of the time-based indexing scheme. It *lseek()* the index information file with the value of *(time - FirstIndexTime) / Interval * sizeof (an Index)* and *read()* the index (i.e., the position in the data stream) from the index file. And then it *lseek()* the data stream file with the index.

3 Implementation Example: DVR

In this study, the time-based indexing scheme on a new file system for a Linux-based DVR is implemented. When it plays back the stored stream, the time-based indexing scheme makes everything simple. With the *locate_stream()* API, the application can locate the exact starting position of the GOP corresponding to the time given by the user and can play the video without any delay. It does not need any scan operation to find the GOP boundaries in the stream in implementing "search by time" and in playing the stream at high speeds such as 10x, 100x.

3.1 The File System for DVR

Though the proposed time-based indexing scheme can be applied to any file system, a new file system to improve the performance, reliability, and maintainability of DVR is designed and implemented.

The basic assumption of most existing file systems is that the file system will be used for server systems which have millions of small-size files on sets of disks, hundreds of online users, and thousands of open files. This assumption leads file system designers to invent more complex indexing schemes (i.e., the meta data structures) and more aggressive buffering schemes to improve performance by utilising the huge amount of main memory. The results are the file systems which can be found in contemporary operating systems, such as *ext2* [4], *ext3* [5], *XFS* [6], *JFS* [7], *NTFS* [8], *reiserfs* [9], and so on. These file systems are too complex to be maintained effectively in terms of engineering cost and have large static/dynamic memory footprints to reside in memory-hungry embedded devices.

In designing a new file system for DVR, maintainability, reliability, performance, and portability are the most important design goals.

We designed a file system with a very simple disk layout to achieve maintainability, which enables the implementation of the file system with less than 10K lines of C code including the utilities (i.e. *mkfs*, *fsck*). It can initialise a 200 GB hard disk in less than two seconds. For many embedded devices, one of the key requirements is reliability. Especially in the surveillance industry, the reliability of a file system means 100% recovery from unexpected power failures. To keep the file system recoverable, the designed file system uses a small amount of non-volatile memory to keep last-minutes critical meta data, instead of maintaining journals which degrade the performance. The file system uses only 1 MB of data buffer, and all the data are passed to the physical disk with minimum delay. Moreover, it immediately writes all the updates in meta data and the indexing information without any delay. For performance, it uses a very large unit of data blocks (multiples of 1 MB). Since many DVRs in the market are equipped with hard disks with capacities bigger than 100 GB, the fragmentation problem which might be caused by the large unit of data block is negligible.

Finally, the file system is designed to be independent from the VFS (Virtual File System) for portability, which is common in modern operating systems. Though most embedded multimedia devices use well-layered operating systems such as Linux [10] or on an embedded form of Microsoft's Windows [11], there is still a high probability that our target systems would be small devices running RTOSes with limited functionalities or even devices without any operating system. In actuality, the designed file system was ported to pSOS [12], a well known real-time operating system, as well as Linux and Microsoft's Windows operating system.

Regarding the indexing scheme, the DVR file system maintains a system-wide index information rather than a stream by stream index file. Figure 3 shows the index information. In the middle of the figure, there is an index information stream which contains the periodic indices (a vertical column in the index information) for all other data streams, where *n* is the number of data streams in the system. In the DVR file system, the index interval was set to 15 seconds, and the stream positions were pointed by 8 byte-structures holding a data block number and an offset in the data block.

To improve the searching functionality and the performance of the DVR, additional information is periodically written in a separate file, which consists of the flags indicating data existences for channels and the bit-wise sensor inputs of the DVR during the index interval.

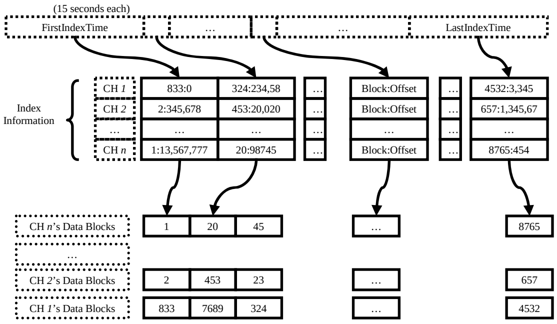


Figure 3: The index information for DVR.

3.2 APIs for DVR Implementation

Since system-wide index information in DVR was used as in figure 3, there is no need for the parameter *fd* or *write_fd* in *start_indexing(), stop_indexing(), check_indexing(),* and *stat_indexing()*. In addition, system-wide *FirstIndexTime* and *LastIndexTime* refer to the time when *start_indexing()* and *stop_indexing()* are called for the first and last times, respectively. In DVR application, the epoch UTC time[4] (*time_t*) was used rather than relative time or *struct timeval* because the device is running based on the real time clock, and the indexing interval is in the second order rather than in the micro second order.

To store continuously generated multimedia stream in limited hard disk space, one more API function for DVR-specific application is added:

The ***recycle(time_t time)*** recycles the data area written before the given time to make room for future data. This function moves the start positions of all data stream file to the positions pointed by the indices corresponding to the time. It does the same thing to the index information itself. After this function returns, all the data blocks used before the time are invalidated, and the occupied disk space becomes available to new data. This function makes the whole disk a set of big circular buffers.

3.3 The Performance Test

To verify the effectiveness of the time-based indexing scheme together with the performance of the file system, its performance is compared with that of the *ext2* file system. The experiments were done on a commercial Linux DVR.

Table 1 shows the experimental setup. Just 1 MB out of the main memory was allocated as the buffer area for the implemented DVR file system, while the *ext2* file system used all the unallocated main memory space as the buffer cache. The *ext2* file system was prepared by the '*mkfs.ext2*' command with default arguments, while the new file system was initialised to have 14 data files with 16 MB unit data block, with an indexing interval of 15 seconds.

Table 1: Experimental setup.

CPU	SOC based on PowerPC (running 166 MHz)
Main memory	32 MB
OS	Linux 2.4.2
HDD	Capacity: 40 GB Interface: UDMA100 RPM: 7200 Avg. seek time: 8.5 msec

The experiments were to measure the elapsed times for four or eight thread mixes of sequential/random reads with sequential writes. Each thread reads or writes 1,875 MB (*128 KB * 15,360*) of data from different files. During the test, the video/audio encoding and decoding processes are blocked from running in order to observe only the performance of the file system and the indexing scheme. In the *ext2* experiments, the writing threads generated index information files as described in section 2.1, and reading threads use the information files to access the data streams randomly (in the sense of time). In the experiments on the new file system, the same indexing operation is performed by the time-based indexing daemon in kernel, and the random read uses the *locate_stream()* API.

Figures 4 and 5 show the results. In the figures, the Y-axis refers to the elapsed time, in seconds, for all the threads to finish the read/write. The times from all 10 experiments are averaged. The X-axis is the ratio of the number of reading threads to the number of writing threads.

In figure 4, the new file system shows superior performance in terms of sequential access. It

[4] The time as seconds elapsed since midnight, January 1, 1970.

outperforms the *ext2* by a maximum of 38.7% (when the ration is 7:1) and an average of 32.5%. This is mainly due to the large unit of data block in our file system.

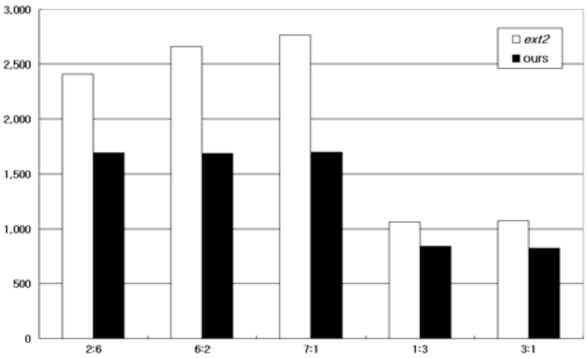


Figure 4: Sequential access performance.

It can be found that the elapsed time of the *ext2* increases when the number of reading threads increases. This is because the *ext2* relies on the buffer cache for its read-ahead scheme. With a limited amount of main memory as in the DVR, when the number of reading threads increases, the contention for the main memory also increases. However, the new file system shows almost the same elapsed times.

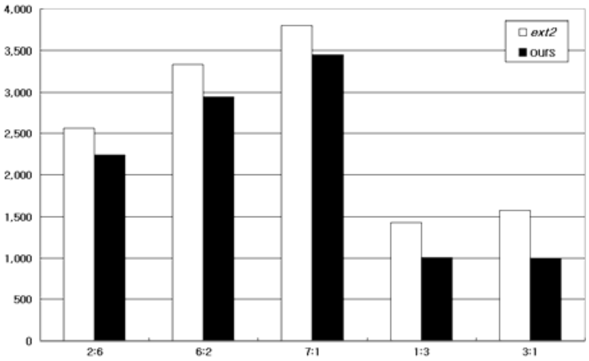


Figure 5: Random access performance.

Figure 5 shows the mixed performance of random reads with sequential writes. In this experiment, a read thread jumps to a random (in time, not in location) position in the data stream file every time before calling *read()*. In the *ext2* file system, this random access is done with an *lseek()* in the data stream with the location acquired from the index information file, again by *lseek()* with a random time as a parameter. In the new file system, however, we *read()* the data stream after the *locate_stream()* to locate the file read position corresponding to a random time. In both file systems, the more random read threads there are, the more movements of disk head can be expected.

The new file system shows better performance by a maximum of 36.6% (when the ratio is 3:1) and an average of 16.2%. Again, this is mainly due to the contiguousness of data because the time to access data dominated the total elapsed time, and the performance gains for the ratio 2:6, 6:2, and 7:1 are 12.3%, 11.7%, and 9.2%, respectively. Though the gains are still positive for all cases, these are decreasing due to the index information file size. The size of generated index information in the *ext2* experiments was only about 8 KB which can fit nicely in a Linux buffer cache. This means that there will be no additional movements of disk head to do the *lseek()* after the kernel loads the whole index information files on the buffer caches at the first *lseek()* and *read()*. In the file system where only 1 MB is used as the buffer space, every *locate_stream()* call incurs more real disk access, and the same situation happens in the writing indices. If indices are generated more frequently (rather than 15 seconds), the index information size will increase, and the *ext2* file system may not enjoy this performance advantage caused by the buffer cache mechanism of Linux.

4 Conclusions

In this study, the time-based indexing scheme was proposed, which simplifies random access, in the sense of time and not location, to the stored continuous data stream. To utilise the proposed indexing scheme, a set of API functions was also defined. The indexing scheme stores the locations of data streams periodically. By maintaining the relation between time and the corresponding data location, the indexing scheme reduces CPU processing requirements and power consumption in most of the multimedia systems, as well as those that generate and store data streams continuously and process the stored data later.

The proposed indexing scheme was embedded in a new file system on a DVR which stores and plays the compressed video/audio streams. The file system was optimised to store sequential data streams as it searches and plays the multimedia data streams using the time-based indexing scheme. The performance of the implemented file system was compared with that of the *ext2* file system in which the former outperformed *ext2* in the experiments with sequential access by a maximum of 38.7%. In the experiments of random read and sequential write mix, the new file system's performance gain reached a maximum of 36.6% and an average of 16.2%.

5 References

[1] C. Fogg, D.J. LeGall, J.L. Mitchell and W.B. Pennebaker, *MPEG Video Compression Standard*, Springer (1996).

[2] T. Ebrahimi and F. Pereira, *The MPEG-4 Book* (1st edition), Prentice Hall PTR (2002).

[3] Advanced Television Systems Committee, *ATSC Digital Television Standard (A/53)*, Revision E with Amendment No. 1 (2006).

[4] R. Card, T. Ts'o and S. Tweedie, "Design and implementation of the second extended filesystem", *Proceedings 1st Dutch International Symposium on Linux*, Amsterdam, (1994).

[5] S. Tweedie, “EXT3 journaling filesystem”, *Proceedings Ottawa Linux Symposium*, Ottawa, (2000).

[6] SGI, “XFS: A high performance journaling fileystsem”, http://oss.sgi.com/projects/xfs, visited on 13/7/2006.

[7] JFS Core Team, “Journaled file system technology for Linux”, http://jfs.sourceforge.net/, visited on 13/7/2006.

[8] H. Custer, *Inside the Windows NT File System*, Microsoft Press (1994).

[9] R. Bryant, R. Forester and J. Hawkes, “Filesystem performance and scalability in Linux 2.4.17”, *Proceedings USENIX 2000 Annual Technical Conference (USENIX 2000)*, San Diego, pp 259-274 (2002).

[10] Linux Kernel Organization, Inc., “Linux Kernel Archives”, http://www.kernel.org, visited on 13/7/2006.

[11] Microsoft, “Microsoft Windows Embedded Developer Center”, http://msdn.microsoft.com/embedded/, visited on 13/7/2006.

[12] Wind River, “pSOSystem”, http://www.windriver.com/, visited on 13/7/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

A Conscious Robot with Emotions

Yusuke Shirakura, Tohru Suzuki, Junichi Takeno
Robot and Science Laboratory, Department of Science and Technology
Meiji University, Kawasaki, Japan
weises_lagerhaus@yahoo.co.jp, t_suzuki@cs.meiji.ac.jp, takeno@cs.meiji.ac.jp

Abstract

The present paper reports on the study of a robot with functions similar to human-like consciousness and emotions. The robot is designed to repeat cognition and behaviours using the consciousness system that basically consists of a combination of two neural networks of reason and feelings systems. The authors discuss how to avoid deadlocks experienced by the reason system due to unknown obstacles when using the feelings system. The consciousness system activates the feelings system upon finding an obstacle encountered by the reason system and cognises discomfort caused by the obstacle. The consciousness system then plans and executes a scheme to change the discomfort to comfort to negotiate the obstacle. When robots possessing the conscious system can imitate others, the robots can realise self-cognition. The consciousness system proposed by the authors can be the base for developing a robot possessing human-like consciousness and feelings.

Keywords: consciousness, emotions, neural network, self-cognition, robotics, imitation behaviour

1 Introduction

Consciousness has been studied extensively in brain science, neural science, psychology, philosophy, other academic fields and recently in robotics. Various robots, including industrial robots, that can communicate with consciousness and feeling functions have been reported by many researchers and companies.

The Actroid robot is one of them [1]. But it cannot talk to others cleverly because it lacks human-like consciousness and feelings.

The WE-4R II robot of Waseda University reportedly has the functions for feelings [2]. Emotional equation is used in this robot system. This robot is different from the robot proposed by the authors which expresses emotion using artificial neural models of the consciousness system. Moreover, the authors report not only emotion but also the function of consciousness.

Both human consciousness and emotion are being actively studied and interesting research has been reported in various fields. Nevertheless, no paper has ever elucidated the relationship between emotions and consciousness. The present paper proposes a computational model capable of realising the function of emotion and reports on the experiments from which robots learn the imitation behaviour. The authors think emotions can be cognised by imitating others behaviour. This will be able to cause advanced self-cognition and other-cognition. The relationship between consciousness and imitation behaviours, self-cognition, and other-cognition is written in the next chapter.

2 Consciousness

What is consciousness? Consciousness is generally considered to be the state when one is paying attention to something, is thinking or is awake.

With a focus on the imitation function of humans, the authors define consciousness to arise from “a consistency of cognition and behaviour.” Based on this belief, the authors have devised a module for the consciousness system named Module of Nerves for Advanced Dynamics (MoNAD). In addition, by the MoNAD, most of Husserl’s ten functions of consciousness can be accounted for [3].

The authors have found a strong relationship between consciousness and imitation based on four important instances of consciousness [3]: mirror neuron [4], mimesis theory [5], medical cases of imitation behaviour [6] and a study of imitation behaviour in neonates by Meltzoff, Moore et al. [7][8].

The immediate objective of the present study is the realisation of self-consciousness. As the first step in the study, the authors attempted to realise the function of imitation and verify whether self-cognition could be realised based on feedback information regarding the condition of self and the other [9]. An implementation of self-consciousness must need self-cognition.

First, other and self conditions are represented by imitating others behaviours with the consciousness system. Next, the representations of self and others are compared. If other representation is similar to self representation, the other is more-self-like.

3 Emotions

During the imitation experiments for the consciousness system, the authors took a great interest in how an autonomous robot behaved when it confronted difficulty caused by harmful obstacles. This theme has a close relationship with emotion.

Emotion is generally said to have the following three features:

1. Emotion is evoked by internal information representing a change in the condition of the body or by external information. For example, one feels pain upon hitting something, or feels bad when one’s stomach is upset.
2. Emotion plays the role of an adjuster in the body. This is particularly important for understanding the homeostasis function [10].
3. Emotion helps people to reason and make choices.

Based on these observations, with regard to feelings of discomfort, the authors have proposed a new hypothesis in which emotions are generated by internal or external information to the body; they make people pay attention to the cause of the discomfort; and assist the person to avoid the discomfort-causing behaviour, eventually enabling the person to avoid harmful damage using this assistance. (see the observations of Damasio [11]).

By implementation of the emotional function, self-cognition and other-cognition will be able to make further progress. This will enable robots to possess human-like sociality.

4 Outline of the Consciousness System

4.1 MoNAD Structure

The consciousness module "MoNAD" is a consciousness computational model using neural networks [9]. Shown in figure 1, MoNAD consists of the cognition system (a), behaviour system (b), primary representation (c) which is common area of cognition and behaviour systems, and symbolic representation (d), and input/output units (A/B).

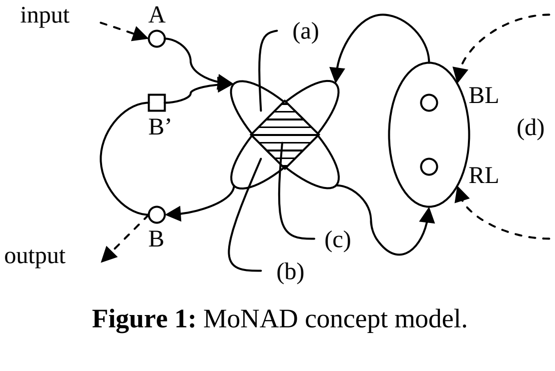


Figure 1: MoNAD concept model.

Symbolic representation has cognition representation RL, which represents what MoNAD cognises now, and behaviour representation BL, which represents symbol how MoNAD will behave next. These two representations serve as the communication terminals for the higher-level modules. Without information transmitted from the higher-level module, the information of cognition representation RL is basically copied to behaviour representation BL.

The most important feature of MoNAD is primary representation, the common area for the cognition and the behaviour system. Primary representation enables the system to learn behaviour while it cognises and, conversely, to learn cognition while the system behaves. It is also possible to realise artificial thought and expectation using MoNAD [3]. Because the feedback B’ as somatic sensation is in this system, therefore, self-condition can be grasped more.

4.2 General Concept of the System

Figure 2 shows a conceptual model of the consciousness system proposed by the authors.

An important feature of the system is that it consists of multiple MoNADs. The major components of the consciousness system are the reason, feelings and association system.

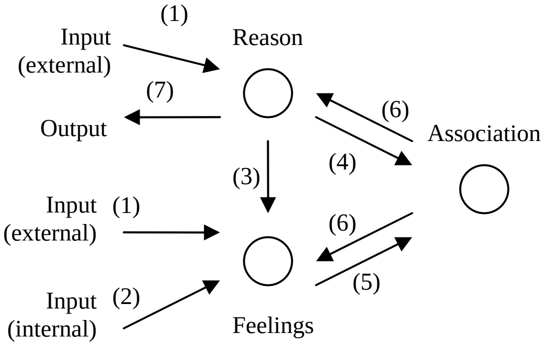


Figure 2: Consciousness system concept model.

The reason system cognises the external environment based on information received from the input unit (figure 2 (1)). The output unit performs the decided behaviour. The reason system has a hierarchical structure of MoNADs.

The feelings system represents feelings based on information from the external environment and the internal environment reflecting bodily changes (figure 2 (1), (2)). The feelings system is also a hierarchical structure of MoNADs. The highest layer of the feelings system holds two modules corresponding to comfort and discomfort. Another important feature of the consciousness system is that information from the reason system is also used in cognising comfort or discomfort. Information from the reason system (cognised language label) is input into the comfort and discomfort modules (figure 2 (3)).

The reason system and the feelings system exchange information between higher and lower levels using their hierarchical structures.

The association system receives two inputs: information cognised in the reason system and the condition of self as understood by the feelings system (figure 2 (4), (5)). Based on these two pieces of information, the association system determines the behaviour that will make self comfortable. To reflect the decision on the behaviour of the robot, the association system outputs information to the reason system and the feelings system (figure 2 (6)). This functioning of the association system modifies the representation of both the reason and feelings systems and, eventually, the reason system outputs the command to perform a certain behaviour that makes the robot comfortable (figure 2 (7)).

At this time, the association system is not a so-called homunculus because the MoNAD of the association system is driven by information from the lower-level MoNADs.

5 Outline of the Consciousness System

This section describes the learning functions of each of the reason, feelings and association systems.

Fundamental construction of the MoNAD is shown in figure 3. As shown in figures 4, 5 and 6, the MoNAD is applied to the implementation of the emotional and associating function. Figure 3 also shows the network in the reason system to imitate for representing self and other condition.

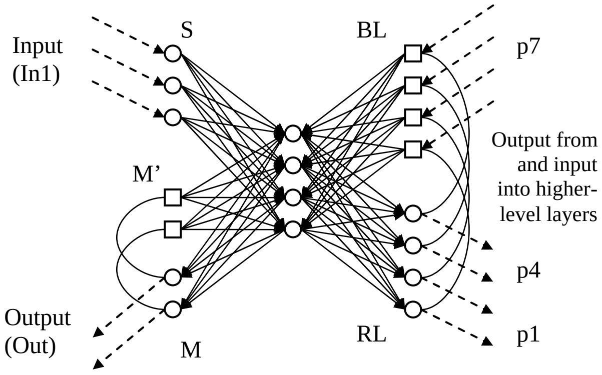


Figure 3: MoNAD computational model and imitation behaviour module.

Figure 4 shows the lower-level networks in the feelings system to represent feeling of pain from internal information. Figure 5 shows the higher-level network in the feelings system to represent feeling of comfort from the lower-level emotional representation. Figure 6 shows the network in the association system to give support to decide next action from the comfortable and uncomfortable representation, the representation of self-cognition and other-cognition. The authors are just describing the differences of the network form.

The networks in the feelings system don't have output unit M because they are used mainly to represent self condition. But an information flow circulates from primary representation to symbolic representation, and it can change emotional condition and the representation. This change will generate emotional thinking.

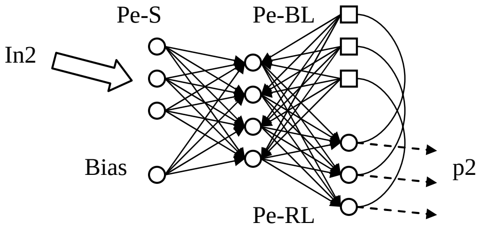


Figure 4: Pain module of consciousness system.

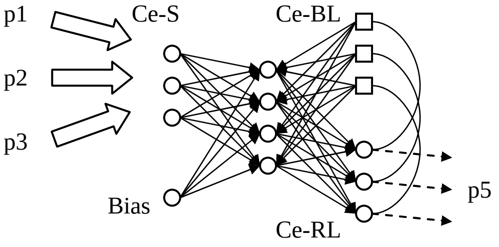


Figure 5: Comfortable module of consciousness system.

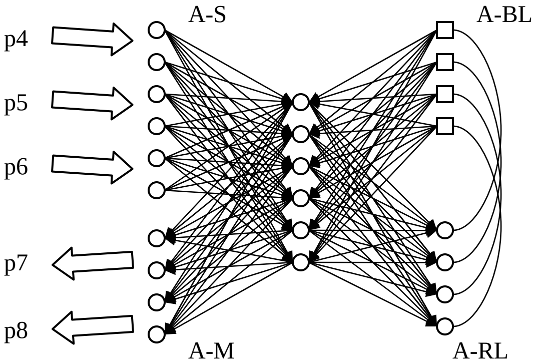


Figure 6: The association module of the consciousness system.

The association network doesn't have feedback, M'. The reason is that the association network is the higher-level MoNAD and doesn't have motors as somatic sensation. This network also has circulation, so is capable of the implementation of thought.

Parts of networks in the feelings system are omitted here for reasons of space.

5.1 Consciousness System for Self-Cognition

5.1.1 Overall Image of the System

Figure 7 shows an overall image of the consciousness system developed for conducting experiments on mirror image cognition. Module A, the reason system, imitates behaviour based on external information from the infrared (IR) sensor and represents self-condition and other-condition. Modules B through E make up the feelings system: B represents "pain" based on internal information or an error detected by PID control; C represents "solitude" from the value of the IR sensor; D and E receive the values of representation from modules A, B and C and represent the emotions of "comfort" and "discomfort", respectively. Module F is the association system.

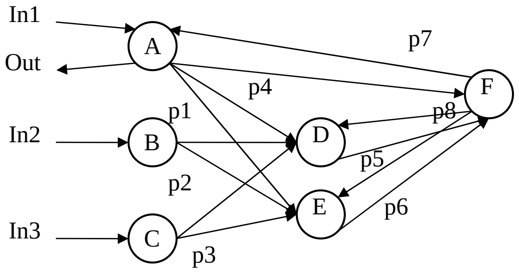


Figure 7: Consciousness system for self-cognition.

Three behaviours of advance, stop and retreat are used in the robotic experiments. Just one module A is used in the reason system but two or more modules may be used for robots requiring more complex motions. The robot used in the experiments is Khepera2. These robots have infrared sensors and motors of PID control.

5.1.2 Sensation and Emotion of the Robot

The robot is assumed to have two sensations of pain and solitude in the experiment. The reason why the error detected by PID control gives pain to the robot and why the sensor condition generates solitude is described below.

The authors focused on the fact that emotion plays a role of adjuster in the body. When no error is detected by PID control, the robot moves at the set speed. When the error detected by PID control increases, some fault such as friction is present and the actual speed decreases to below the set speed. This increases the load on the motor, and the robot thinks the speed must be adjusted, which gives pain to the robot. The present consciousness system is designed to imitate behaviours and learn self-cognition using emotions. If the sensor does not react, the robot cannot learn, making self-cognition impossible. If no information arrives at the sensor, discomfort results causing solitude. Pain and solitude prevent the robots from imitating for self-cognition, so which is discomfort.

5.1.3 Flow of Information through the Consciousness System

This section describes the flow of information through the consciousness system that performs self-cognition (figure 7).

The value of the infrared sensor for capturing external information is input into terminals In1 and In3. Any error detected by PID control representing internal information is input into terminal In2. In the symbolic representation of imitation module A, behaviour of self and the other is cognised from the external information input into In1 and from the somatic sensation of A. Using information from In2 and In3, the degree of pain and solitude is matched to one of four stages of language labels at the symbolic representations B and C, respectively.

As a result of the cognition of behaviour of self and the other at A, the degree of pain and solitude determined at B and C are respectively input as comfort and discomfort into modules D and E via p1, p2 and p3. Similarly, in the comfort and discomfort modules, the degree of comfort and discomfort is matched to one of four stages of language labels at the symbolic representation.

The result of the cognition by the module of imitation at A and the degree of comfort and discomfort respectively determined at D and E are eventually input into module F via p4, p5 and p6 to associate information from the reason system and the feelings system. The association system module F represents at A-RL (figure 6) the behaviour of self and whether self is comfortable or not at the present moment as determined by the result of the cognition by the imitation module and the degree of comfort and discomfort determined. This information from cognition representation A-RL is copied to the behaviour representation A-BL (figure 6).

As a result, the association system module F transmits information to module A via p7, enabling it to change the "expectation of behaviour" generated in the cognition-by-imitation process of module A. This means that the reason system is dictated to prepare the next behaviour according to the condition of the feelings system of self at the present moment.

If A-RL of the module F represents discomfort, module F transmits to the lower-level module information by which the lower-level reason module will invite comfort as a result of behaviour. If A-RL of module F represents comfort, module F transmits to the lower-level module information by which the lower-level reason module will allow comfort to continue as a result of behaviour. Information determined by the association system module F is transmitted to the feelings systems D and E via p8. This information affects the feedback on comfort and discomfort. The authors think this feedback can cause a change in emotional thinking without input.

Assume the result of the cognition by imitation at the reason system is advance for both self and the other and the robot is imitating the specified behaviour smoothly. If at this time the degree of discomfort determined by the feelings system is high, the consciousness system judges that advance is a discomfort. In response, the association system determines a new step and instructs the robot to retreat in the next behaviour.

The timing of the above determination is described below. In the mirror image cognition experiments, the robot developed by the authors repeatedly imitated the motion of its image in the mirror. When the robot imitates the motion of its image in the mirror, and if the robot itself advances, the other or the mirror image always advances. By repeating this imitation of the advance behaviour, the robot eventually collides with the mirror. The collision causes an error detected by PID control and the consciousness module represents pain while the discomfort module E represents discomfort. The association system module F transmits representation of retreat to the reason unit. Behaviour to avoid discomfort and invite comfort is implemented to continue imitation for self-cognition.

5.2 Neural Network Learning Method

The neural network learning method used in the present study is supervised learning with the back propagation method.

Specifically, we prepared bit strings for S, M’ and B as the learning inputs for the MoNAD of the reason system (figure 3). Bit strings for S and B were prepared for the feelings systems. Here are each bit strings. Table 1 shows input in the imitation module.

Table1: Bit strings of input of the system.

Bit strings	Condition
000	Both advances
001	One advance
010	Both stop
100	One retreat
111	Both retreats

Table 2: Bit strings of robotic behaviour.

Bit strings	Condition
00	Advance
01	Stop
11	Retreat

Table 2 shows output in the imitation module. Bit strings in table 2 are used in self and other cognition, e.g. if both self and other advance, bit strings in symbolic representation are 0000. Tables 3 and 4 show representations of pain and comfort modules.

The imitation module A (figure 4) has 27 learning patterns. Each of the feelings systems B through to E has 32 and the association system has 72 patterns. In the connection from lower-layer to higher-layer in the feelings system, the trilaminar neural network is used and has 32 learning patterns. R and M at the reason and association system become the outputs when a value is entered into the neural network of the respective MoNAD (figure 3). R is the only output in the feelings systems.

Table 3: Bit strings of representation of pain.

Bit strings	Condition
000	No pain
100	A little pain
011	Pain
111	Lots of pain

Table 4: Bit strings of representation of comfort.

Bit strings	Condition
000	No comfortable
001	A little comfortable
100	Comfortable
110	Lots of comfortable

The system learns by calculating the sum of mean square errors for the number of patterns using outputs R and M. The difference with the corresponding teacher signals then sequentially reduces the determined error. Learning continues until the sum of mean square errors is reduced to below a certain value (0.01).

5.3 Result of Learning and Observation

Figure 8 shows the learning of the imitation module, the association module, the pain and solitude modules. Figure 9 shows the learning of the connections, and the discomfort and comfort modules. The x-axis of the graph is the learning number (order) and the y-axis is the mean square error (error).

The graphs in figure 8 show that learning converges to about 0.01 with about 900 orders for the imitation module, the pain module, the solitude module, and the association module. The graphs in figure 9 show that learning converges to about 0.01 with about 400 orders for the comfort and discomfort module. But in the connections to comfort and discomfort learning converges to about 3,000 and 40,000 orders. Learning in the MoNAD completed after a relatively small number of times because the processing efficiency of the MoNAD neural network is good. The results also show that each MoNAD has a function to imitate and cognise emotion. It is further shown that the functions of consciousness and emotion as defined by the authors can be realised by combining the respective MoNADs.

By structuring the consciousness system as described above, when mounted on a robot the system can imitate and cognise emotion, and if discomfort is determined, select behaviour to remove the discomfort. Self-cognition can continue for the implementation of self-consciousness.

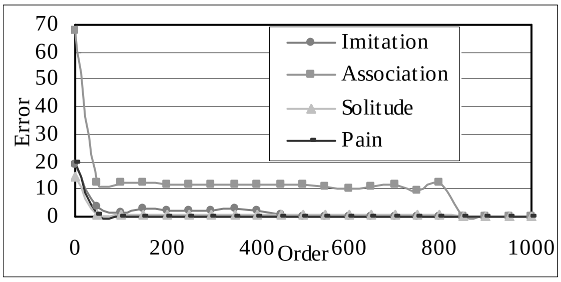


Figure 8: Results of learning experiment of imitation, association, solitude, and pain.

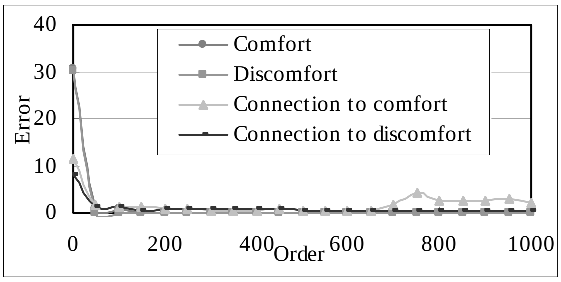


Figure 9: Results of learning experiment of comfort, discomfort, and the connections.

6 Future Study

In the learning experiments using the consciousness system developed by the authors, the robot is considered capable of avoiding harmful obstacles if the robot is given an emotion function. The authors will mount the consciousness system on a robot to verify that the robot actually evades obstacles using the emotion function and that the robot is capable of self-cognition. For robotic experiments, four objects to be imitated are prepared for. The following is four objects: a robot taking action which is advance, stop or retreat at random, a robot possessing the consciousness system (a conscious robot), a robot controlled via cables from a conscious robot, and a mirror image of a conscious robot.

A conscious robot will imitate these four objects. And then the authors will show the robot can have self-consciousness by comparing representations of self-condition to representations of other-condition in the system proposed by the authors.

7 Summary

Consciousness and emotion have been extensively researched recently to understand the mind and feelings. Systematisation of consciousness and emotions will further accelerate the development of robots and it will become possible to give consciousness and emotions to various types of robots.

The present paper reported on research into functions that are intrinsic to humans such as mirror neuron and mimesis theory and defined a new consciousness. The authors have also established a definition of the robot's emotions based on human emotions. Using this definition, the authors have developed a consciousness system and successfully taught it with emotions utilising artificial neural networks. These results in the leaning experiments will enable robots with the consciousness system developed by the authors to possess human-like sociality.

8 References

[1] Kokoro Company, “Home Page”, http://www.kokoro-dreams.co.jp, visited on 10/7/2006.

[2] H. Miwa, K. Itoh, M. Matsumoto, M. Zecca, H. Takanobu, S. Roccella, M. C. Carrozza, P. Dario and A. Takanishi, “Effective emotional expressions with emotion expression humanoid robot WE-4RII,” *Proc. 2004 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Miyagi, Japan, pp 2203-2208 (2004).

[3] T. Suzuki, K. Inaba and J. Takeno, “Conscious robot that distinguishes between self and others and implements imitation behavior,” *Proc. IEA/AIE*, Bari, Italy, pp 101-110 (2005).

[4] V. Gallese, L. Fadiga, L. Fogassi and G. Rizzolatti, “Action recognition in the premotor cortex,” *Brain*, 119, pp 593-609 (1996)

[5] M. Donald, *Origins of the Modern Mind*, Harvard University Press, Cambridge (1991).

[6] University of Maryland, “Hearing and Speech Sciences”, http://www.bsos.umd.edu/hesp/ index.htm, visited on 20/12/2004.

[7] A.N. Meltzoff and M.K. Moore, “Imitation of facial and manual gestures by human neonates,” *American Association for the Advancement of Science*, 198, pp 75-78 (1977).

[8] K. Ikegami, “Effect of extrauterine and intrauterine experiences on tongue protruding imitation in premature infants and full-term neonates,” *Annual Report of Grant-in-aide for Scientific Research, Ministry of Education, Science, Sports and Culture: The Emergence of Human Cognition and Language*, 3, pp 9-13 (1996).

[9] J. Takeno, K. Inaba and T. Suzuki, “Experiments and examination of mirror image cognition using a small robot,” *Proc. 6th IEEE Intl. Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland, pp 493-498 (2005)

[10] A.R. Damasio, *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*, Harcourt Brace, Orlando, FL (1999).

[11] A.R. Damasio, *Descartes' Error: Emotion, Reason and the Human Brain*, Quill, New York (1994).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Naive + Naive = Smart Bayes?

J.M. Martínez-Otzeta, B. Sierra, E. Lazkano, E. Jauregi and Y. Yurramendi
Department of Computer Science and Artificial Intelligence
University of the Basque Country, Donostia, Spain
ccbmaotj@si.ehu.es

Abstract

Naive Bayes is a well-known and studied algorithm both in statistics and machine learning. Bayesian learning algorithms represent each concept with a single probabilistic summary. This paper presents a novel method to construct a hierarchy of Naive Bayes classifiers, in which two levels of Naive Bayes classifiers are applied. The underlying motivation is to try to correct the mistakes made by a single Naive Bayes classifier applying Naive Bayes again over its output. We have tested this algorithm over 20 UCI datasets. The results suggest that the chance of obtaining a better performance than with the original Naive Bayes approach are much greater than the opposite.

Keywords: data mining, Naive Bayes, combination of classifiers

1 Introduction

In the supervised learning task [1], the main goal is to induce a classification model from a given training database for which we know the labelling of every sample. Existing paradigms come from the area of the Artificial Intelligence and are grouped in the family of *Machine Learning* (ML).

The goal of supervised learning is to predict the class labels of examples that have not been seen; to achieve this goal a number of different techniques have been developed, among them Bayesian models, decision trees, decision rules, neural networks or SVMs.

The Naive Bayes algorithm is one of the oldest and most well-known machine learning approaches [2]. To make the probability based classification computation feasible, Naive Bayes assume independence among the variables, in a *naive* fashion. Such assumption in real problems is certainly a simplistic one. But, despite this objection, Naive Bayes works rather well when tested on actual databases, particularly when combined with some attribute selection procedure.

On the other hand, several lines of research have dealt with the best way of combining classifiers in order to improve the accuracy of a single classifier.

There are very different manners to combine classifiers, most of which have been found to improve accuracy over single classifiers [3]. Several approaches have been studied, among them: bagging [4], that uses more than one model of the same paradigm in order to reduce errors; boosting [5], in which a different weight is given to different training examples looking for a better accuracy; bi-layer classifiers [6] where several models from different paradigms are combined in a parallel mode, and after the individual decision of each one is used as predictor variable for a new classifier which takes the final decision; or some other combination approach in serial or semi-parallel architectures [7, 8].

In this paper a new variant of the Naive Bayes algorithm is presented and tested. A hierarchy consisting of two levels of Naive Bayes classifiers is built, where the classifiers in the second level act over the predictions of the first level. The idea behind this procedure is that the mistakes a classifier makes are suitable to be the input to another classifier, in such a way that this one could discover some patterns in them.

Experimental results have been obtained over 20 UCI datasets, and show that the presented paradigm outperforms standard Naive-Bayes in 12 out of the 20 datasets.

The rest of the paper is organised as follows. Section 2 describes the Naive Bayes classifier; in section 3 the new proposed approach is presented. Experimental results are shown in section 4. Final section is dedicated to conclusions and points out the future work.

2 Naive Bayes

Given a database of cases consisting of a set of predictor variables $X_1, X_2, \cdots, X_n$ and a special variable Y which value must be predicted, being each of the N cases in the form

$$\text{Case}_i : X_1 = x_{1i}, X_2 = x_{2i}, \cdots, X_n = x_{ni}, Y = y_l$$

and being, without lost of generality, $Y = y_1, Y = y_2, \cdots, Y = y_m$ the m possible values the class variable can take (the m categories considered in the classification problem), theoretically, Bayes' rule minimises error by selecting the class y_j with the largest posterior probability for a given example $\mathbf{X}$ of the form $X =< X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n >$, as indicated below:

$$P(Y = y_j|X) = \frac{P(Y = y_j)P(\mathbf{X}|Y = y_j)}{P(\mathbf{X})}.$$

Since $\mathbf{X}$ is a composition of n discrete values, one can expand this expression to:

$$P(Y = y_j|X_1 = x_1, ..., X_n = x_n) =$$

$$\frac{P(Y = y_j)P(X_1 = x_1, ..., X_n = x_n|Y = y_j)}{P(X_1 = x_1, ..., X_n = x_n)}$$

where $P(X_1 = x_1, ..., X_n = x_n|Y = y_j)$ is the conditional probability of the instance $\mathbf{X}$ given the class y_j; $P(Y = y_j)$ is the a priori probability that one will observe class y_j; $P(\mathbf{X})$ is the prior probability of observing the instance $\mathbf{X}$. All these parameters are estimated from the training set. However, a direct application of these rules is difficult due to the lack of sufficient data in the training set to reliably obtain all the conditional probabilities needed by the model. One simple form of the previous model has been studied [2] that assumes independence of the observations of feature variables $X_1, X_2, ..., X_n$ given the class variable Y, which allows us to use the next equality

$$P(X_1 = x_1, ..., X_n = x_n|Y = y_j) \propto$$

$$P(Y = y_j)\prod_{i=1}^{n} P(X_i = x_i|Y = y_j)$$

where $P(X_i = x_i|Y = y_j)$ is the probability of an instance of class y_j having the observed attribute value x_i. In the core of this paradigm there is an assumption of independence between the occurrence of features values, that is not true in many tasks; however, it is empirically demonstrated that this paradigm gives good results in many real domains, typically in medical tasks. Some modifications of the Naive Bayes have been proposed, like hybrid methods [9], training set edition [10], or new attributes construction [11] [12]. A review of the health of Naive Bayes method at its fortieth birthday can be found in [13].

3 Smart Bayes

In this section we present the way a two-level hierarchy of Naive Bayes classifiers works, as well as the method we have used to construct it. We have named this approach Smart Bayes, as it is intended to overcome some limitations of Naive Bayes.

3.1 Smart Bayes at Work

In a hierarchy of Naive Bayes classifiers single classifiers are arranged in a tree, as depicted in figure 1. Thereafter, a new case runs the following path: starting at the root node, the instance is classified by the top level Naive Bayes classifier; according to the result, a branch is selected and the instance is classified again according to the Naive Bayes model corresponding to that branch and the final outcome is returned.

The main idea of the presented approach is to take into account the fact that a Naive Bayes classifier (as any other ML paradigm) performs different when classifying cases as belonging to a class or to another. In other words, for the presented positive-negative example, the amount of false positives and false negatives could be different. Therefore, another Naive Bayes classifier can be used to correct some of the errors made in the previous classification, i.e., to turn true negatives some of the false positives (classified as *Yes* in the first attempt) and to turn also true positives some of the false negatives (badly classified as *No* in the first layer).

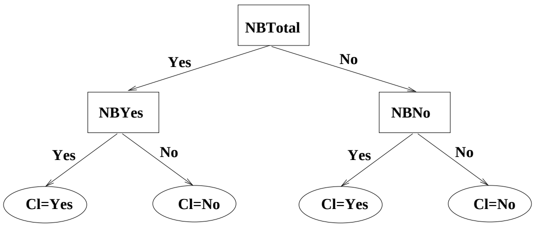


Figure 1: A hierarchy of Naive Bayes classifiers in a Yes-No problem.

3.2 Smart Bayes under Construction

To construct a two-level classifier it needs to be inferred, from a unique training set, all the models that form the hierarchy. While it is more or less obvious how to build a suitable model for the root node (just using the whole training set), it is not so obvious how to get the model for non-root nodes. When a case reaches a node, that means that it has been classified by the classifier in the parent node, as belonging to some class (say *Yes*). Therefore, to build the classifier at that node, a training set fulfilling the previous constraints (cases classified as *Yes* for the parent node) is needed. The best approach would be to perform a leave-one-out validation over the training set, in order to have an accurate estimation of the cases that the classifier assigns to class *Yes*. But, as this is quite time-consuming, in our experiments we use a n-fold cross validation [14]; in order to study the behaviour of Smart Bayes with different values for n, we have experimented with the values from $n = 2$ to $n = 9$ as shown in section 4.

The following example will help to understand the construction process. Given the hierarchy in figure 1, let

Table 1: Accuracy of Naive Bayes single classifier and Smart Bayes.

Database	**NB**	**SB 2**	**SB 3**	**SB 4**	**SB 5**	**SB 6**	**SB 7**	**SB 8**	**SB 9**	**Worst**	**Best**
Banding	78.57	**71.41**	70.18	70.54	68.08	*70.16*	70.96	69.71	70.94	↓70.16	↓71.41
Breast	96.14	*92.99*	**93.56**	93.42	93.42	93.42	**93.56**	93.28	**93.56**	↓92.99	↓93.56
Cars1	68.84	*72.24*	74.04	**74.78**	74.03	74.28	72.76	74.03	72.49	↑72.24	↑74.78
Chess	87.64	91.65	91.52	**91.68**	*91.49*	**91.68**	91.58	*91.49*	91.58	↑91.49	↑91.68
Churn	88.42	*89.50*	**89.70**	89.56	89.62	89.54	89.62	89.52	89.58	↑89.50	↑89.70
Cleve	83.51	81.54	*80.86*	80.88	*80.86*	**82.84**	81.86	*80.86*	82.53	↓80.86	↓82.84
Corral	84.42	*81.28*	82.05	**84.42**	83.59	82.05	83.59	83.59	83.59	↓81.28	=84.42
Crx	77.68	*79.85*	80.29	80.00	80.00	80.29	**80.58**	80.14	80.14	↑79.85	↑80.58
Diabetes	75.66	**74.36**	73.06	*72.66*	73.32	73.57	73.44	73.45	73.57	↓72.66	↓74.36
Flare	80.67	*81.89*	82.26	82.73	82.55	82.45	82.64	82.73	**82.92**	↑81.89	↑82.92
Hepatitis	83.88	85.87	*84.62*	*84.62*	85.92	85.92	85.87	**87.21**	86.54	↑84.62	↑87.21
Horse-colic	78.49	81.23	**83.42**	81.24	*80.97*	81.78	81.50	81.51	81.25	↑80.97	↑83.42
Lenses	63.33	63.33	63.33	63.33	63.33	63.33	63.33	63.33	63.33	=63.33	=63.33
Sonar	68.79	*76.02*	77.90	77.90	77.90	77.95	76.48	**78.45**	76.48	↑76.02	↑78.45
Tic-tac-toe	69.72	*71.08*	72.33	**72.96**	72.33	72.33	72.54	72.33	72.02	↑71.08	↑72.96
Vehicle	45.41	*60.99*	61.24	63.02	62.41	62.65	62.89	**63.60**	63.12	↑60.99	↑63.60
Vote	90.34	*94.01*	94.93	94.94	94.93	94.71	94.70	**95.40**	**95.40**	↑94.01	↑95.40
Vote1	87.33	89.88	89.64	**90.11**	**90.11**	*89.41*	**90.11**	**90.11**	*89.41*	↑89.41	↑90.11
Wine	**97.19**	*96.63*	*96.63*	**97.19**	**97.19**	**97.19**	**97.19**	**97.19**	**97.19**	↓96.63	=97.19
Zoo	87.09	87.09	87.09	87.09	87.09	87.09	87.09	87.09	87.09	=87.09	=87.09

us suppose that we are working in a two-class problem, being the two classes *Yes*, *No*. It is necessary to build the classifier model for each of the nodes in the tree that forms the hierarchy. Let us name *Examples.data* the datafile where all the available instances corresponding to our classification problem are collected. The procedure is as follows:

- The classifier in the root node of the hierarchy is constructed using all the cases in *Examples.data*; let us name the classifier model *NBTotal*.
- The classifier in every branch just below the root is constructed in the following way:
  - Get the database which was used to build the classifier at the root node.
  - Divide the database into *n* folds of the same size.
  - For each fold,
    classify each case in the fold using the Naive Bayes induced by the remaining *n-1* folds.
  - If *class of case = yes*, add it to *Examplesyes.data*
    else add it to *Examplesno.data*.
  - If the node is reached from the root node through a *Yes* edge, then use *Examplesyes.data* to build the model associated at that node (*NBYes*). If it is not (that is, if the node is reached through a *No* edge), use *Examplesno.data* to build the model (*NBNo*).

4 Experimental Setup and Results

In order to evaluate the performance of the proposed approach, we have performed an experiment over 20 databases from the UCI repository [15]. In table 2 the characteristics of these databases are drawn. The number of cases ranges from 24 to 5000, the number of attributes from 4 to 59 and the number of classes from 2 to 7, so a wide variety of problems are represented in these databases.

In our experimental setup we have made use of MLC++ libraries [16] to implement Naive Bayes.

Let us remember that in the process of construction of this two-level classifier it is necessary to choose a number of folds in order to build the databases below the root node. In our experiments we have worked with *n* ranging from 2 to 9. The estimation of the accuracy of our proposed hierarchy has been obtained by means of a ten-fold cross validation.

In table 1 is shown the performance of Naive Bayes over the databases we have worked with, along with the performance of Smart Bayes with folds in the range (2..9). *NB* stands for Naive Bayes and *SB n* stands for Smart Bayes with *n* folds. The column labelled *Best* is the best performance among all the number of folds partitions we have considered, while the column labelled *Worst* is the worst performance among all the folds partitions. An upward arrow means that the performance is better than Naive Bayes, while a downward arrow means the opposite.

Table 2: Characteristics of the databases used in the experimental setup.

Database	#Cases	#Attributes	#Classes
Banding	238	29	2
Breast	699	10	2
Cars1	392	7	3
Chess	3196	36	2
Churn	5000	20	2
Cleve	303	14	2
Corral	129	6	2
Crx	690	15	2
Diabetes	768	8	2
Flare	1066	10	2
Hepatitis	155	19	2
Horse-colic	368	28	2
Lenses	24	4	3
Sonar	208	59	2
Tic-tac-toe	958	9	2
Vehicle	846	18	4
Vote	435	16	2
Vote1	435	15	2
Wine	178	13	3
Zoo	101	18	7

Table 3: Summary of the results of table 1.

NB versus SB	Wins	Ties	Losses
Best fold partition	12	4	4
Worst fold partition	12	2	6

Let us note that, taking into account even the worst partition, Smart Bayes outperforms Naive Bayes in 12 out of 20 databases, tying twice and losing in 6 databases. When we choose the best partitions these figures raise to 12 wins, 4 ties and 4 losses. From these numbers we can conclude that whatever the number of folds partitions we choose the chances of Smart Bayes outperforming Naive Bayes are much greater than the opposite. These results are summarised in table 3.

So far we have shown that choosing at random the value of the number of folds (n), even in the worst case the behaviour of Smart Bayes is better than Naive Bayes. Anyway, it is always desirable to know for which value of n our chances of obtaining the best results are the highest. In this way, as n is the only parameter of Smart Bayes, this process can be seen as parameter tuning. Thus, to achieve the goal of selecting the best number of folds, we have constructed the table 4, in which the cell in row i and column j contains two numbers, being the first one the number of times (over the 20 databases) Smart Bayes with i folds outperforms Smart Bayes with j folds, and the second one the number of times the result is the opposite. This sum is not 160 (20 databases X 8 number of folds) because there are some ties.

As it was expected, the best performance is achieved with a high number of folds. Performances are similar to $n = 7, 8$ or 9. Let us note that for $n = 4$ the performance is very good too. The best result is achieved with $n = 7$. The number of wins is the same than with $n = 9$, but the number of losses is one less; in the particular match $n = 7$ wins $n = 9$ for $(7, 6)$.

Once the parameter n is selected, the comparison of individual classifiers can be done. As shown in table 5, the hierarchy constructed by using 7 fold cross-validation in order to obtain the low-level classifier, outperforms standard Naive Bayes classifier in 12 out of 20 databases, losing in four of them. If the mean of the 20 accuracies is used (only for confirmation purposes, as it is not considered a remarkable value for the authors), a slight increase can be seen also, from the 79.66% of the Naive Bayes to the 81.61% mean obtained by the presented new approach.

Table 5: Comparison between Naive Bayes and Smart Bayes for $n = 7$.

Database	NB	SB (n=7)	SB - NB
Banding	78.57	70.96	-7.61
Breast	96.14	93.56	-2.58
Cars1	68.84	72.76	+3.92
Chess	87.64	91.58	+3.94
Churn	88.42	89.62	+1.20
Cleve	83.51	81.86	-1.65
Corral	84.42	83.59	-0.83
Crx	77.68	80.58	+2.90
Diabetes	75.66	73.44	-2.22
Flare	80.67	82.64	+1.97
Hepatitis	83.88	85.87	+1.99
Horse-colic	78.49	81.50	+3.01
Lenses	63.33	63.33	0.00
Sonar	68.79	76.48	+7.69
Tic-tac-toe	69.72	72.54	+2.82
Vehicle	45.41	62.89	+17.48
Vote	90.34	94.70	+4.36
Vote1	87.33	90.11	+2.78
Wine	97.19	97.19	0.00
Zoo	87.09	87.09	0.00
Mean	79.66	81.61	+1.96

From these figures we conclude that for the presented experiment the best election is $n = 7$.

5 Conclusions and Further Work

In this paper we have presented a new way to combine several Naive Bayes classifiers in order to improve the performance of a single Naive Bayes. The method has been implemented and tested over 20 databases from the UCI repository.

It has been shown that over the databases tested the new method outperforms single Naive Bayes in a significant number of them.

Table 4: Results of matches between Smart Bayes with *i* folds and Smart Bayes with *j* folds.

	SB 2	**SB 3**	**SB 4**	**SB 5**	**SB 6**	**SB 7**	**SB 8**	**SB 9**	**Total**
SB 2	(0, 0)	(6, 11)	(4, 14)	(5, 13)	(3, 15)	(3, 14)	(4,14)	(4,14)	(29, 95)
SB 3	(11, 6)	(0, 0)	(5, 11)	(7, 7)	(6, 9)	(5, 12)	(7, 9)	(7, 10)	(48, 64)
SB 4	(14, 4)	(11, 5)	(0, 0)	(10, 3)	(9, 6)	(8, 8)	(8, 7)	(6, 11)	(66, 44)
SB 5	(13, 5)	(7, 7)	(3, 10)	(0, 0)	(5, 9)	(4, 10)	(2, 9)	(5, 11)	(39, 61)
SB 6	(15, 3)	(9, 6)	(6, 9)	(9, 5)	(0, 0)	(8, 9)	(9, 7)	(7, 8)	(63, 47)
SB 7	(14, 3)	(12, 5)	(8, 8)	(10, 4)	(9, 8)	(0, 0)	(7, 8)	(7, 6)	(67, 42)
SB 8	(14, 4)	(9, 7)	(7, 8)	(9, 2)	(7, 9)	(8, 7)	(0, 0)	(7, 7)	(61, 44)
SB 9	(14, 4)	(10, 7)	(11, 6)	(11, 5)	(8, 7)	(6, 7)	(7, 7)	(0, 0)	(67, 43)

As future work it would be interesting to extend the current two-level model to a more general model with a higher number of levels. In this work we have worked with Naive Bayes classifiers, but other paradigms, as decision trees or neural networks could be used too.

6 Acknowledgements

This work has been supported by the Ministerio de Ciencia y Tecnología under grant TSI2005-00390 and by the AMIGUNE ETORTEK Project.

7 References

[1] T. Mitchell, *Machine Learning*, McGraw Hill, New York, USA (1997).

[2] M. Minsky, “Steps toward artificial intelligence”, *Proceedings of the Institute of Radio Engineers*, 49, pp 8-30 (1961).

[3] T.K. Ho, J.J. Hull, and S.N. Srihari, “Decision combination in multiple classifier systems”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1), pp 66-75 (1994).

[4] L. Breiman, “Bagging predictors”, *Machine Learning*, 24(2), pp 123-140 (1996).

[5] Y. Freund and R.E. Schapire, “A short introduction to boosting”, *Journal of Japanese Society for Artificial Intelligence*, 14(5), pp 771-780 (1999).

[6] D.H. Wolpert, “Stacked generalization”, *Neural Networks*, 5, pp 241-259 (1992).

[7] J.M. Gama, *Combining Classification Algorithms*, PhD Thesis, Universidade do Porto (2000).

[8] V. Gunes, M. Ménard, P. Loonis, and S. Petit-Renaud, “Combination, cooperation and selection of classifiers: a state of the art”, *International Journal of Pattern Recognition*, 17, pp 1303-1324 (2003).

[9] R. Kohavi, “Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid”, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, USA, pp 202-207 (1996).

[10] J.M. Martínez-Otzeta, B. Sierra, E. Lazkano, M. Ardaiz, and E. Jauregi, “Edited naive bayes”, *Ibero-American Journal of Artificial Intelligence*, 10(31), pp 63-69 (2006).

[11] I. Kononenko, “Semi-naive bayesian classifier”, *European Working Session on Learning-EWSL91*, Porto, Portugal, pp 206-219 (1991).

[12] M. Pazzani, “Constructive induction of cartesian product attributes”, *Proc. Conf. Information, Statistics and Induction in Science (ISIS96)*, Melbourne, Australia, pp 66-77 (1996).

[13] D.D. Lewis, “Naive (bayes) at forty: The independence assumption in information retrieval”, *Proceedings of the 10th European Conference on Machine Learning*, Chemnitz, Germany, Springer-Verlag, pp 4-15 (1998).

[14] M. Stone, “Cross-validation choice and assessment of statistical procedures”, *Journal of the Royal Statistical Society*, 36, pp 111-147 (1974).

[15] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, “UCI repository of machine learning databases”, http://www.ics.uci.edu/∼mlearn/MLRepository.html, visited on 24/07/2006.

[16] R. Kohavi, D. Sommerfield, and J. Dougherty, “Data mining using MLC++ : A machine learning library in C++”, *International Journal on Artificial Intelligence Tools*, 6(4), pp 537-566 (1997).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Distributed Markov Localisation for Probabilistic Behaviour Activation

E. Jauregi, E. Lazkano, B. Sierra, A. Astigarraga, J.M. Martínez-Otzeta, Y. Yurramendi
Department of Computer Science and Artificial Intelligence
University of the Basque Country, San Sebastian, Basque Country

Abstract

Probabilistic methods offer the necessary tools with a sound theoretical basis for handling self localisation but they are generally applied to rigid environment representations and thereby, they are hardly capable of coping with dynamic environments. Our current research effort aims to narrow the gap between behaviour based navigation and probabilistic methods. This paper presents a distributed self-localisation system in semi-structured environments.

Keywords: Markov localisation, behaviour-based navigation, topological maps

1 Introduction

Even though great technical progress has been made in the area of mobile robotics, some fundamental control problems, such as autonomous navigation, remain unresolved. Without such ability the robot would not be able to avoid dangerous obstacles, reach energy sources or come back home after an exploration of its environment. Although many animals have shown that they are very good at navigating, autonomous navigation is a complicated task for engineered robots. Therefore, research efforts have been aimed at incorporating biologically inspired strategies into robot navigation models [1, 2]. The control architecture being developed by the authors [3] fully commits with the behaviour-based (BB) approach [4] and the taxonomy of biomimetic navigation strategies proposed in [5]. In this context, navigation consists of determining and maintaining a trajectory to the goal.

Probabilistic methods offer the necessary tools with a sound theoretical basis for handling self localisation but they are generally applied to rigid environment representations and thereby, they are hardly capable of coping with dynamic environments. On the other hand, robots developed within the BB paradigm only make use of a subset of environmental properties needed for localisation, showing a higher degree of adaptability to dynamic environmental changes. Thus, our current research effort aims to narrow the gap between behaviour based navigation and probabilistic methods. In our opinion, it is possible to take profit of the soundness and mathematical foundation of probabilistic techniques within the field of BB robotics.

In this paper we present a attempt of integrating a self-localisation subsystem based in probabilistic methods, with which the robot is able to localise itself without any prior knowledge about its position. The proposed approach is an application of the well known Markov Localisation (ML) algorithm to our topological navigation system. The resulting distributed localisation method has been tested in simulation for the procedural map of the second floor of the Computer Sciences Faculty.

2 Procedural Environment Description

BB systems advocates for a functional bottom-up decomposition of the control problem in independent processes called behaviours. Matarić [6] proposed a topological representation that instead of being a declarative model, was fully procedural in the sense that each node in the graph that formed the map was a completely functional unit responsible of guiding the robot when the current location matched the node. From this point of view, the topological "map" should be composed of tightly coupled behaviours, specific to the meaningful locations.

In [3] we proposed an extension to Matarić's work where a map is formally defined as a directed and connected graph, and each node consists of:

1. A set of inputs (from landmark identification subsystems) and outputs.
2. A set of predecessors and successors. These links to neighbouring nodes include information about the direction θ in which the i-th succession occurs. In that way, each successor of a node is a successor in a fixed orientation distribution.
3. A signature that uniquely identifies the node from its immediate predecessors and successors: $signature_i$.
4. A function α_i to be executed when the node is active and that will output the action to be performed

at the node specific current state. The behaviour the robot will show can be different depending on the location, and so can the associated function of several nodes be.

The overall "map" is then composed of sets of behaviours, each launched on a different thread, with an associated functionality and differing in the node signature that makes them distinguishable. Nodes behave according to a finite state automata (FSA) that uniquely determines, by means of some activation/deactivation signals, the node active at each time step. Each node has also assigned a location identifier in terms of coordinates ((x_i, y_i, θ_i)) that gives spatial relationship among the nodes and helps to extract information about the direction in which the goal can be achieved. However, the most important drawback of the previous approach resides in the deterministic nature of the localisation system. A single node remains active at each time step and therefore, only position tracking is allowed, assuming that the initial position is known.

Next sections describe how the navigation approach is extended in such a way that the robot is able to localise itself and to maintain its location in a more robust manner.

3 Probabilistic Behaviour Activation

Nowadays the field of probabilistic robotics offers a full set of algorithms that provide the necessary tools to maintain belief distributions over the state space [7]. The key idea of probabilistic robotics is to explicitly represent the uncertainty associated to robot localisation by means of probability calculus theory, thus, representing the ambiguity in a sound manner. The ML algorithm (see figure 1) addresses the global localisation problem, the position tracking problem and the kidnapped robot problem, although it is normally applied in static environments.

```
Markov localisation($bel(x_{t-1}), u_t, z_t, m$)
 for all $x_t$ do
     $\overline{bel(x_t)} = \int p(x_t|u_t, x_{t-1}, m) bel(x_{t-1}) dx$
     $bel(x_t) = \mu p(z_t|x_t, m) \overline{bel(x_t)}$
 endfor
return($bel(x_t)$)
```

Figure 1: General form of the ML algorithm

This algorithm is the adaptation of the Bayes Filter for the robot localisation problem and as such, it is a recursive algorithm.

In order to apply the Markov Localisation algorithm, two probability distributions must be defined: the probability of observing a sensor reading at each state (*sensor model*) and the state transition probability or *action model*.

3.1 Sensor Model

The control architecture is composed by several landmark identification subsystems that output a confidence level for each type of landmark. These values are filtered through node signatures, giving at each time step the node activation level according to the sensor readings. More specifically, these identificators are:

1. Corridor identification (cl^{corr})
2. Mean compass orientation (cl^{θ})
3. Left/right wall (useful in non corridor places) (cl^{wall})

The perception of the robot at time t is then:

$$z_t = \left\{ cl_t^{corr}, cl_t^{wall}, cl_t^{\theta} \right\}$$

Then the probability of seeing a concrete landmark k in a node x_i is a function of the expected values (signature of the node) and the confidence level values returned by the landmark identification subsystems:

$$P(cl_t^k|x_i) = f(cl_t^k, signature_i) \in (0, \cdots, 1]$$

And assuming independence among the different landmarks:

$$P(z_t|x_i) = \prod_k P(cl_t^k, x_i)$$

3.2 State Transition Probability

In general, no totally position independent method will be able to distinguish between two landmarks of the same qualitative type and compass bearing. It is common to use odometry measurements for calculating the robot's motion over time, although it is only available after the robot has moved and suffers from accumulative error.

In our case, the state transition probability function should reflect the probability that the transition occurs if the robot has travelled the distance (translational and rotational) accumulated by the odometry system.

In the *odometry motion model* defined in [7], any robot movement is decomposed in three steps: initial rotation (a^1), translation (d) and final rotation (a^2) (see figure 2). It is assumed that each motion variable is perturbed by an independent error source.

If according to two consecutive odometry measurements the robot's movement vector is $(\overline{d}, \overline{a}^1, \overline{a}^2)$ and the movement vector associated to transition from

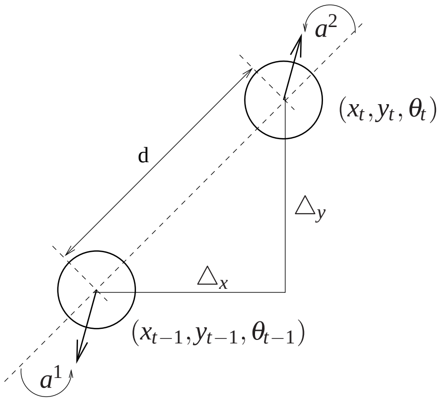


Figure 2: Odometry motion model

state x_j to state x_i is (d, a^1, a^2), the state transition probability is then the product of the component probability functions. Each component function is described as a zero mean normal distribution based on the difference between the occurred value and the theoretical value. For the presented approach, we decomposed the transition probability just into two components, the translational (p_{trans}) and the rotational components (p_{rot}).

For the translational component, instead of defining d as the Euclidean distance, we used the Manhattan distance (or Minkowsky's L_1 distance): $d = \triangle_x + \triangle_y$. This distance measures the distance between two points along axes at right angles and has shown to be more appropriate for the given environmental setup. The translational and rotational differences are then:

$$\begin{cases} \delta = d - \overline{d} \\ \theta = |a^1 - \overline{a}^1| + |a^2 - \overline{a}^2| \end{cases}$$

The transition probability is calculated assuming independent error sources for each component:

$$P(x_i|x_j, u_t) = p_{trans} * p_{rot} \tag{1}$$

where[1]:

$$\begin{cases} p_{trans} = N(\delta, \sigma_{trans}) \\ p_{rot} = N(\theta, \sigma_{rot}) \end{cases}$$

Opposite to grid models, the topological representation approach does not imply a uniform distribution of locations. Behaviours are maintained for different space scales, depending on the environment the robot moves in. Moreover, in grid-based approaches the "do-nothing" action reflects the probability of remaining in a node. Instead, for the topological approach, we need to define the probability of **remaining** in a node as a function of the space represented by that node. Again, $P(x_i|x_i, u_t)$ is computed as in equation (1) but the component probability values are calculated as follows:

$$\begin{cases} p_{trans} = \begin{cases} 0.99 & \text{if } d \leq \text{length} \\ \frac{0.99}{2*d} & \text{otherwise} \end{cases} \\ \\ p_{rot} = N(\theta, \sigma'_{\theta}) \text{ where } \sigma'_{rot} = 2 * \sigma_{rot} \end{cases}$$

3.3 Behavioural Organisation

The previous subsections described how the probabilistic models were defined. But still, how the nodes manage to calculate and maintain their beliefs should be clarified. Figure 3 shows the behavioural organisation of the localisation subsystem of the architecture.

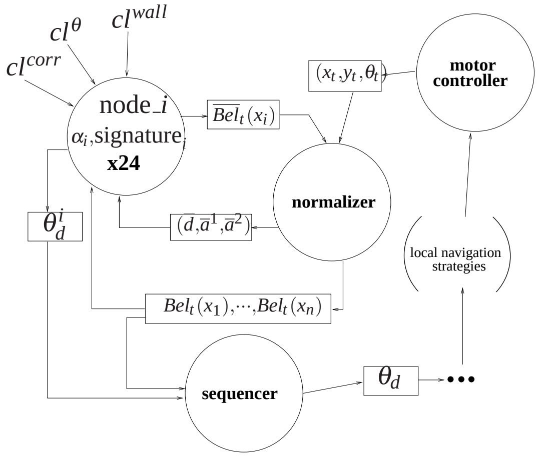


Figure 3: Localisation subsystem

Each node i stores the information necessary to calculate $P(x_i|x_j, u_t), \forall j.$, i.e the parameters (d, a^1, a^2) that need to occur to arrive to that node from everywhere else. In this manner, every node receives as input the values $(\overline{d}, \overline{a}^1, \overline{a}^2)$ calculated from odometry readings and it is able to calculate the non-normalised belief $\overline{Bel_i}$ of being in that node. An extra thread (*normaliser*) is responsible of reading individual beliefs and normalising them.

The main peculiarity of the system being developed is the qualitative nature of the actions performed by the robot. The nodes decide which compass orientation must follow the robot to reach the goal while avoiding obstacles. This property is of high relevance because makes the robot behaviour robust in dynamic environments. The fact that no deterministic action set is defined makes it difficult to determine the effect of an action and therefore, to know when the transition probability must be updated. To cope with this problem, we accumulate odometric information until there is a high degree of certainty about robot's location, i.e, the belief of being in a node exceeds certain threshold. That normaliser process is also responsible of deciding when to accumulate distance and when to restart the reference point, according the normalised belief values.

Finally, the *sequencer* receives as inputs the normalised beliefs and the orientation proposals of the nodes and

[1] $N(0, \sigma)$ represents a zero centred normal distribution with deviation σ

acts as a conflict solver, selecting the orientation suggested by the most probable node.

4 Experimental Evaluation

The system has been developed and tested using Player/Stage [8]. Figure 4 shows the map of 24 processes (nodes) associated to the experimental environment in figure 5. Several kinds of nodes have been identified for the given environment.

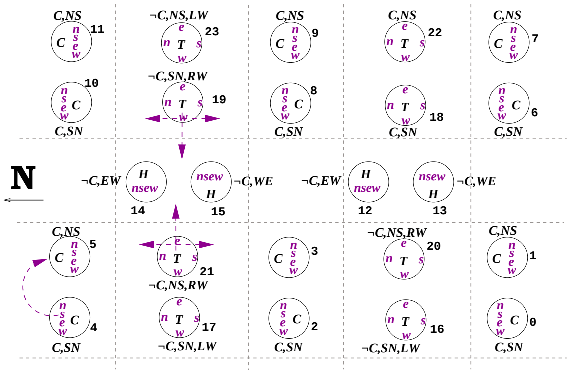


Figure 4: Procedural map of nodes

- 12 corridors (C), 6 South to North (right to left in the figure) and 6 North to South (left to right in the figure). Each corridor has two different nodes inside, depending on the direction followed by the robot.

- 4 Halls (H), two East-West (Up-to-Down in the figure) and other two West-East (Down-to-Up in the figure). As happens with the corridors, each hall has two nodes.

- 8 Transitions (T); four North-South and another four South-West.

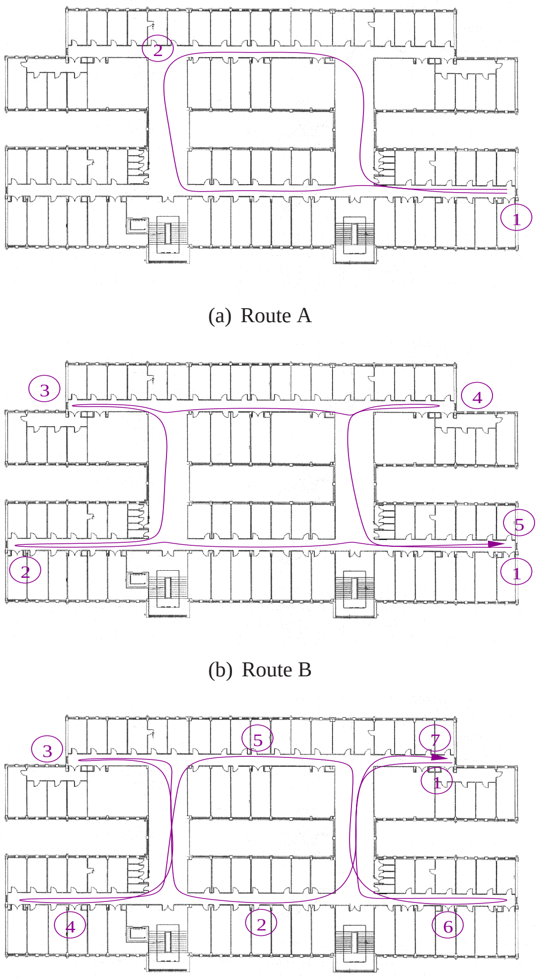
Note that nodes are duplicated. This fact is due to the local navigation strategies defined. The robot always moves on a privileged compass orientation and therefore, the orientation is part of the behaviour shown by the robot. Note also that no link information is used for the localisation task. In principle, all transitions are possible. The high degree of symmetry of the environment facilitates the development of clone functions for different locations.

Two more landmark identification subsystems are needed for action triggering (orientation changing):

- Crossroad recognition: we need to somehow identify crossroads because it is crucial to choose the correct orientation when different alternative ways can be taken. Functions associated to hall nodes make use of the output of this identification.

- Dead ends: some corridors are dead end paths, there is no other exit possible than turning around 180° and retracing one's steps. Dead ends can be identified using the sonar ring, more specifically the 4 front sonars, and measuring the distance to the walls that surround the robot. These are only considered within the corridor nodes.

Several local navigation strategies (compass-following, corridor-free-space-balancing, ...) allow the robot to navigate towards a goal while avoiding obstacles. The whole control architecture is then composed by 38 threads that communicate among them.



(a) Route A

(b) Route B

(c) Route C

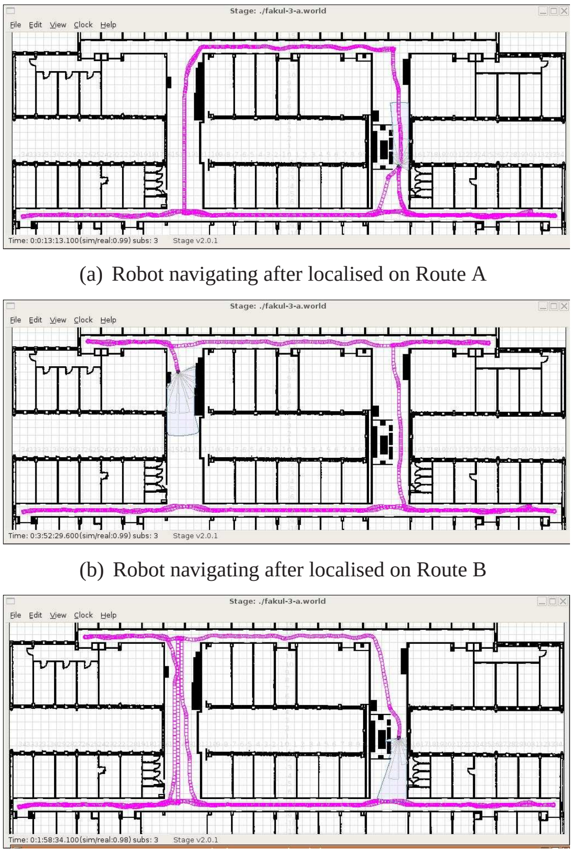
Figure 5: Different routes and intermediate goals

The robot can be launched at any *unknown* position of the environment, i.e, the belief distribution is initialised uniformly. It is important to note that landmarks are *ambiguous*; that is, the robot cannot localise itself simply by observing a single landmark. Rather, the robot must undertake some relatively complex series of actions, and may have to travel a considerable distance, in order to unambiguously determine its location [9].

Therefore, the robot first needs to localise itself and fulfil the trajectory afterwards. The *localizator* thread is responsible of initially performing a simple strategy, looking for free space into the four nominal orientations to guide the robot to the closest corridor and, afterwards following it until the localisation subsystem is stabilised.

In order to test the developed localisation subsystem, three different routes (robotic tasks) have been defined, and some intermediate goals are included in each route. As it can be seen in Figure 5, the route A has two intermediate goals, the first one in the right-down side (node 0) and the second one in the left-up side (node 19); similarly, the route B defines 5 intermediate points as goals, while route C has 7 locations to visit.

The same experiment has been made in the A, B and C routes, starting in an unknown place for the robot, localising itself and performing the routes three times and showing persistence. The robot performs well its task in all the cases. Figure 6 shows the robot performing these routes after localised.


Stage: ./fakul-3-a.world
File Edit View Clock Help

Time: 0:0:13:13.100(sim/real:0.99) subs: 3   Stage v2.0.1

(a) Robot navigating after localised on Route A

Stage: ./fakul-3-a.world
File Edit View Clock Help

Time: 0:3:52:29.600(sim/real:0.99) subs: 3   Stage v2.0.1

(b) Robot navigating after localised on Route B

Stage: ./fakul-3-a.world
File Edit View Clock Help

Time: 0:1:58:34.100(sim/real:0.98) subs: 3   Stage v2.0.1

(c) Robot navigating after localised on Route C

Figure 6: Robot performing routes A, B and C

Figure 7 shows how the robot performs the localisation strategy and starts fulfilling route A afterwards, starting from a rather complicated initial position (zoomed in the circled area) and verifying the adequateness of the initial localisation strategy (*localizator* module).

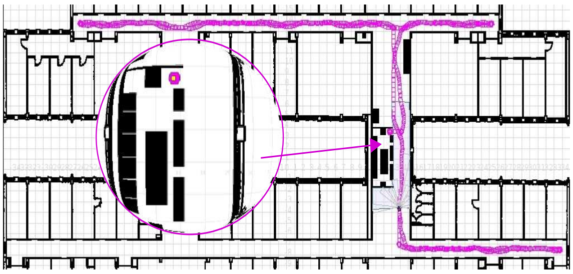


Figure 7: Robot localising before route completion

5 Related Work

Different paradigms can be distinguished for environment modelling. Occupancy grids are used for Simultaneous Localisation and Mapping (SLAM) approaches. The main weaknesses of this approach are the computational burden and the relevance of the odometry drift. On the other hand, topological approaches typically represent the environment as a sequence of landmarks or distinct places. Hybrid approaches combine both representations, usually in a hierarchical manner, in which topological map are used for large scale localisation and each node has an associated grid for small scale localisation. For example, Howard and Kitchen [9] present a vision-guided robot navigating around the corridors at a University building. Two coordinate systems are used; the model is defined in a global coordinate system (GCS) and the odometry measurements are made in a local one (LCS), whose origin is arbitrary. The localisation process consists of looking for the an invertible transformation function between the GCS and the LCS.

We found some references that bear resemblance with our approach. In the work by Barber and Salichs [10], an event driven navigation method is presented. The topological map (*Navigation Chart*) is not made up of environmental element successions, but of a succession of tasks to be executed during navigation, which means going through all the possible plans. Alike, Egido et al. [11] propose to represent the environment by a succession of elementary skills. Concerning the metric encoding, Lankenau and Röfer [12] present a self-localisation system where maps are represented as route graphs containing (geo-)metric data about the length of the corridors as well as about the included angles.

Related to the action model, in the work by Tomatis et al. [13] a combination of topological and metric approaches for SLAM is presented, using a global topological map which contains a metric map associated to each of the nodes. The topological localisation system is also Markovian but within the action model, they condition the probability that a state transition occurs to the observation of an opening. If no opening is visible, then the probability of remaining in a node is high while transitions to other states are almost improbable. Otherwise, when an opening is visible, the most probable transition is the one closer according to the distance travelled while the rest of the transitions are low probable.

Finally, in [14] a landmark-based topological representation is used in a behaviour-based navigation system. The topological map is a connected graph where each node represents a landmark completely characterised by a set of attributes. It is mentioned that state transition probabilities are estimated from the reachability information contained in the topological map, but no detailed explanation is given about the process. Moreover, navigation is solely based on wall following.

6 Conclusions and Further Work

This paper describes a preliminary approach of how the Markovian localisation can be used in a distributed system. Experimental results, although simulated, are promising. The proposed approach requires low storage and computational resources and is, in our opinion, more adequate to be applied in dynamic environments. The coordinate information included in the nodes help to accommodate a common action model used in probabilistic approaches and could be easily acquired within an exploration strategy. The topological nature of the system makes it less sensitive to odometry errors. However, experiments in the real robot/environment system are mandatory. Moreover, the granularity of the environment should be increased in order to reach more interesting goals such as offices. We plan to incorporate a door identification module and enrich the behaviour associated to several nodes with door crossing abilities. It is worth mentioning that the sequencer has been added only for testing the localisation subsystem and its going to be replaced by a belief propagation scheme in the near future, using the predecessor/successor information and thereby, taking profit of the topological link relationship embedded in the map.

7 Acknowledgements

This work was supported by the MCYT (TSI2005-00390), and the Gipuzkoako Foru Aldundia (OF200/2005).

8 References

[1] O. Trullier, S. I. Wiener, A. Berthoz and J. A. Meyer, "Biologically-based artificial navigation systems: review and prospects", *Progress in Neurobiology*, 51, pp 483–544 (1997).

[2] H. A. Mallot and M. A. Franz, "Biomimetic robot navigation", *Robotics and Autonomous System*, 30, pp 133–153 (2000).

[3] E. Lazkano, A. Astigarraga, E. Jauregi, B. Sierra and J. M. Martínez-Otzeta, "Environment representation by behavior decomposition", *Proceedings Workshop de Agentes físicos* (WAF2006), Las Palmas de Gran Canaria, pp 101–108 (2006).

[4] R. A. Brooks, "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, 26, pp 14–23 (1986).

[5] O. Trullier and J. A. Meyer, "Biomimetic navigation models and strategies in animats", *AI Communications*, 10(2), pp 79–92 (1997).

[6] M. Matarić, *A Distributed Model for Mobile Robot Environment: Learning and Navigation*, Technical Report MIT-AITR 1228, MIT (1992).

[7] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, MA (2005).

[8] B. P. Gerkey, R. T. Vaughan and A. Howard, "The player/stage project: tools for multi-robot and distributed sensor systems", *Proceedings of the International Conference on Advanced Robotics (ICAR2003)*, Coimbra, pp 317–323 (2003).

[9] A. Howard and L. Kitchen, "Navigation using natural landmarks", *Robotics and Autonomous Systems*, 26, pp 99–115 (2001).

[10] R. Barber and M. A. Salichs, "Mobile robot navigation based on event maps", *Proceedings of Field and Service Robotics (FSR2001)*, Helsinki, pp 61–66 (2001).

[11] V. Egido, R. Barber, M.J.L. Boada and M. A. Salichs, "A planner for topological navigation based on previous experiences", *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles (IAV2004)*, Lisbon, (2004).

[12] A. Lankenau and T. Röfer, "Mobile robot self-localization in large-scale environments", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2002)*, Washington, D.C., pp 1359–1364 (2002).

[13] N. Tomatis, I. Nourbakhsh and R. Siegwart, "Combining topological and metric: A natural integration for simultaneous localization and map building", *Proceedings of the 4th European Workshop on Advanced Mobile Robots (Eurobot2001)*, Lund, (2001).

[14] F. Zanichelli, "Topological maps and robust localization for autonomous navigation", *Proceedings of the IJCAI Workshop on Adaptive Spatial Representation on Dynamics Environments*, Stockholm, (1999).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Experiments with A Mobile Cluster for Real-time Object Detection

Andre L. C. Barczak and Ravi Chemudugunta
Institute of Information and Mathematical Sciences,
Massey University, Auckland, New Zealand
a.l.barczak@massey.ac.nz

Abstract

This paper describes the development of a mobile parallel platform (using the concept of the cluster of computers) for real-time object recognition based on Viola-Jones method. The mobile platform was built with off-the-shelf components (hardware and software). This platform could be fitted in a robot, car, backpack etc. A simple application protocol based on sockets and UDP was developed to deal with the communication between the nodes. This paper describes the prototype hardware and software as well as the details of the protocol developed. Experimental results regarding the performance of the communication and the processing capacity of the prototype when running Viola and Jones detection algorithm are presented.

Keywords: Real-time object detection, parallel systems, clusters. mobile cluster

1 Introduction

This paper presents the development and test of a mobile parallel system prototype for computer vision applications. The platform was built using off-the-shelf components and was tuned to deal with the real-time detection method developed by Viola and Jones [1].

The prototype was built targeting the following characteristics:

- Cheap to build using commodity components.
- Easy to deploy and maintain.
- Low power consumption.
- Fast communication infra-structure among the processors.
- Allow a USB camera to be directly attached to the system.

Related work include the use of parallel classifiers for hand detection and angle determination [2], which could potentially be used in gesture recognition. A theoretical model for the behaviour and performance of a parallel platform running detection algorithms based on Viola and Jones method was presented in [3]. On the subject of parallel systems for computer vision and image processing there is a revelant discussion in [4].

The rest of the paper is organised as follows: the next section presents the parallelisation of the Viola-Jones detector. The hardware and the software descriptions are presented, as well as the details about the protocol developed especially for this prototype. Finally experiments with real-time classifiers are presented and performance issues are discussed.

2 Viola-Jones Method and its Parallelisation Strategy

Viola and Jones proposed a method that encompasses training and detection. For the details about the training and the detection methods refer to [1]. In this section we will discuss the characteristics of the detection phase of Viola and Jones method and its consequences for the parallelisation of the algorithm. When running the detection algorithm, each frame needs a special data structure called *Integral Image* [5] to speed up the feature extraction based on Haar-like features. Once classifiers are produced by the training process, the detection carries out the following summarised steps [6]:

1. The Integral Image are computed for the current frame.
2. Each sub-window of the frame is submitted to a classification process
3. For a given classifier, Haar-like features are extracted according to the classifier rules. Each classifier has several stages (cascades).

One difficulty with the parallelisation is that each stage of a certain classifier need the response from a previous stage. If different objects need to be detected or the same object needs to be detected at random angles, it is possible to train separate classifiers that share the same integral images. The parallelisation strategy is then based on the following assumptions:

- The number of classifiers is always larger than the number of nodes.
- A Master/slave paradigm is used, where the master acquires images and broadcasts them to the slaves.
- Each slave responds with the position of object(s) found on a certain frame by sending the points to the master.

A node can use the same integral image to extract the features needed for all the classifiers it is running. Figure 1 shows an example of parallel classifiers applied to a angle detection problem.

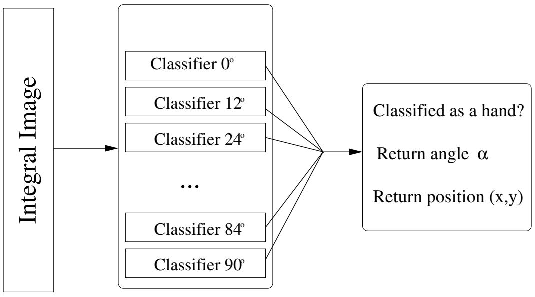


Figure 1: Detecting objects (e.g. hands) and their in-plane angle.

3 The Platform

The cluster was built using mini-itx boards manufactured by VIA. These boards are relatively small (17x17 cm) and have a very small power consumption (25 watts at full speed). The prototype has currently only 4 nodes, but it can easily be extended to several nodes. Figure 2 shows a schematic of the cluster. Figure 3 shows an actual photo of the prototype. The system can use a 12 volts source for all its components and therefore can be powered by batteries.

Each board is fitted with a 1Ghz processor (CPU Via C3/Eden 1000 mhz), 512 MB RAM, dual fast Ethernet adaptors and USB. The prototype is diskless, with the only persistant storage being a flash disk on the master node which stores the operating system. Once the master has booted, the other nodes boot via the network. The operating system consists of a the custom built linux kernel 2.6. OpenCV [7] is the library used for the development of the computer vision algorithms. Sockets are used to develop the special protocols to transfer information between the nodes.

4 The Communication Pattern

Two separate processes in each node are responsible for the communication and for the computation. The master processes and the slave processes differ and are briefly described next. Because the VIA boards are fitted with two Ethernet adapters each, they can be used to build two separate networks, one for broadcasting the original frame, the other dedicated to inform the master about the results of the classification process.

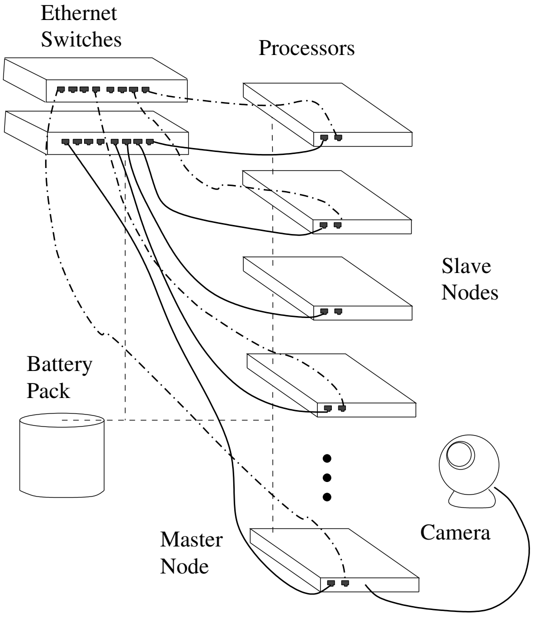


Figure 2: The mobile cluster platform.

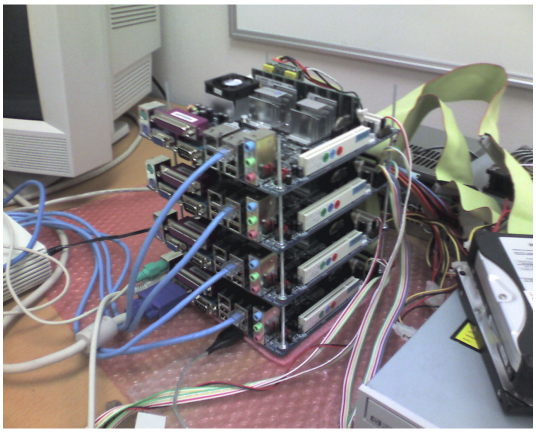


Figure 3: The mobile cluster platform.

The first master's process broadcasts frames (images) acquired by the web camera by the master to the nodes. Although there are libraries that provide calls for these services, usually they are not very efficient due to high latency. For example, MPICH (a generic implementation of MPI [8]) provides blocking calls that have a tree like strategy and do not necessarily use a true network broadcast. Experiments showed that such generic libraries are not always suitable for real-time applications due to the high latency [4] [9].

The second master's process listens and receives the responses from the slaves. The responses might contain the position, angle and type of object found.

The first slave's process receives the frame, computes the integral images and sends the position, angle and object type to the second slave's process. The transfer of this data is very fast because the processes share the slave's memory.

The second slave's process sends the position, angle and type of any object found. Although some delay is tolerated, the accuracy can be compromised if the answer refers to a frame that is several seconds old.

A slight variation on this pattern is possible. As the integral images used by the nodes are often identical their computation could be done on the slaves rather than on the master. In this scenario it would be important that the broadcasting time is shorter than the time to compute the integral images. However, the Integral Images are usually larger than the images themselves because they need more space to store large sums of pixels. The choice of scenarios will depend on both hardware (bandwidth, latency and processing speed) and software (mainly the characteristics of the protocol). The two scenarios are show in figures 4 and 5.

4.1 Real-time Constraints

Real-time constraints such as deadlines are application dependent. We have defined two types of deadline: application-specific deadline and a performance-penalty deadline. An application-specific deadline refers to the maximum time before the answer is invalid, while performance-penalty refers to maximum time before the throughput of the system is reduced (reducing the number of frames per second).

Figure 6 shows a scenario where the performance-penalty deadline is bypassed. While the answer is still valid because it has met the application-specific deadline, it incurs a performance penalty. The process misses a few frames until it is able to receive a new one and proceed to compute the II and the classifiers. It is anticipated that detection tasks such as face and gesture detection/recognition the performance-penalty is not so critical. For other detection tasks where the objects being detected are in rapid motion (such as cars) one may need faster CPUs to be able to complete the task within real-time constraints.

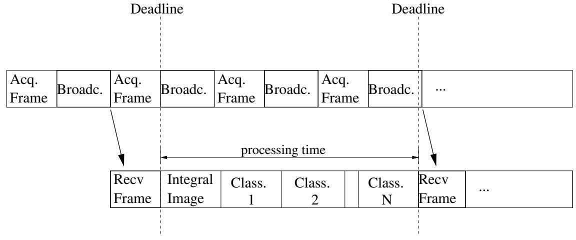


Figure 6: Real-time deadlines.

4.2 A Simple and Fast Broadcast Protocol

The mobile-cluster uses standard fast Ethernet connectivity between the nodes of the cluster. The latency between each of the nodes is on the order of milliseconds. Because the size of job is considerably small in proportion to the latency, inter-processor communication must be minimised.

We propose the 'unattended broadcast' method where point-to-point communication and synchronisation are eliminated to favour speed. In this method the master node broadcasts a video sequence constantly and does not perform any error-recovery or flow-control functions. No acknowledgements are sent back from the receivers. This works well because the adapters are within a dedicate network switch so packet losses due to network problems are negligible. Also the receivers need only a fraction of the total number of frames available to determine the existence and position of objects in the scene.

UDP is used as the transport protocol. Due to the limitation imposed by ipv4 on the size of the payload, the image is fragmented and reassembled at the receiving end. A special header is attached to each fragment to achieve this. A customised simple error detection is carried out using this header, out of sequence fragments can be detected and dropped. Figure 7 shows the headers and the packet structures.

{ IP { UDP { [Context ID] [Frame ID] [Offset][Size][Hash][Data ...]}}}

Figure 7: Packet layout.

In the next section we analyse various aspects of the performance of this protocol.

Table 1: Characteristics of the protocol.

Image Size	Number of Fragments	Header (bytes)	Total Size (bytes)
320x240	2	40	76840
640x480	5	100	307300
800x600	8	160	480160
1024x768	13	260	786692

5 Experimental Results

In this section the broadcast performance of our image transfer protocol is compared to MPI's broadcast and the runtimes for the various phases of the algorithm and for the communication are measured. Experiments were carried out to measure performance characteristics and to determine which communication scenario (Figures 4 and 5) is more appropriate for this prototype.

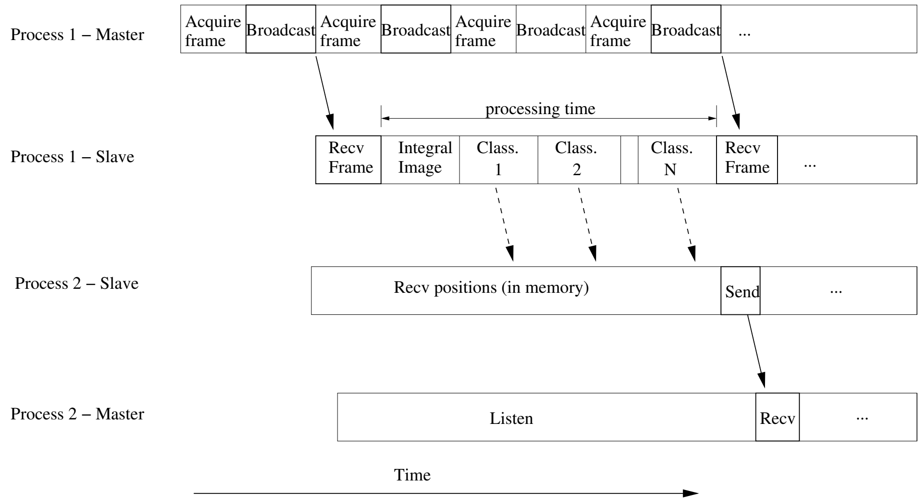


Figure 4: Scenario 1 - communication pattern with the Integral Images computed on the slaves.

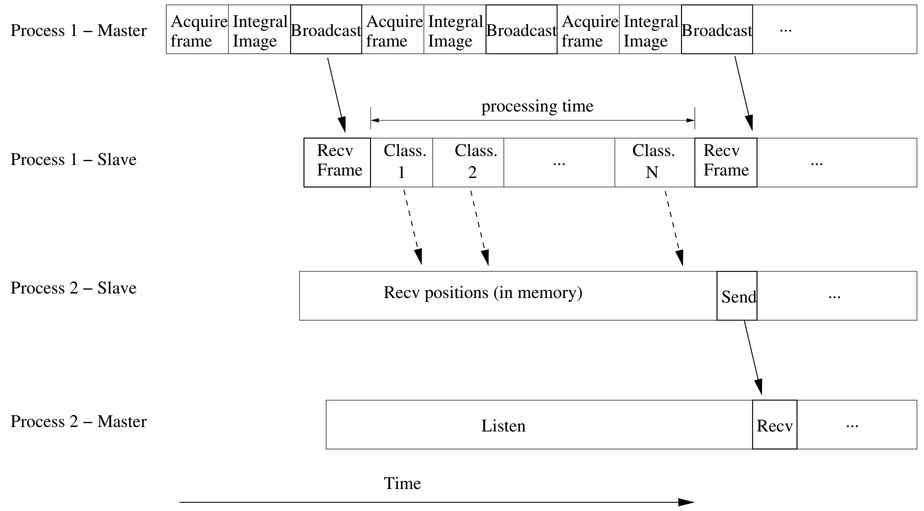


Figure 5: Scenario 2 - communication pattern with the Integral Images computed on the master.

5.1 Broadcast Performance

We present results for the broadcast using MPI [8] to compare with the our image transfer protocol.

Experiments were carried out transferring various image of different sizes. In order to obtain results for a larger number of nodes, we used another cluster of computers available to us. The results are significant because they use the same time of switch used in the prototype. Figure 8 shows the results obtained.

The results show that the MPI broadcast call is not suited to real-time applications. There are two reasons for that. Firstly the MPI broadcast, as implemented in MPICH available for the Linux OS, is not a true broadcast. It uses a tree like strategy to send the packet to each node involved in the collective call. Secondly, the MPI broadcast call is of the blocking type and it awaits for an acknowledgement, which makes the latency too large in relation to the sending time.

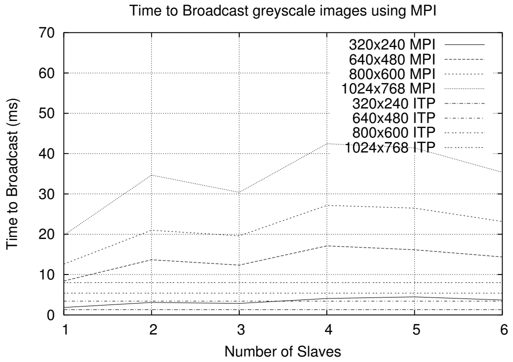


Figure 8: Broadcasting images using MPI calls and image transfer protocol (ITP).

5.2 The Image Transfer Protocol

One needs to know how this particular implementation of the protocol running on the cluster's prototype behaves. Four figures are very important to determine the performance:

- Broadcasting time
- Integral Images computation time
- Classifiers computation time
- Sending/receiving time

The broadcast time was obtained by repeating the process for several seconds. An average transfer time shows what the network infrastructure is able to deliver in practice. The perceived rate (in frames per second) indicates the maximum rate at which no classifiers would be running. It is lower than the theoretical capacity mainly due to latency, which is inherent to the hardware available.

Table 2: Broadcasting time.

Image Size	Transfer Time (ms)	Theor. (fps)	Perc. (fps)
320x240	11.08	162.7	90.2
640x480	27.40	40.7	36.5
800x600	43.29	26.0	23.1
1024x768	70.92	15.9	14.1

The original Viola and Jones method used only one Integral Image. The OpenCV library's implementation of Viola and Jones method uses three Integral Images. The additional ones are used to compute the variance (it stores the squared sum of the pixels) and a tilted version of the Integral Image. The last is used to compute an extended set of Haar-like features proposed in [10]. Table 3 shows that the runtimes are very short, so the computation of the Integral Images are not critical.

Table 3: Integral Images runtime for various image sizes (average of three measurements in ms).

	II runtime	equivalent Rate (fps)
320x240	0.83	120.48
640x480	3.35	29.88
800x600	5.26	19.01

The runtimes for the classifiers are more critical. They involve the computation of Haar-like features and the comparison with thresholds, which is very fast for a single sub-window. However, many sub-windows (in some cases hundreds of thousands) have to be examined. The total number of sub-windows is dependent of the scaling factor used. Figure 9 shows results for scaling factors from 1.1 to 1.6.

Finally, the times to send/receive the responses from the slaves are measured and analysed. Because the amount of data coming from the slaves in each response is negligible (a few bytes at most), this time tends to be equal to the latency. The latency was estimate at an average of 6 ms.

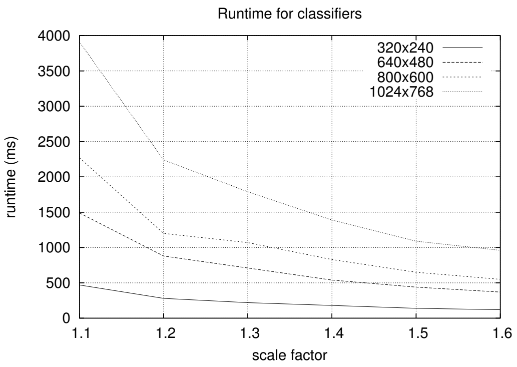


Figure 9: Classifiers' runtimes.

5.3 An Example: Hands Detection

A typical application for this system is to detect objects at different angles. Classifiers were trained according to Viola and Jones method in a previous work ([2]). Figure 10 show some of the results for hand detection using a single gesture. Each node is responsible for finding certain angles, and they respond to the master with the position and angle detected. A rectangle is drawn to indicate the detection and the approximate angle.

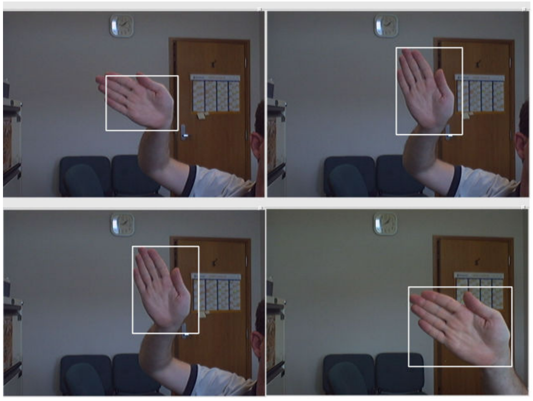


Figure 10: An example of application for this system: hand detection.

The detector is colour independent, as the Haar-like features are measuring differences in contrast among areas of the image. Notice that the door and the arm, even though have the same colours, are not detected.

5.4 Discussion of the Results

The system is loosely synchronised allowing the communication protocol to be very fast in comparison to other parallel communication libraries such as MPI [8]. Even without taking into consideration such features omitted in our protocol such as error-recovery and flow control, this shows the difficulty that parallel libraries face at not being able to capitalise on the characteristics of the underlying network topology to keep their code topology independent.

The run times for a particular classifier are dependant on the input image itself, i.e., images with objects being classified take longer than pictures with no objects. A tightly-coupled system where all nodes are synchronised would lead to some nodes being idle. Our approach further improves by having the master continuously broadcast an image resulting in a start-up latency time of approximately $\frac{1}{2fps}$ compared to a request-response mechanism with at least RTT to the receiving of the frame.
The runtimes for the Integral Images and for the send/receive operations are very short when comparing with the runtimes of the classifiers and the time to complete a broadcast. This indicates that this prototype might have a small limit of number of classifiers that can run on a given node.
These results also indicate the for any image size it is better to use scenario 1. In other words, to send the raw images to the slaves, so each slave computes its own Integral Images. Although there is computation that is repeated in the slaves, this takes less time than trying to send whole Integral Images across. It is important to state that scenario 2 might be better if using a Gigabit Ethernet.

6 Conclusions and Future Work

In this paper a prototype of a parallel system for real-time object detection in images was built using off-the-shelf components. We described the implementation of a customised broadcast function to provide fast image transfer between the nodes using UDP and broadcasting over Ethernet. Between two possible scenarios for the distribution of tasks among the nodes, we concluded that scenario 1 (figure 4) is more appropriate for this prototype.

For future work we intend to measure more characteristics of the network, processors and overall system and to implement applications such as a more complete gesture recognition system and vehicle detection. Other topics include exploring other applications where a mobile platform could bring benefits, the measurement of power utilisation characteristics and its effect on mobility (for e.g. battery weight vs length of time before a recharge is required) and experimenting with detection based decision systems.

7 Acknowledgements

We would like to thank Dr. Martin J. Johnson and Dr. Chris M. Messom for their valuable suggestions and support for this work and the MURF's committee for the funding that allowed us to assemble the hardware for this project. We also acknowledge the use of IIMS's cluster for the training of the classifiers.

8 References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR01)*, Kauai, HI, pp 511-518 (2001).

[2] A.L.C. Barczak, F. Dadgostar and M.J. Johnson, "Real-time hand tracking using the Viola and Jones method", *Proceedings 7th IASTED International Conference on Signal and Image Processing (SIP 2005)*, Honolulu, HI, pp 336-341 (2005).

[3] A.L.C. Barczak, M.J. Johnson and C.H. Messom, "A mobile parallel platform for real-time object recognition", *Proceedings 12th Electronics New Zealand Conference (ENZCon05)*, Auckland, NZ, pp 153-158 (2005).

[4] J. Barbosa, J. Tavares and A.J. Padilha, "Parallel image processing system on a cluster of personal computers", *Proceedings 4th International Meeting Vector and Parallel Processing (VECPAR 2000)*, Porto, Portugal, pp 439-542 (2001).

[5] F.C. Crow, "Summed-area tables for texture mapping", *Computer Graphics*, 18(3), pp 207-212 (1984).

[6] P. Viola and M.J. Jones, "Robust real-time face detection", *International Journal of Computer Vision*, 57, pp 137-154 (2004).

[7] G. Bradski, "The OpenCV library", *Dr. Dobb's Journal*, 25(11), pp 120-126 (2000).

[8] B. Wilkinson and M. Allen, *Parallel programming*, Prentice Hall, New Jersey (1999).

[9] W. Gropp and E.L. Lusk, "Reproducible measurements of MPI performance characteristics", *Proceedings 6th European PVM/MPI Users' Group Meeting on Recent Advances in PVM and MPI*, London, UK, pp 11-18 (1999).

[10] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection", *Proceedings IEEE Conference on Image Processing 2002 (ICIP02)*, Rochester, NY, pp 900-903 (2002).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Mobile SOKUIKI Sensor System: Accurate Range Data Mapping System with Sensor Motion

Tatsuro Ueda, Hirohiko Kawata, Tetsuo Tomizawa, Akihisa Ohya and Shin'ichi Yuta
Intelligent Robot Laboratory,
University of Tsukuba, Tsukuba, Japan
t_ueda@roboken.esys.tsukuba.ac.jp, kwt@roboken.esys.tsukuba.ac.jp,
tomys@roboken.esys.tsukuba.ac.jp, ohya@roboken.esys.tsukuba.ac.jp,
yuta@roboken.esys.tsukuba.ac.jp

Abstract

2D laser range finders are commonly used for constructing range data maps in robotics. In order to construct such maps, sensor's position and posture when it is performing each scan should be estimated accurately. However, in conventional laser scanning system the sensor is not synchronised with the modules, which estimate its position and posture. Therefore, correct position and posture of the sensor cannot be acquired if the sensor moves while it is scanning. This causes many errors in map construction. This paper first describes the timing registration method that makes the sensor synchronised with position estimation modules. Then, effectiveness and usefulness of this method are evaluated by the 2D and 3D scanning experiments.

Keywords: SOKUIKI sensor, range data map, timing registration

1 Introduction

2D laser range finders are commonly used for constructing range data maps or obstacle avoidance in robotics. In order to construct range data maps, the robot on which the sensor is mounted moves around so that the sensor scans[1] the whole environment. To construct maps by using these scanned range data, the positions and postures of the sensor when it measures each range data are needed. Generally, in order to acquire the positions and postures, exact time when the sensor measures each range data should be acquired. There are two steps shown below in order to acquire accurate timing.

1. Detect a period of time while the sensor perform a scan.

2. Find the exact moment when the sensor measures each range data in the period of time detected in first step.

The second step is not difficult because the time intervals between each measurement of range data are constant. However, first step is not easy because the sensor is not synchronised with the robot and the response time[2] of the sensor is ambiguous. In most of conventional scanning method, in order to acquire the positions and postures of the sensor, the robot stops every time when the sensor performs scan. For this reason, constructing range data map takes long time.

We develop a system which solves this time registration problem by using a SOKUIKI sensor which is a super small sized 2D laser range finder. With this system, the sensor accurately scans the environment even if it moves fast. We call this system Mobile SOKUIKI Sensor System. This paper describes the algorithm and structure of the system in section 2. Experiment of 2D scanning which proves effectiveness of the system is treated in section 3 and a new 3D scanning method is explained in section 4.

2 Mobile SOKUIKI Sensor System

2.1 Structure of the System

Figure 1 shows appearance of a SOKUIKI sensor URG-04LX manufactured by HOKUYO AUTOMATIC ([1, 2]). Table 1 shows main specification of the sensor.

This sensor outputs a synchronous signal at a specific scanning angle when it finishes every scan. The system uses this signal to acquire the exact time when the sensor scans.

Structure of Mobile SOKUIKI Sensor System is shown in figure 2. The system consists of a SOKUIKI sensor, a PC and a timing registration module which we developed. The SOKUIKI sensor is carried by a motion mechanism. After it finishes scanning, the sensor outputs a synchronous signal to the timing registration

[1] In this paper, we call it "scan" that a laser of the laser range finder rotates 360 degrees and call it "measure" that the sensor measures a range of a direction in one scan.

[2] The time which the sensor takes to send the range data after data request.

Figure 1: SOKUIKI sensor URG-04LX

Table 1: Specification of URG-04LX

Detectable distance	0.02 to 4[m]
Accuracy	0.02 to 1[m]:±10[mm] 1 to 4[m]: ±1% of distance
Resolution	1[mm]
Scanning angle	240[degrees]
Angle resolution	Approx. 0.36[degrees]
Scanning time	100[ms]
Weight	Apporox. 160[g]
External dimension	50×50×70[mm]

module. The module estimates the position and posture of the sensor by counting encoder pulses from the motor of the motion mechanism and stores them as logs. The PC constructs range data map using the range data and sensor's position and posture.

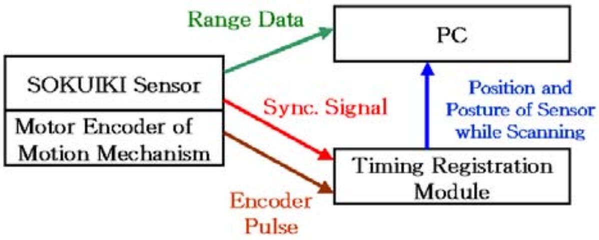

Range Data
PC
SOKUIKI Sensor
Motor Encoder of Motion Mechanism
Sync. Signal
Position and Posture of Sensor while Scanning
Timing Registration Module
Encoder Pulse

Figure 2: Signal flow of the system

2.2 Architecture of the Timing Registration Module

The architecture of the timing registration module is shown in figure 3. The system clock of the CPU called M16C is 24MHz and it counts three encoder pulse simultaneously. A USB controller enables the module to realise USB communication.

The module estimates the position and posture of the sensor every 5ms and stores them. When the module receives synchronous signal, it picks up the sets of the position and posture data when the sensor had been scanning out of logs and sends them to the PC.

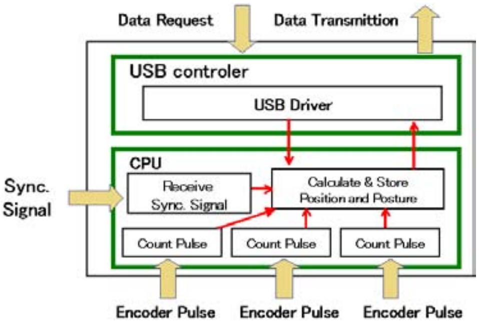


Figure 3: Architecture of the timing registration module

2.3 Procedure of Timing Registration

Figure 4 shows the procedure how each position and posture data is matched to each measurement point. The timing registration module estimates and stores position and posture of the sensor every 5ms. The SOKUIKI sensor repeats measurement periods and transaction periods continuously. In measurement period, the sensor measures ranges. Then it calculates the measured range data in transaction period. The measurement period and transaction period take 66.7ms and 33.3ms respectively. At the moment when measurement period finishes, the sensor outputs a synchronous signal to the module. If the sensor receives request for range data from the PC, the sensor start sending range data after the transaction period. Range data is sent through DMA channel so that this process does not use CPU power. The procedure of timing registration is explained as follows.

1. The PC sends request to the SOKUIKI sensor for range data.
2. The sensor outputs synchronous signal to a timing registration module at the moment when measurement period finishes.
3. The SOKUIKI sensor sends the range data to the PC.
4. The PC sends request to the module for the positions and the postures of the sensor, and then, the timing registration module sends the sets of the positions and the postures at which the SOKUIKI sensor is scanning.
5. Each position and posture is matched to each range data on the PC.

While sensor's positions and postures are estimated every 5ms, time intervals between each measurement point are 0.1ms. Therefore, the positions and postures are interpolated for matching to every measurement point.

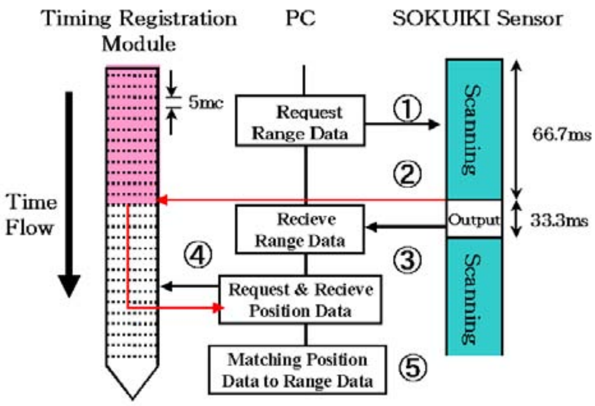


Figure 4: Procedure of matching the position and posture data to range data

3 2D Scanning Experiment

The effectiveness of the Mobile SOKUIKI Sensor System in constructing 2D range data map is evaluated in this section. In order to construct 2D range data maps, the sensor needs to move around the environment. Therefore, a mobile robot is used as the motion mechanism of Mobile SOKUIKI Sensor System. Figure 5 shows the mobile robot on which a Note PC, timing registration module and a SOKUIKI sensor are mounted. The sensor is attached to a robot in such a way that its scan plane is parallel to horizontal floor. The position and posture of the sensor is equivalent to those of the robot. the position and posture of the robot are estimated at the timing registration module by counting encoder pulses of wheels of the robot.

We performed an experiment in which the robot constructs 2D range data map in the following three ways. The robot moves at a speed of 300mm/s and turns 30rpm at most.

1. A robot with a conventional SOKUIKI sensor system scans environment. It stops for every scanning.

2. A robot with a conventional SOKUIKI sensor system scans environment. It does not stop for scanning.

3. A robot with the Mobile SOKUIKI sensor system scans environment. It does not stop for scanning.

figure 6a shows the environment for the experiment. Figure 6b, 6c and 6d are 2D range data maps constructed by the first, second and third procedures respectively. Green points in the middle of the maps are the positions at which the robot scans. Red points are the measurement points. In the first scanning procedure, the robot stops and estimates its position and posture before scanning so that the map is not affected by the timing registration problem and does not have any errors. In the second scanning procedure, because of lack of synchronization between the robot and the sensor it cannot estimate the exact position and posture at the moment when the sensor scans. Instead, the robot uses the position and posture when it receives the range data from the sensor. The positions and the postures used for map construction are not correct so the map has a lot of errors obviously. Figure 6d shows the map constructed by the third procedure. Though the robot does not stop for scanning, accuracy of the map is as same as figure 6b because of synchronization.

Another experiment is set to evaluate the system in a more detailed way. In the experiment, the sensor rotates horizontally at a speed of about 60rpm and scans the corridor once. Figure 7a shows the result of scanning without synchronization and figure 7b shows the result of scanning with the Mobile SOKUIKI Sensor System. Though the scanning results should be shown like two paralleled lines, the lines of figure 7a are distorted. The reason is that the robot treats the measurement points as if they measured at the same time though they are not. In figure 7b, the measurement points figured the corridor accurately. These two experiments prove the advantage of the 2D mapping system with the Mobile SOKUIKI Sensor System.

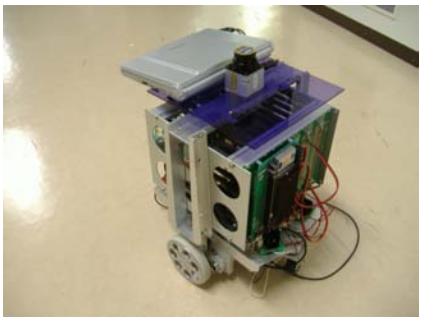


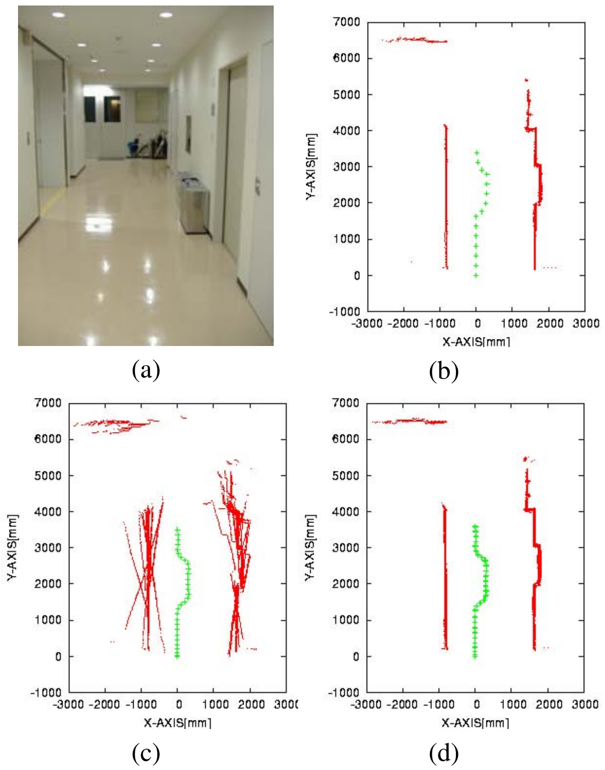
Figure 5: Robot equipped with the Mobile SOKUIKI Sensor System

4 Constructing 3D range data map

4.1 Rotating Mechanism for 3D Scanning

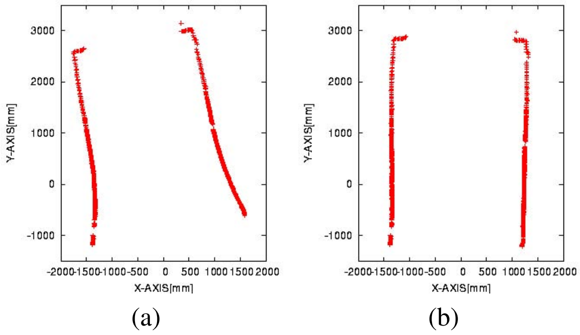
In constructing 3D range data map with a 2D laser range finder, changing the angle of scan plane is a common way for acquiring 3D range data. There are some ways to change its angle[3]. In our research, we rotate the sensor as it is shown in figure 8. This rotating method has vertical scan plane and rotate the plane around upright z-axis. With this method, measurement density focuses on z-axis (see figure 8b).

The rotating device is shown in figure 9a. A DC motor manufactured by Maxon Motor rotates the sensor. A slip ring manufactured by Tsubame Radio Co.,Ltd. is



(a) (b) (c) (d)

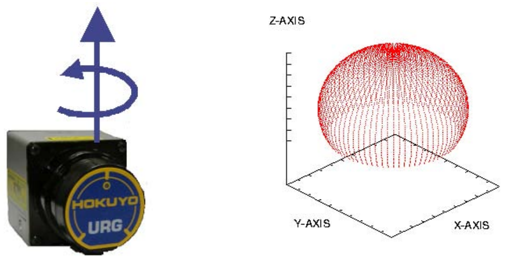
Figure 6: Scanning experiment of 2D environment: (a) environment, (b)result of scanning with halts for every scanning, (c) result of scanning without halt, (d) result of scanning without halt by using Mobile SOIKUIKI Sensor System



(a) (b)

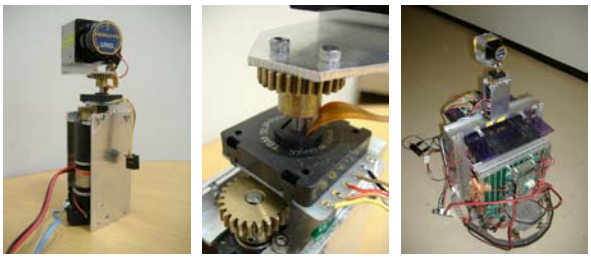
Figure 7: Comparision of scanning result: (a) result of one scanning without the Mobile SOKUIKI Sensor System, (b) result of one scanning with the Mobile SOKUIKI Sensor System

attached to the rotating shaft and all the cables from the sensor are though it so that the sensor rotates unlimitedly (see figure 9b). Note that the maximum rotating velocity of the slip ring is 60rpm. We mount this unlimitedly rotating device on a mobile robot (see figure 9c). The timing registration module estimates the sensor's position and posture by counting the pulses from the three motor encoders. Two of them are attached to the motors of wheels and the other is to the unlimitedly rotating device.



(a) (b)

Figure 8: Scanning method: (a) rotating method, (b) measurement density distribution



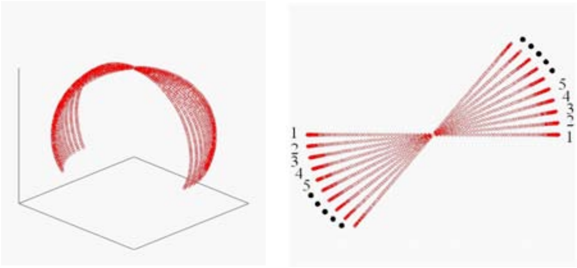
(a) (b) (c)

Figure 9: Motion mechanism: (a) unlimitedly rotating device with the sensor, (b) slip ring, (c) robot with the rotating device

4.2 Rotating Velocity of the Unlimitedly Rotating Device

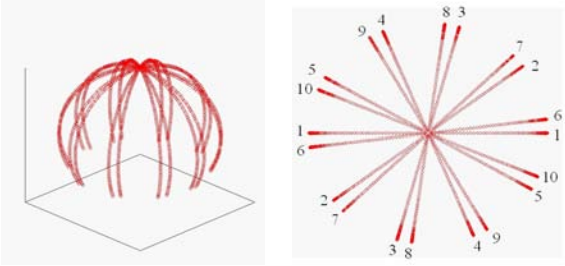
In most of conventional method for constructing 3D range data map with a SOKUIKI sensor, the scan plane of the sensor is rotated slowly around 5rpm because of timing registration problem. If it is rotated fast, the scanning result includes a great amount of errors. Figure 10a shows an image of the change of measurement density scanned by rotating the sensor slowly. Figure 10b is a top view of figure 10a. Numbers in figure 10b represent scanning order. We see from figure 10 that the measurement points increase in two directions. If the robot changes its moving direction before the sensor rotates 180 degree, the range data map has two big holes in which there is no measurement point.

In order to solve this problem, we rotate the sensor fast, around 60rpm in our research, to construct 3D range data maps. The change of measurement density is shown in figure 11. If we see figure 11 we will see that the measurement points increase in many directions. This rotating method decreases holes in the map. When rotating the sensor fast, the velocity of rotation should be determined carefully. If the sensor rotates exactly at a velocity of 60rpm, it scans at only 5 different angles.



(a) (b)

Figure 10: Change of measurement point density in rotating slowly: (a) side view of measurement point density, (b) top view of measurement point density



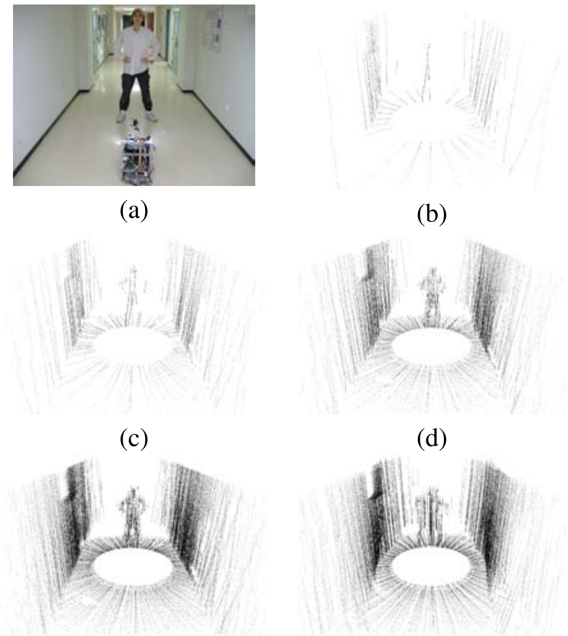
(a) (b)

Figure 11: Change of measurement point density in rotating fast: (a) side view of measurement point density, (b) top view of measurement point density

4.3 3D Scanning Experiment

Two experiments were performed to evaluate the effectiveness of our 3D scanning method. First, the robot does not move and only uses rotating device to scan environment. The sensor rotates at a speed of 58rpm. The result of experiment is shown in figure 12. Figure 12a, 12b, 12c, 12d and 12e each shows the experimental environment, result of scanning in 2s, 5s, 15s and 30s. As time passed, the density of the measurement points increases relatively uniformly. Figure 12f shows the result of scanning in 30s without the Mobile SOKUIKI Sensor System. We see from figure 12f that the human in the middle of the corridor is blurred and the walls are curved. Without the system, the range data map has a number of errors.

In the second experiment, the robot moves and the sensor scans 30s with the sensor rotating in the corridor, which is the same environment as the previous experiment. The robot moves along the corridor at a speed of 300mm/s and the sensor rotates at a speed of 58rpm. The result of the experiment is shown in figure 13. Figure 13a shows the 3D range data map from the viewpoint in the middle of the corridor. Figure 13b shows the result from a different viewpoint. The measurement points of the ceiling are not displayed in figure 13a and those of the ceiling and the left wall are not displayed in figure 13b to facilitate clear view of the scan data. Even if the robot moves around the environment, there are less error and no big holes in the map.

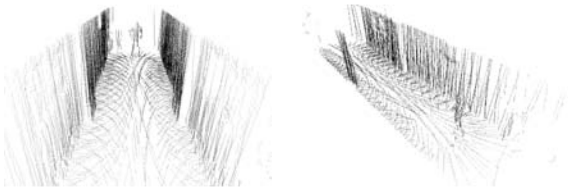


(a) (b)

(c) (d)

(e) (f)

Figure 12: Robot does not move and scans corridor: (a) environment, (b) scanning 2s, (c) scanning 5s, (d) scanning 15s, (e) scanning 30s, (d) scanning 30s without Mobile SOKUIKI Sensor System



(a) (b)

Figure 13: Robot moves and scans Corridor: (a) from the viewpoint in the corridor, (b) side view

5 Conclusion

This paper proposes a sensing system based on time synchronization between a position estimation module and a SOKUIKI sensor. The synchronization is realised by using synchronous signal from the SOKUIKI sensor. Our system can construct an accurate range data map of an area even if the sensor itself changes its position and posture continuously. The effectiveness of the system is shown through experiments. The accuracy of the system is mainly based on accuracy of the position and posture of the sensor. If the accumulative error of the position and posture has increased, the accuracy of the map is not assured. To solve that problem, another method is applied together.

The timing registration problem restricts the action of robots and manipulators on which 2D laser range finders are mounted in map construction. Our proposed system removes this restriction, so future systems could be implemented using Mobile SOKUIKI sensor system.

6 Acknowledgement

We would like to thank Mr. Luis Yoichi Morales Saiki of The University of Tsukuba for his assistance in assistance in preparing this manuscript.

7 References

[1] H. Kawata, W. Santosh, T. Mori, A. Ohya and S. Yuta, “Development of ultra-small lightweight optical range sensor system”, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2005),* Edmonton, pp 3277-3282 (2005).

[2] M. Alwan, M.B. Wagner, G. Wasson, and P. Sheth, “ Characterization of infrared rangefinder PBS-03JN for 2-D mapping”, *Proceedings IEEE Int. Conf on Robotics and Automation (ICRA2005),* Barcelona, pp 3947-3952 (2005).

[3] O. Wulf and B. Wagner, “Fast 3D scanning methods for laser measurement systems”, *Proceedings International Conference on Control Systems and Computer Science (CSCS2003),* Bucharest, pp 312-317 (2003).

[4] A. Diosi and L. Kleeman, “Laser scan matching in polar coordinates with application to SLAM” , *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS2005),* Edmonton, pp 3317-3322 (2005).

[5] K. Ohno, T. Tsubouchi and S. Yuta, “Outdoor map building based on odometry and RTK-GPS positioning fusion”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA2004),* New Orleans, pp 684-690 (2004).

Contaminated Floor Cleaning Robotic System for Radioactive Environment Application

Kiho Kim[1], Jangjin Park, Hohee Lee, Keechan Song, Chaeyoun Oh[2]
[1]Korea Atomic Energy Research Institute, Daejeon, Korea
[2]School of Mechanical and Aerospace System Engineering,
Automobile Hi-Technology Research Centre, ERI
Chonbuk National University, Chonbuk, Korea
khkim5@kaeri.re.kr, ohcy@chonbuk.ac.kr

Abstract

In this paper, we present the development of a RObotic Contamination Collection System (ROCCS) for use in a radioactive zone of the isolation room of the DUPIC Fuel Development Facility at the Korea Atomic Energy Research Institute. ROCCS employing a vacuum cleaning method is designed to clean up the contaminated floor of the isolation room in a remote manner, thereby completely eliminating a human interaction with the hazardous radioactive contaminants. ROCCS that is constructed in modules for an easy remote maintenance consists of five replaceable submodules – a mobile module, a cleaning module, a sensing module, a collection module and a cover module. ROCCS is operated either by a remote control or by an autonomous control in conjunction with a graphical simulator, by which the human operator can monitor and intervene in the ROCCS cleanup tasks in the isolation room. We also describe the environmental, mechanical design considerations and control system of ROCCS. Finally, we present its performance test results for a mockup of the isolation room.

Keywords: cleaning robot, remote cleanup, remote maintenance, radioactive contamination

1 Introduction

The use of robotics and remote technologies in the nuclear industry has been driven by the need to decrease the personnel radiation exposure limits. Reducing personnel exposure to as low as reasonably achievable in a radiation field where radiation and contamination levels are high requires human workers to be replaced by robotic devices or other remote devices.

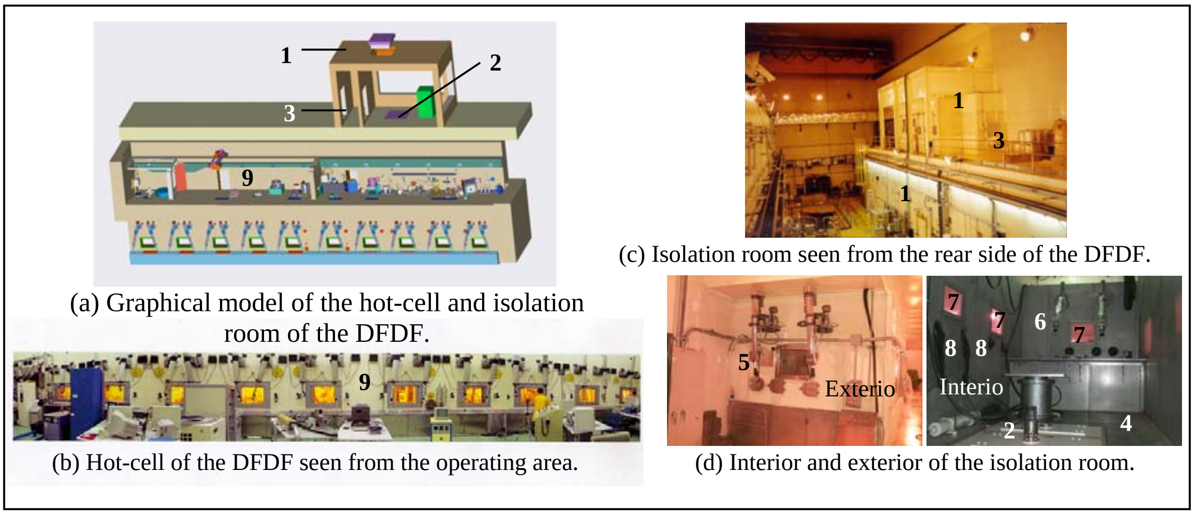
The Korea Atomic Energy Research Institute (KAERI) is developing DUPIC (Direct Use of spent PWR fuel in CANDU reactor) fuel cycle technology [1] which reuses spent PWR (pressurised water reactor) fuel. DUPIC technology is being developed in the completely shielded cell of the DUPIC Fuel Development Facility (DFDF) at KAERI because of the nature of the high radioactivity of spent PWR fuel. As the DFDF cell is active, direct human worker's access to the in-cell is not possible. Therefore, all DUPIC fuel fabrication processes and equipment operations are conducted in a remote manner.

The DFDF consists of the hot-cell and isolation room. The isolation room is located above the hot-cell. In the case where the fabrication equipment or devices are damaged or broken, its electrical or mechanical parts need to be properly repaired or exchanged. Such maintenance work is conducted in the isolation room. Undesirable products such as spent nuclear fuel powder debris and contaminated waste are inevitably produced during the maintenance process, thus contaminating the floor of the isolation room. Such undesirable radioactive contaminants should be cleaned to prevent the contamination from spreading inside the isolation room. A robotic system is needed to clean the contaminated floor of the isolation room, thus eliminating a human's interaction with hazardous radioactive contaminants.

The objective of this work is to present a RObotic Contamination Cleaning System (ROCCS) being capable of cleaning the contaminated floor of the isolation room without endangering the human workers.

2 Design Considerations

The isolation room that ROCCS is operated in is shown in figure 1 – the graphical representation of the hot-cell and isolation room (figure 1a), the hot-cell front seen from the operating area (figure 1b), the isolation room seen from the service area (figure 1c), and the interior of the isolation room (figure 1d). The isolation room located above the hot-cell has a length of 6 m, a width of 2.6 m and a height of 2.6 m. The isolation room has an entrance only from the rear side of it and is not seen from the front operating area of the DFDF. This room includes four shielding windows located on the wall, eight gloves installed below the shielding windows, one overhead crane mounted on tracks, one pair of master-slave manipulators, and a roof door. The shielding windows



(a) Graphical model of the hot-cell and isolation room of the DFDF.

(b) Hot-cell of the DFDF seen from the operating area.

(c) Isolation room seen from the rear side of the DFDF.

(d) Interior and exterior of the isolation room.

Figure 1: Isolation room and hot-cell of the DFDF at KAERI. 1. isolation room, 2. roof door, 3. entrance, 4. floor to be cleaned, 5. master manipulator, 6. slave manipulator, 7. shielding window, 8. glove, 9. hot-cell, 10. rear side of hot-cell.

provide the inside information or situations of the isolation room to the human operator who is located outside the isolation room. The roof door (2) in figure 1 was installed between the ceiling of the inside of the hot-cell and the floor of the isolation room. It is used as a channel for connecting the isolation room and the interior of the hot-cell. Failed equipment or devices in the hot-cell that need repair or replacement are transmitted to the isolation room through the roof door.

Cleanup or decontamination operation of a contaminated floor where, as the isolation room is active, a direct human access to the interior is not possible due to a high radioactivity must be conducted by means of robotic devices in a remote manner. Therefore, three mutually dependent design elements should be considered in the design development of ROCCS for use in a radioactive zone of the isolation room of the DFDF. These design elements are the isolation room facility and the remote or autonomous cleanup operation of ROCCS and its remote maintenance when necessary.

First design element, the isolation room facility, is an important factor in determining the size, mobile means, and cleanup tools of ROCCS. The design development involves the environmental and spatial limitations of the task environment of the isolation room – the geometric structure and the accessibility to the target to be cleaned. As to the other design elements, the design concept should take into account the remote manipulation strategies, remote repair procedures, and the remote handling devices that are available at the isolation room. Control means significantly affects the design development of ROCCS. ROCCS should be controlled either by a remote control or by an autonomous control relying on the cleanup task conditions because the workers can have no access to the inside isolation room. The design should also include the considerations of an interface with a human operator, modular construction for an easy maintenance, power transmission for a control, and the radiation effects of the materials to be used. Hence, a compromise needs to be made between these design elements in the design process.

3 ROCCS

3.1 System Description

ROCCS was developed to clean up the contaminated floor of the DFDF isolation room. ROCCS employs the vacuum cleaning method [3] in order to clean up and collect the loose spent nuclear fuel debris and other radioactive waste placed on the floor. ROCCS has a configuration of 465x465 (HxD) mm. As shown in figure 2, it mainly consists of five replaceable submodules – a mobile module for navigation, a cleaning module for dislodging and sucking contaminated waste, sensing module for avoiding obstacles, a collection module for storing waste acquired through the cleaning module, and a cover module for protecting the collection module.

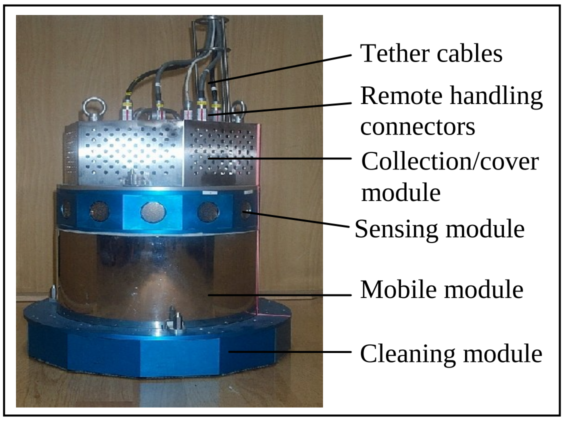
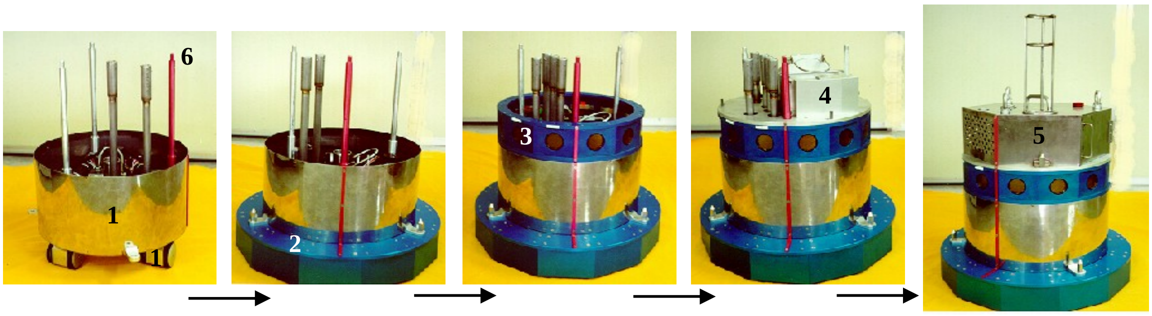


Figure 2: Developed ROCCS.



1. mobile module, 2. cleaning module, 3. sensing module, 4. collection module, 5. cover module, 6. assembly guide.

Figure 3: Assembly order of each module of ROCCS.

A mobile module that employs a synchro-drive wheel assembly was designed so that ROCCS can navigate the floor of the isolation room with a welded seam. The synchro-drive wheel assembly has three synchro-drive wheels, which were connected together with precise timing belts. These wheels are driven by two dc servomotors with resolvers – one for steering and the other for driving.

The sensing module that utilises ultrasonic sensors was designed to detect the walls of the isolation room and identify the initial position and orientation of ROCCS. Twelve ultrasonic sensors were installed at 30^o intervals in the shape of a ring around ROCCS in order to cover the entire area in all directions. A single drive module was used to simultaneously fire twelve ultrasonic sensors at a single time. In this work, a firing sequence of the four sensor intervals was used to reduce the undesirable crosstalk [2] occurring by firing the multiple sensors by a single drive module. Sensory data measured from each ultrasonic sensor during an operation is sent to the controller via an USB in a real time.

The cleaning module was designed to dislodge contaminated waste debris fixed or placed on the floor surface and to suck up mobilised waste particles, which were captured by the collection module. This cleaning module consists of twelve compact, rotary brush tools. Each rotary brush tool consists of a small rotary brush and a rectangular shaped aluminium cover housing the brush. A thin bronze fringe was installed around the base of the aluminium cover such that it increases the cleaning ability without sacrificing the mobility of ROCCS. Each rotary brush tool was connected together in series, forming a circular shape. Each aluminium cover was, via each corresponding tube, connected to the reservoir, which was also connected to the collection module via a single tube. The collection module was designed to capture and store waste particles mobilised by the cleaning module. The collection module installed above the sensing module consists of an electrical blower and a filter box. This filter box made of a HEPA (high efficiency particulate air) filter can capture particles as small as 0.3 μm. It was designed so that a used filter box can be easily exchanged for a new one in a remote manner. The cover module installed on the top of ROCCS was designed to protect the collection module and dissipate the heat generated from the blower motor of the collection module.

Modular construction of ROCCS facilitates its remote maintenance when necessary. Each module can be easily assembled and separated by a remote manipulation, by using a crane, master-slave manipulators and other auxiliary devices installed in the isolation room. Figure 3 shows the assembling process of each module of ROCCS by using a master-slave manipulator.

3.2 Control

3.2.1 Overview of Control System

The control console provides a control location for ROCCS located inside the isolation room. The wall of the isolation room acts as a barrier between the operator (located outside the isolation room) and ROCCS. The control console is the interface device between the operator and ROCCS. All functions for controlling ROCCS are contained in it. Figure 4 shows the overall control system of ROCCS. The control console mainly includes a two-axis handcontroller for navigation in a remote control mode, operable knobs for activating the cleanup tools, and a graphic simulator for a monitoring and path generation. The controller, circuitry, power supplies, and necessary software are also installed within a control console.

ROCCS is powered and controlled via a tether. This form of power and control was selected because powering ROCCS via a tether allows for reliable operations of unlimited duration and it eliminates the needs for batteries. In addition the transmission of the control and power signals through a tether ensures a reliable control of ROCCS, regardless of the remote distance or barrier between the operator and ROCCS.

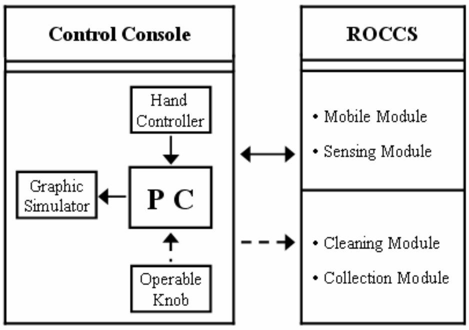


Figure 4: Control structure of ROCCS.

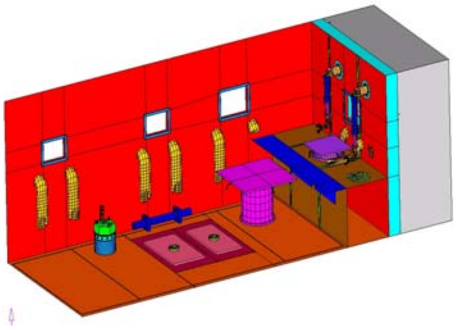


Figure 5: Graphical representation of the isolation room.

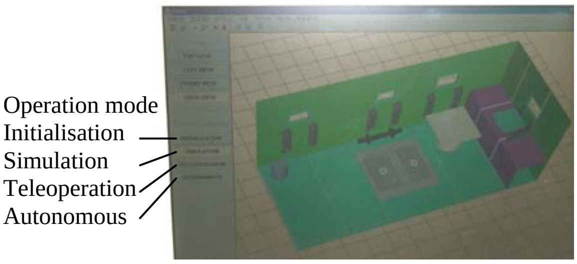


Figure 6: Modelled task environment of the isolation room generated by the graphic simulator.

3.2.2 Graphic Simulator

The shielding windows installed on the wall of the isolation room provide the human operator with inside information on the status of the isolation room. The human operator located outside the isolation room monitors the cleanup operation of ROCCS by relying on the visual information provided through the shielding windows. In this work, a graphic simulator was developed to provide the operator with added visual information and a more useful means for developing an improved ability to simulate and control ROCCS.

The graphic simulator mainly consists of a graphical model of ROCCS and a graphical model of the isolation room. Figure 5 shows the detailed graphical representation of the isolation room with its structural auxiliaries. The graphic simulator used Visual C++ as a programming language and OpenGL as an application programmer's interface. Figure 6 shows the modelled task environment of the isolation room, which is generated by the graphic simulator. The screen on the graphic simulator displays the useful tools for operating ROCCS.

This graphic simulator allows the human operator to simulate any desired cleanup tasks that ROCCS will perform inside the task environment of the isolation room. In addition the graphic simulator interfaces with the control system of ROCCS, thereby providing the operator with the additional visional information during an operation. The cleanup paths of ROCCS during an operation are displayed in a real-time on the graphic simulator. The operator can supervise and intervene in the cleanup operations of ROCCS through the graphic simulator without continuously seeing the interior of the isolation room through the shielding windows, thereby reducing the operator's physical and mental burdens. Consequently, the operator can have the inside operative information of ROCCS from both the shielding windows and the graphic simulator, enhancing the operational efficiency of ROCCS.

3.2.3 Cleanup Path Searching

For generating the cleanup path during an operation, ROCCS employs a left weight method and the A* algorithm. Either of these methods is selected before ROCCS searches a path for a cleaning. While searching a path, ROCCS might arrive at a position at which it can not find a path to move any further. When such a situation occurs, the A* algorithm is used to find an open space where a path searching has not been conducted as yet [4, 5]. If the A* algorithm can not find an open space, it is considered that the path search is complete. The path generated by the path searching scheme is composed of cells. ROCCS can not move easily if every cell on the generated path is used as a point for computing its movement. It is because the distance between two cells is very short compared with the dimension of ROCCS. A proper distance is specified as a minimum distance to be maintained by two adjacent points in order to avoid such an awkward movement of ROCCS. In this work, the cells that do not have the specified minimum distance are not used as points for computing the movement of ROCCS.

For the mockup test, it is assumed that the cleanup workspace of ROCCS is rectangular and there are no obstacles in it. The centre of the rectangular workspace is set to the origin of the coordinate system for positioning ROCCS. Twelve ultrasonic sensors installed around ROCCS are used to determine the position and heading direction of ROCCS. Values measured from two sensors which are located in an opposite direction are summed. The smallest and largest summed values then represent the lengths of the shorter and longer sides of the rectangular workspace, respectively. The differences between the values of the two sensors representing the lengths of

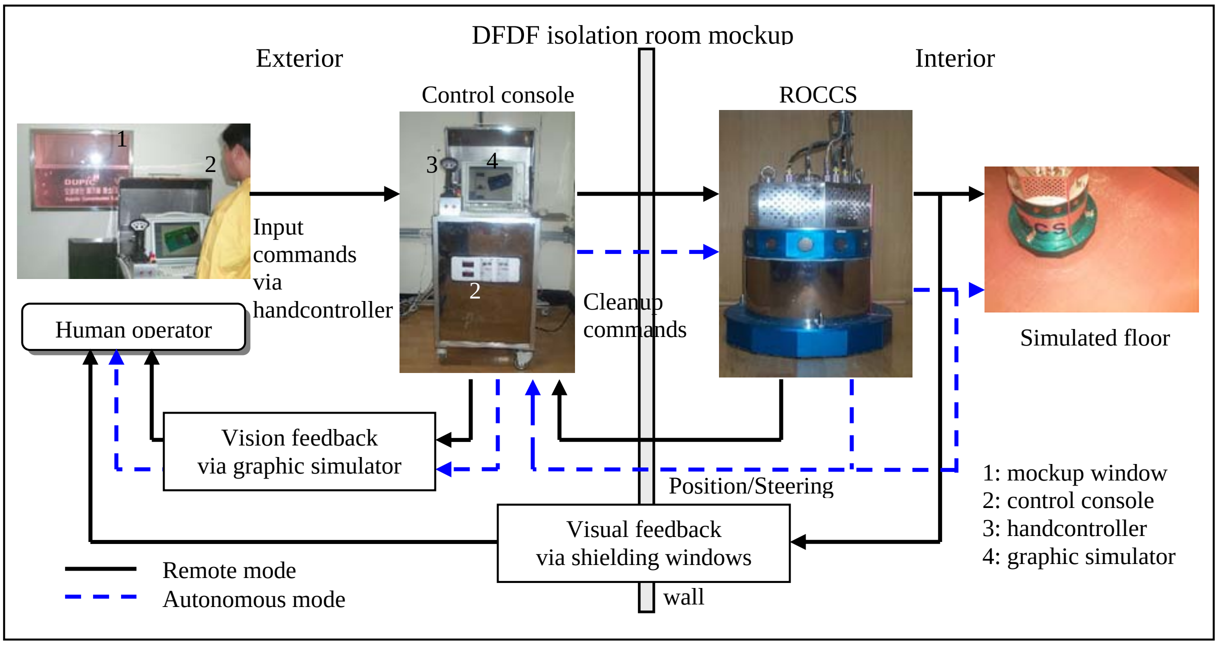
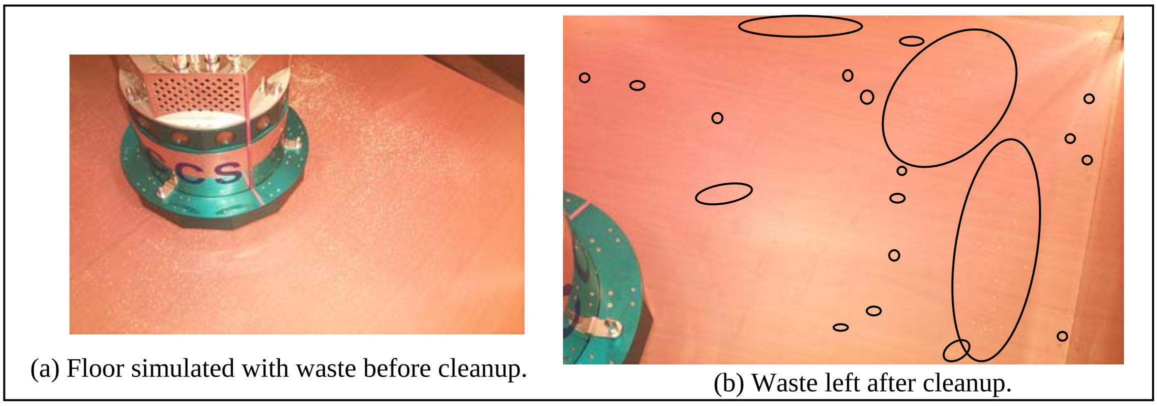


Figure 7: Functional connections of ROCCS for its mockup test.


(a) Floor simulated with waste before cleanup.

(b) Waste left after cleanup.

Figure 8: Cleanup operation of ROCCS in the autonomous control mode.

the shorter and longer side are used to compute the x and y coordinates, representing the position of ROCCS. The heading direction of ROCCS is also represented by an angle with respect to the coordinate system. The angle is computed with the current and previous position's coordinates of ROCCS.

3.2.4 Mockup Cleanup Test

ROCCS has been developed and tested to verify its performance and capabilities in a mockup of the isolation room. As with a real application, ROCCS is powered and controlled via a tether. ROCCS is operated either by a remote control or by an autonomous control. The mockup test environment and functional connections of ROCCS for the floor cleanup operation are shown in figure 7, where a solid line denotes the remote control, and a dashed line the autonomous control. The human operator and control console are located outside the mockup of the isolation room, and ROCCS is located inside it. In this work, only a part of the isolation room's floor was designed and used for the cleanup test of ROCCS in the mockup.

The graphic simulator is activated in both the remote and autonomous control modes, thus displaying the cleaning path of ROCCS in a real-time. The cleaning paths generated from the graphic simulator are also used as a command input to operate ROCCS. In the autonomous mode, ROCCS, at the initial stage, identifies its current position and orientation before it moves. ROCCS then moves to clean the floor following the path generated by the graphic simulator. In the autonomous mode, ROCCS, however, has limitations in its cleaning workspace in terms of the mobility because of the nature of the ultrasonic sensors installed on ROCCS, as shown in figure 8. Such an inaccessible area around the inside wall in the autonomous mode can be cleaned by switching the control to the remote control mode and by controlling ROCCS to move close to the wall edge, as shown in figure 9.

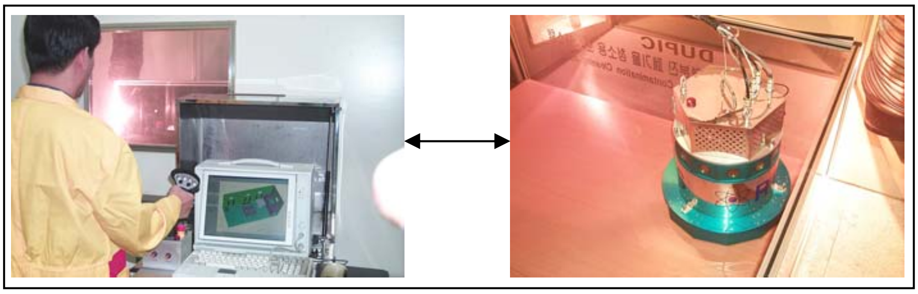


Figure 9: Cleanup operation of ROCCS in the remote control mode.

In the remote control mode, the operator manipulates, via a handcontroller, ROCCS to move it into the desired locations for a cleanup. The operator then controls the operable knobs to activate both the cleaning and collection module. In this mode, control command is transmitted unilaterally from the handcontroller to ROCCS. For both the remote and autonomous operations the cleaning path that ROCCS moves is always displayed on the graphic simulator in a real-time. Figure 10 shows the cleaning path generated from the graphic simulator during the mockup test in the autonomous mode.

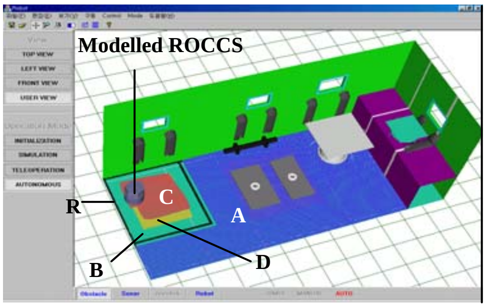


Figure 10: Cleaning path generated by the graphic simulator in the autonomous mode.

In figure 10, ***A*** denotes the entire area of the floor of the isolation room to be cleaned at the very beginning of the cleanup. The shape of a rectangle, ***R***, drawn by a bold line is the test area to be cleaned, and it corresponds to the rectangular floor of the isolation's mockup which is used for the cleanup test. ***B*** denotes the area that, in the autonomous mode, ROCCS can not reach. ***C*** denotes the completely cleaned area of the test floor – swept area. ***D*** denotes the area that ROCCS doest not pass through yet. As ROCCS moves, the trail that ROCCS followed on the floor is changed to a red colour, thereby differentiating the swept area ***C*** and the unswept area ***D*** clearly. In both the remote and autonomous modes, the operator can monitor all the cleanup operations of ROCCS through the mockup window and the graphic simulator.

From the mockup test, the results showed that ROCCS has a maximum speed of 0.1 m/sec, an ability to collect particles up to 0.3 um, as well as an ability to clean up a floor in all directions relative to the bottom of the cleaning module. Currently, ROCCS is under performance tests in order to acquire its reliability and stability before it is put into service in the isolation room during an overhaul period of the DFDF.

4 Conclusion

ROCCS developed in this work demonstrated its robotic applications for performing cleanup tasks in a highly radioactive zone of the DFDF. The significance of this development is that ROCCS can clean up a contaminated floor of the DFDF isolation room, thus eliminating a human interaction with its hazardous contaminants. The cleanup operation in the contaminated isolation room using the developed ROCCS will have the benefits of an improved worker safety, increased facility soundness, and a reduced personnel exposure dose rate.

5 Acknowledgements

This work was performed under the long-term Nuclear R&D programme sponsored by the Korea Ministry of Science and Technology.

6 References

[1] M.S. Yang, *Direct Use of Spent PWR Fuel in CANDU Reactor: DUPIC Manufacturing and Process Technology*, KAERI/RR-2022/99, Korea Atomic Energy Research Institute, Daejeon (2000).

[2] J. Borenstein and Y. Koren, "Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance", *IEEE Trans. Robotics and Automation*, 11, pp 132-138 (1991).

[3] K.Kim, H.S. Park, J.J. Park and M.S. Yang, "Remotely operated vehicle for collecting contamination in hot-cell", in D. Ruan, H.A. Abderrahim, P. D'hondt and E.E. Kerre (eds), *Intelligent Techniques and Soft Computing in Nuclear Science and Engineering*, World Scientific, Singapore, pp 535-542 (2000).

[4] T. Auyeung, "Popular micromouse algorithms, Part IV: A* algorithm", *Robot Science and Technology Magazine*, pp 14-19 (1999).

[5] S.J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (2nd edition), Prentice Hall, New Jersey (2002).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Design of an Evolutionary Controller and its Application

Kazuo Kawada[1] and Toru Yamamoto[2]
[1]Takamatsu National College of Technology, Kagawa, Japan /
[2]Graduate School of Education, Hiroshima University, Hiroshima, Japan
kawada@takamatsu-nct.ac.jp / yama@hiroshima-u.ac.jp

Abstract

PID control schemes still continue to be widely used for most industrial control systems. This is mainly because PID controllers have simple control structures, and are simple to maintain and tune. However, it is difficult to find a set of suitable control parameters in the case of time-varying and/or nonlinear systems. For such a problem, the robust controller has been proposed. Although it is important to choose the suitable nominal model in designing the robust controller, it is not usually easy. In this paper, a new robust PD controller design scheme is proposed, in which the suitable nominal model is designed using a real-coded genetic algorithm.

Keywords: real-coded GA, PD control, pole-assignment, controller design, sway control, crane

1 Introduction

In recent years, many complicated control algorithms can be developed owing to the rapid progress of electronic computers. Therefore, lots of advanced and intelligent control schemes have been considered. On the other hand, PID control schemes[1-3] have been widely employed to real control systems. The reasons are summarized as follows: (1) PID controllers have simple control structures. (2) It is relatively easy to understand the physical meanings of the control parameters. (3) It is possible to maintain and tune PID parameters based on operators' experience. However, it is difficult to find a suitable set of control parameters in case of time-varying and/or nonlinear systems.

For such systems, some design schemes of robust controllers have been already proposed, and the effectiveness has been discussed[4]. However, it is difficult to find the nominal model and the perturbed models in designing the robust controller. Especially, the control performance strongly depends on the nominal model. Therefore, it is necessary to consider the scheme so that the desired nominal model can be easily designed.

In this paper, a new robust PD controller design scheme is proposed for a nonlinear system. The necessary and the sufficient condition for the robust stability is usually employed in designing the robust controller. However, since it is difficult to design the nominal model suitably in the crane system to be considered in this paper, it is not utilized in this work. According to the newly proposed control scheme, some typical models for the controlled object are first designed, followed by the nominal model using the genetic algorithm (GA)[5, 6].

This paper is organized as follows. First, the mobile crane system as a controlled object is briefly explained, and the physical model is derived. Next, based on the pole-assignment scheme, the PD controller is designed using the nominal model parameters. However, it is difficult to design the nominal model as mentioned above. And also, the pole-assignment controller includes a user-specified parameter. Then, the nominal model and the user-specified parameter are suitably sought by employing a real-coded GA. Using the nominal model parameters and the user-specified parameter, the control parameters, *i.e.* PD parameters are calculated. Finally, the effectiveness of the newly proposed control scheme is experimentally evaluated on the sway control of a lifted load for a mobile crane.

2 Modelling of Crane System

2.1 Physical Model

The simulation model of a mobile crane system to be considered in this paper, is shown in figure 1. The control objective in this paper is to regulate the sway of a lifted load. The revolving super structure of a mobile crane, which affects the behaviour of a lifted load, is modeled.

In order to derive the differential equation, suppose the following assumptions:

1 The lifted load is a particle.

2 The jib and rope are rigid.

The differential equation of motion based on the jib angle ν, the jib rotaion angle ϕ, the sway angle θ and the sway rotation angle β of the lifted load is derived for the purpose of examining the behaviour of the lifted load during the jib derricking motion and the jib rotation motion. In Figure 1, m is the mass of a lifted load; L_J is the jib length and L_R is the rope length. The sway angle θ and the sway rotaion angle β of the lifted load, is controlled by manipulating the jib angular acceleration $\ddot{\nu}$ and the jib rotation angular acceleration $\ddot{\phi}$. The following differential equation of motion corresponding to figure 1 can be obtained:

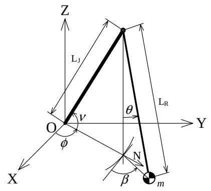


Figure 1: Simulation model of a mobile crane system.

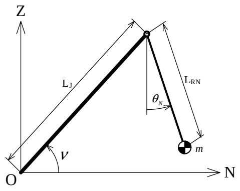


Figure 2: Simulation model of a mobile crane system in the normal direction to the jib revolution.

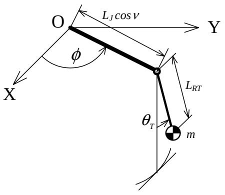


Figure 3: Simulation model of a mobile crane system in the tangent direction to the jib revoluion.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} A_{13} \\ A_{23} \end{bmatrix}. \tag{1}$$

The model shown figure 1 is considered by decomposing to the the normal direction and the tangent direction to the jib revoluion. Then, figure 2 and figure 3 show the simulation models of the mobile crane system in the the normal direction and the tangent direction to the jib revoluion, respectively. The following assumptions are introduced in order to linerize the differenctial equations corresponding to figure 2 and figure 3.

1 The sway of the lifted load that is controlled in the condition in which the jib angle ν and the jib rotation angle ϕ are so small. That is, the sway is controlled near the desired jib angle ν_d and the desired jib rotation angle ϕ_d.

2 The sway angles of the lifted load θ_N and θ_T are so small.

3 The jib angular velocity $\dot{\nu}$, the jib rotation angular velocity $\dot{\phi}$ and the sway angular velocity of the lifted load $\dot{\theta}_N$ and $\dot{\theta}_T$ are so small.

Then (1) can be simplified to

$$L_{RN}\ddot{\theta}_N = -g\theta_N + L_J\ddot{\nu}\sin\nu_d, \tag{2}$$

$$L_{RT}\ddot{\theta}_T = -g\theta_T + L_J\ddot{\phi}\cos\nu. \tag{3}$$

Moreover, the jib angle, the jib angular velocity, the jib angular acceleration, the jib rotaion angle, the jib rotation angular velocity and the jib rotation angular acceleration are related to the following equations:

$$\ddot{\nu} = \frac{d}{dt}\dot{\nu} = \frac{d^2}{dt^2}\nu, \tag{4}$$

$$\ddot{\phi} = \frac{d}{dt}\dot{\phi} = \frac{d^2}{dt^2}\phi. \tag{5}$$

2.2 Control System

Based on the physical model (2) and (4), the controlled object is described as the following state space model:

$$\left.\begin{aligned} \dot{\boldsymbol{x}}_N(t) &= \boldsymbol{A_N}\boldsymbol{x_N}(t) + \boldsymbol{b}_N u_N(t) \\ \boldsymbol{y}_N(t) &= \boldsymbol{C_N}\boldsymbol{x_N}(t), \end{aligned}\right\} \tag{6}$$

where $\boldsymbol{A}_N$ denotes the system matrix which depends on the rope length L_{RN}. $\boldsymbol{b}_N$ denotes the control vector which depends on the rope length L_{RN}, the jib length L_J and the desired jib angle ν_d. $\boldsymbol{C}_N$ denotes the output matrix. Thus, these matrices are given by the following equations.

$$\left.\begin{aligned} \boldsymbol{A}_N &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\dfrac{g}{L_{RN}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ \boldsymbol{b}_N &= \begin{bmatrix} 1 & 0 & \dfrac{L_J}{L_{RN}}\sin\nu_d & 0 \end{bmatrix}^T \\ \boldsymbol{C}_N &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}\right\} \tag{7}$$

$\boldsymbol{x}_N(t)$, $u_N(t)$ and $\boldsymbol{y}_N(t)$ are denoted the state variable vector, the control input and the output vector, respectively, which are given by

$$\left.\begin{array}{l} \boldsymbol{x}_N(t) = \left[\begin{array}{cccc} \dot{\nu}(t) & \nu(t)-\nu_d & \dot{\theta}_N(t) & \theta_N(t) \end{array}\right]^T \\ u_N(t) = \ddot{\nu}(t) \\ \boldsymbol{y}_N(t) = \left[\begin{array}{cc} \nu(t)-\nu_d & \theta_N(t) \end{array}\right]^T. \end{array}\right\} \quad (8)$$

By the similar way, the state space model with respect to the sway angle θ_T in the tangent direction to the jib revolution can be obtained as follows:

$$\left.\begin{array}{l} \dot{\boldsymbol{x}_T}(t) = \boldsymbol{A_T}\boldsymbol{x_T}(t) + \boldsymbol{b}_T u_T(t) \\ \boldsymbol{y}_T(t) = \boldsymbol{C_T}\boldsymbol{x_T}(t), \end{array}\right\} \quad (9)$$

3 Robust PD Controller Design Using a Real-Coded GA

3.1 PD Control Law

In the normal direction to the jib revolution, the pole-assignment of the closed-loop system is realized by the following state feedback control law:

$$\begin{aligned} u_N(t) &= -\boldsymbol{k}_N{}^T\boldsymbol{x}_N(t) \\ &= -k_{1_N}\dot{\nu}(t) - k_{2_N}\{\nu(t)-\nu_d\} \\ &\quad -k_{3_N}\dot{\theta}_N(t) - k_{4_N}\theta_N(t). \end{aligned} \quad (10)$$

Then, the following closed-loop characteristic polynomial $T_N(s)$, can be obtained:

$$T_N(s) = \det\{s\boldsymbol{I} - (\boldsymbol{A}_N - \boldsymbol{b_N}\boldsymbol{k_N}^T)\}. \quad (11)$$

Therefore, by designing the desired characteristic polynomial $T^*(s)$ and matching the coefficients so that $T_N(s) = T^*(s)$ is satisfied, the pole-assignment control system can be constructed, where $T^*(s)$ is the user-specified polynomial[7] given by

$$T^*(s) := s^4 + t_1 s^3 + t_2 s^2 + t_3 s + t_4, \quad (12)$$

where t_i is given based on minimizing the ITAE (Integral of Time squared Absolute Error) criterion, that is,

$$t_1 = \frac{5}{\sigma},\ t_2 = \frac{50}{3\sigma^2},\ y_3 = \frac{100}{3\sigma^3},\ t_4 = \frac{100}{3\sigma^4}. \quad (13)$$

In (13), σ corresponds to the convergence velocity, and is set by taking account of the regulation property.

(10) can be rewritten by

$$\begin{aligned} u_N(t) = &-k_{C1_N}\{(\nu(t)-\nu_d) + T_{D1_N}\dot{\nu}(t)\} \\ &-k_{C2_N}\{\theta_N(t) + T_{D2_N}\dot{\theta}_N(t)\}, \end{aligned} \quad (14)$$

where

$$k_{C1_N} = k_{2_N},\ T_{D1_N} = \frac{k_{1_N}}{k_{2_N}},$$
$$k_{C2_N} = k_{4_N}, T_{D2_N} = \frac{k_{2_N}}{k_{4_N}}. \quad (15)$$

By rearranging the control law (10) as PD control law (14), it is easy to comprehend the physical meaning of the control parameters. Furthermore, PD parameters can be expressed as the following equations by using system parameters L_{RN}, L_J and ν_d, and user-specified parameters $t_i (i = 1, 2, 3, 4)$:

$$\left.\begin{array}{l} k_{C1_N} = \dfrac{L_{RN}}{g}t_4 \\ T_{D1_N} = \dfrac{t_3}{t_4} = \sigma \\ k_{C2_N} = -\dfrac{g^2 - gL_{RN}t_2 + L_{RN}^2 t_4}{gL_J \sin\nu_d} \\ T_{D2_N} = \dfrac{L_{RN}(L_{RN}t_3 - gt_1)}{g^2 - gL_{RN}t_2 + L_{RN}^2 t_4}. \end{array}\right\} \quad (16)$$

3.2 Overview of Robust Control

First, (6)-(8) can be rewritten by the following transfer functions:

$$G_1(s) = \frac{1}{s^2} \quad (17)$$

$$G_2(s) = \frac{\frac{L_J}{L_{RN}}\sin\nu_d}{s^2 + \frac{g}{L_{RN}}}. \quad (18)$$

Then, $G_1(s)$ and $G_2(s)$ denote the transfer function of the jib angle system, and the transfer function of the sway angle system in the normal direction to the jib revolution, respectively.

Suppose the system parameters of the mobile crane, L_{RN}, L_J and ν_d, change within the following ranges:

$$0.3 \leq L_{RN} \leq 0.9[\mathrm{m}] \quad (19)$$

$$0.6 \leq L_J \leq 1.2[\mathrm{m}] \quad (20)$$

$$10 \leq \nu_d \leq 80[\mathrm{deg}]. \quad (21)$$

Then, note that the parameters included in $G_2(s)$ vary owing to the change of system parameters. For such a time-varying system, it is well-known that the robust control scheme is effective. According to the conventional robust control scheme, it is necessary to design the nominal model suitably in order to obtain good control performances. However, it is not easy to obtain it. Then, a new design scheme of the nominal model is proposed, in which the real-coded GA is employed. Furthermore, the robust controller is designed by choosing the suitable σ which satisfies the robust stability. Therefore, σ is also adjusted by using the real-coded GA. PD parameters are calculated using the nominal model parameters and σ. The proposed robust tuning scheme of PD parameters are discussed in detail in the following section.

3.3 Robust Tuning Scheme of PD Parameters Using GA

Outline

The schematic figure of the proposed genetic tuning scheme of robust PD parameters is shown in figure 4.

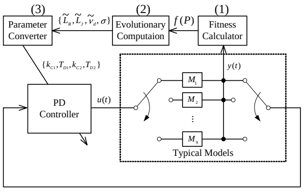


Figure 4: Schematic figure of the genetic tuning scheme of the robust PD parameters.

The upper part in figure 4 shows the procedure of the GA. On the other hand, the lower part shows the general feedback control system with PD structure. Some typical models (M_1 thru M_9) are first designed within the perturbed ranges shown in (19)-(21). These models are utilized in evaluating the fitness value in the GA. The parameters included in the typical models for $G_2(s)$ are summarized in table 1, and the gain plots corresponding to these models are shown in figure 5. Therefore, as the stability of the colsed-loop system is verified in all typical models, the robust stability is guaranteed even if the system dynamics is changed. Note that these typical models cover all purturbed ranges given by (19)-(21). Based on the sum of control errors for all typical models, the fitness values are calculated. Next, the evolutionary computation, *i.e.*, selection, crossover and mutation, is performed based on the fitness values. Furthermore, nominal model parameters $\tilde{L}_{RN}$, $\tilde{L}_J$ and $\tilde{\nu}_d$, and the user-specified parameter σ included in each individual are converted to PD parameters. After suitable iterations, PD parameters corresponding to the superior individual are adopted.

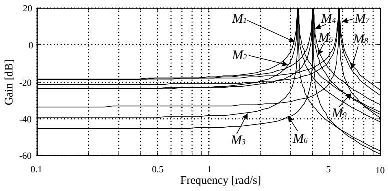


Figure 5: Bode diagram of typical models.

Table 1: System parameters included in typical models.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9
L_{RN}	0.9	0.9	0.9	0.6	0.6	0.6	0.3	0.3	0.3
L_J	1.2	0.9	0.6	1.2	0.9	0.6	1.2	0.9	0.6
ν_d	80	45	10	80	45	10	80	45	10

Real-coded GA

The nominal model parameters, $\tilde{L}_{RN}$, $\tilde{L}_J$ and $\tilde{\nu}_d$, and the user-specified σ are arranged as cells. The real-coded GA is employed, which is explained as follows.

1) Initialization
Set the number of generation, G, and product the initial individuals with real numbers at random within the perturbed ranges.

2) Selection
Calculate the fitness value $f(P)$, and select α percent individuals with superior fitness values. Here, the fitness function f can be freely designed by taking account of the desired control performance. α percent superior individuals are saved in the next generation.

3) Crossover
The $(100-\alpha)$ percent remains are generated by the crossover. Choose two individuals a and b among α percent superiors. Then, the new individual c is generated by

$$c = \frac{I(\max\{f(a), f(b)\}) + (a+b)/2}{2}, \tag{22}$$

where $I(\max\{f(a), f(b)\})$ means the individual with superior fitness value, *i.e.*, a or b. Note that this procedure is employed for every cell included in a and b.

4) Mutation
β percent cells for all new individuals are replaced with random values in the perturbed ranges.

The procedure from 2) thru 4) is repeated for generations which are specified in advance. Fig.6 shows the calculation flow of the evolutionary computation.

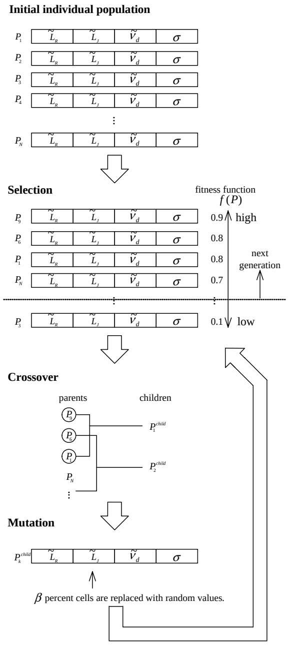


Figure 6: Calculation flow of the evolutionary computation.

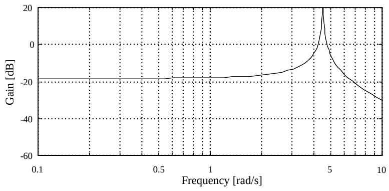


Figure 7: Bode diagram of the nominal model which is obtained by employing the GA.

Fitness function

The fitness function to be considered in this paper is given by the following equation:

$$f(P) = \frac{1}{\displaystyle\sum_{j=1}^{9} \int_{t_0}^{t_1} \tilde{y}_{Ni}^2(t)dt}, \tag{23}$$

where $\tilde{y}_{Ni}(t)(j = 1, 2, \cdots, 9)$ denotes the output signal corresponding to the typical model M_j. Furthermore, $[t_0, t_1]$ denotes the evaluation horizon.

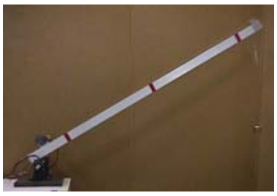


Figure 8: Experimental scale model of a mobile crane system.

By the similar way, the state space model with respect to the sway angle θ_T in the tangent direction to the jib revolution can be also obtained.

4 Control Results

The effectiveness of the proposed robust PD control scheme is experimentally evaluated on the sway control. Here, due to the page limit, the only control results in the normal direction are shown. The scale model is shown in Fig.8, which is prepared for experimentally evaluation. The jib angle and the sway angle of the scale model are controlled by manipulating the jib angular acceleration. The scale model have two DC motors as the actuators, and two rotary encoders as angle sensors.

The strategy of the GA and parameters are summarized as follows:

[Individual]
The individual is composed of 4 cells corresponding to $\{\tilde{L}_{RN}$, $\tilde{L}_J$, $\tilde{\nu}_d$ and $\sigma\}$. And also, the number of population, N, is set as 20.

[Selection]
The number of generation, G, is set as 200. And the rate of selection, α, is set as 50[%].

[Fitness function]
The fitness function shown in (23) is employed.

[Crossover]
The remainder, *i.e.*, 10 individuals, are generated by using (22).

[Mutation]
The rate of mutation, β, is set as 20[%].

[Seek range]
$0.3 \leq L_{RN} \leq 0.9$, $0.6 \leq L_{RN} \leq 1.2$,
$10 \leq \nu_d \leq 80$, $0.1 \leq \sigma \leq 1.0$

For the purpose of comparison, the conventional PD control scheme was employed, in which the nominal model parameters were set as $\tilde{L}_{RN} = 0.6$[m], $\tilde{L}_J = 0.9$[m] and $\tilde{\nu}_d = 45$[deg]. Furthermore, the user-specified parameter was set as $\sigma = 0.55$. These parameters were in the centre of the perturbed ranges. Then, PD parameters

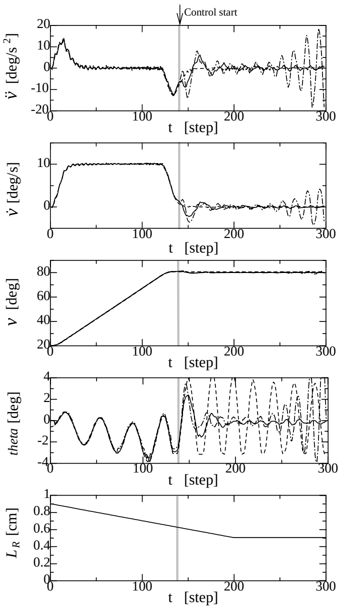


Figure 9: Experimental control results, where L_{RN} is changed from 0.9[m] to 0.5[m] and $L_J = 1.2$[m].

were calculated as $k_{C1} = 22.3$, $T_{D1} = 0.55$, $k_{C2} = 15.5$ and $T_{D2} = -0.19$.

On the other hand, by employing the above procedure of GA, the optimal nominal model parameters were obtained as $\tilde{L}_{RN} = 0.49$[m], $\tilde{L}_J = 1.1$[m] and $\tilde{\nu}_d = 46.3$[deg]. The gain plot of the nominal model with these parameters is shown in Fig.7. Furthermore, the user-specified parameter σ was obtained as $\sigma = 0.6$. Then, PD parameters were calculated as $k_{C1_N} = 12.8$, $T_{D1_N} = 0.6$, $k_{C2_N} = 8.3$ and $T_{D2_N} = 0.046$.

The experimental control results are summarized in Fig.9, where $L_J = 1.2$[m] and L_{RN} is changed from 0.9[m] thru 0.3[m] at the first 200 steps. And also, the jib angle ν is changed from 20[deg] to desired jib angle $\nu_d = 80$[deg]. The sampling interval is set as $T_S = 0.05$[sec]. In Fig.9, the solid line shows the proposed robust PD control result, the broken line shows the free response without the sway control, and the dashed and dotted line shows the conventional PD control result whose PD parameters are calculated based on the nominal model ($L_{RN} = 0.6$[m], $L_J = 0.9$[m] and $\nu_d = 45$[deg]) and the user-specified parameter . According to the newly proposed control scheme, it is clear that the sway control of the lifted load effectively works adequately after the jib derricking motion. From these control results, the effectiveness of the newly proposed robust PD control algorithm is verified.

5 Conclusion

In this paper, a new robust PD controller design scheme has been proposed, whose control law is derived based on the pole-assignment control scheme. It is important to design the nominal model suitably in designing the robust controller, because the control performance strongly depends on the nominal model. According to the newly proposed control scheme, the suitable nominal model is designed using the GA. The effectiveness of the newly proposed robust PD control scheme has been experimentally evaluated on the sway control system of a lifted load for a mobile crane. The main feature of the proposed scheme is that the necessary and sufficient condition with respect to the robust stability is not employed in designing the proposed robust controller. The control results in the tangent direction will be shown in that day.

6 References

[1] J. G. Ziegler and N. B. Nichols, Optimum settings for automatic controllers. *Trans. ASME 64*, pages759-768, (1942).

[2] K. L. Chien, J. A. Hrones and J. B. Reswick, On the automatic control of generalized passive systems, *Trans. ASME 74*, pages175-185, (1952).

[3] N. Suda, *PID control.* Asakura Publishing Co. Ltd., Tokyo (in Japanese) (1992).

[4] M. J. Chen and C. A. Desoer, Necessary and sufficient condition for robust stability of linear disturbed feedback systems. *Int. J. Control 35*, pages255-267, (1982).

[5] L. J. Eshleman and J. D. Schaffer, Real-coded genetic algorithms and interval-schemata. in L. D. Whitley (Ed.), *Foundation of Genetic Algorithms 2*, pages187-202, (1993).

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley, (1992).

[7] T. Shigemasa, Y. Takagi, Y. Ichikawa and T. Kitamori, A practical reference model for control system design. *Trans. SICE 19*, pages592-594, (in Japanese) (1993).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Autonomous Agents and Fuzzy Behavioural Maps: A Flexible Development Framework for Complex Behaviour

Ana M. Gonzalez de Miguel
Faculty of Computing,
Complutense University of Madrid, Spain
ana.gonzalez@fdi.ucm.es

Abstract

This paper explores the behaviour-oriented nature of intelligence and presents the definition and use of a flexible development framework called *Fuzzy Behavioural Maps* (FBMs) for providing complex adaptive behaviour to *Autonomous Agents* (AAs). The research work elaborates some of the fundamentals of the Knowledge-Based and Behaviour-Based approaches to Artificial Intelligence, identifies relevant contributions from AAs research and, summarises some recent implications and issues around the realisation of truly adaptive agents. The FBM-F represents a modular top-down approach to derive the behaviour-oriented models using generic FBMs, *levels of abstraction* and *refinement stages*. FBMs are appropriate tools for prototyping behaviour models and networks. The paper provides some relevant interpretations and uses of FBMs to develop large-scale AAs. The major scope of the FBM-F is to capture and model the behavioural dynamics of the agent at different levels of abstraction and, through aspect-oriented refinement stages.

Keywords: autonomous agents, behaviour-based models, fuzzy systems, complexity, software platforms.

1 Introduction

This paper presents the *Fuzzy Behavioural Maps Framework* (FBM-F, firstly introduced in [10]) as a generic methodology that allows the developers of *Autonomous Agents* (AA) to produce complex adaptive behaviour-based models in a flexible and efficient way.

In the last few years, there has been a wide interest in the development of intelligent agents inspired, largely, by the behaviour viewpoint (see, for example, [3]-[5], [10] and [15]. From the research carried out so far, it seems clear that in order to produce complex adaptive behaviour, the agents will need to acquire and develop their own knowledge, and the underlying computational models, architectures and methods will need to adapt to the demands of truly complex environments.

A number of Artificial Intelligence (AI) techniques have been proposed and explored as providing appropriate methods to build behaviour-based systems for adaptive agents (as explained, for example, in [15]). However, the development of complex models does not seem to be resolved using any of these approaches [4] and [5]). Indeed, the development of AAs appears to become unwieldy as the number and complexity of the behaviours and the interactions involved increase, and this leads to difficulties in the application of the AI techniques that make some behaviour models very difficult to foresee (see discussion [18] and [5]).

This paper presents our work to date exploring the role of our FBM-F as a *flexible development methodology* to try to handle complexity and adaptation more effectively. More generally, this research work represents our own understanding about how to develop *intelligent software platforms* to computationally support complex behaviours and related knowledge-oriented structures, focusing on five major aspects of “development” and “adaptation”:

a. the analysis of real adaptive agents,
b. the design of large-scale behaviour models,
c. the design of dynamic knowledge structures,
d. the design of computational models and,
e. the implementation of these using appropriate AI techniques and technologies.

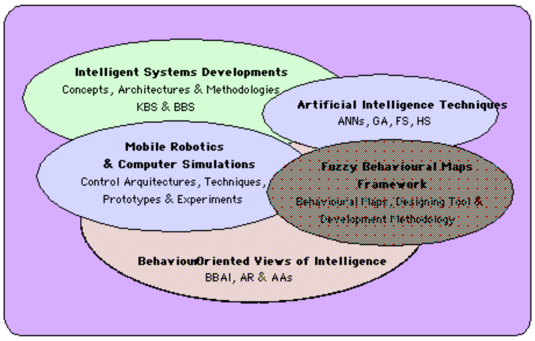
2 Objectives and Explorations

Our major research objective is to define and use a generic and flexible methodology to develop complex models for adaptive agents. More concrete tasks are:

- investigating the latest AI and AAs insights,
- identifying relevant contributions (ideas, tendencies and, perspectives) made by AAs research groups,
- underlying the role played by Computer Science,
- presenting our initial ideas for solving “complexity” and “adaptation” in an intelligent and effective way,
- analysing AAs in real environments,
- exploring and evaluating the key benefits of our FBM approach to design large-scale models,
- planning our future research work to produce and test a number of FBM architectures towards the development of more complex, and more natural intelligent systems.

Figure 1 illustrates our current investigations in AI and AAs. Basically, this research context includes:

- *Behaviour-Oriented Views of Intelligence* (as presented in [3]-[5], [10], [13]-[15] and [18]),
- *Intelligent Systems Developments*: Concepts, Software Architectures and Methodologies (as reviewed, for example, in [6] and [10]),
- *Mobile Robotics and Computer Simulations*: Control Architectures, Techniques, Prototypes and Experiments [3]-[6], [14]-[15],
- *AI Techniques* like Fuzzy Systems [6], [10], [19]-[20], ANNs [2], [12], Genetic Algorithms [11] or, Hybrid Systems [10], [20]) and,
- our *FBM-F* (see central nodes in figure 1).


Intelligent Systems Developments
Concepts, Architectures & Methodologies
KBS & BBS
Artificial Intelligence Techniques
ANNs, GA, FS, HS
Mobile Robotics
& Computer Simulations
Control Arquitectures, Techniques,
Prototypes & Experiments
Fuzzy Behavioural Maps
Framework
Behavioural Maps, Designing Tool &
Development Methodology
BehaviourOriented Views of Intelligence
BBAI, AR & AAs

Copyright© Ana M. Gonzalez de Miguel

Figure 1: Research Context.

The explorations into the behaviour-oriented nature of intelligence provide an overall view of what has been achieved and what remains to be done in order to prove the validity of the latest approach to AI to develop intelligent agents (see section 3).

Our key objective is studying "intelligence" by introducing and using non-conventional models. We investigate new types of problems, adopt new techniques and search for new solutions to design and implement the systems from a hybrid (knowledge and behaviour) viewpoint. One of the main advantages of using the behaviour-based approach is that no explicit model of the working environment of the agent is required to develop the intelligent systems. Thus behaviour-based AI can provide the designer of the software platforms with flexible techniques and methods for copying with the complexity of adaptive, autonomous agents.

Our FBM-F has been inspired, mainly, by the definition and use of *behaviour-oriented maps* [10] and, some *software architectures* [7]-[8] and *computational methodologies* [9] that we have designed in a number of technology-oriented projects like, for example, Mobile e-Services Solutions (e.g. [8]), Middleware and Software Platforms (like [7]) and, IT Service Management Solutions (e.g. [9]). In these projects, we have worked on truly complex functional requirements for financial services and mobile network operators. We have designed innovative service-oriented platforms and used some of the latest technologies. Further, we have recently analysed some effective ways for evolving and/or adapting these architectures into business markets and some scientific knowledge areas, adding more "intelligence" into our higher-level designs and definitions, respectively.

Nowadays, we are trying to bring the behaviour-based style into biological AAs and enterprise systems, as to integrate these with conventional AI [16]-[17]. In this paper, we provide some of our initial ideas around the suitability and usability of both (knowledge and behaviour) approaches to develop intelligent, adaptive agents using our FBMs.

3 Behaviour-Oriented Views of Intelligence

The goal of AI researchers has always been around the modelling and building of forms of intelligence. They have done much work to study intelligence by making models of human activities and, applying these models to construct the systems.

3.1 Intelligent Systems

Behaviour-based AI investigates the development of new types of intelligent systems (see e.g. [3], [4], [10], [13]-[15]) that have challenged conventional knowledge-based AI [16], [17] to construct AAs.

The research of behaviour-based AI communities reflects, mainly, into the field of Robotics, a benchmark area of AI for longer than thirty years. This practical work is being devoted to the investigation of robots (artificial creatures with physical bodies) able to perform some activities within their surroundings. These investigations lead to the institution of a school known as *Autonomous Robotics* that concentrates on the realisation of *autonomous robots* while making strong connections between the biological inspiration of the systems [4], [14], [18], the use of the "physical basis" of ANNs (e.g. [2], [12]) and other Artificial Life techniques [1].

The AI paradigm shifted in 1986 from the study and replication of high level knowledge-oriented forms of intelligence to the analysis and synthesis of lower level behaviour-based ones. Indeed, the literature reflects that, while the researchers of the "good-old-fashioned" AI focused on *knowledge models*, the recent communities of AI study *behaviour models*.

The researchers of the early AI associated the concept *intelligence* to the term *knowledge* [17]. Their work focused on the study of forms of intelligence such as "reasoning" or, "problem solving", motivated by the fact that computers could mimic human thoughts [16]. They built artificial intelligent systems by identifying, formalising, representing and processing some specific knowledge. They had many successful applications in areas where the level of machine intelligence could be measured in terms of the amount and significance of knowledge that the systems were able to store and manipulate (e.g. Game Playing). More recent AI communities use a concept of intelligence that is strongly related to the term “natural behaviour” (behaviour displayed by living organisms). Their work represents a general move towards the study of behaviour-based forms of intelligence such as *adaptation*, *learning* or, *interaction dynamics*, mainly, because of two general beliefs. First, the knowledge-oriented concept of intelligence is “ill-defined” in the sense that it does permit “viable machine intelligence with clear biological inspiration” [3]. Second, knowledge models and systems are not realistic enough to advocate AI. Nowadays, humans and other natural beings (animals) can be said to be intelligent not because they can store and process some knowledge but, because they can intelligently respond to their environments [4]. They can determine their own representations and investigate new (appropriate) ones. They are observed to adapt their actions to the changes of their environments and learn from this. Intelligence is determined by the dynamics of the interaction with the world [4], [18]. It is “an emergent property of the interaction of the physically embodied agent with the real world” [3].

3.2 Building Models

Building a knowledge model using, for example, a classical Sense-Model-Plan-Act architecture means constructing a system is such a way that, when it is placed in a determined situation (well-known environment) and, it is equipped with appropriate symbolic representations (goals and copies of the world), it can recognise certain aspects of its surroundings (sensing) and plan actions for achieving its goals (thinking) before performing these (acting). It is a matter of designing and implementing a system based on three specific-purpose units:

- a *sense unit* (that collects the symbols about the objective features of the environment),
- a *planner unit* (or central engine that delivers a *plan* according to the input coming from the sense unit and an internal copy of the environment) and,
- an *act unit* (that performs the set of actions delivered by the planner).

Further, the effort is concentrated on the logical operations of the planner unit. In fact, the system can only perform the required functionality if its planner unit “knows” how to correlate the sensor unit’s information with its internal, engineered copy of the outside world (see, for example, [17]).

By contrast, *building a behaviour model* means constructing a system in such a way that it can adapt its actions to (or learn from) the real world situations that it encounters. It requires solving issues such as how to design *behaviour-producing modules* (in a non-centralised and self-contained manner), and make these working within the real world situations, not how to symbolically represent and process some symbols to respond to a given (well-known) problem domain. The *behaviour-based architectures* are not intended to plan some actions according to an engineered environment. They have to be designed using *open architectures* because the ability of the system to adapt and/or learn depends on its capabilities to dynamically interact with the real world [3], [13]. They have to relate mutually to their natural (non-engineered) domains. They have to solve many problems at a time. Their designers need to worry about the “environmental pressures” [18].

3.3 Autonomous Agents

“Autonomous Agent” is a concept that has been proposed by researchers wishing to explore what acting intelligently means; a self-sufficient, adaptive and organised intelligent creature; a creature able to sequence its activities without need of human intervention; a concept that is associated (and sometimes identified) with many types of systems such as: software agents, robotic agents, cognitive structures and hybrid models.

More generally, we understand AAs as “a group of research communities” that investigate a more complex paradigm than that of behaviour-based AI. We believe that AAs represent the result of many years of investigations within a wide variety of disciplines (e.g., Computer Science, Cognitive Science, Neuroscience, Biology, Ethology, Developmental Psychology, Mechanical Engineering, Artificial Life) that look for the establishment of general theories about the study and construction of creatures (agents) able to act autonomously. Indeed, the literature reflects a group that appeals to a great diversity of ideas, formalisms and architectures that can be unified by the following common concerns:

- searching for general principles to analyse, design and build the biological models,
- trying to find new development methodologies and techniques (e.g. bottom-up methods or, *dynamical* systems techniques),
- defining and using new characterisations of agents (e.g. principled agent-environment interactions in robot systems and programmes) and,
- investigating the use of recent formal-frameworks like, for example, hybrid techniques,

in order to guide

- the study of living organisms: their morphology, underlying mechanisms and, abilities (innate and evolved behaviours),

- the experimental work with artificial creatures, focusing on the physical setups of robots,
- the investigation of behaviour-related capabilities such as: autonomy, self-sufficiency, cognitive control, emotions, motivations, learning, evolution, cooperation, evolution, etc. and,
- the design and synthesis of new artificial intelligent agents.

The general success of the behaviour viewpoint of AI requires some *cognitive*, *biological* and *technological* tendencies investigated within AAs. These tendencies are necessary to re-define concepts, ideas and methods to model and build complex forms of intelligence. AAs study a wide variety of creatures (e.g. animals, brain-like models or, cognitive structures) and help to smash the old dilemma of AI into a number of aspects of intriguing interest for the AI objectives. They investigate a very complex paradigm and work on the establishment of quite complete theories. They use scientific basis that are necessary to consolidate the validity of the behaviour viewpoint.

4 The FBM Framework

The FBM-F is an intelligent and flexible development framework of behaviour models for AAs. It provides an efficient way to identify, design and integrate behaviour-based models and knowledge-based structures. It helps to develop hybrid architectures.

4.1 Fuzzy Behavioural Maps

FBM [10] is a concept fundamentally motivated by: Cognitive Mapping Techniques (e.g. [12]), Non-Hierarchical Behaviour Architectures [3], [4], [15] and, Behaviour-Based Networks (like [13]).

These behaviour-oriented maps do not state explicit equations about the competences of the autonomous creatures. They are not alternative techniques to current algorithmic and/or dynamic approaches to design behaviour-based control mechanisms. FBMs are suitable tools for modelling behaviour-based systems before identifying any control architecture to implement these.

We have defined a number of FBMs with varying degrees of complexity. A simple *FBM* is a directed graph where the nodes represent *behaviour-producing modules* the designer is hoping to implement within an AA. It is a network that draws pictures about the behaviours of the agent and how the firing degrees of these can be causally related with *symbolic edges* showing either positive or negative cause-effect relations.

Every time we connect two behaviours with one of these edges, we represent a *causal relationship* between their degrees of activation. Further, we can associate a *qualitative label* (a weighting word such as *little* or, *more or less*). Thus, FBMs provide quantitative and qualitative data about causal relationships of their behaviour nodes.

The cause-effect connections provide a view of interaction between the behaviour-oriented nodes of the map. But the main idea is that both the levels of activation of the behaviour nodes and the inter-connections among these should be expressed in terms of Fuzzy Logic [19], [20] using, for example, *linguistic variables* and *fuzzy sets* or, *fuzzy inference mechanisms* to determine the level of activation of a particular node of a FBM.

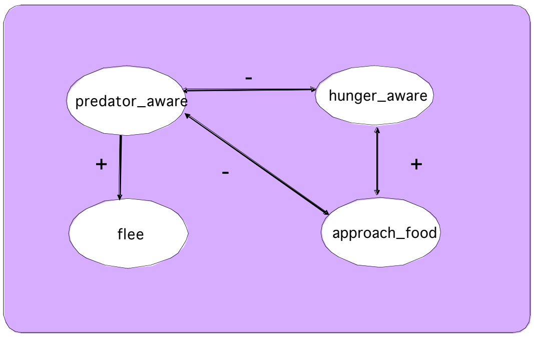


Figure 2: A Fuzzy Behavioural Map.

Figure 2 illustrates an example of a FBM. This map represents the following relationships among four behaviours of an agent:

- as the degree to which *awareness-of-hunger* increases, the tendency to *approach-food* increases,
- as *approach-food* is highly fired (the agent approaches food), *hunger* increases (through a sense of anticipation),
- as *hunger* increases (and food is approached), *awareness-of-predator decreases* and,
- as *awareness-of-predators* increases, the urge to *flee* rises, the *awareness-of-hunger* drops and the urge to *approach-food* falls.

Interpreting this diagram in Hopfield Net terms, it is not difficult to see that if at some instant, the node *hunger_aware* is activated, then *approach_food* becomes activated. If the *predator_awareness* node is then switched on, the *flee* behaviour is the only behaviour which is active.

The fuzzy interpretation is more subtle since we can add both quantitative and qualitative data to define the map. The nodes and edges of the map can be defined in terms of membership functions of fuzzy sets so that fuzzy granularity is added to the interpretation provided above. For example, if we represent the nodes with fuzzy sets and, *hunger_aware* is "highly" activated, the *approach_food* also activates in a "high" fashion since both nodes are connected using a positive edge. Using this fuzzy granularity (in nodes and edges) means that the influences among the behaviour nodes are fuzzy too. Therefore, the degree of activation of each node (e.g. *approach_food*) becomes affected by the communication degree among them.

4.2 Levels of Abstraction and Refinement

In essence, what all these behaviour-oriented modules (sub-modules) and interaction processes tell us is that we can address the overall performance of the FBM through a number of stages that might include (but are not limited to):

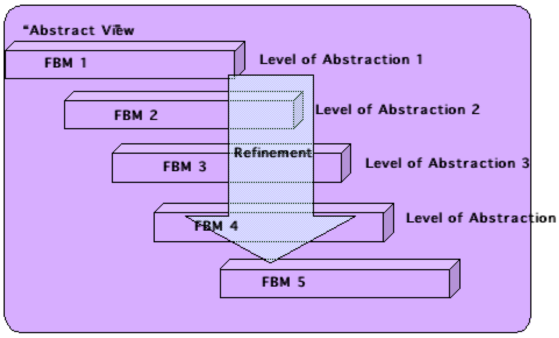
1. Wait for *firing conditions* from external components and modify degrees of activation accordingly
2. Drive the internal causal flow until *terminating condition* is reached
3. Wait for *behavioural performance of the agent* ("behaviour execution" that could be part of the computations of the map)
4. Drive *feedback connections* to external components and modify them (execute "consequences of behaviours") accordingly
5. Wait for the *completion of all consequences* of behaviour execution and, return to 0.

During step 1, the FBM is in a resting position (state). All its nodes have the potential of being activated or, causally effected, by the external components that determine the *current situation of the agent*. Then, once *something happens*, it starts driving the causal links that its edges represent and keeps on swirling until a terminating condition (e.g. maximum degree of activation in one of the behaviours) is reached. In the next step (2), the FBM waits for behaviour execution.

The key concepts explored with our FBM-F are: "level of abstraction" and "refinement stages". These extend the mapping of FBMs onto fuzzy architectures considering that it is possible designing FBMs at different levels of abstraction and, refining these by identifying lower behaviour-oriented design parameters rather than linguistic variables, fuzzy sets or, fuzzy inference mechanisms.

In figure 3, the first level of abstraction represents a *qualitative net* of behavioural nodes. It is a top-level, dynamic, and interaction-based model of *potential behaviours* based on the behaviour-oriented structures that we have identified for abstract, simple FBMs. The complexity of this (first) FBM model depends on the decisions made by the designer. But it is important to emphasise the fact that it is not intended to be a physical model, namely, because this is the purpose of the refinement stages that characterise the framework.

An FBM is abstracted from the selection of behaviours and the first (top level) interactions among these. The designer does not need to worry about what the behaviour producing modules should be (i.e. goal-attainment modules) or, what the agent is equipped with. Most behaviour-based systems are designed so that the morphological set up of the agent and the control actions (that are required to perform the required functionality) are identified first. This is not considered using FBMs as abstract views of behaviour systems. The morphological set up of the agent(s), control actions and other *parameters* can be identified in the latest stages of development.



Copyright© Ana M. Gonzalez de Miguel

Figure 3: The FBM Framework.

In practice, the designer of the FBM might map the behaviour nodes of the FBM onto specific tasks or, non-central control actions that allow the agent exhibiting the required functionality (e.g. we can model task-achieving behaviours). These computations can be time consuming and make the behaviour map stopping from further evolution. More importantly, all the processes must be completed in such a way that the causal flow and components of the FBM are appropriately changed.

We use the FBM-F to "map" some required behaviours into connection structures of behaviour-producing modules that are causally related. But the main idea is following as many refinement stages as required to complete the development process. They facilitate the definition of the behaviours that the AA designer selects from first levels of abstraction. Further, we can take the FBM-F into *implementation levels* that are required to build behaviour-based control architectures using some appropriate AI techniques and related standard technologies. In other words, the FBM-F allows the AA developer to produce large-scale behaviour models using any suitable computational technique and/or technology at the lowest levels of abstraction (at the implementation levels). The nodes and edges of a FBM can change from one level to another. Indeed, once the top-level behaviour maps have been refined, it might occur that the overall structure and its functioning change. FBMs help to adopt abstract views of behaviours as to design very low-level behaviour-oriented processes such as the sensor-motor activities. That is, we can start up the development process of behaviour models with high-level abstractions of behaviours (and their interactions) and, move down to lower level abstractions by a number of refinement stages.

5 Conclusions

The paper has outlined the definition of our FBM-F as a generic and flexible top-down development framework for AAs. The FBM-F has been presented as a contribution to knowledge in the modelling of large-scale behaviour models. This generic development framework has been described in terms of levels of abstraction and refinement stages that facilitate the identification and description of behaviour units and their interactions. The most important benefits of this work are:

- FBM-F allows flexible, modular and incremental building of intelligent behaviour models for AAs,
- FBM-F is AI technique independent (any AI technique could be applied) and,
- FBM-F is technology independent (any suitable technology could be incorporated at the appropriate level of abstraction).

6 Future Research Work

We are planning to continue proving the validity and efficiency of our FBM-F to develop complex adaptive behaviour. In this future work, we will implement several *computational models* and *case studies* for testing and evaluating the flexibility of our framework. This work will serve to emphasise the key benefits of our FBM-F as to facilitate the adaptation of autonomous creatures either in simulation or in enterprise systems. As part of this future work, we will be adding more intelligence into our FBM-F, solving truly complex and adaptive models. We will implement a number of behaviour maps, at different abstractions, to intelligently support the FBM models.

7 Acknowledgements

I am grateful for discussions and comments from my friends, colleagues and supervisors: P. Collingwood (SHU, UK), P. Vacher (SHU, UK) and R. Osborn.

8 References

[1] R.D. Beer, H. Chiel and L. Sterling, *A Biological Perspective on Autonomous Agent Design*, Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, edited by P. Maes, MIT Press, Cambridge, MA, (1994).

[2] N.K. Bose and P. Liang, *Neural Network Fundamentals with Graph, Algorithms, and Applications*, McGraw-Hill, New York (1996).

[3] R.A. Brooks, "Intelligence without representation", *Artificial Intelligence*, 47, pp 139-160 (1991).

[4] R.A. Brooks, "Artificial life and real robots", *Towards A Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, MIT Press, Paris (1991).

[5] R.A. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. Kemp, U.M. O'Reilly, E. Torres-Jara, P. Varshavskaya and J. Weber, "Sensing and manipulating built-for-human environments", *International Journal of Humanoid Robotics*, 1(1), pp 1-28 (2004).

[6] A.M. Gonzalez de Miguel, P. Vacher, P. Collingwood and R. Osborn, "An avoidance fuzzy logic controller for the mobile robot Khepera", *International Workshop on Advanced Robotics and Intelligent Machines*, University of Salford, Manchester, UK (1997).

[7] A.M. Gonzalez de Miguel, *J2EE Solution, Application Prototype and Development Methodology*, Hewlett Packard Consulting, Madrid (2002).

[8] A.M. Gonzalez de Miguel, *MES: A Mobile e-Services Platform for Telefónica Móviles España*, Hewlett Packard, Madrid (2003).

[9] A.M. Gonzalez de Miguel, *The Definition and Deployment of an ITSM Solution for Telefónica*, Hewlett Packard Consulting, Madrid (2003).

[10] A.M. Gonzalez de Miguel, *Explorations into the Behaviour-Oriented Nature of Intelligence: Fuzzy Behavioural Maps*, PhD Thesis, Sheffield Hallam University, UK (2003).

[11] J.R. Koza, *Genetic Programming*, MIT Press, Cambridge, MA (1992).

[12] J.R. Levenick, "NAPS: a connectionist implementation of cognitive maps", *Connection Science*, 3(2) (1991).

[13] P. Maes, "How to do the right thing", *Connection Science*, 1(3) (1989).

[14] P. Maes and R.A. Brooks, "Learning to coordinate behaviours", *Proceedings of AAAI-90: The American Conference on Artificial Intelligence*, Boston, MA, pp 796-802 (1990).

[15] M.J. Mataric, *Interaction and Intelligent Behaviour*, MIT EECS PhD Thesis, MIT Lab Tech Report AITR-1495 (1994).

[16] M. Minsky, *The Society of Mind*, Simon & Schuster, New York (1986).

[17] A. Newell, "The knowledge level", *Artificial Intelligence*, 18, pp 87-127 (1982).

[18] L. Steels, "The artificial life roots of artificial intelligence", *Artificial Life*, 1(1), pp 75-100 (1994).

[19] L.A. Zadeh, "Fuzzy sets", *Information and Control*, 8, pp 338-353 (1965).

[20] L.A. Zadeh, "The roles of fuzzy logic and soft computing in the conception and deployment of intelligent systems", *BT Technological Journal*, 14(4), pp 32-36 (1996).

Optimisation of Cooperative Motion for a 24-DOF Humanoid Robot using a Genetic Algorithm

[1]Il-Hwan Park, [1]Kook-Jin Choi, [1]Dae Sun Hong, [2]Man-Wook Han, [2]Peter Kopacek
[1]Department of Mechanical Design and Manufacturing Engineering,
Changwon National University, Changwon, Gyeongnam, Korea
[2]Intelligent Handling and Robotics,
Vienna University of Technology, Vienna, Austria
hanahwan@gmail.com, choi@changwon.ac.kr, dshong@changwon.ac.kr, e318@ihrt.tuwien.ac.at

Abstract

When a humanoid robot is to move in cooperation with external environment, it is essential to adequately control torques of each joint for achieving desirable robot motion. For such purpose, this study proposes a method to find an optimal posture of a humanoid robot using a genetic algorithm (GA) in such a way that the surplus torque ratio for all joints is maximised when the robot is pushing an object by hands. This method also adopts structure analysis by ANSYS for computing the force and moment exerted on the leg part while the desired joint angle and the generated torque for each arm and leg joint are calculated using forward and inverse kinematics and Jacobian. Then, such torques found are reflected to an objective function which is to be optimised, i.e. maximised, by a GA. As a result of successive maximisation process, an optimised posture having a large surplus joint torque is finally found. To show the effectiveness of the proposed method, a 24-DOF humanoid-robot's pushing motion is considered, and a number of simulations are carried out. The simulation result shows that the proposed method can be adopted to the control of torques for humanoid robot cooperative motion with external environment.

Keywords: humanoid robot, optimisation, genetic algorithm, cooperative motion

1 Introduction

Many people expect that humanoid robots can perform various works in addition to walking motion. In this case, humanoid-robot motion needs to be more complex and sophisticated, and the joints of the robots need to be more precisely controlled for achieving a desirable motion or trajectory. Generally, a humanoid robot consists of many joints, each of which is a combination of an actuator and reducer. So, when a robot performs a cooperative motion with external environment, it is essential to adequately control torques of the joint actuators of the robot.

As an example of a cooperative motion, this study considers a robot's action of pushing an object. If a torque required at any joint of a humanoid robot exceeds its maximum capacity for achieving a desirable task, the robot cannot perform a desirable motion or some joints might be damaged. From this point of view, our approach involves finding optimal postures that maximise the working capability of humanoid robots by controlling the joint torques in such a way that the surplus torque for all joints is maximised.

Until now, a number of research works showing a growing interest in humanoid-robot motion have been reported. The works [1-2] find optimal configurations in terms of the Cartesian force generation and energy consumption using nonlinear programming technique. However, the works are not directly related to the humanoid-robot research. The work [3] optimises configuration of the posture of humanoid robot by Simulated Annealing Algorithm. However, it is only applicable to the arm part of a humanoid robot. The study for developing a whole body cooperative dynamic walking control method [4] and effective balancing method [5] also reported, but they are restricted to stabilisation of walking mode. Recent successful development can be seen in mobile manipulation of humanoid robots [6-9] and optimisation of its posture using genetic algorithms [10]. However, most of the methods rely on approximate or restricted force information on a humanoid robot instead of precise information.

To provide realistic force information with humanoid-robot cooperative motion, this study proposes a method to find an optimal posture of a humanoid robot using a genetic algorithm (GA) in such a way that the surplus torque ratio for all joints is maximised when the robot is pushing an object by hands. Also, to assist structure, ANSYS programme is used. The desired joint angle and the generated torque at each arm and leg joint are calculated using forward and inverse kinematics and Jacobian, and the force and moment exerted on the leg part is calculated using ANSYS. The torques thus found are reflected on an objective function which is to be maximised by a GA. As a result of successive maximisation process, an optimised posture having a large surplus joint torque is finally found.

To show the effectiveness of the proposed method, a 24-DOF humanoid-robot's pushing motion is considered, and a number of simulations are carried out. The simulation result shows that the proposed method can be used to the control of torques for humanoid-robot cooperative motion with external environment.

2 Modelling and Analysis of Humanoid Robots

2.1 Degree of Freedom (DOF) and Coordinate System

This study considers a humanoid robot consisting of two arms and two legs. Each arm and leg has 6 DOF of motion, thus the robot has total 24 DOF of motion. Figure 1 shows the kinematic diagram of the robot assumed to be 1.2m in height and 36kg in weight.

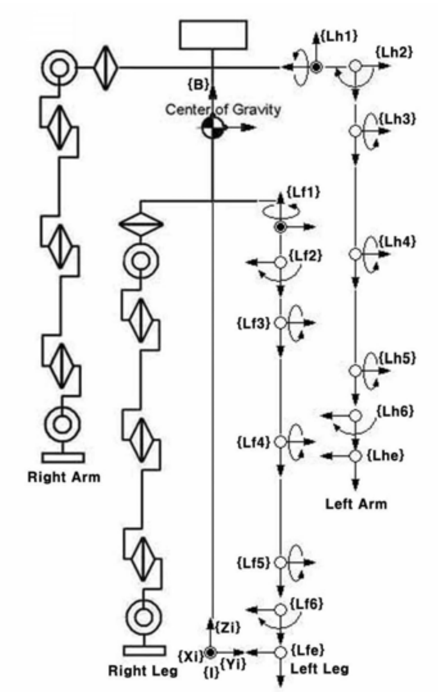


Figure 1: Degree of freedom and the coordinate system.

As shown in the above figure, the coordinate system is divided into four frames: {Lh} for the left hand, {Rh} for the right hand, {Lf} for the left foot and {Rf} for the right foot.

2.2 Forward and Inverse Kinematics

Forward and inverse kinematics is to be derived with respect to four parts: two arm parts and two leg parts. This study takes the homogeneous transformation consisting of the position vector about the origin of the reference frame, and the rotation matrix described by X-Y-Z Eular Angle. Each joint angle range is assumed to be given as

$$\begin{aligned}&-180 < (\theta_1, \theta_2, \theta_3, \theta_5, \theta_6) < 180,\\ &0 < \theta_4 < 180 \quad or \quad -180 < \theta_4 < 0.\end{aligned} \tag{1}$$

Here, the index i of each joint angle θi follows the same number of each arm and leg part shown in figure 1. θ_4 corresponds to each knee and elbow of the robot. Frame {B} is attached on the centre of gravity of the robot, so the homogenous transformation ${}^I_BT$, {B} with respect to {I}, can be defined. Also, each link transformations, ${}^B_0T$, ${}^6_eT$ is defined as the coordinate system. Here, the position and rotation of frame {0} is coincident with that of frame {1}. Thus, ${}^0_6T$ is given by

$${}^0_6T = {}^B_0T^{-1} \times {}^I_BT^{-1} \times {}^I_eT \times {}^6_eT^{-1}. \tag{2}$$

For obtaining the close-form solutions to the inverse kinematics, two methods are adopted: algebraic approach and geometric approach. In figure 1, each joint angle with respect to frame {Lh1}, {Lh2} and {Lh3} is obtained by algebraic approach, while each joint angle with respect to frame {Lh4}, {Lh5} and {Lh6} is easily found by geometric approach.

The obtained joint angles are verified by the MSC.visualNastran4D, and figure 2 shows the 3D-modelling of an initial posture by the software.

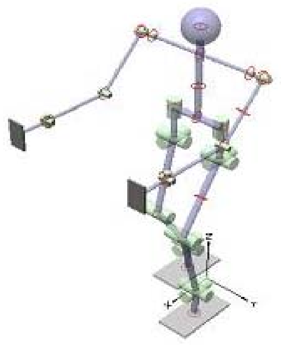


Figure 2: Modelling of the humanoid robot's initial posture using MSC.visualNastran4D.

2.3 Force and Torque Analysis for the Arm Part

To find the joint torques, Jacobian J relating joint velocity $\dot{\theta}$ to Cartesian velocity $\dot{X}$ is used on the basis of calculated angles at the static situation. The Jacobian is defined by

$$\dot{X} = J \times \dot{\theta}. \tag{3}$$

The Jacobian matrix is expressed by a 6×n (n: the number of joints) matrix: the upper three rows for linear velocity and the lower three ones for angular velocity. The Jacobian matrix deriving from the forward kinematics is expressed by the following equation:

Table 1: Parameters for the initial posture.

<table>
<tr><td rowspan="6">$\Theta_{ij,leg}$[°]
If i=1,
for right.
If i=2,
for left.
J=joint number.</td><td colspan="2">Θ_{11}, Θ_{21}</td><td colspan="2">0.0000</td><td colspan="2">0.0000</td><td rowspan="6">$\Theta_{ij,arm}$ [°]
If i=3,
for right.
If i=4,
for left.
J=joint number.</td><td>Θ_{31}, Θ_{41}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td colspan="2">θ_{12}, θ_{22}</td><td colspan="2">0.0000</td><td colspan="2">0.0000</td><td>θ_{32}, θ_{42}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td colspan="2">θ_{13}, θ_{23}</td><td colspan="2">14.6965</td><td colspan="2">-33.6212</td><td>θ_{33}, θ_{43}</td><td>45.0000</td><td>-45.0000</td></tr>
<tr><td colspan="2">θ_{14}, θ_{24}</td><td colspan="2">-48.3177</td><td colspan="2">48.3177</td><td>θ_{34}, θ_{44}</td><td>45.0000</td><td>-45.0000</td></tr>
<tr><td colspan="2">θ_{15}, θ_{25}</td><td colspan="2">33.6212</td><td colspan="2">-14.6965</td><td>θ_{35}, θ_{45}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td colspan="2">θ_{16}, θ_{26}</td><td colspan="2">0.0000</td><td colspan="2">0.0000</td><td>θ_{36}, θ_{46}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td rowspan="3">Θ_B [°]
For frame {B} about {I}</td><td colspan="2">α_B</td><td colspan="4">0.0000</td><td rowspan="3">P_B [m]
For frame {B} about {I}</td><td>P_{Bx}</td><td colspan="2">0.0750</td></tr>
<tr><td colspan="2">β_B</td><td colspan="4">0.0000</td><td>P_{By}</td><td colspan="2">0.0000</td></tr>
<tr><td colspan="2">γ_B</td><td colspan="4">0.0000</td><td>P_{Bz}</td><td colspan="2">0.7500</td></tr>
<tr><td rowspan="3">$\Theta_{e1,2}$ [°]
For frame {e} about {I} at leg</td><td colspan="2">α_{e1}, α_{e2}</td><td colspan="2">180.0000</td><td colspan="2">180.0000</td><td rowspan="3">$P_{e1,2}$ [m]
For frame {e} about {I} at leg</td><td>P_{e1x}, P_{e2x}</td><td>0.0000</td><td>0.1500</td></tr>
<tr><td colspan="2">β_{e1}, β_{e2}</td><td colspan="2">-90.0000</td><td colspan="2">-90.0000</td><td>P_{e1y}, P_{e2y}</td><td>-0.0800</td><td>0.0800</td></tr>
<tr><td colspan="2">γ_{e1}, γ_{e2}</td><td colspan="2">0.0000</td><td colspan="2">0.0000</td><td>P_{e1z}, P_{e2z}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td rowspan="3">$\Theta_{e3,4}$ [°]
For frame {e} about {I} at arm</td><td colspan="2">α_{e3}, α_{e4}</td><td colspan="2">90.0000</td><td colspan="2">90.0000</td><td rowspan="3">$P_{e3,4}$ [m]
For frame {e} about {I} at arm</td><td>P_{e3x}, P_{e4x}</td><td>0.5164</td><td>0.5164</td></tr>
<tr><td colspan="2">β_{e3}, β_{e4}</td><td colspan="2">0.0000</td><td colspan="2">0.0000</td><td>P_{e3y}, P_{e4y}</td><td>-0.2000</td><td>0.2000</td></tr>
<tr><td colspan="2">γ_{e3}, γ_{e4}</td><td colspan="2">0.0000</td><td colspan="2">0.0000</td><td>P_{e3z}, P_{e4z}</td><td>0.7086</td><td>0.7086</td></tr>
<tr><td rowspan="3">$F_{f1,2}$ [N]
$M_{f1,2}$ [Nm]
at leg</td><td>F_{f1x}, F_{f2x}</td><td>M_{f1x}, M_{f2x}</td><td>-80</td><td>-80</td><td>-7.2</td><td>7.2</td><td rowspan="3">$F_{h3,4}$ [N]
at arm</td><td>F_{h3x}, F_{h4x}</td><td>-80</td><td>-80</td></tr>
<tr><td>F_{f1y}, F_{f2y}</td><td>M_{f1y}, M_{f2y}</td><td>0</td><td>0</td><td>3.312</td><td>3.312</td><td>F_{h3y}, F_{h4y}</td><td>0</td><td>0</td></tr>
<tr><td>F_{f1z}, F_{f2z}</td><td>M_{f1z}, M_{f2z}</td><td>-180</td><td>-180</td><td>3.2</td><td>-3.2</td><td>F_{hz3}, F_{h4z}</td><td>0</td><td>0</td></tr>
<tr><td rowspan="6">$\tau_{ij,leg}$
[Nm]
i and j imply the same at θ.</td><td colspan="2">τ_{11}, τ_{21}</td><td colspan="2">3.2000</td><td colspan="2">-3.2000</td><td rowspan="6">$\tau_{ij,arm}$
[Nm]
i and j imply the same at θ.</td><td>τ_{31}, τ_{41}</td><td>11.3137</td><td>-11.3137</td></tr>
<tr><td colspan="2">τ_{12}, τ_{22}</td><td colspan="2">7.2000</td><td colspan="2">-7.2000</td><td>τ_{32}, τ_{42}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td colspan="2">τ_{13}, τ_{23}</td><td colspan="2">-12.6880</td><td colspan="2">12.6880</td><td>τ_{33}, τ_{43}</td><td>-11.3140</td><td>11.3140</td></tr>
<tr><td colspan="2">τ_{14}, τ_{24}</td><td colspan="2">-43.4501</td><td colspan="2">54.2588</td><td>τ_{34}, τ_{44}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td colspan="2">τ_{15}, τ_{25}</td><td colspan="2">-35.1880</td><td colspan="2">62.1880</td><td>τ_{35}, τ_{45}</td><td>0.0000</td><td>0.0000</td></tr>
<tr><td colspan="2">τ_{16}, τ_{26}</td><td colspan="2">7.2000</td><td colspan="2">-7.2000</td><td>τ_{36}, τ_{46}</td><td>0.0000</td><td>0.0000</td></tr>
</table>

$$
\begin{aligned}
{}^{0}_{6}J &= \begin{bmatrix} {}^{0}_{6}J_{v,leg-arm} \\ {}^{0}_{6}J_{\omega,leg-arm} \end{bmatrix} \\
&= \begin{bmatrix} {}^{0}_{1}Rk \times ({}^{0}_{e}d - {}^{0}_{1}d) & \cdots & {}^{0}_{6}Rk \times ({}^{0}_{e}d - {}^{0}_{6}d) \\ {}^{0}_{1}Rk & \cdots & {}^{0}_{6}Rk \end{bmatrix}.
\end{aligned} \tag{4}
$$

Using the principle of virtual work W, each joint torque is derived by the following equations:

$$
\dot{W} = \tau^{T} \times \dot{\theta}, \; F^{T} \times \dot{X} = \tau^{T} \times \dot{\theta}, \; \tau = J^{T} \times F . \tag{5}
$$

Here, F is the 6×1 force and moment vector applied at the robot with respect to frame {I}.

2.4 Force and Torque Analysis for the Leg Part

In general, the reaction force and moment applied at the leg part of a humanoid robot cannot be simply calculated when frame {B} is varied. So this study uses a structure analysis programme ANSYS. As the first step, the position of each joint angle with respect to frame {I} is determined from forward and inverse kinematics and is to be used for modelling by ANSYS. In the process of modelling, the acting force F=[-80 0 0] is applied on each palm and F=[0 0 -360] is applied on frame {B}, where two kinds of the force are presented with respect to frame {I}.

As a result of structure analysis, the reaction force and moment at frame {0} located on the robot's waist of leg-part is found. In the same way as the arm part, joint torques at the leg part are computed by such obtained force and moment. Figure 3 shows the modelling of ANSYS for the initial posture, while table 1 shows joint angles, joint torques, input force for each part and the position and rotation of frame {B} at the initial posture.

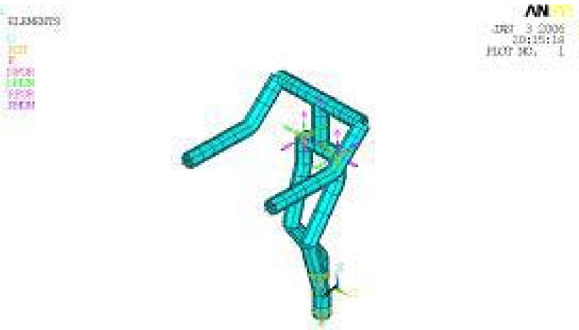


Figure 3: Modelling of the humanoid robot's initial posture using ANSYS.

3 Optimisation using a Genetic Algorithm (GA)

3.1 Parameters and Objective Functions

GA has been known to perform generally well when searching large spaces with many local optima, and formulates evolution process of survival by the fitness. This paper adopts a simple genetic algorithm (SGA) with roulette wheel selection, simple crossover and mutation. In addition, elitist strategy in which individual with the strongest fitness would be survived in the next generation is used for enhancing performance of the algorithm. Table 2 shows parameters used in this algorithm.

Table 2: Parameters of a genetic algorithm.

Parameters	**Value**
Maximum generation	300
Population size	100
Number of variables	6
Probability of crossover, Pc	0.8
Probability of mutation, Pm	0.01

To increase overall acting force through maximising the overall surplus joint torque ratio, the objective function F(x) is defined by

$$F(x)=\sum_{i=1}^{n}\left(1-\left|\frac{\tau_i}{\tau_{\max_i}}\right|^2\right). \qquad (i=1,\ldots,6) \qquad (6)$$

Here, τ_i is each joint torque and $\tau_{\max i}$ is maximum allowable torque of each joint.

3.2 Constraints in a GA

The following shows the constraints for achieving optimisation by the GA.

1. Each position and rotation of the palm and the sole is fixed at the initial posture.
2. The force acting on the palm and on the centre of gravity of the robot keeps constant.
3. The torque at each joint should be lower than its allowable maximum torque shown in table 3.

Table 3: Maximum allowable torques at joints.

Joint No.	**T_{max} [Nm]**	**Joint No.**	**T_{max} [Nm]**
Joint11(θ_{11}), Joint21(θ_{21})	57.8	Joint31(θ_{31}), Joint41(θ_{41})	57.8
Joint12(θ_{12}), Joint22(θ_{22})	189.8	Joint32(θ_{32}), Joint42(θ_{42})	57.8
Joint13(θ_{13}), Joint23(θ_{23})	189.8	Joint33(θ_{33}), Joint43(θ_{43})	57.8
Joint14(θ_{14}), Joint24(θ_{24})	227.8	Joint34(θ_{34}), Joint44(θ_{44})	57.8
Joint15(θ_{15}), Joint25(θ_{25})	227.8	Joint35(θ_{35}), Joint45(θ_{45})	57.8
Joint16(θ_{16}), Joint26(θ_{26})	189.8	Joint36(θ_{36}), Joint46(θ_{46})	57.8

Table 4: Allowable range of {B}.

Range of Px [m] ({B} of robot)	$0 \le Px \le 0.15$
Range of Py [m] ({B} of robot)	$-0.1 \le Py \le 0.1$
Range of Pz [m] ({B} of robot)	$0.5 \le Pz \le 0.8$
Range of Rx [°] ({B} of robot)	$-10 \le Rx \le 10$
Range of Ry [°] ({B} of robot)	$-10 \le Ry \le 10$
Range of Rz [°] ({B} of robot)	$-10 \le Rz \le 10$

4. The rotation and position of frame {B} should follow the allowable range shown in table 4.

3.3 Optimisation Algorithm

Figure 4 shows the flow chart of the proposed optimisation algorithm.

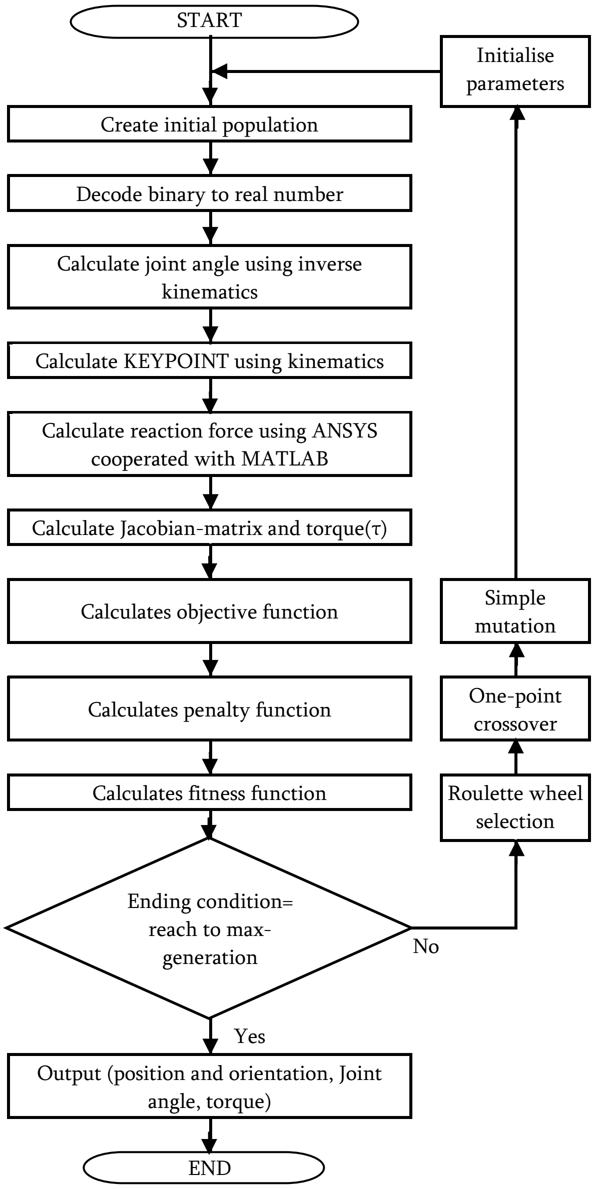

START
Initialise parameters
Create initial population
Decode binary to real number
Calculate joint angle using inverse kinematics
Calculate KEYPOINT using kinematics
Calculate reaction force using ANSYS cooperated with MATLAB
Calculate Jacobian-matrix and torque(τ)
Calculates objective function
Simple mutation
Calculates penalty function
One-point crossover
Calculates fitness function
Roulette wheel selection
Ending condition= reach to max-generation
No
Yes
Output (position and orientation, Joint angle, torque)
END

Figure 4: Flowchart of the proposed optimisation algorithm.

4 Simulation Results

Using the proposed approach, a number of simulations are carried out to show the effectiveness of the algorithm.

Figure 5 shows the change of the average fitness function and the maximum value of the objective function. The figure show that values tend to increase through the simulation.

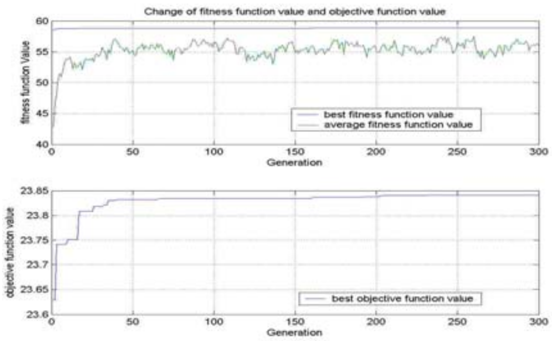


Figure 5: Change of the fitness function and the objective function.

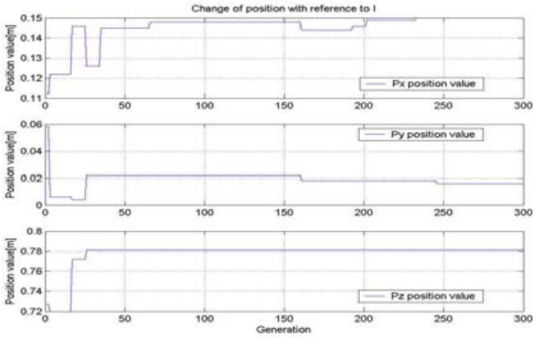


Figure 6: Change of position with reference to {I}.

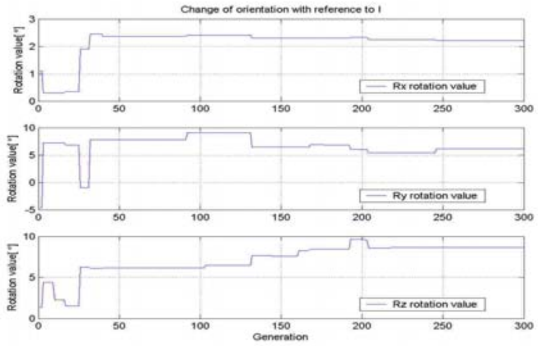


Figure 7: Change of rotation with reference to {I}.

Figures 6 and 7 show the position (Px, Py, Pz) and rotation (Rx, Ry, Rz) of frame {B} is varying in progress of generation with respect to frame {I}. The figure shows that values (Px, Py, Pz, Rx, Ry, Rz) tend to converge into a single value through the simulation.

Figures 8 and 9 show the comparisons between the initial joint torque ratio and the final joint torque ratio. Each joint torque ratio at the final posture is less than that at the initial posture, as shown in figure 8 and figure 9. This means that surplus joint torque at final posture is larger than that at the initial posture.

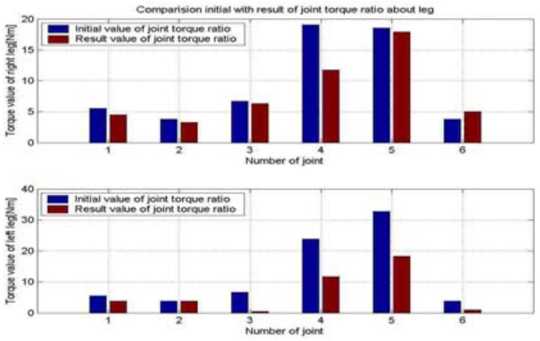


Figure 8: Comparison of the joint torque ratio between the initial and the final posture at the leg part.

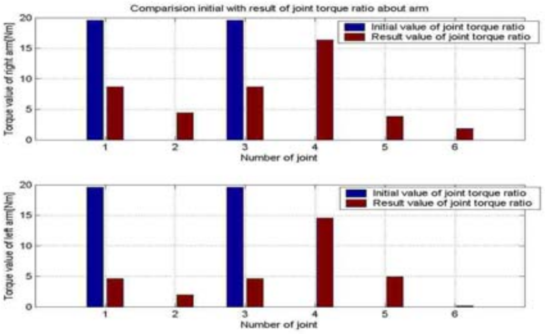


Figure 9: Comparison of the joint torque ratio between the initial and final posture at the arm part.

It can be seen that sum of the each joint torque is 300.9Nm at the initial posture while 207.5Nm at the final posture. This means the humanoid robot has more surplus torque by 93.3Nm than it of initial posture. Table 5 shows the simulation results at the

Table 5: Simulation results of optimal posture.

$\Theta_{ij,leg}$ [°] If i=1, for right. If i=2, for left. J=joint number.	Θ_{11}, Θ_{21}	9.2083	9.1767	$\Theta_{ij,arm}$ [°] If i=3, for right. If i=4, for left. J=joint number.	Θ_{31}, Θ_{41}	81.7883	-79.4531
	θ_{12}, θ_{22}	-5.1509	-4.8602		θ_{32}, θ_{42}	-12.7198	-10.9853
	θ_{13}, θ_{23}	-6.1560	-22.5192		θ_{33}, θ_{43}	-67.1485	48.0091
	θ_{14}, θ_{24}	-8.0116	24.7964		θ_{34}, θ_{44}	92.0997	-64.0467
	θ_{15}, θ_{25}	20.3628	-8.4696		θ_{35}, θ_{45}	-10.6342	-0.5651
	θ_{16}, θ_{26}	2.9610	2.6686		θ_{36}, θ_{46}	3.7247	1.9254
Θ_B [°] For frame {B} about {I}	α_B	2.22		P_B [m] For frame {B} about {I}	P_{Bx}	0.150	
	β_B	6.17			P_{By}	0.016	
	γ_B	8.65			P_{Bz}	0.781	
$\tau_{ij,leg}$ [Nm] i and j imply the same at θ.	τ_{11}, τ_{21}	2.6156	-2.2315	$\tau_{ij,arm}$ Nm] i and j imply the same at θ.	τ_{31}, τ_{41}	-5.0062	2.6511
	τ_{12}, τ_{22}	6.2494	-7.2037		τ_{32}, τ_{42}	2.5589	1.1189
	τ_{13}, τ_{23}	-12.0265	0.9187		τ_{33}, τ_{43}	5.0338	-2.6631
	τ_{14}, τ_{24}	-26.7286	26.5393		τ_{34}, τ_{44}	-9.4526	8.3913
	τ_{15}, τ_{25}	-33.9862	34.6487		τ_{35}, τ_{45}	-2.2310	2.8452
	τ_{16}, τ_{26}	9.5074	-1.7913		τ_{36}, τ_{46}	-1.0702	0.0776

optimised posture. Figure 10 shows the humanoid robot's final posture from the simulation results in table 5.

This result means that surplus torque can be enlarged as much as reduced. Thus, it is concluded that proposed method can be efficiently used to the control of torques for humanoid robots cooperative with external environment.

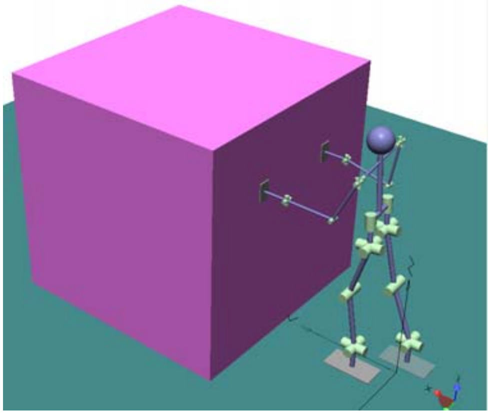


Figure 10: Optimised configuration as a result of the simulation posture.

5 Conclusion

It is essential to adequately control torques of the joint actuators of a humanoid robot for achieving desirable motion especially when the robot is to move in cooperation with external environment. For such purpose, this study proposed a method of optimisation of a humanoid robot cooperative motion using a genetic algorithm (GA).

The robot with 24 degree of freedom is considered, and forward and inverse kinematics is derived on the basis of modelling. Then, the torque of all joints is calculated from the relation of forward and inverse kinematics and Jacobian. Then ANSYS programme is used to find the acting force and moment at the leg part. Such obtained torques are reflected to an objective function which is to be minimised by a genetic algorithm (GA). Here, the position and the rotation angle of the centre of gravity of the robot are considered as the design variables. As a result of successive maximisation of the objective function by the GA, an optimised posture having a maximum surplus torque for the cooperative motion is finally found. And to show the effectiveness of the proposed method, a number of simulations are carried out. The simulation result shows that the proposed method can be adopted to the control of torques for humanoid robot cooperative motion with external environment.

6 References

[1] S. Masaaki, I. Mariko, M. Satoshi and S. Satoshi, "Optimal posture control for object recognition on active vision robot", *Proc. of the International Conference on Control Applications*, pp 1141-1146 (2004).

[2] F. Yuan and Qichao Yin, "Coordinating multilimbed robots for generating large cartesian force", *IEEE Transactions*, 20(4), pp 849-857 (1990).

[3] V. Riffard and P. Chedmail, "Optimal posture of a human operator and CAD in robotics", *Proc. of the International Conference on Robotics and Automation*, pp 1199-1204 (1996).

[4] Y. Jin'ichi, S. Eiji, I. Sadatoshi and T. Atsuo, "Development of a bipedal humanoid robot: control method of whole body cooperative dynamic biped walking", *Proc. of the International Conference on Robotics and Automation,* pp 368-374 (1999).

[5] T. Sugihara and Y. Nakamura, "Whole-body cooperative balancing of humanoid robot using COG Jacobian", *Proc. of the International Conference on Intelligent Robots and Systems,* pp 2575-2580 (2002).

[6] Y. Haruyuki, I. Kenji, A. Tatsuo and M. Yasushi, "Mobile manipulation of humanoid robots: a method of adjusting leg motion for improvement of arm manipulability", *Proc. of the International Conference on Advanced Intelligent Mechatronics*, pp 266-271 (2001).

[7] Y. Haruyuki, I. Kenji, A. Tatsuo and M. Yasushi, "Mobile manipulation of humanoid robots: optimal posture for generating large force based on statics", *Proc. of the International Conference on Robotics and Automation,* pp 2271-2276 (2002).

[8] I. Kenji, N. Yusuke, A. Tatsuo and M. Yasushi, "Mobile manipulation of humanoid robots: body and leg control for dual arm manipulation", *Proc. of the International Conference on Robotics and Automation,* pp 2259-2264 (2002).

[9] T. Tomohito, I. Kenji, S. Kotaro, M. Yasushi and A. Tatsuo, "Mobile manipulation of humanoid robots: control method for COM position with external force", *Proc. of the International Conference on Intelligent Robots and Systems,* pp 1180-1185 (2004).

[10] K.J. Choi, Y.K. Hwang and D.S. Hong, "Optimization of cooperative motion for humanoid robot using a genetic algorithm", *Proc. of the International Conference on Mechatronics and Information Technology,* pp 60426Q1-60423Q6 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Concurrent and Real-Time Task Management for Self-Reconfigurable Robots

Harris Chi Ho Chiu and Wei-Min Shen
Information Science Institute
University of Southern California, Los Angeles, USA
chichiu@usc.edu, shen@isi.edu

Abstract

We present a concurrent and real-time task management method for distributed control of modules in a self-reconfigurable robot. This method is essential for modules to simultaneously control multiple behaviours in real-time and supporting a concurrent programming style that will greatly ease the development of control software for large-scale reconfigurable robots. Although real-time operating systems and concurrent programming have been around for many years, it is only recently that the technology is miniaturised enough for embedded systems and self-reconfigurable robots. We have successfully implemented the method on our new SuperBot modules and have demonstrated its utility through the programming of locomotion gaits with the real modules.

Keywords: reconfigurable robots, real-time task management, concurrent programming

1 Introduction

A self-reconfigurable robot is a new type of modular robot that can change its shape, size and function to accomplish difficult tasks in dynamic and unforeseen environments. Such robots are made of many autonomous, intelligent, and self-reconfigurable modules that provide the basic units for the transformation. These modules are extreme examples of "design for reuse." For space applications, for example, one such robot can reconfigure its modules into an extended arm for extravehicular inspection and maintenance, or unloading payloads after landing. On the surface, the robot may morph into a set of mini-rovers to explore the environment, or a set of climbers to go down and up a crater, or become a single, large mobile platform to perform applications such as drilling, building, or sample collection. In many cases, such a robot may self-detect unexpected failures and repair malfunctions by reconfiguration of its modules.

Due to the fact that modules may reconfigure dynamically and have heterogeneous controllers, communication devices, sensors, actuators and connectors, the control of modules is a difficult problem. It must be flexible to satisfy both spatial and temporal constraints with limited computational resources. For the spatial constraints, the control must be *totally distributed* among modules. This is because centralised control is too vulnerable to module failures and partially distributed control must rely on certain non-local knowledge to function. An example for a partially distributed controller is a controller that requires unique global identifiers for modules. This would unnecessarily complicate the reconfiguration process because a trivial local modification in the system, such as a replacement of a single module, may cause many non-local changes that are expensive to implement. In this case, all non-local modules that use the replaced identifier must be informed to modify their programmes.

For the temporal constraints, the control of modules must be *concurrent and real-time*. This is inevitable for a self-reconfigurable robot because a single module is often required to simultaneously perform and manage multiple behaviours that are all time critical. Such behaviours include sensor reading, motor controlling, communication, synchronisation, and collaboration with other modules. The sequential programming style is inadequate for this purpose for several reasons. First, it is generally difficult to interleave multiple time-critical tasks correctly into a sequential programme. For example, consider two behaviours that must be performed periodically at every x and y units of time, respectively. To sequence them properly, a programmer must calculate every time point that is critical for interleaving them in a proper order. This can be very hard if x and y do not have common denominators. The resultant programme would be very long and extremely difficult to debug. Second, when there are many concurrent behaviours, say $\{b_1, b_2, \ldots, b_n\}$, to be managed, a programmer must consider all possible subset combination of these behaviours. This is very hard to manage when n is large. Third, interleaving time-critical behaviours is not always possible. This is especially true when the starting times or the duration of certain behaviours are not known in advance and must be dynamically triggered by external events that are beyond the control of the module. Both spatial and temporal constraints impose difficulties on programming global gait for all the modules. The spatial constraints require totally distributed control among modules, which limits a module to only communicate to its neighbouring modules. The temporal constraints add complexities in synchronising modules to perform global gait. The programming of global gait has to consider both the protocol for synchronisation and inter-module communication in addition to the timing control of various devices in individual module. Due to these difficulties, we believe it is absolutely necessary to have concurrent and real-time task management for programming self-reconfigurable robots with many modules.

To date, there is not yet any controller for self-reconfigurable robot that is totally distributed (i.e. ID free), and concurrent in real-time. A concurrent programming would bring many advantages for programming self-reconfigurable modules that are not possible in sequential programming. For example, it will greatly ease the addition or subtraction of time critical behaviours in a module's management portfolio. It will increase the robustness of the control software because behaviours can be organised into intuitive and modularised software units. It will enhance the adaptability of the control programme to different hardware because behaviours can encapsulate underlying hardware through a well-defined software interface. Furthermore, it will provide a natural interface for human operators to control the robot without being distracted by the detailed timing issues in the modules.

Inspired by the above motivation, this paper introduces a concurrent and real-time task management framework for behaviours/device control that is free from global unique identifiers. This method has been implemented and tested on a set of new self-reconfigurable modules called *SuperBot* that are developed for space applications. The utilities of this real-time programming framework are demonstrated through SuperBot locomotion where modules concurrently monitor and execute multiple behaviours simultaneously in real-time.

The rest of the paper is organised as follows. Section 2 discusses the related work for concurrent and real-time task management. Section 3 describes the complexity of task management for self-reconfigurable modules. Sections 4 to 6 describe the detailed notion of tasks and their real-time management and communication, as well as an implementation that divides tasks into the system level and the application level. Section 7 illustrates the utility of this approach by applying it to a new implementation of distributed control without assuming any global identifiers for the modules. Section 8 concludes the paper with future works.

2 Related Work

Although real-time operating systems [1] and concurrent programming [2] have been around for many years, it is only recently that technology is miniaturised enough for embedded systems and self-reconfigurable robots. Good examples of these small real-time operating systems include: QNX [3], LynxOS/BlueCat [4], TinyOS [5], XMK [6], NutOS [7], FreeRTOS, [8], and AvrX [9].

However, the current research in self-reconfigurable robot is mostly focused on hardware design and algorithm development. How to programme the modules effectively and efficiently is a relatively new topic. The current state of the art is to have a sequential, special-purpose programme for every different thing a modular robot needs to do, and this simple approach has served its purpose well so far because the complexity of behaviours of these modules are mostly single threaded.

CANbus was initially developed by Robert Bosch for in-vehicle data transfer and was defined in 1984. Silicon became available in 1987 and CAN was first used in cars in 1992.The draft international standard was introduced in 1991 and this became a full standard (ISO 11898) in 1994. Several self-reconfigurable robots have used this technology, but the drawback is that it has only small addressing space so the number of modules in the system must be limited.

Zhang et al. [10] have proposed a software architecture and implemented it on a real-time operating system for modular reconfigurable robots. However, inter-module communication requires a bus and therefore global identifiers are required for addressing. If a module is added or replaced, the control programme must be rewritten or the new module must be reprogrammed to retain the same global address. Shen et al. [11] provides a hormone-inspired, distributed control mechanism without global addressing by having point-to-point communication between neighbouring modules. Programme in every module is identical and can be replicated. Therefore, modules can replace one another without any non-local changes. This ensures robustness and scalability. However, the implementation of this method has been single-threaded and it is inadequate for more complex behaviours, where devices must be controlled simultaneously in real time.

3 Tasks for Reconfigurable Modules

To illustrate the complexity for task management of self-reconfigurable modules, we consider the SuperBot [12,13] as an example. Shown in figure 1, SuperBot is a new self-reconfigurable robot developed for NASA space applications. Figure 1 shows two individual SuperBot modules, and six connected modules in a human-like configuration. Each SuperBot module has six dock faces that can be connected to other modules. A single SuperBot module is designed to be totally autonomous, and it has power suppliers, controllers, motors, sensors, communication devices, and docking guidance mechanisms. A single SuperBot module can move and turn autonomously to any direction it desires to. It can travel on batteries up to 500 meters on carpet in an office environment. Multiple-module locomotion has also been demonstrated for a variety of configurations and gaits, such as a serpentine gait, a caterpillar gait, a butterfly stroke, rolling tracks, and gaits for legged configurations. Movies of these behaviours can be seen at the website http://www.isi.edu/robots/superbot/.

Inside a SuperBot module, there are batteries, two Atmega128 micro-controllers, nine motors, seven communication devices, six docking guidance mechanisms, and many types of sensors. The internal hardware architecture is shown in figure 2. As we can see, there are many devices that must be managed by a single SuperBot module.

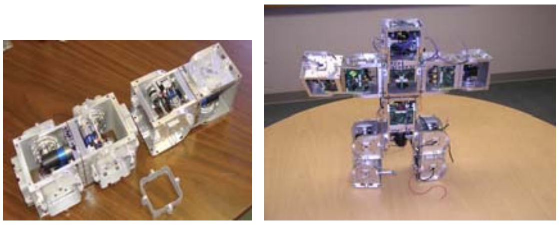


Figure 1: SuperBot modules and configurations.

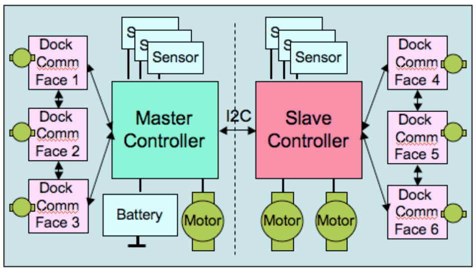


Figure 2: Necessary tasks for a SuperBot module.

To programme SuperBot module properly, software tasks must be created and performed. Each device has different timing requirements and therefore each device requires one task to manage. There are seven communication tasks, nine actuator tasks, nineteen sensor tasks and one or more behaviour tasks. Since many of these tasks are time critical, it is a nontrivial problem to manage them properly. Although it is possible to combine similar tasks into a single task and organise the tasks into a hierarchy, the complexity of managing them in a timely fashion is indeed a challenge. Furthermore, the onboard computational resource is highly limited due to the small energy budget and that increases the challenge even further. This analysis is not only true for SuperBot, but for any deployable self-reconfigurable modules in general. Thus, a concurrent and real-time management is essential.

4 Concurrent Tasks and Real-Time Management

In this paper, a concurrent task is a fundamental software unit for real-time management. Similar to a thread in an operating system, a concurrent task has its own stack and memory space for static variables and it appears to have a complete CPU control independent of other tasks. In our implementation, we have adopted the notion of task from AvrX.

AvrX [9] is an extremely small but powerful real-time kernel for Atmel AVR series of micro controllers, developed by L. Barello. AvrX contains approximately 40 API functions in six categories: Tasking, Semaphores, Timer Management, Message Queues, Single Step Debugging support, and Byte FIFO support with synchronisation. The Kernel is written in assembly and is extremely small footprint – about 1500 bytes in code size and about 30 bytes SRAM per task. AvrX is a real-time operating system and maintains state information for the each task. So a programmer can write each task in as sequential code as if the task has the CPU for its own. It is easier to develop independent modules that can be wired together later on.

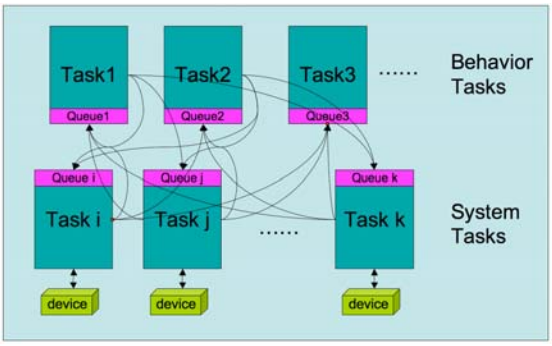


Figure 3: Tasks and their communication.

To manage the computational resources well in real-time, we use AvrX to allocate CPU resources among tasks. At the same time, AvrX also provides us a very useful tool for software modularity. It provides a common interface for communicating with different hardware devices and a common mechanism for handling timing constraints. We use AvrX's mechanism for tasking, semaphores, timer management and message queues. The speed is also sufficient, it takes less than 20ns for context/task switching when running on Atmega128 at 16 MHz with 128kbytes flash memory and 4kbytes SRAM. All tasks are written in C language. Associated with each task is a message queue. Tasks can communicate with each other by placing messages into each other's queue. Tasks can be set up to run periodically or to be run "on demand." A task receives a message from other tasks by checking its own queue as shown in figure 3.

To perform locomotion and reconfiguration successfully, both control software and hardware control have to meet real-time constraints. For example, a motor needs to be periodically monitored within certain period to hold its position while control software has to send out messages regularly to keep control information updated and synchronised.

Task-oriented real-time programming also helps to maximise resource utilisation. The scheduler runs tasks whenever the CPU is not busy. Timing can also be set independently in software modules. Execution conflicts among device software can be resolved systematically by using system utilities, like semaphores, timers and mutual exclusion (mutex). Therefore, real-time scheduler makes the change of timing constraints in tasks easier for the programmer, and devices can be controlled concurrently.

In our implementation, we divide the tasks in a module into two classes: the system tasks, and the behaviour tasks. A high-level distinction between these two classes is that a system task encapsulates and manages a hardware device and can be shared by many behaviours tasks. A behaviour task is to use the system tasks to control the module for locomotion, navigation, reconfiguration, and other functions. Behaviour tasks can reuse the system tasks without any reprogramming.

5 System Tasks

System tasks are low-level software modules that hide the details of low-level control of the hardware from the high-level behaviour software programmer. Figure 4 shows a simplified diagram of the tasks running on a SuperBot module, where two AvrX kernels, Master and Slave, are running on the two micro-controllers in each module.

The Master and Slave controllers use I2C serial communication to send messages to each other. The communication with other modules via the docks is handled by the IR tasks. Every dock has its own IR tasks, but for simplicity, the IR related tasks for only one dock on the Master and Slave are shown. Although the large number of tasks seems to add significant complexity, it actually minimises the time that the CPU is blocked waiting on a task or resource.

The handling of incoming data through IR and I2C is interrupt-driven. The sending and receiving of data is therefore wrapped into single task: in this case the task switching cost is expected to be higher than the cost of not being able to send and receive simultaneously. The motor task implements a PID controller, which is being executed every one millisecond.

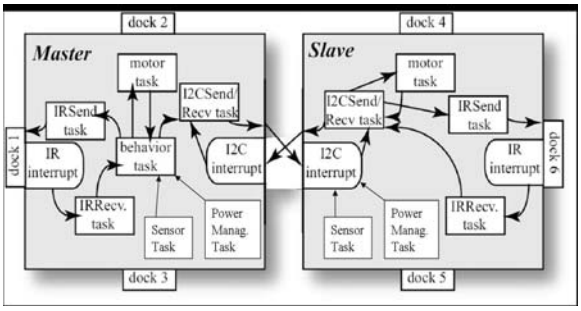


Figure 4: Tasks running on a SuperBot module.

The IR communication is much slower and tends to be noisier. For IR communication we have implemented the stop-and-wait ARQ (Automatic Repeat reQuest) protocol [14]. A task on a neighbouring module cannot directly send a message to a low-level task on a module, but only to a behaviour task. So a message received on any of the docks is routed to a behaviour task. If the destination task specified in the header of the message does not run on a receiving module, then the message is simply ignored.

For each communication link, there are two tasks for sending and receiving messages. The sending task checks its own message queue and only sends when there is message in the queue. For interrupt driven communication, received data is collected in the interrupt. A message using a dynamic memory buffer is formed and put into the message queue of the receiving task. The receiving task picks up messages from the queue and delivers it to destination task.

The sensor tasks are executed by the behaviour tasks. Once activated, a sensor task reports the status of the on-board sensors to the behaviour task directly or through I2C channel. The power management task is responsible for checking each connector current, the status of the battery, charging the battery, and set/resetting the power switches in each docking face.

The tasks for controlling the docking motors are not shown in this figure, and they are being developed along with the mechanics and electronics development for the connectors.

6 Behaviour Tasks

Behaviour tasks are responsible to run high-level control software using the functions provided by systems tasks. They have no direct access to the low-level devices but they call the interfaces provided by the system tasks. This provides an abstraction for programmers so that when they are writing the behaviour tasks, they do not need to worry about the details of timing the hardware devices. Thus, behaviour tasks are less time critical in general, and they are given a lower execution priority than the system tasks.

Behaviour tasks are typically designed for high-level jobs such as power management, locomotion, manipulation, navigation, and self-reconfiguration. Since modules may have heterogeneous hardware, it is important for behaviour tasks to be abstracted from the hardware details. For this purpose, behaviours tasks are generally written through a set of API functions that are implemented by the underlying system tasks. This way, behaviours tasks can be portable to different modules even though the implementation of these API may be different from module to module.

7 Distributed Control without Global-IDs

Due to the nature of self-reconfiguration, it is highly desirable to have distributed controllers that are free from any non-local knowledge (e.g. free from global unique identifiers for modules). This is essential for quick adaptation to dynamic and uncertain changes in the topology of reconfigurable robot. The real-time task management described in the paper can actually support this id-free approach for controlling a self-reconfigurable robot. The key idea is to provide messaging without the need of global addressing, thus enabling the development of high-level control to be independent from any specific arrangement of the modules.

To demonstrate the utility of the proposed concurrent and real-time task management framework, we describe one detailed example of a controller for the locomotion of a large centipede configuration that contains 100 modules without any global identifiers for the modules. This example will be done using the hormone-inspired distributed controller and the Adaptive Communication (AC) protocol described in [11]. To implement this control programme using the task management framework, we introduce three behaviour tasks: AC_SEND, AC_RECV and HORMONE, shown in figure 5. All modules in the robot will run the same three behaviours tasks.

```
The AC_SEND Task:
    Repeatedly probe neighbours to update the local topology.

The AC_RECV Task:
    Receive probes and update the local topology of the module.

The HORMONE Task:
    For each received hormone message:
        Select and execute the proper local actions based on
            (a) the local topology,
            (b) the local sensor inputs,
            (c) the local state/timer information,
            (d) the received hormone message;
        Propagate the hormone message to other connectors.
```

Figure 5: Pseudo-code for the hormone-inspired controller.

The AC_SEND task is to continuously update the connection information with neighbouring modules by periodically sending probe messages to all six dock faces through the corresponding system-level IR communication tasks. If a neighbouring module is present, then the AC_RECV task will receive a reply from a system-level communication task and thereby determine the local topology of the current module. The AC_RECV is continuously updating the local topology of the module and sends the result to the HORMONE task. Recall that IR communication is carried out by a system task that checks its queue and uses the IR device to send to its neighbour module. On the neighbouring module, the receiving task of corresponding communication interface will pick up the message and deliver it to the AC_RECV task. The AC_RECV task gets update from its queue and sends corresponding topological information to the HORMONE task. The HORMONE task also receives messages from sensor tasks and timer tasks at the system level (for items (b) and (c) in figure 5).

The HORMONE task checks its message queue for hormone messages from other modules and topology information from the AC_RECV task on the same module. Upon receiving a hormone message, the HORMONE task can determine what actions to perform and what hormone to propagate to neighbouring modules using a set of “receptor” rules. Actions can be executed by putting messages to the queues of corresponding actuator tasks and the propagation of hormone messages can be achieved by putting message into the message queues of the sending task of the desired communication interface. Since the AC_RECV task constantly updates the local topology based on the results of probing neighbours, and a module acts differently according to the four factors (items a-d in figure 5), the robot can run without reprogramming even when modules are reshuffled in the body.

Table 1: The action rules for a centipede.

Module Type	Local Timer	Received Hormone Data	Perform Action	Send Hormone
Head	0		Straight	[CP, A, {l,r,b}]
Head	0.5*Period		Straight	[CP, B, {l,r,b}]
Spine		A	Straight	[CP, B, {b}]
Spine		B	Straight	[CP, B, {b}]
Branch		A	Straight	[CP, B, {l,r,b}]
Branch		B	Straight	[CP, A, {l,r,b}]
Left Leg		A	Swing	
Right Leg		A	Holding	
Left Leg		B	Holding	
Right Leg		B	Swing	

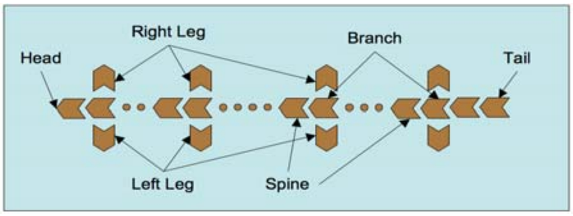


Figure 6: A centipede configuration of 100 modules.

Figure 6 shows an example of a centipede configuration with 100 modules. We assume each leg is one module long and the number of legs and the distribution of legs on the body are arbitrary. Each leg can either swing forward or hold in place to sustain the body. Notice that in this configuration, there are only six types of modules with respect to their local topology: head, tail, right leg, left leg, link torso, and branch torso. A *head* is a module that has only one connection and its back dock is connected to the front dock of another module. A *tail* is a module that has only one connection and its front dock is connected to the back of another module. A *right leg* (*left leg*) is a module has only one connection and its back dock is connected to the right (left) dock of another module. A *spine* is a module that has two connections, its front is connected to the back of another module and its back is connected to the front of a different module. A *branch* is a module that has four connections, its front, left, right docks are connected to the back of other modules respectively, and its back dock is connected to the front of another module. Each module can dynamically discover its own type, among the six local topologies described, by using the AC_SEND and AC_SEND tasks discussed in figure 5.

To control the movement of this centipede, the HORMONE task will use the rules listed in table 1. The hormone message is named as CP. We use set notation such as {l,b,r} as shorthand for sending a message to the left, back, and right dock, respectively. The action *Straight* means to hold all motors at 0 degree. The action *Swing* means to lift a leg module, swing the module forward, and then put the module down on the ground. The action *Holding* means to hold a leg module on the ground while rotating the hip to compensate the swing actions of other legs.

The first two rules indicate that the head module is to generate two new CP hormones with alternative data (A and B) for every cycle of period. This hormone propagates through the spine modules as it is, but alternates its data field (A↔B) at every shoulder module, and reaches the leg modules, which will determine their actions based on their types (left or right). The result is that every other left and right legs are swinging while the rest legs are holding, and if a left leg is swinging, then its corresponding right leg is holding, and vice versa.

This control mechanism is robust to changes in configurations. For example, one can dynamically add or delete legs from this robot, and the control will be intact. The speed of this gait can be controlled by the duration of period, which determines the frequency of hormone generation from the head module. Notice also that this set of rules works for configurations that have arbitrary pairs of legs and is scalable to large configurations.

8 Conclusion

This paper presents a concurrent, real-time task management framework for programming modules in self-reconfigurable robots. Compared to sequential programming, this method eliminates the difficulties of interleaving many time critical tasks that are necessary for the modules, modularises the software in modules, and eases the integration and modification of software and hardware components. Furthermore, it supports an ID-free distributed controller. The framework has been implemented on the SuperBot self-reconfigurable modules, and demonstrated its utilities in several control programmes for locomotion tasks. Our future work includes extension of the framework for reconfiguration, navigation, and other functions of self-reconfigurable robots.

9 Acknowledgements

This research is supported in part by NASA Cooperative Agreement NNA05CS38A, and in part by US Army Research Office under the grants W911NF-04-1-0317 and W911NF-05-1-0134. We are also grateful for other members in our Polymorphic Robotics Laboratory, especially Dr. Mark Moll and Dr. Behnam Salemi, for their useful comments on the drafts of the paper.

10 References

[1] R. Williams, *Real-Time Systems Development*, Butterworth-Heinemann, UK (2005).

[2] D. Bustard, *Concurrent Program Structures*, Prentice Hall, UK (1987).

[3] “QNX Real-Time OS”, http://www.qnx.com

[4] “LynxOS/BlueCat”, http://www.lynuxworks.com

[5] “TinyOS”, http://www.tinyos.net/

[6] “XMK - extremely minimal kernel”, http://www.shift-right.com/xmk/

[7] “NutOS”, http://www.ethernut.de/en/software.html

[8] “FreeRTOS”, http://www.freertos.org/

[9] L. Barello, “AvrX Real Time Kernel”, http://www.barello.net/avrx

[10] Y. Zhang, K.D. Roufas and M. Yim, “Software architecture for modular self-reconfigurable robots”, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, 4, pp 2355-2360 (2001).

[11] W.M. Shen, B. Salemi and P. Will, “Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots”, *IEEE Trans. on Robotics and Automation*, 18(5), pp 700-712 (2002).

[12] W.M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein and J. Venkatesh, “Multimode locomotion for reconfigurable robots”, *Autonomous Robots*, 20(2), pp 165-177 (2006).

[13] B. Salemi, M. Moll and W.M. Shen, “SUPERBOT: a deployable, multifunctional and modular self-reconfigurable robotic system”, *Proc. 2006 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Beijing (2006).

[14] “ARQ protocol”, http://www.erg.abdn.ac.uk/users/gorry/eg3567/arq-pages/saw.html

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Human Machine Cooperative Tele-Drive by Path Compensation and Evaluation of its Traversability

Yasuharu Kunii, Manabu Miyazaki, Masahiko Suzuki
Department of Electrical, Electronic and Communication Engineering,
Chuo University, Tokyo, Japan
kunii@hmsl.elect.chuo-u.ac.jp

Abstract

In operation environment with communication time delay, it's difficult to compose a closed loop control structure between a master and a slave system, and some scheme to achieve the stability is thus necessary. In this paper, we discuss and evaluate our proposed human machine cooperative tele-drive system that consists of global and local path-planning, for long range traversability. The operator can command any desired path as a sequence of waypoints by using a 3D terrain model measured as DEM by the on-board sensor of a rover. The data is transmitted to the ground and evaluated to obtain a dangerousness map. To cope with an unknown obstacle, a conventional autonomous path-planning algorithm is applied between each waypoint. In addition, a rover is continuously updating its knowledge about the environment. By continuously recalculating the difference between the original terrain data used for initial path generation on the ground and the most recent data acquired by the rover, waypoint compensation can be achieved. Therefore, we have to compensate waypoints by using the latest measurement data which can be assumed more reliable than previous data, for corresponding to the difference automatically. Here, the difference is assumed as the distortion between each data set, and compensated by using a distortion compensation matrix which is the mapping between the old and new terrain data sets. Simulation and experimental results and its evaluations by using the rover test-bed are mentioned in the paper.

Keywords: tele-operation, human machine cooperation, field robotics, planetary rover

1 Introduction

In 1970, the first unmanned mobile explorer called Lunokhod has just knocked on the door to the next stage of planetary missions, which was opened by Sojourner in 1997 and by MERs cruising the surface of Mars these days. [1][2][3] It was just the beginning of the new era of planetary surface explorations by using a rover for engineering and scientific applications. Toward the next step, we have to discuss and solve many research subjects. Efficient tele-driving scheme should be one of the most important topics for long range traversability on planetary surface. [4][5][6]

In operation environment with communication time delay, it's difficult to compose a closed loop control structure between a master and a slave system, some scheme to achieve the stability is thus necessary. The supervisory control system might be one of the solutions. A high-level supervisory control system also demands higher system performances, such as a computational power and sensors. However, in many cases actual space systems have difficulties to install high performance computers and sensors used on the ground, due to its problems of the harsh environmental system requirements, the weight controls for equipments and so on. Therefore, high-level autonomy will not be expected on the system, but one can accept the system based on human direct and continuous tele-control with some low-level intelligence on the rover.[6][7]

In this paper, we discuss and evaluate our proposed human machine cooperative tele-drive system consisted of global and local path-planning, for a long range traversability. The operator can command any desired path as a sequence of waypoints by using a 3D terrain model which was measured as DEM by an on-board sensor of a rover and transmitted to the ground. The received data is evaluated and a dangerousness map is generated. In order to react to and avoid an unknown obstacle, a conventional autonomous path-planning algorithm is applied between each waypoint. On the other hand, measurement data of some area is supposed to be more reliable, when a rover is getting closer to this area. However, the rover calculates the difference between terrain data sets used for initial path planning by the operator and data sets acquired by the on-board sensors of the rover. This means that the trajectory by waypoints is going to collide to an obstacle because of the difference. Therefore, we have to compensate waypoints by the latest measurement data, to correspond to this difference automatically. Here, the difference is assumed as the distortion between each data set, and compensated by using a distor-

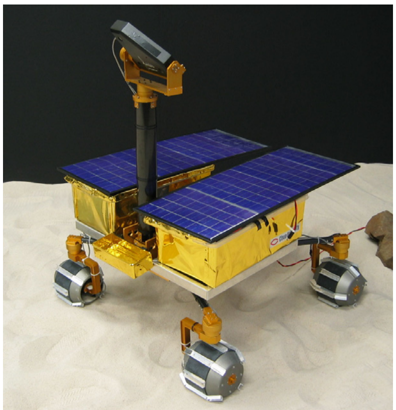


Figure 1: Micro5-05 rover test-bed system.

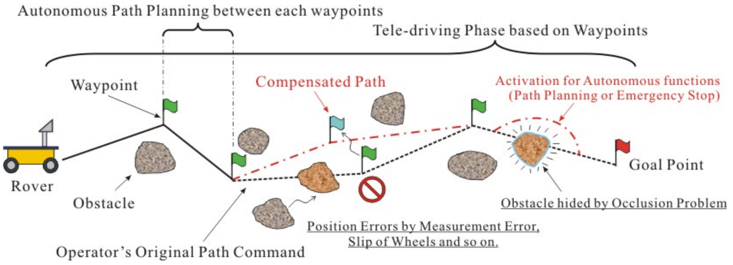

Autonomous Path Planning between each waypoints
Tele-driving Phase based on Waypoints
Waypoint
Compensated Path
Activation for Autonomous functions
(Path Planning or Emergency Stop)
Rover
Obstacle
Goal Point
Obstacle hided by Occlusion Problem
Position Errors by Measurement Error,
Slip of Wheels and so on.
Operator's Original Path Command

Figure 2: Tele-driving system for exploration rover.

tion compensation matrix which is the mapping between the old and new terrain data sets.

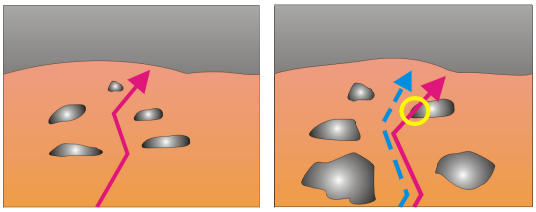
In section 2, the proposed tele-driving system is discussed along with the command path compensation algorithm. Simulation and experimental results and their evaluations by using the rover test-bed called Micro-5 which has a five-wheeled suspension and a gimbal sensor-mast as shown in figure 1, are discussed in sections 3 and 4.

2 Human Machine Cooperative Tele-driving

2.1 Tele-driving System for Planetary Rover

In the case of a remote environment time-delay occurs between the master and slave system due to their distance and the limited capacity of communication bandwidth. It is thus difficult to compose a closed loop control structure between a master and slave system. Conventional teledriving methods cause some strange behaviour called "Move & Wait" to a movement of a rover, because of the time-delay. A rover has to wait for commands while the operator's planning the path, which is carefully considered and as a consequence, is time consuming. Moreover, to avoid collision between the waypoint path and obstacles, a rover requests the operator to regenerate its waypoint path, which causes further delay until new path data is received. Therefore, some tele-driving scheme is necessary for efficient and continuous driving of the rover, which should be a low-level intelligence that can understand human intentions in the operator's path command for obstacle avoidance.[8][9]

Here, a human machine cooperative tele-drive system is discussed, consisting of a global and a local path-planning, for long range traversability, as shown in figure 2. The operator can create any desired command-path as a sequence of waypoint by using a 3D terrain model measured as DEM by an on-board sensor, transmitted to the ground station. A dangerousness map is then calculated using the received terrain data. However, the measured terrain model may include some errors and cause some problems. For example, generally, data measured by a sensor has proportional errors depending on the distance from a sensor to a measured target, and unknown obstacles might be found by moving on the path, because of an occlusion problem of sensors such as stereo cam. Of course, a rover itself also causes position estimation errors and dead reckoning errors, because of slips of wheels and so on. For corresponding to an unknown obstacle, a conventional autonomous path planning algorithm is a solution, and it can be applied for short range path planning between each waypoint (figure 2).



(a) old map (b) new map

Figure 3: Distortion compensation.

On the other hand, a rover is continuously updating the environment data set, and calculates the difference between original terrain data sets used for initial path planning done by the operator and the data sets acquired by the on-board sensors of the rover. The original path may result the rover to follow a trajectory that might cause a collision to obstacles, due to the difference between the distorted original and the more accurate, recently acquired data sets, as shown in figure 3. Therefore, we have to compensate waypoints by using the latest measurement data which can be assumed to be more reliable than previous data sets. This is because the measurement data of a certain area is more reliable, when a rover is getting closer to that area. Here, let's assume that the difference between those terrain maps is the distortion between data sets, and the path would be compensated by using a distortion compensation matrix which is the mapping between the old and new terrain data sets. This command path compensation and its compensation matrix will be mentioned in the next subsection.

2.2 Command Path Data Compensation

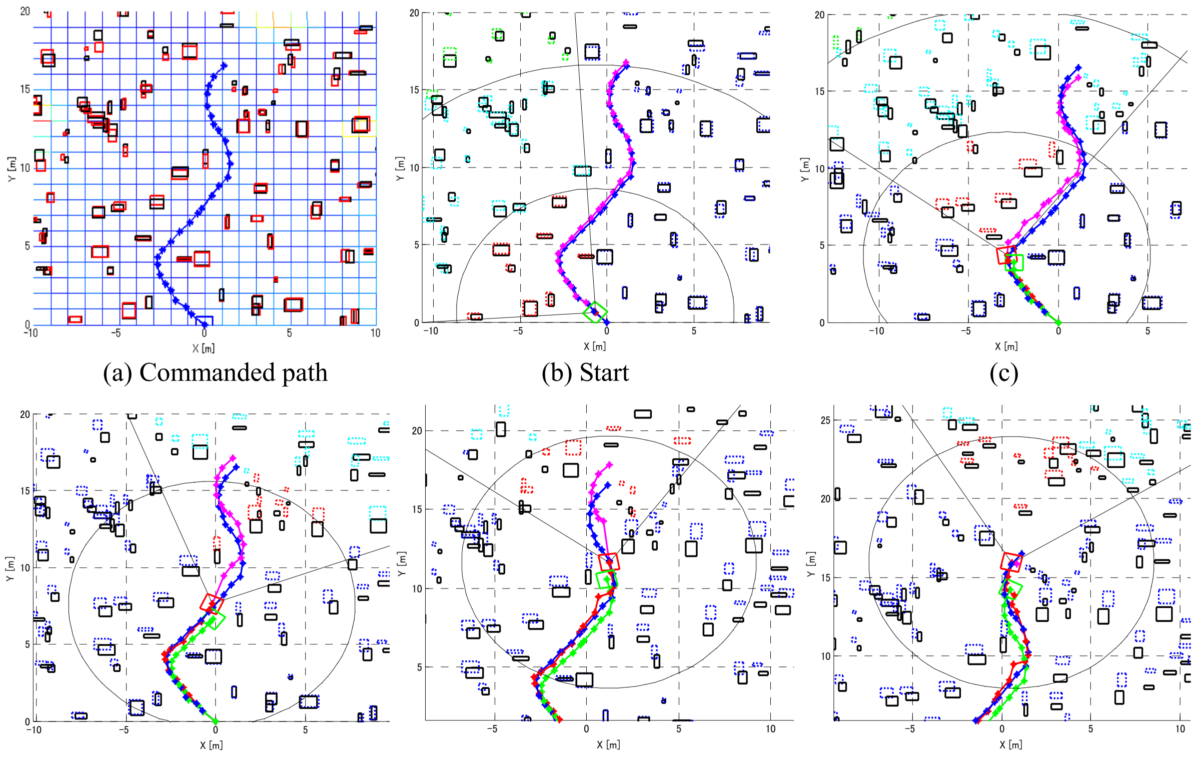
DEM data is a 2 dimensional scalar field with elevation values in each position *(x, y)*, according to the terrain elevation *z*. The map data distortion is due to the measurement error and the position estimation error of the rover, and can not be calculated using only one parameter and in many cases it is also non-linear. The compensation of 3D non-linear distortion, however, makes the on board algorithm of the rover more complicated. Therefore, in this paper we assume that the distortion has two dimensional and linear properties, and a distortion compensation algorithm of the camera-lens is applied on the compensation. The Command path Data Compensation (CDC) transformation is given by

$$\overline{X} = XA \tag{1}$$

where the rows of the matrix $\boldsymbol{X}$ are consisted of position data sets of landmarks(LMs) in the old terrain data, $\overline{\boldsymbol{X}}$ is also LM data of the latest environmental data, and $\boldsymbol{A}$ is the distortion compensation matrix. Actually, only three pairs of the LM data sets ($\boldsymbol{X}$ and $\overline{\boldsymbol{X}}$) are required for the acquisition of $\boldsymbol{A}$, if these measurement data have high accuracy. However, measurement data usually includes a nonlinear noise because of measurement errors and position estimation errors of the rover. So, a sufficient number of LM sets are required to obtain a suitable linear solution by using least mean square techniques.

Here the pseudo-inverse of X is used to obtain the matrix A, defined as

$$X^{+} = [X^{T} X]^{-1} X^{T} . \tag{2}$$



(a) Commanded path

(b) Start

(c)

(c)

(d)

(e) Goal

Figure 4: Simulation result of path tracking with command path data compensation.

Equation (2) is applied on (1) and the matrix $\boldsymbol{A}$ is given as

$$A = X^{+}\overline{X}\,. \tag{3}$$

Finally, the new compensated waypoint set is obtained from

$$Wp_{new} = Wp_{old}A\,, \tag{4}$$

where Wp_{old} is the waypoint data matrix created by the operator, and Wp_{new} is the compensated waypoint data matrix.

2.3 Simulation Results for Long Range Travers

Simulation results of the continuous and long-range tele-driving are shown in figure 4. The on-board sensor randomly has 10% of measurement error, proportional to the distance between a rover and each LM. The measurement area of the sensor is 10m long and 90° wide. The slip rate is 10%, proportional to the movement length of the rover. A rover is measuring the position of LMs and compensating the given path, at each waypoint. A rover is instructed to move on the blue curve, which is the original commanded path, shown in figure 4a. On the figure, the rover is the blue box initially located at the origin. In this simulation, the length of the command path is more than 20m at a time.

In figure 4b - 4e, the rover is located at the centre of the circle which indicates the sensing range of the on-board sensor. The two black lines indicate the field of view of the sensor. Black lined boxes are the obstacles at their real positions, boxes, which are dotted and coloured, indicate positions of measured obstacles. The green curve indicates the trace of the rover without any compensation method, and the rover is shown as the green box on the top of this curve. The rover on the compensated path is indicated by a series of red boxes over the red curve along the compensated motion path. The pink curve means a predicted and traveled result at this time. Finally, the rover is safely following the trajectory to reach the goal waypoint. In the final position of the rover the distance from the goal point is defined as final error. CDC tele-driving has proved to yield lower errors than conventional simple methods because of the efficiency of the proposed algorithm.

3 Feature Landmark Selection for CDC

During the path compensation process, the path might be changed beyond expectations, because measurement error on each LandMark (*LM*) is uneven and a larger error has more influence on the compensated path. Therefore, feature points for the compensation matrix should be chosen from *LM*s, by considering the influence on the movement of the rover, as follows:

1. *LM*s that are near to the rover.
2. *LM*s that may possibly collide with the rover.
3. *LM*s that have a proven reliability based on measurement history.

3.1 LM Selection based on Area

First of all, *LM*s are chosen from the existence area shown in figure 5, and then selected *LM*s are evaluated by the three criteria of distance, size and reliability. Figure 5 shows the structure of areas used for *LM* selection. *LM*s are selected from Area-1 & 2, and *LM*s in Area-1 have a higher priority. If a certain *LM* enters to Area-1, it is measured with high precision and is registered as a reliable *LM* in the measurement history list.

3.2 LM Selection Based on Weight Functions

3.2.1 Based on Distance from Rover

The distance from the rover is one of the most important factors, because measurement error is generally increasing by the distance from a sensor, and also a *LM* closer to the

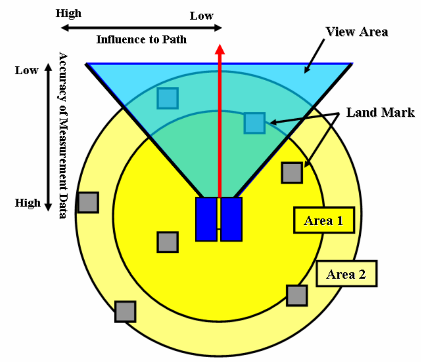


Figure 5: Landmark selection based on area.

rover has a higher risk of collision, and thus considered to be more dangerous. For example, a stereo imaging sensor has an error proportional to the square of the distance from the sensor to a measured target. Therefore, the reliability of *LM* is in inverse proportion to the distance. Let the distance from the rover to *LM* i be l_i ($i = 3 \cdots m$), and the weight parameter w_i be defined as $w_i = l_i^2$ in the case of the stereo imaging sensor. The weight matrix *L* is given as

$$L_{m\times m} = diag\left(w_1^{-1}, w_2^{-1}, \ldots, w_m^{-1}\right), \tag{5}$$

where m is the maximum number of feature *LMs*.

3.2.2 Based on Size of Obstacles

In the sensor-acquired data set, each *LM* is represented in a certain area due to its size and the measurement error. Therefore, the size of the *LM* should also be considered for the *LM* selection. Here, the weight matrix *S* is given using the ratio of the size s_i of each *LM* as

$$S_{m\times m} = \left(\sum_{i=1}^{m} s_i\right)^{-1} diag\left(s_1, \ldots, s_m\right) \tag{6}$$

3.2.3 Based on Reliability of Position

Once *LM* goes out of the sensing range, the uncertainty of measured *LM* position is increasing. It can be said that *LMs* kept being measured are more reliable. So it is necessary to use the *LMs* about which the rover has as much recent measurement data as possible. The reliability of an *LM* can be evaluated by using the passed distance l_{r_i} of the rover from the point where the *LM* had been measured for the last time. To select reliable *LMs* from the list, the weighting matrix *H* is introduced such that

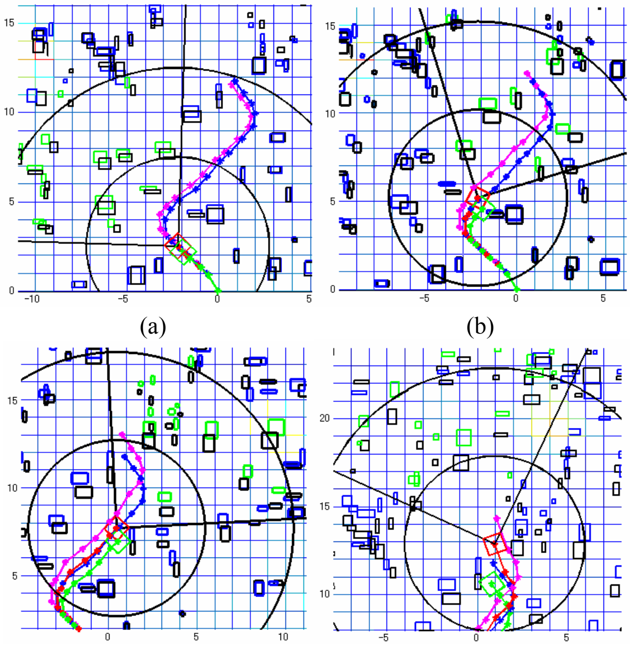
$$H_{m\times m} = \left(\sum_{i=1}^{m} l_{r\_i}\right)^{-1} diag\left(l_{r\_1}^{-1}, l_{r\_2}^{-1}, \ldots, l_{r\_i}^{-1}\right). \tag{7}$$

3.2.4 CDC by Using Selected Landmark

By using the above mentioned weight matrices, the total weight matrix for CDC is defined as

$$W_{m\times m} = \beta_1 L_{m\times m} + \beta_2 S_{m\times m} + \beta_3 H_{m\times m}, \tag{8}$$

where β_1, β_2 and β_3 are arbitrary design parameters. This weight matrix W is applied on (1), and the improved path compensation matrix is given as



(a) (b)

(c) (d)

Figure 6: Simulation result of CDC with LM selection.

$$W\bar{X} = W X A'. \tag{9}$$

The improved compensation matrix A' is derived as

$$A' = P^{+}\left(W\bar{X}\right), \tag{10}$$

where the matrix *P* is defined as *P*=*WX* and P^+ is the pseudo-inverse matrix of *P*. Finally, (10) is applied on (2) and the improved compensated way points are obtained.

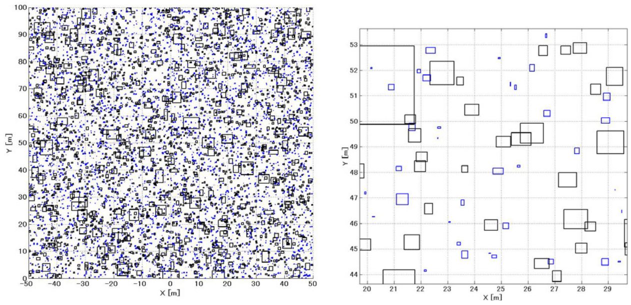
3.2.5 Simulation Results of LM Selection

The simulation result of the LM-selection with same condition of figure 4 is shown in figure 6. Obstacles located at the true position are shown as black boxes. Measured obstacles are given as blue boxes and green boxes which are also selected for the path compensation algorithm. The red box and the green box are rovers applied with and without the LM selection, respectively, than, the red line and the green line are trajectories yielded by each algorithm. The pink line is a commanded path generated by the operator, while the blue one is a compensated path. The path improved by the proposed method with LM selection almost coincides with the commanded path, and it successfully reaches to the goal point. Therefore the effectiveness of the LM selection and the path compensation algorithm are confirmed.

4 Evaluation of CDC Tele-driving Method

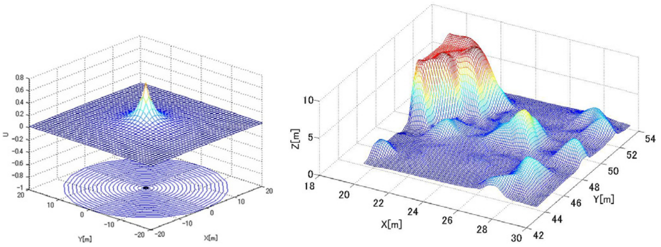
In above sections, CDC Tele-driving method already indicates its performance. Here it is analysed more precisely by simulation. Figure 7 shows virtual lunar field generated by Monte Carlo method based on rock distribution profile measured by Lunar Surveyor 7 in 1968[10]. Ten test fields are randomly extracted from figure 7a, for example figure 7b, and used for performance evaluations.

An operator creates command path on each test field and



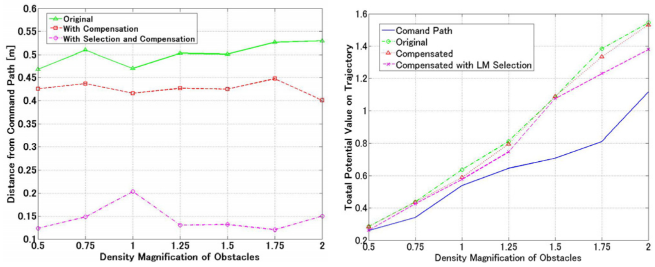
(a) virtual lunar field (b) sampled simulation field

Figure 7: Virtual field based on data of Surveyor-7.



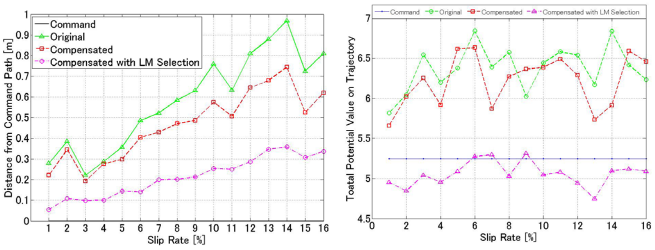
(a) electrical charge (b) test field

Figure 8: Potential field for trajectory evaluation.



(a) distance from command path (b) potential value

Figure 9: Evaluation based on density of obstacles.



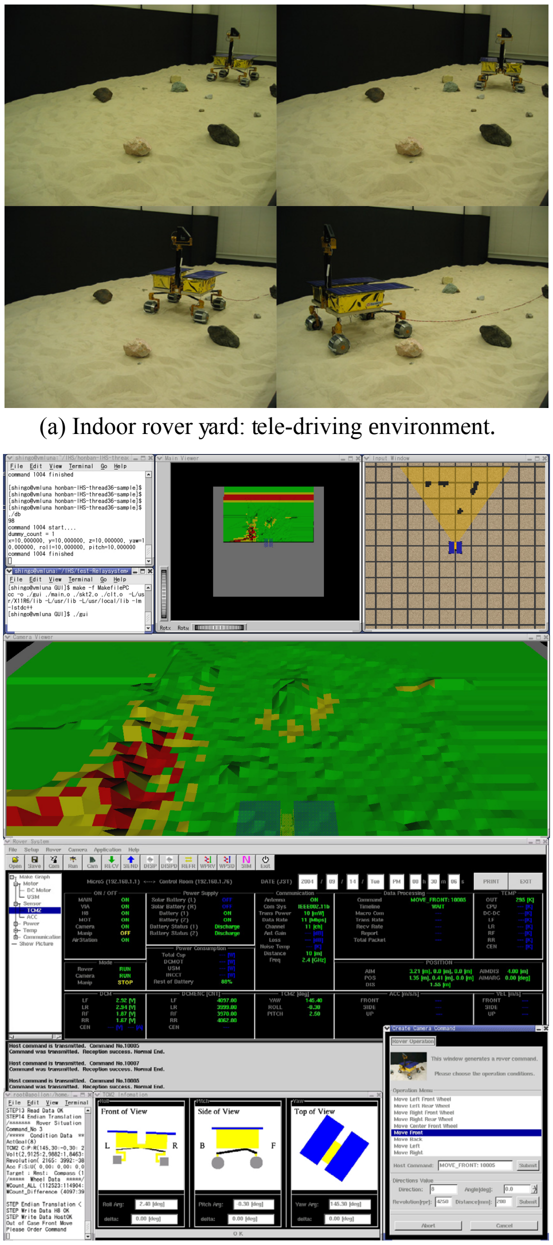
(a) distance from command path (b) potential value

Figure 10: Evaluation based on slip rate of wheels.

virtual rover is going on them by CDC tele-driving algorithm. For evaluation of its trajectories, virtual electrical potential field is applied. Virtual electrical charge is located on each obstacle, its potential field indicates dangerousness in proportion to the distance from obstacle, as shown in figure 8a, and (b) shows the result of the summation of virtual potential from some obstacles, and each result is the average of ten samples.

The evaluations of trajectories are discussed with transition of each trajectory from command path, and total amount of potential along each trajectory.

Figure 9 and 10 show influences of the density of obstacle and the slip rate on each path. CDC algorithm mostly performs well, and its performance is obviously improved by the landmark selection. The fluctuation on each result is caused by shortage of the number of samples, and they will be stable with enough samples in the future work.



(a) Indoor rover yard: tele-driving environment.

(b) GUI windows for rover control

Figure 11: Tele-driving test facilities.

5 Experimental Results of Tele-Driving with Command Path Data Compensation

Experiments for tele-driving are conducted by using a developed rover test-bed and its control system using a command path data compensation algorithm. Figure 11 shows our rover control GUI windows and the scenery of tele-driving experimental setup in our indoor rover yard. Rover yard consists of sand and several rocks as obstacles, and has a rectangular shape of 4 x 8 metres (figure 11a). The rover test-bed can communicate with the control station by using wireless LAN, and it's operated by using the GUI window (figure 11b). To acquire terrain data as DEM, the stereo camera unit on the top of the sensor mast in the middle of the rover is used (figure 1). Figure 12 indicates measurement terrain data measured by the camera unit. Obstacles are recognised from this DEM data, and then the operator considers, plans and decides a path by choosing a series of way-points while looking at the DEM image on

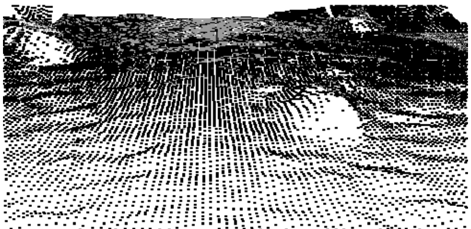
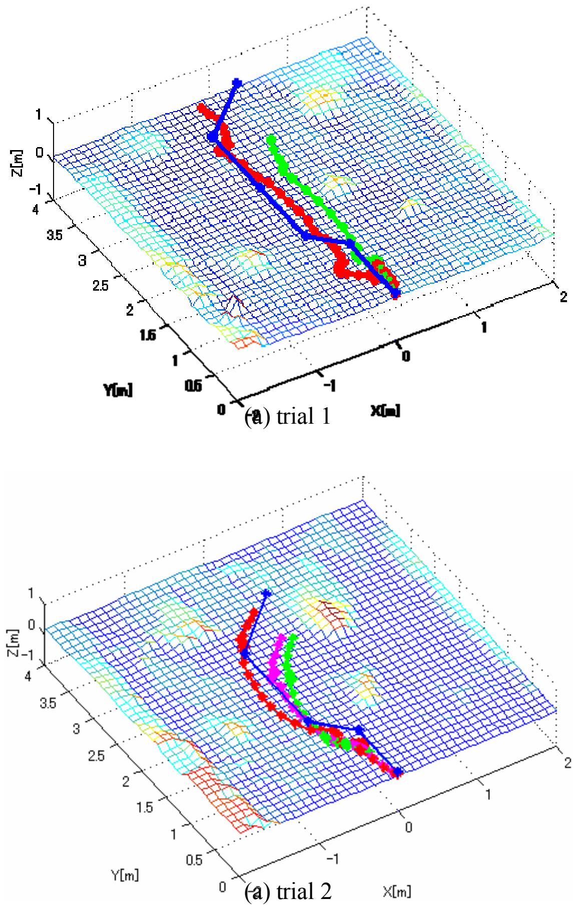


Figure 12: Example of digital elevation map.



(a) trial 1

(a) trial 2

Figure 13: Experimental results of CDC tele-driving.

the GUI. The generated way-points are finally transmitted to the rover.

In figure 13, experimental results are shown. The blue line indicates a command path made by the operator, and the green line is the path without any compensation. The red line in (a) and the pink line in (b) are compensated by CDC using all LMs. On the other hand, the red line in (b) is compensated by CDC with selected LMs. Here, trajectory is measured by using a polhemus sensor (Fastrak). Collisions are caused on green lines; however, compensated paths are safely passed to the goal. In addition, the LM selection works very well in figure 13 (b). In this experiment, though the rover has 23% error of slip of wheels, the rover safely completed its mission. Therefore, the effectiveness of the proposed algorithm is confirmed.

6 Conclusion

The science on the moon has required the wide area of investigations. But it is difficult to realise a long driving in an environment with communication delay. One of the solutions is the application of tele-drive systems with a virtual world simulator. However, we have a difference between environmental map data of an operator and a rover, because of time-delay. This difference makes a reliability of command path lower. In this paper, we discussed about a Tele-driving method with Command Data Compensation (CDC), for a long driving. The CDC is a kind of mapping to compensate a distortion of a space, and compensates command data according to the latest environmental data measured by a rover. The algorithm was evaluated by simulation and experiment. As a result, the reliability of the command path is improved. We confirmed the effectiveness of CDC by some simulations and experiment by using a developed rover test-bed.

Finally, this research is supported by the joint research project in the Institute of Science and Engineering of Chuo University.

7 References

[1] Mars Exploration Rover Mission, "Home Page", http://marsrovers.jpl.nasa.gov/, visited on 2/9/2006.

[2] R. Volpe, J. Balaram, T. Ohm, R. Ivlev, "The Rocky 7 Mars rover prototype", *Proc. of Int. Conf. on Intelligent Robot and Systems*, Osaka, pp 1558-1564 (1996).

[3] R. Volpe, "Rover functional autonomy development for the Mars mobile science laboratory", *Proc. of the 2003 IEEE Aerospace Conference*, pp 643-652 (2003).

[4] R. Simmons, L. Henricksen, L. Chrisman, G. Whelan, "Obstacle avoidance and safeguarding for a lunar rover", *Proc. of AIAA Forum on Advanced Developments in Space Robotics*, Madison (1996).

[5] E. Krotkov, R. Simmons, F. Cozman, S. Koenig, "Safeguarded teleoperation for lunar rovers", *Proc. of SAE 26th Int. Conf. on Environmental Systems*, Monterey, SAE paper # 961585 (1996).

[6] J.L. Loch, D. Desai, "Moose on the loose: toward extended mission autonomy for robotic exploration of planetary surface", *Proc. of International Conference on Advanced Robots*, Tokyo, pp 97-101 (1993).

[7] G. Giralt, "Remote intervention robot autonomy and real world application cases", *Proc. of IEEE Int. Conf. on Robotics and Automation*, Atlanta, pp 541-547 (1993).

[8] R. Chatila, R. Alami, S. Lacroix, J.Perret, C. Proust, "Planet exploration by robots: from mission planning to autonomous navigation", *Proc. of International Conference on Advanced Robotics*, Tokyo, pp 91-96 (1993).

[9] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, T. Litwin, "Mars microrover navigation: performance evaluation and enhancement", *Autonomous Robots Journal*, 2(4), pp 291-312 (1995).

[10] G. Heiken, D. Vaniman, B.M. French, *Lunar Sourcebook: A User's Guide to the Moon*, Cambridge University Press (1991)

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

The AR-CAVE: Distributed Collaborative Augmented Reality and Immersive Virtual Reality System

Si-Jung Kim[1], Denis Gračanin[2], Woodrow W. Winchester[1], Tae-Yong Kuc[3]
[1]Department of Industrial and Systems Engineering, [2]Department of Computer Science
Virginia Tech, Blacksburg, Virginia, USA
[3]School of Information and Communications
Sungkyunkwan University, Suwon, South Korea
hikim@vt.edu, gracanin@vt.edu, wwwinche@vt.edu, tykuc@yurim.skku.ac.kr

Abstract

Combining augmented reality (AR) environments with immersive virtual environments (VEs) can reduce interaction gaps and provide embodied interactions in distributed collaborative works. It can also provide many opportunities and degrees of freedom for collaboration in both real and virtual worlds. A gesture based AR environment provides variety of input and output modalities to networked and distributed immersive VEs. In this paper, we present the AR-CAVE (Augmented Reality connected CAVE) platform that enables embodied interaction and tangible manipulations from a set of AR remote site to integrate physical free body interactions in a CAVE system. As a case study of the AR-CAVE, a simple 3D ball hitting task called the Ting Ting Together (T3) is demonstrated under a networked immersive VE and a computer vision based AR system.

Keywords: virtual reality, augmented reality, virtual environments, collaboration, distributed, embodied

1 Introduction

An augmented reality (AR) technology based collaboration system provides physically challenging realistic experiences [2], and it also gives many opportunities for integration of different user interfaces and interaction techniques. Distributed and collaborative virtual reality technologies were used for distributed virtual environments (DVEs) since early 1990s (e.g. simple networked games and simulations) and it still faces many challenges, such as latency, dead reckoning and consistence [5]. In distributed collaborative interactions, interaction techniques are divided into several sub areas according to types and domains. One such area includes interactions among virtual objects within a local virtual environment (VE), and the other area includes interactions between two or more VEs distributed connected. A few application programme interfaces (API) such as DIVERSE and VR-Juggler have been developed to facilitate design and implementation of DVEs [1][6]. Some researchers have investigated on collaborative playful spaces and work place using a VE or an AR interaction technique. For example, a playful space, where VE and real environment are mixed, is introduced, especially for children, based on the concept of embodied interaction and tangible interaction techniques to enhance children's physical and social activities [7]. A tool for interactively mediated reality is presented using a head-worn video mixing display for painting, grabbing and gluing [4], and a 3D table for texture painting works is showed by using the AR Toolkit with several markers as they address the limitations of physically changing textures [8]. Some researchers have investigated on user techniques from the points of spectators' view [9]. From the review, most interaction techniques have been investigated in a single and local environment. Typical AR systems are monolithic and designed for a non-immersive augmented application in a single location. In general, the required programme and data in AR are self-contained to enhance the user's perception lacking in immersive effects. On the contrary, general immersive VEs systems are standalone and do not have collaborative configurations to collaborative with other multi modalities such as gesturers and tangible manipulations.

In this paper, we present the AR-CAVE (Augmented Reality connected CAVE) platform, a combination of an AR environment with an immersive VE and show an example of the AR-CAVE called Ting Ting Together (T3), which is a simple 3D ball hitting game under the AR-CAVE platform. The T3 consists of two different working platforms to perform particular tasks. One is AR platform which is mainly controlled by human hands or physical objects using a computer vision system, and the other platform is an immersive VE mainly controlled by a wand in the VT-CAVE. To connect two different working environments and to share data of them, an open source virtual reality toolkit DIVERSE is used with the Internet network protocols. This paper divided into three sections. In the first section, motivation and goal is represented, and system architecture is showed in the second section. In the last of two sections, a case study is introduced and conclusion with future works is described.

2 Motivation and Goal

Two main goals are motivated in this research. The first one addresses the needs to provide tangible manipulation input methods to immersive VEs from the outside of its working range. The second one is to make physical activity based collaboration environment between different interaction environments. To share and access each different VE simultaneously, distributed and collaborative virtual technologies were considered for communication with remote users in VEs. According to above motivations, two specific goals are considered in terms of tangible manipulations and AR coupled immersive VEs.

2.1 Tangible Manipulations to VEs

One goal is set to provide a tangible manipulation [10] into immersive VEs from an AR environment and we call this system as the AR-CAVE. Although there are some research results describing how to integrate input and output devices into immersive VEs, mostly have been done in combining sensor dependent components with CAVE locally. There were not remotely coupled with input methods providing gestures or tangible manipulations into the CAVE. In other words, the goal of this part is to expand input and output range of CAVE system, because if users are not in the CAVE, they can not input to the CAVE as well as the results are displayed only within CAVE walls. In addition, most AR and immersive VEs are working separately.

2.2 Embodied Interactions in the CAVE

The other one is utilising the AR-CAVE platform to enable a user to make physical activity. The specific goal of this part is to show an embodied interaction with the AR-CAVE platform to expand virtual space [3]. Most applications based on the CAVE system are static. Usually it is being used for navigation and visualisation tasks. However, some applications running on the CAVE are needed dynamic task needing natural inputs such as body gestures or physical movements from users who are locating outside of its working range. For example, a CAVE system will be used for a playful space for children, and to make collaborative interactions with real agents such as robots.

In this study, a few things are not considered such as shadow occlusions, network latency and scalability between immersive VEs and an AR environment. VT-CAVE and the Ting Ting platform are used as an immersive VE and an AR space respectably.

3 Designing the AR-CAVE

3.1 Interaction Layers

The purpose of the AR-CAVE platform is to enable collaboration between geographically and functionally separated different VR environments and to support tangible manipulations. To realise this requirement, we adapted a layer concept in terms of data and contents. The one layer is a platform layer and the second is a manipulation layer. The platform layer plays a role of inter-platform interactions between an AR and an immersive VE, and the manipulation layer acts as a facilitator for interactions between a user and each platform by providing unique modalities. The conceptual model representing the relationships between the layers is shown in figure 1. In the platform layer, motion data such as position and orientation is transacted, and in the manipulation layer, image data captured by a computer vision and sensed hands gesture data by a Wand in the VT-CAVE is exchanged.

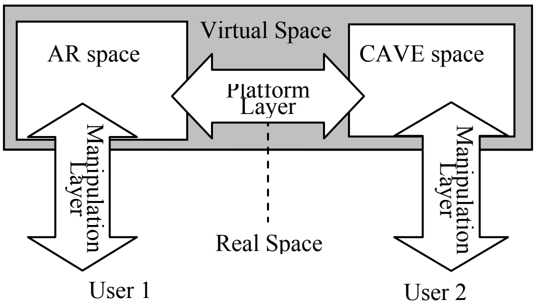


Figure 1: The relationships between interaction platforms and user layers.

Regarding inputs and outputs modalities on each manipulation layer, human hands and tangible objects are used as major input methods in the AR space and a 3D wand is used a main manipulation method by sensing its position and orientation in VT-CAVE space. Figure 2 represents the input and output modalities in each space. The data in the AR space is being displayed by one project and can be seen without any particular materials. On the contrary, in the VT-CAVE space, data is being displayed on three walls by three projects and can be seen with a 3D shutter glasses.

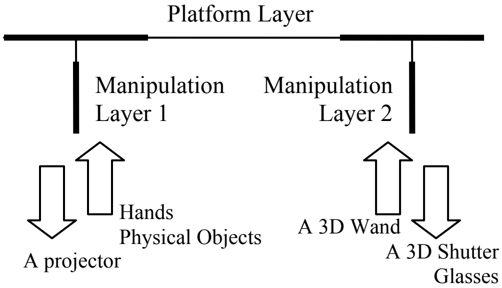


Figure 2: Interaction modalities on each layer.

3.2 Connectivity

To connect the two space, between AR and the VT-CAVE, there needs a particular connection concept to enhance data transactions between two different platforms. A semi-coupled connection structure is used where one part has TCP/IP and the other part has UDP. TCP/IP is used to send motion trajectory data of 2D objects on a large screen in AR space to the VT-CAVE. However, collision detection data generated for a 3D object and a wand in the VT-CAVE space is transferred by UDP network protocol. To share and deliver the motion trajectory and collision data, an asynchronous shared memory called DIVERSE is used between the two environments. The concept of a connectivity of the AR-CAVE platform is represented in figure 3.

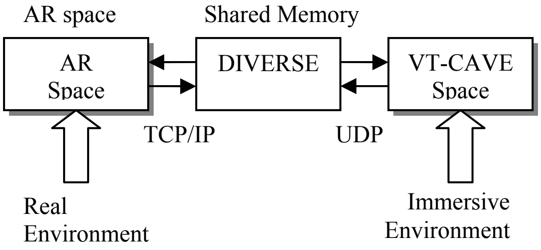
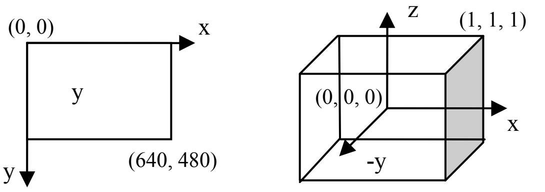


Figure 3: The relationships of a platform and a content layer.

3.3 Coordinate Systems

Since the AR and the VT-CAVE has different coordinate systems as shown in figure 4, we need to match them.



(a) AR coordinate (b) VT-CAVE coordinate

Figure 4: Coordinate systems.

To translate between two different coordinate systems, we projected the AR coordinate to the VT-CAVE coordinate using a normalisation function represented by equations (1) and (2). The equation (1) represents mapping the x axis of the AR space to the x axis of the VT-CAVE and the equation (2) represents mapping the y axis of the AR to the z axis of the VT-CAVE. During this mapping process, the y axis of CAVE is set as a constant variable. According to above matching coordinate systems, the expected object's motion trajectory created from the AR space will be displayed within +x and +z area in the VT-CAVE space.

$$CAVE_x = \frac{1}{AR_{x(\max)}} r_x, \; where \; 1 \le r_x \le 640 \qquad (1)$$

$$CAVE_z = \frac{1}{AR_{y(\max)}} r_y, \; where \; 1 \le r_y \le 480 \qquad (2)$$

4 Implementation

The proposed the AR-CAVE system was implemented by combining the Ting Ting AR environment with the VT-CAVE through the TCP/IP and UDP Internet network protocols. The AR space is built under an IBM compatible Pentium 4 PC with a computer vision system and the VT-CAVE space is used configuring by Linux system with four RGB projectors. An image grabber is used in the AR space in order to capture user's motions at 30 frames rate per second. In the VT-CAVE, a 3D positioning device, a Wand is used to input 3D data to the VT-CAVE system.

In the AR space, four main functions include: 1) generating motion trajectory, 2) sending/ receiving motion data, 3) collision detection, 4) feedback effects. Similarly, five main functions include in the VT-CAVE space: 1) sending/receiving motion data, 2) storing data into a shared memory, 3) collision detector, 4) collision detection effecter, 5) generating motion trajectory. The implemented major characteristics of the AR-CAVE platform are represented in table 1.

Table 1: The AR-CAVE characteristics.

Items	**AR Space**	**CAVE Space**
Input methods	Human hands and physical objects	3D wand
Input detection	Computer vision	DTK (DIVERSE toolkit)
Rendering	Direct draw	DPF (DIVERSE performer)
Output way	1 DLP projector	3 RGB projectors
Shared memory	none	DIVERSE
Data sending	TCP/IP	UDP
Collision detection	Colour based	Position based
User feedback	Changing background colour	Changing 3D object's colour

5 Case Study: Interaction

A simple hitting ball game called Ting Ting Together (T3) is demonstrated under the implemented AR-CAVE. To play the T3 game, one player is in the VT-CAVE room wearing a stereo shutter glasses and holding a Wand and the other player is in the AR space located outside of the VT-CAVE room. Human hands and physical objects are used to change the motion trajectory of the 2D object in the AR space and a 3D object is displayed in the VT-CAVE according to the motion path of the 2D object of the AR space. If the 3D object, the clone of the 2D object of the AR space, is hit by the Wand, the background colour is changed as cyan colour in order to notify to a player in the AR space that their object is captured by the opponent player in the VT-CAVE space. The snapshots are shown in figure 5.

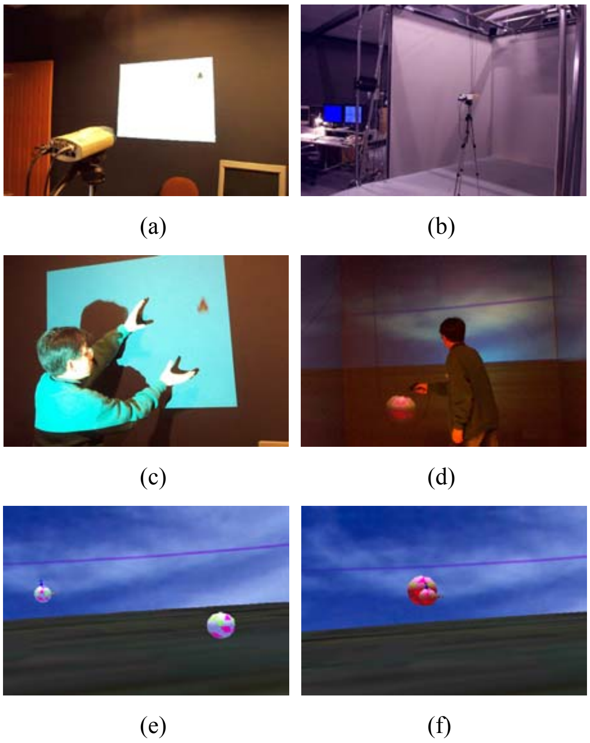


Figure 5: Snapshots of the T3 game: (a) AR space; (b) VT-CAVE space; (c) playing T3 in the AR; (d) playing T3 in the VT-CAVE; (e) display an object of the AR in the VT-CAVE space; (f) object captured by the wand in the VT-CAVE.

6 Conclusion and Future Work

The AR-CAVE (Augmented Reality connected CAVE) platform allowing tangible manipulations and embodied interactions is implemented by incorporating two different VEs such as an AR and the VT-CAVE. The implemented AR-CAVE platform enables networked group collaborations, and provided more mobility and a larger working space compared to the traditional CAVE applications, because the AR-CAVE platform supported users from the outside of working range of CAVE system.

An illustrative example called T3 is implemented and tested to show the interactive collaboration ability of the AR-CAVE platform. The player playing in the VT-CAVE space felt user awareness considering that a 3D ball in his place is controlled by another player in the AR space. In the T3 game, a 3D object in the VT-CAVE system is controlled by gesture inputs or tangible objects from the AR space, and embodied interactions are performed in the AR sides. One of things that we found from the test by accidentally is that social interactivity and collaboration performance can be increased in VEs if they are inputting data by using their gestures with physical manipulators instead of not only sensing devices such as a Wand.

However, calibration issues had to be resolved to synchronise coordinate systems for the two dimensional AR space and the immersive VT-CAVE space. Feedback effects also had to be considered to give better instructive messages to users. We will improve the AR-CAVE platform not only to support various types of tangible and embodied inputs, but also to make collaborative social interactions and works between human and agents such as robots.

7 Acknowledgements

We would like to thank Patrick Shinpaugh for helping with the VT-CAVE system.

8 References

[1] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker and C. Cruz-Neira, “VR Juggler: s virtual platform for virtual reality application development”, *Proc. IEEE VR Conference 2001*, Yokohama, Japan, pp 89-96 (2001).

[2] F. Brooks, “What’s real about virtual reality?”, *IEEE Computer Graphics and Applications*, 19(6), pp 16-27 (1999).

[3] P. Dourish, *Where The Action Is: The Foundations of Embodied Interaction*, MIT Press, Cambridge, MA (2001).

[4] R. Grasset, J.D. Grascuel and D. Schmalstieg, “Interactive mediated reality”, *Proc. International Symposium on Mixed and Augmented Reality*, Tokyo, pp 302- 303 (2003).

[5] C. Joslin, G.D. Thomas and N. Magnenat-Thalmann, “Collaborative virtual environments: from birth to standardization”, *IEEE Communication Magazine*, 42(4), pp 28-33 (2004).

[6] K. Kelso, S.G. Satterfield, L.E. Arsenault, P.M. Ketchan and R.D. Kriz, “DIVERSE: a framework for building extensible and reconfigurable device-independent virtual environments and distributed asynchronous simulations”, *Presence*, 12(1), pp 19-36 (2003).

[7] S.J. Kim, M.S. Jang and T.Y. Kuc, “A development of interactive game Ting Ting using real and virtual objects”, *Proc. IEEE International Conference on SMC,* The Hague, pp 1191-1197 (2004).

[8] K. Matkovic, T. Psik, I. Wagner and D. Gracanin, “Dynamic texturing of real objects in an augmented reality system”, *Proceedings IEEE VR 2005*, Bonn, Germany, pp 245-248 (2005).

[9] S. Reeves, S. Benford, C. O’Malley and M. Fraser, “Designing the spectator experience”, *Proc. ACM Conference on Computer-Human Interaction (CHI)*, Portland, pp 741-750 (2005).

9 B. Ullmer and H. Ishii, “Tangible bits: towards seamless interfaces between people, bits and atoms”, Proc. ACM Conference on Computer-Human Interaction (CHI), Atlanta, pp 234-241 (1997).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Delay-dependent Criteria for Robust Stability of Uncertain Fuzzy Systems with Time-varying State Delay

Yanbo Li[1] , Yonggui Kao[1,2] and Cunchen Gao[1]
[1]Ocean University of China, Qingdao, China
[2]ZaoZhuang University, ZaoZhuang, China
apples729@163.com

Abstract

The stability problem of Takagi-Sugeno fuzzy systems with time-varying delay and parameter uncertainties is considered. New delay-dependent stability criteria are given in terms of linear matrix inequality by using the Lyapunov-Krasovskii functional method and by taking the relationship between the terms in the Leibniz-Newton formula into account. Since free weighting matrices are used to express this relationship and appropriate ones are selected by means of linear matrix inequalities, the new criteria are more computationally effective and less conservative.

Keywords: T-S fuzzy model, time-delay, uncertain system, robust stability, LMI

1 Introduction

Nonlinear systems with time-delay and uncertainty are very common in various industrial fields, such as robotic control. Recently, fuzzy logic control based on Takagi-Sugeno (T-S) model becomes more and more popular for nonlinear systems [1]-[23]. It is an effective approach to represent complex nonlinear systems by fuzzy sets and fuzzy reasoning, and has many successful applications in industrial processes. It is to combine some simple local linear dynamic systems with their linguistic description to represent highly nonlinear dynamic systems, and is feasible because in many situations, human experts can provide linguistic descriptions of local systems in terms of IF-THEN rules. It gives a way to smoothly connect local linear systems to form global nonlinear systems by fuzzy membership functions.

Stability is one of the most important problems in analysis and design of control systems. The stability of T-S model fuzzy systems without delays has been widely studied; many results are reported [2-10]. However, as we all know, delays appear in many dynamic systems. Generally speaking, the dynamic behaviours of systems with delays are more complicated than that of systems without any delays. Fuzzy delayed systems of T-S model, which were first introduced and studied in [19], provide a method of using local linear delayed systems combined with fuzzy linguistic descriptions to achieve nonlinearity. The existing stability results for time-delay systems can be classified into two types: delay-independent stability; and delay-dependent stability. The delay-independent stability criterion is not affected by the size of the delay. On the other hand, the delay-dependent stability criterion is concerned with the size of the delay and usually provides an upper bound of the delay such that the system is stable for any delay less than the upper bound. In general, the delay-dependent stability criterion is considered to be less conservative than the delay-independent case. More recently, delayed fuzzy systems were studied in [18-23]. The Lyapunov-Krasovskii approach and the Lyapunov-Razumikhin method [25] have been used to study the stability of delayed fuzzy systems [18, 19]. In [20] the output feedback robust H_∞ control of uncertain fuzzy dynamic output systems with time-varying delays was studied. In [21], the global exponential stability of fuzzy systems with bounded uncertain delays was investigated using the method of functional differential inequalities analysis. These stability conditions are all delay independent. [22] and [23] obtained delay dependent Criteria, but [23] didn't consider uncertainty and the result of [22] is conservative, because [22] didn't consider the relationship between the terms in the Leibniz-Newton formula [24].

In this paper, we intend to consider the robust stability problem of uncertain Takagi- Sugeno fuzzy systems with time-varying delays. Delays in this paper are assumed to be any uncertain bounded continuous functions. We successfully develop a delay-dependent robust stability criterion in terms of linear matrix inequalities (LMIs) [25] by using the Lyapunov-Krasovskii functional method [26] and by taking the relationship between the terms in the Leibniz-Newton formula into account.

In the following, we will use, W^T, W^{-1} to denote, respectively, the transpose and inverse of a square matrix W .We use $W > 0 (< 0)$ to denote a positive (negative) definite matrix W .

2 System Descriptions and Preliminaries

Consider a continuous-time fuzzy system with time-varying delay, which is represented by a Takagi-

Sugeno fuzzy model, composed of a set of fuzzy implications, where each implication is expressed by a linear system model. The *i*th rule of this fuzzy model is of the following form:

Rule *i*: If $z_1(t)$ is $M_{i1}$and … and $z_p(t)$ is M_{ip} then

$$\Sigma_0 : \begin{cases} \dot{x}(t) = A_i x(t) + B_i x(t-d(t)) \\ x(t) = \phi(t), t \in [-\tau, 0], i = 1, \cdots, r \end{cases} \qquad (1)$$

where $z_1(t), \cdots, z_p(t)$ are the premise variables and each $M_{ij}\,(j = 1, \cdots, p)$ are fuzzy sets, $x(t)$ is the state variable, r is the number of if-then rules, $0 \le d(t) \le \tau$ is the bounded time-varying delay in the state and it is assumed that $\dot{d}(t) \le \beta < 1$, i.e. the derivative of the time-varying delay function is continuous and bounded. $\phi(t)$ is the continuous vector-valued initial function of $t \in [-\tau, 0]$.

Though the use of fuzzy 'blending', the final output of the fuzzy system (1) is inferred as follows:

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} w_i(z(t))\left[A_i x(t) + B_i x(t-d(t))\right]}{\sum_{i=1}^{r} w_i(z(t))}$$
$$= \sum_{i=1}^{r} \mu_i(z(t))\left[A_i x(t) + B_i x(t-d(t))\right] \qquad (2)$$

with

$$w_i(z(t)) = \prod_{l=1}^{p} M_{il}(z(t)),\ \mu_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^{r} w_i(z(t))},$$

in which $M_{il}(z_l(t))$ is the grade of membership of $z_l(t)$ in M_{il}. It is assumed that

$$w_i(z(t)) \ge 0,\ i = 1, \cdots, r,\ \sum_{i=1}^{r} w_i(z(t)) > 0$$

for all t. Therefore,

$$\mu_i(z(t)) \ge 0 \text{ and } \sum_{i=1}^{r} \mu_i(z(t)) = 1$$

for all t.

When the system contains time-varying structured uncertainties, it can be described by

Rule *i*: If $z_1(t)$ is $M_{i1}$and … and $z_p(t)$ is M_{ip} then

$$\Sigma_1 \begin{cases} \dot{x}(t) = (A_i + \Delta A_i(t))x(t) + (B_i + \Delta B_i(t))x(t-d(t)) \\ x(t) = \phi(t), t \in [-\tau, 0], i = 1, \cdots, r \end{cases} \qquad (3)$$

The parametric uncertainties $\Delta A_i(t)$, $\Delta B_i(t)$ are time-varying matrices with appropriate dimensions, which are defined as follows:

$$\left[\Delta A_i(t) \quad \Delta B_i(t)\right] = DF(t)\left[E_{ai} \quad E_{bi}\right], \forall t, i = 1\cdots, r \qquad (4)$$

where D, E_{ai}, E_{bi} (for $i = 1, \cdots, r$) are known constant real matrices with appropriate dimensions and $F_i(t)$ is unknown real time-varying matrix with Lebesgue measurable elements bounded by:

$$F^T(t)F\ (t) \le I, \forall t \quad i = 1, \cdots, r \qquad (5)$$

Though the use of fuzzy 'blending', the final output of the uncertain fuzzy system (3) is inferred as follows:

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} w_i(z(t))\left[\overline{A}_i x(t) + \overline{B}_i x(t-d(t))\right]}{\sum_{i=1}^{r} w_i(z(t))}$$
$$= \sum_{i=1}^{r} \mu_i(z(t))\left[\overline{A}_i x(t) + \overline{B}_i x(t-d(t))\right] \qquad (6)$$

where

$$\overline{A}_i = A_i + \Delta A_i(t),\ \overline{B}_i = B_i + \Delta B_i(t) \qquad (7)$$

others are defined as (1) and (2).

The following lemma is employed to handle the time-varying structured uncertainties in the system.

Lemma 1 [27]: Given matrices $Q = Q^T > 0, H, E$ and $R = R^T > 0$ of appropriate dimensions,

$$Q + HFE + H^T F^T E^T < 0$$

for all F satisfying $F^T F \le R$, if and only if there exists some $\lambda > 0$ such that

$$Q + \lambda H H^T + \lambda^{-1} E^T R E < 0$$

The Lyapunov functional candidates for (1) and (3) are chosen to have the same form and are given by

$$V(x_t) := x^T(t)Px(t) + \int_{t-d(t)}^{t} x^T(t)Qx(t)ds$$
$$+ \int_{-\tau}^{0}\int_{t+\theta}^{t} \dot{x}^T(t)Z\dot{x}(t)dsd\theta \qquad (8)$$

where $P = P^T > 0, Q = Q^T > 0$ and $Z = Z^T > 0$ are to be determined.

3 Main Results

First, we will derive a delay-dependent robust stability condition for the nominal fuzzy system (4), which can be recast as:

$$\dot{x} = \sum\nolimits_{i=1}^{r}\mu_i(z(t))A_i x(t) + \sum\nolimits_{i=1}^{r}\mu_i(z(t))B_i x(t-d(t))$$
$$= Ax(t) + Bx(t-d(t)) \qquad (9)$$

where

$$A = \sum\nolimits_{i=1}^{r}\mu_i(z(t))A_i,\ B = \sum\nolimits_{i=1}^{r}\mu_i(z(t))B_i \qquad (10)$$

We now summarise the main result for the robust stability of (1) in the following theorem.

Theorem 1: If there exist symmetric positive-definite matrices $P = P^T > 0, Q = Q^T > 0$ and $Z = Z^T > 0$, a symmetric semi-positive-definite matrix $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \geq 0$, and any appropriately dimensioned matrices Y and T such that the following LMIs are true:

$$\dot{V}(x_t) \leq x^T(t)\left[PA + A^T P + Q\right]x(t) + 2x^T(t)PBx(t-d(t))$$

$$\Phi_i = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \tau A_i^T Z \\ \Phi_{12}^T & \Phi_{22} & \tau B_i^T Z \\ \tau Z A_i & \tau Z B_i & -\tau Z \end{bmatrix} < 0 \qquad (11)$$

$$\Psi = \begin{bmatrix} X_{11} & X_{12} & Y \\ X_{12}^T & X_{22} & T \\ Y^T & T^T & Z \end{bmatrix} \geq 0 \qquad (12)$$

where

$$\Phi_{11} = PA_i + A_i^T P + Y + Y^T + Q + \tau X_{11}$$

$$\Phi_{12} = PB_i - Y + T^T + \tau X_{12}$$

$$\Phi_{22} = -T - T^T - (1-\beta)Q + \tau X_{22}$$

for $i = 1, \cdots, r$. Then the fuzzy system (1) is asymptotically stable for any time delay $0 \leq d(t) \leq \tau$ and $\dot{d}(t) \leq \beta < 1$.

Proof: Using the Leibniz-Newton formula, we can write

$$x(t-d(t)) = x(t) - \int_{t-d(t)}^{t}\dot{x}(s)ds \qquad (13)$$

Then, for any appropriately dimensioned matrices Y and T, we have the following equation

$$2\left[x^T(t)Y + x^T(t-d(t))T\right]\times\left[x(t) - \int_{t-d(t)}^{t}\dot{x}(s)ds\right.$$
$$\left. -x(t-d(t))\right] = 0 \qquad (14)$$

where the free weighting matrices Y and T denote the relationship between the terms in the Newton- Leibniz formula. On the other hand, for any semi-positive-definite matrix $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \geq 0$, the following holds

$$\tau\xi^T(t)X\xi(t) - \int_{t-d(t)}^{t}\xi^T(t)X\xi(t)ds \geq 0 \qquad (15)$$

where $\xi(t) = \left[x^T(t) \quad x^T(t-d(t))\right]^T$.

Then, using (14) and (15) and calculating the derivative of $V(x_t)$ along the trajectory of system (9), we get

$$\dot{V}(x_t) \leq x^T(t)\left[PA + A^T P + Q\right]x(t) + 2x^T(t)PB$$
$$x(t-d(t)) - (1-\dot{d}(t))x^T(t-d(t))Qx(t-d(t))$$
$$+\tau\dot{x}^T(t)Z\dot{x}(t) - \int_{t-\tau}^{t}\dot{x}^T(s)Z\dot{x}(s)ds$$
$$\leq x^T(t)\left[PA + A^T P\right]x(t) + 2x^T(t)PBx(t-d(t))$$
$$-(1-\beta)x^T(t-d(t))Qx(t-d(t)) + \tau\dot{x}^T(t)Z\dot{x}(t)$$
$$-\int_{t-d(t)}^{t}\dot{x}^T(s)Z\dot{x}(s)ds + 2\left[x^T(t)Y + x^T(t-d(t))T\right]$$
$$[x(t)\ - \int_{t-d(t)}^{t}\dot{x}(s)ds - x(t-d(t))] + \tau\zeta^T(t)X\zeta(t)$$
$$-\int_{t-d(t)}^{t}\zeta^T(t)X\zeta(t)ds + x^T(t)Qx(t)$$
$$< \zeta^T(t)\Pi\zeta(t) - \int_{t-d(t)}^{t}\zeta^T(t,s)\Psi\zeta(t,s)\ \mathrm{ds} \qquad (16)$$

where

$$\zeta(t,s) = \left[x^T(t) \quad x^T(t-d(t)) \quad \dot{x}^T(s)\right]^T$$

$$\Pi = \begin{bmatrix} \Theta_{11} + \tau A^T ZA & \Theta_{12} + \tau A^T ZB \\ \Theta_{12}^T + \tau B^T ZA & \Theta_{22} + \tau B^T ZB \end{bmatrix}$$

$$\Theta_{11} = PA + A^T P + Y + Y^T + Q + \tau X_{11}$$

$$\Theta_{12} = PB - Y + T^T + \tau X_{12}$$

$$\Theta_{22} = \Phi_{22}$$

in which Φ_{22}, A and B are defined in (9), others and Ψ is defined in (12). If $\Pi < 0, \Psi \geq 0$, then $\dot{V}(x_t) < 0$ for any $\zeta(t) \neq 0$. In fact,

$$\Phi = \sum\nolimits_{i=1}^{r}\mu_i(z(t))\Phi_i$$
$$= \begin{bmatrix} \Theta_{11} & \Theta_{12} & \tau A^T Z \\ \Theta_{12}^T & \Theta_{22} & \tau B^T Z \\ \tau ZA & \tau ZB & -\tau Z \end{bmatrix} < 0$$

Applying the Schur Complement, it is obvious that Φ is equivalent to Π, that is, (11) implies $\Pi < 0$. So system (1) is asymptotically stable if LMIs (11) and (12) are true. This completes the proof.

Now, we will extend Theorem 1 to systems with time-varying structured uncertainties yields the following theorem.

Theorem 2: If there exist symmetric positive-definite matrices $P = P^T > 0, Q = Q^T > 0$ and $Z = Z^T > 0$, a symmetric semi-positive-definite matrix $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \geq 0$ and any appropriately dimensioned matrices Y and T such that LMI (12) the following LMI are true:

$$\Omega_i = \begin{bmatrix} \Phi_{11} + E_{ai}^T E_{ai} & \Phi_{12} + E_{ai}^T E_{bi} & \tau A_i^T Z & PD \\ \Phi_{12}^T + E_{bi}^T E_{ai} & \Phi_{22} + E_{bi}^T E_{bi} & \tau B_i^T Z & 0 \\ \tau Z A_i & \tau Z B_i & -\tau Z & \tau Z D \\ D^T P & 0 & \tau D^T Z & -I \end{bmatrix} < 0 \tag{17}$$

where $\Phi_{11,} \Phi_{12}$ and Φ_{22} are defined in (11) for $i = 1, \cdots, r$. Then the fuzzy system (3) is asymptotically stable for any time delay

$0 \leq d(t) \leq \tau$ and $\dot{d}(t) \leq \beta < 1$.

Proof: Replacing A_i, B_i in (11) with $A_i + DF(t)E_{ai}$, $B_i + DF(t)E_{bi}$, respectively, we find that (11) for system (3) is equivalent to the following condition:

$$\Phi + \begin{bmatrix} PD \\ 0 \\ \tau ZD \end{bmatrix} F(t) \begin{bmatrix} E_{ai} & E_{bi} & 0 \end{bmatrix} + \begin{bmatrix} E_{ai}^T \\ E_{bi}^T \\ 0 \end{bmatrix} F^T(t) \begin{bmatrix} D^T P & 0 & \tau D^T Z \end{bmatrix} < 0$$

By Lemma 1, a sufficient condition guaranteeing (11) for system (3) is that there exists a positive number $\lambda > 0$ such that

$$\lambda \Phi + \lambda^2 \begin{bmatrix} PD \\ 0 \\ \tau ZD \end{bmatrix} \begin{bmatrix} D^T P & 0 & \tau D^T Z \end{bmatrix} + \begin{bmatrix} E_{ai}^T \\ E_{bi}^T \\ 0 \end{bmatrix} \begin{bmatrix} E_{ai} & E_{bi} & 0 \end{bmatrix} < 0 \tag{18}$$

Replacing $\lambda P, \lambda Q, \lambda Z, \lambda X, \lambda Y$ and λT with P, Q, Z, X, Y and T, respectively, applying the Schur Complement [27], it is obvious that (18) is equivalent to (17). So system (3) is asymptotically stable if LMIs (17) and (12) are true. This completes the proof.

4 Conclusion

This note deals with the robust stability problem for time-delay fuzzy systems with time-varying uncertainties. Based on the Takagi-Sugeno model and linear matrices inequalities, some free weighting matrices that express the influence of two terms are determined. Some sufficient conditions for robust stability are obtained by solving LMI [25]. So the method proposed in this paper is effective for the system we considered.

5 Acknowledgements

The work was supported by the National Natural Science Foundation of CHINA under Grant 69974032.

6 References

[1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control", *IEEE Trans. Syst., Man, Cybern.*, SMC-15, pp 116-132 (1985).

[2] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems", *Fuzzy Sets Syst.*, 45, pp 135-156 (1992).

[3] K. Tanaka and M. Sano, "Fuzzy stability criterion of a class of nonlinear systems", *Inf. Sci.*, 70(1), pp 3-26 (1993).

[4] F. Cuesta, F. Gordillo, J. Aracil and A. Ollero, "Stability analysis of nonlinear multivariable Takagi-Sugeno fuzzy control systems", *IEEE Trans. Fuzzy Syst.*, 7, pp 508-520 (1999).

[5] J. Joh, Y. H. Chen and R. Langari, "On the stability issues of linear Takagi-Sugeno fuzzy models", *IEEE Trans. Fuzzy Syst.*, 7, pp 402-410 (1998).

[6] J. Joh, Y. H. Chen and R. Langari, "On the stability issues of linear Takagi-Sugeno fuzzy models", *IEEE Trans. Fuzzy Syst.*, 6, pp 402-410 (1998).

[7] K. Tanaka and M. Sano, "A robust stabilization problem of fuzzy control systems its application to backing up control of a truck-trailer", *IEEE Trans. Fuzzy Syst.*, 2(2), pp 119-134 (1994).

[8] J.C. Lo and Y.M. Chen, "Stability issues on Takagi-Sugeno fuzzy model-parametri approach", *IEEE Trans. Fuzzy Syst.*, 7, pp 597-607 (1999).

[9] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents", *IEEE Trans. Fuzzy Syst.*, 7, pp 201-224 (1999).

[10] S.G. Cao, N.W. Rees and G. Feng, "Stability analysis and design for a class of continuous-time fuzzy control systems", *Int. J. Control*, 64(6), pp 1069-1087 (1996).

[11] K. Tanaka, T. Ikeda and H.O. Wang, "Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs", *IEEE Trans. Fuzzy Syst.*, 6(2), pp 250-265 (1998).

[12] G. Calcev, "Some remarks on the stability of Mamdani fuzzy control systems", *IEEE Trans. Fuzzy Syst.*, 6(3), pp 436-442 (1998).

[13] M. Feng and C.J. Harris, "Piecewise Lyapunov stability conditions of fuzzy systems", *IEEE Trans. Syst. Man Cybern.*, 31(2), pp 259-262 (2001).

[14] E. Kim and D. Kim, "Stability analysis and synthesis for an affine fuzzy systems via LMI and ILMI: discrete case", *IEEE Trans. Syst. Man Cybern. B. Cybern.*, 31(1), pp 132-140 (2001).

[15] E. Kim and S. Kim, "Stability analysis and synthesis for an affine fuzzy systems via LMI and ILMI: continuous case", *IEEE Trans. Fuzzy Syst.*, 10(3), pp 391-400 (2002).

[16] G. Feng, "Approaches to quadratic stabilization of uncertain fuzzy dynamic systems", *IEEE Trans. Circuits Syst. 1, Fundam. Theory Appl.*, 48(6), pp 760-769 (2001).

[17] H.J. Lee, J.B. Park and G. Chen, "Robust fuzzy control of nonlinear systems with parametric uncertainties", *IEEE Trans. Fuzzy Syst.*, 9(2), pp 369-379 (2001).

[18] Y.Y. Cao and P.M. Frank, "Stability analysis and synthesis of nonlinear time-delay systems via linear Takagi-Sugeno fuzzy models", *Fuzzy Sets Syst.*, 124, pp 213-229 (2001).

[19] Y.Y. Cao and P.M. Frank, "Analysis and synthesis of nonlinear time-delay systems via fuzzy control approach", *IEEE Trans. Fuzzy Syst.*, 8(2), pp 200-211(2000).

[20] K.R. Lee, J.H. Kim, E.T. Jeung and H.B. Park, "Output feedback H_∞ control of uncertain fuzzy dynamic systems with time-varying delay", *IEEE Trans. Fuzzy Syst.*, 8(6), pp 657-664 (2000).

[21] Y. Zhang and P.A. Heng, "Stability of fuzzy control systems with bounded uncertain delays", *IEEE Trans. Fuzzy Syst.*, 10(1), pp 92-97 (2002).

[22] C. Li, H.J. Wang and X.F. Liao, "Delay-dependent robust stability of uncertain fuzzy systems with time-varying delays", *IEE Pro.-Control Theory Appl.*, 151(4) (2004).

[23] X.P. Guan and C.L. Chen, "Delay-dependent guaranteed cost control for T-S fuzzy systems with time delays", *IEEE Trans. Fuzzy Syst.*, 12(2) (2004).

[24] M. Wu, Y. He, J.H. She and G.P. Liu, "Delay-dependent criteria for robust stability of time-varying delay systems". *Automatica*, 40, pp 1435-1439 (2004).

[25] S. Boyd, L. Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA (1994).

[26] J.K. Hale, *Theory of Functional Differential Equations*, Springer-Verlag, New York (1977).

[27] L. Xie, "Output feedback H_∞ control of systems with parameter uncertainty", *Int. J. Control*, 63(4), pp 741-750 (1996).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Remarks on Breath Gas Sensing System and its Application to Man-Machine Interface

Kazuhiko Takahashi[1] and Iwao Sugimoto[2]
[1] Doshisha University, Kyoto, Japan
[2] Tokyo University of Technology, Tokyo, Japan
katakaha@mail.doshisha.ac.jp, sugimoto@cc.teu.ac.jp

Abstract

This paper proposes a smart gas sensing system to achieve nonverbal interface using breath gas information. A quartz crystal resonator with a plasma-polymer film is used as a sensor, and breath gas sensing system is designed under air in indoor atmospheric condition. To evaluate basic characteristics of the sensing system, breath gas test experiments are carried out. Experimental results indicate the feasibility of the sensing system for breath gas information.

Keywords: breath gas, organic films, plasma polymer, quartz crystal resonator, emotion

1 Introduction

Nonverbal information plays an important role in human communication. Especially, people can communicate with each other more smoothly by using information of intention and emotion which are estimated from nonverbal information. It is clear that the exchange of nonverbal information is important in all forms of communication and is sometimes more important than verbal information. This means that nonverbal communication is the basis of human communication. Recently, in addition to human-to-human communication, communication between humans and machines are becoming more and more common. Thus the use of both verbal and nonverbal information will be essential in man-machine interface systems achieve more intimate and human-like interactions between humans and machines. There are various nonverbal information caused from human body, figures, postures and actions and many studies have been undertaken in order to achieve engineering applications. For example, facial expressions have been studied in the field of image processing and computer vision and tone in voice has been handled in the field of speech processing. In this study, we investigate a possibility of using biogas emitted from a human body as a new channel of nonverbal interface system that can estimate human emotions.

Emotion recognition is an interesting but difficult task. People can recognize emotional speeches with about 60% accuracy and emotional facial expressions with about 70-98% [1]. Studies on emotion recognition with computers differ on the number of categories and the kinds of categories to use. Some emotion recognition systems in speech or facial expressions which have been used include several emotional states such as joy, teasing, fear, sadness, disgust, anger, surprise, and neutral. In those studies, emotions that are consciously and purposefully expressed by the subjects are treated since consciously expressed emotions are easier to recognize, control and significantly to gather data on, however recognition rates are 50-60% in emotional speech recognition [2, 3, 4] and 80-90% in facial expressions [5, 6, 7]. Physiological indexes are useful to evaluate emotions since they can be measured physically and objectively and can be easily applied to engineering approaches. Physiological changes according to exciting emotions can be observed on changes of the body surface and/or autonomic nervous system [1]: e.g., skin conductivity, electrocardiogram, electromyogram, and blood volume pressure. Using brain waves seems to be effective [8, 9] since emotions are excited in the limbic system and are deeply related to cognition process. In these cases, however, sensors have to be attached to the human body to collect physiological data and expert techniques of handling the sensors are also required. On the other hand, it is possible that there is a relationship between mental states such as emotion and stress, and physical states of the internal organs or the oral cavity. Studies on a disease diagnosis by breath gas test [10, 11, 12] have been carried out actively in the field of remoteness medical care, Tele-medicine, and Tele-care since it is clear that internal organs disease on a liver, lungs, and blood has bad breath (chemical features) from experience and an analytic chemistry point. As the breath gas test is a non-invasive method of biomarkers without pain, it protects the human body off from the risks of inserting electrodes or sensors. Therefore, in a medical field, the breath gas test is expected as diagnosis technology of the 21st century to inspect an

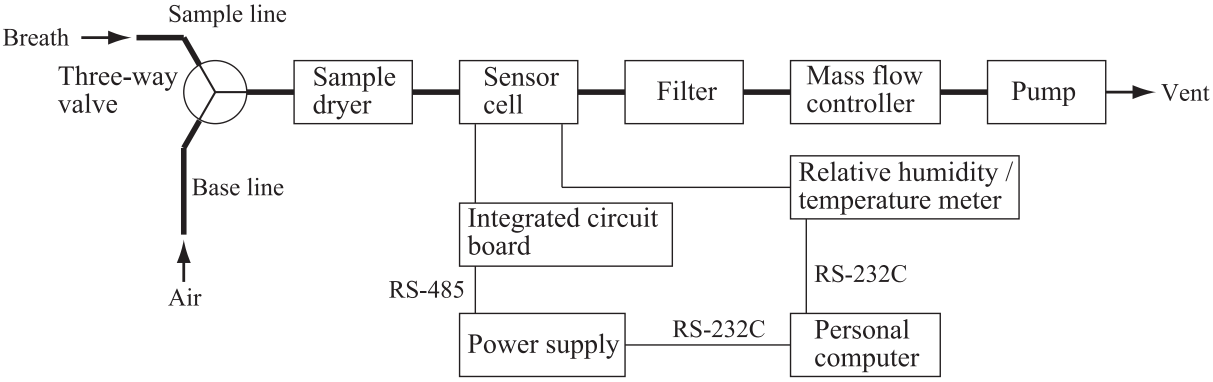


Figure 1: Diagram of breath gas sensing setup.

internal change of human body. Furthermore, if an evaluation method of mental states by using change of breath gas states can be achieved, information from the breath gas may become a new channel of man-machine interface. However, to date few study of estimating mental states such as emotions from breath gas information has been undertaken.

In this paper, as a first step to achieve nonverbal interface using breath gas information we propose a smart gas sensing system for breath gas test. The sensing system uses a quartz crystal resonator sensor, which is a type of mass transducer, since it can detect various gas, however, it is very affected with vapour in atmospheric condition. In order to evaluate whether the breath gas sensing system can design by using the sensor or not, we investigate the characteristics of the sensor response in atmospheric condition. The basic design of breath gas sensing system is described in section 2 and some results of our breath gas sensing experiments are presented in section 3.

2 Breath Gas Sensing System

A block diagram of the setup for breath gas sensing system is shown in figure 1. This sensor system consisted of a sensor module, a power supply, and a personal computer. The sensor module utilized a sensitive sensor device based on a quartz crystal resonator (QCR) with a plasma-polymer film (PPF) where the sensor device can detect volatile organic compounds (VOCs) with parts-per-billion (ppb) levels under the dry air conditions [13, 14, 15]. A PPF was prepared by radio-frequency (RF) sputtering with an organic solid target, which was suitable for mass production and needed no reactive or toxic reagents for processing. A 9 MHz AT-cut QCR, which is 8.5 mm diameter and 0.1 mm thick, was used. PPFs with different chemical structures were prepared by using various target materials including synthetic polymers (s6 $\sim$ s7) and biomaterials (s1 $\sim$ s5) as shown in table 1. The electrode's area of the sensor device was 0.13 cm^2. The relationship between the frequency shift of the QCR Δf (Hz) and mass variation of the organic thin film Δm (ng) is defined with Sauerbrey equation as follows.

Table 1: Materials for organic thin film of sensor cell.

sensor cell	material
s1	D-phenylalanine
s2	D-tyrosine
s3	D-glucose
s4	DL-histidine
s5	Adenine
s6	Polyehylene (PE)
s7	Polychlorotrifluoroethylene (PCTFE)
s8	D-phenylalanine (Sealing) (temperature sensor)

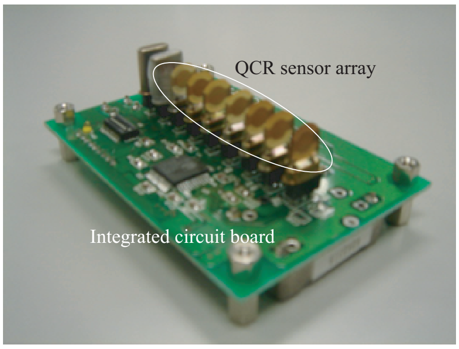


Figure 2: Configuration of sensor module.

$$\Delta m = \quad 1.05\Delta f \tag{1}$$

As shown in figure 2, the PPF-QCRs were placed in a flow sensor cell and simultaneously attached to an integrated circuit board equipped with a custom LSI for oscillation and resonant frequency measurements and a multiport serial interface. This sensor module can mount eight PPF-QCRs. The sensor module was

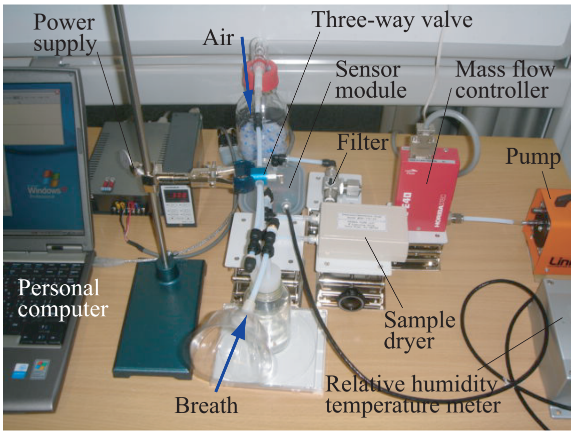


Figure 3: Experimenal setup of breath gas sensing system.

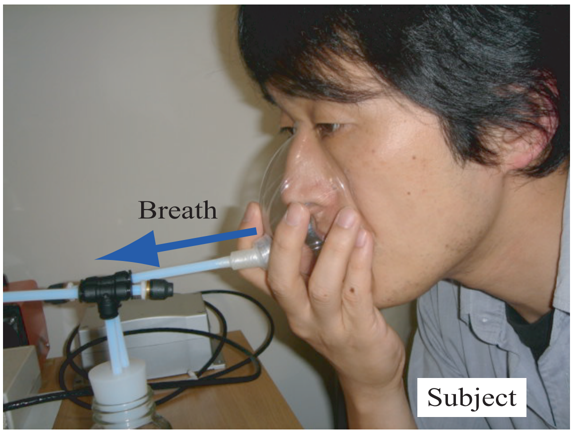


Figure 4: Collection of breath gas.

placed in a freeze sealing case. The relative humidity and temperature in the case were measured by a HMT337 sensor (Vaisala, Finland, accuracy: ±1.0% for 0 to 90 % relative humidity). As shown in figure 1, the breath measurement system consisted of two gas flow lines: a sample line carries breath gas and a reference line carries base gas. In our sensing system, air in indoor atmospheric condition was used as a reference or cleaning gas for establishing the initial state of the QCR sensors. These flow lines were switched by a three-way valve to connect with the gas line of the sensor cell. The flow rates of all flow lines were controlled around 300 mL/min generally by a mass flow controller. The flow line connected to a sensor cell was changed by a three-way valve.

3 Experiments

3.1 Response of Sensor for Breath Gas

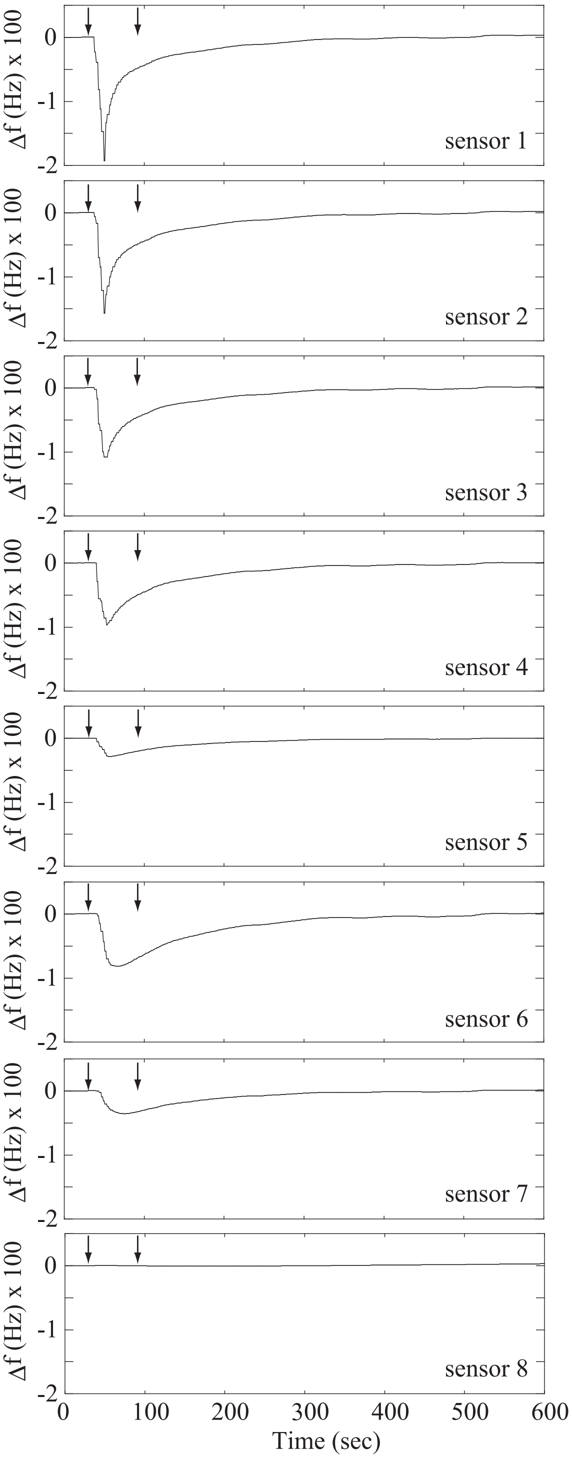


Figure 5: Examples of responses from breath gas sensing system.

The experimental setup for measuring and analyzing breath gas using arrays of PPF-coated QCRs in the laboratory is shown in figure 3. As shown in figure 3, dehumidification by a Silica gel was performed before it was introduced into the reference line. In the experiments, the breath gas was introduced into the sensor cell by switching the three-way valve after the baseline fluctuation had been suppressed to below ±0.1 Hz for 1minute. Data were collected at 1-second interval for the QCR sensor array and the relative humidity/temperature sensor.

In figure 5, examples of responses from each sensor cell when the subject strongly blew breath are indicated. The horizontal axis is time and the vertical axis is the frequency shift. The left arrow in each figure shows the moment that the subject blows breath to the sample line (30 seconds), and the right arrow indicates the moment that the flow line is changed into the reference line (90 seconds) by the three-way valve. Except for the sensor 8, the response of the frequency shift shows a local minimum after breath introduction and then converges to around zero as shown in figure 5. The amplitude of each response curve depends on how strongly the subject blows breath to the sample line, however, the shape of the response curve is not affected.

3.2 Computational Experiment of Emotion Recognition

Next, we collected the response of the sensing system under the condition of emotion recognition experiment. By considering a guide to gathering physiological data for affective recognition described in the affective recognition work [1], we carried out the experiments following an *event-excited*, *lab setting*, *feeling*, *open-recording*, and *emotion-purpose* methodology. In our experiment, two emotions, such as pleasure (positive emotion) and displeasure (negative emotion), and no emotion were considered. A subject imaged situations of feeling the emotion to excite the emotion, then the subject blew breath into the sample line of the sensing system. The experiment was carried out in our laboratory where the illumination, sounds, and room temperature were controlled to maintain uniformity. The subject was a male, native Japanese and the breath gas was collected four times, one time for each of the two emotions. Experimental results are shown in figure 6. The two features, such as peak Δf_p and pulse width Δt_p, are extracted from each response curve. The pulse width is defined by the interval that the output of the sensor crosses over the half of the peak. As shown in figure 6, it is hard to find remarkable differences depending on emotions from both features since all sensors have very similar response data.

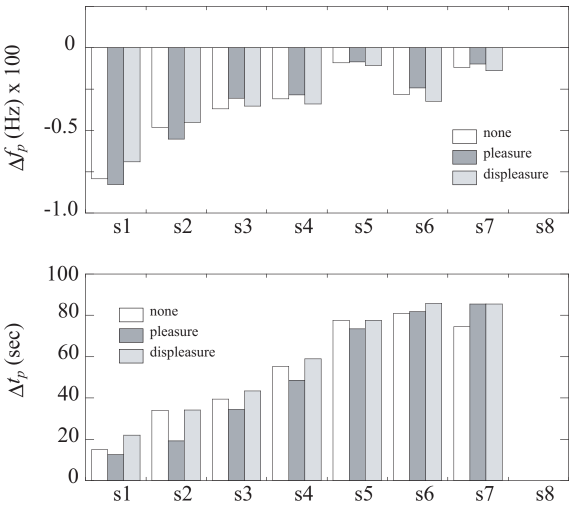


Figure 6: Experimental results of average features under emotional states.

To investigate a possibility of emotion recognition from the breath gas information by using machine learning techniques, support vector machines (SVMs) were applied to the features extracted from the raw signals of the sensing system as a feasibility study. The SVM is a learning algorithm based on statistical learning theory[16]. The basic training principle of the SVM is finding the optimal hyperplane where the expected classification error of test samples is minimized. To find the optimal hyperplane, the VC-dimension reduction principle is implemented. In the two-class pattern recognition problem, given a labelled set of P training data (x_i, u_i), where $x_i \in R^N$, and u_i is the associated label ($u_i \in (\ 1,1)$), the optimal hyperplane $wx+b=0$ can be found by minimizing:

$$\|w\|^2 + C\sum_{i=1}^{l}\xi_i$$

constrained by:

$$\xi_i \geq 0, u_i(w_i\ \ x+b) \geq 1\ \ \ \xi_i\quad (i=1,2,...,l)$$

where ξ_i is the slack variable that is introduced to account for non-separable data, C is the margin parameter that quantifies the trade-off between training error and system capacity. Solving the quadratic programming problem, the optimal hyperplane can be defined as follows.

$$g(x) = \sum_{i=1}^{Q} u_i \alpha_i^* K(x, x_i^*) + b^* \tag{2}$$

where $K(\ ,\ )$ is a kernel function, x_i^* is a support vector that corresponds to a nonzero Lagrange multiplier α_i^*, Q is the number of the support vector, and b^* is a bias parameter. The kernel function is a simple dot product for a linear SVM. In a nonlinear SVM, a nonlinear mapping function of the kernel function projects the data to high dimensional feature space and the optimal hyperplane is found in that space. Several kernel functions such as Gaussian and polynomial functions have been used in the nonlinear SVM. In this study, we used the following polynomial function.

$$K(x,y) = \gamma(x\ \ y+\delta)^d \tag{3}$$

where γ and δ are coefficients and d is the order of the polynomial kernel.

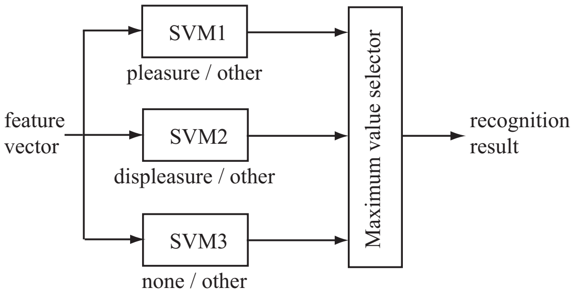


Figure 7: SVM-based emotion recognition system.

Table 2: Emotion recognition results using SVM [%] ($d = 2$).

In \ Out	Pleasure	Displeasure	None
Pleasure	**100.0**	0.0	0.0
Displeasure	25.0	**50.0**	25.0
None	25.0	50.0	**25.0**

Table 3: Emotion recognition results using SVM [%] ($d = 3$).

In \ Out	Pleasure	Displeasure	None
Pleasure	**100.0**	0.0	0.0
Displeasure	25.0	**75.0**	0.0
None	25.0	0.0	**75.0**

Originally the SVM is designed for two classes classification by finding the optimal hyperplane where the expected classification error of test samples is minimized. There are several approaches to apply the SVM for multiclass classification. In this study, the one-vs-all method [17] was implemented. The processing flow of the emotion recognition using SVM is shown in figure 7. Three SVMs that correspond to each of the three emotions were used. The *i*th SVM is trained with all of the training data in the *i*th class with positive labels, and all other training data with negative labels. In the emotion recognition process, the feature vector is simultaneously fed into all SVMs and the output from each SVM is investigated in the decision logic that selects the best emotion; the SVM that gives the positive label is chosen, and the class of the SVM indicates the recognition result.

The input vector of the SVMs was composed of the pulse width $\Delta t_{p_i} (i = 1, 2, \quad, 7)$ since it is hard to be affected by how to blow the breath. In the experiments of emotion recognition, training and testing of the classifier were carried out by the leave-one-out cross-validation method using the collected breath gas data. In the SVM classifier, the kernel function parameter γ was 1, δ was 1, and the margin parameter C was chosen as 1. These parameters were defined by trial and error in order to achieve complete classification rate for training data. The emotion recognition results are shown in tables 2 and 3. The averaged recognition rate defined with the average of the diagonal element is 58.3% in table 2 while the average recognition rate for the teaching data is 100%. As shown in table 3, the averaged recognition rate of 83.3% is achieved and the emotion recognition rates is improved by using the 3rd order polynomial kernel function. These results indicate that using breath gas information is feasible in emotion recognition and the SVM is suited for this task. However, the recognition results are obtained by using only one subject's data. Therefore, the collection of more number of sample data should be required and the feature extraction method from the raw signal of the sensor should be also investigated in order to improve the recognition ability of the classifier.

4 Conclusions

This paper proposed a smart gas sensing system to achieve nonverbal interface using breath gas information. A quartz crystal resonator with a plasma-polymer film was used as a sensor, and breath gas sensing system was designed under air in indoor atmospheric condition. To evaluate basic characteristics of the sensing system, breath gas test experiments were carried out. Experimental results indicated the feasibility of the sensing system for breath gas information .

There is still more work to be done in the field of breath gas sensing system and its application to emotion recognition. Improving how to introduce breath gas into the sensing-system is the important. Possible features of breath gas have to be selected in the feature extraction process, and further trials with different pattern recognition methods may help improve recognition performance.

5 Acknowledgement

This work was supported by JSPS Grant-in-Aid for Scientific Research (C) 18500142.

6 References

[1] R. W. Picard, E. Vyzas and J. Healey, " Toward machine emotional intelligence: analysis of affective physiological state", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), pp 1175-1191 (2001).

[2] L. S. Chen and T. S. Huang, " Emotional expressions in audiovisual human computer interaction", *Proceedings 2000 IEEE International Conference on Multimedia and Expo*, New York, USA, 1, pp 423-426 (2000).

[3] F. Dellaert, T. Polzin and A. Waibel, " Recognizing emotion in speech", *Proceedings Fourth IEEE International Conference on Spoken Language*

Processing, Philadelphia, USA, 3, pp 1970-1973 (1996).

[4] J. Nicolson, K. Takahashi and R. Nakatsu," Emotion recognizing in speech using neural networks", *Neural Computing and Applications*, 9(4), pp 290-296 (2000).

[5] L. C. De Silva, T. Miyasato and R. Nakatsu, " Use of multimodal information in facial emotion recognition", *IEICE Transaction*, E81-D(1), pp 105-114 (1998).

[6] I.A. Essa and A. P. Pentland, " Facial expression recognition using a dynamic model and motion energy", *Proceedings Fifth International Conference on Computer Vision*, Cambridge, USA, (1995).

[7] Y. Yacoob and L. Davis, " Recognizing human facial expressions from log image sequences using optical flow", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6), pp 636-642 (1996).

[8] K. Ishino and M. Hagiwara," A feeling estimation system using a simple electroencephalograph", *Proceedings 2003 IEEE International Conference on Systems, Man, and Cybernetics*, Washington, D. C., USA, pp 4204-4209 (2003).

[9] K. Takahashi, " Remarks on emotion recognition from multi-modal bio-potential signals", *The Japanese Journal of Ergonomics*, 41(4), pp 248-253 (2005).

[10] K. M. Dubowski," Breath analysis as a technique in clinical chemistry", *Clinical Chemistry*, 20, pp 966-972 (1974).

[11] A. Manolis, " The diagnostic potential of breath gas", *Clinical Chemistry*, 29, pp 5-15 (1983).

[12] M. Phillips," Breath tests in medicine", *Scientific American*, 267, pp 74-79 (1992).

[13] I. Sugimoto, M. Nakamura, S. Ogawa, M. Seyama and T. Katoh," Petroleum pollution sensing at ppb level using quartz crystal resonators sputtered with porous polyethylene under photoexcitation", *Sensors and Actuators*, B64, pp 216-223 (2000).

[14] I. Sugimoto, M. Seyama and M. Nakamura," Detection of petroleum hydrocarbons at low ppb levels using quartz crystal resonators sensors and instumentation of a smart environmental monitoring system", *Journal of Environmental Monitoring*, 1, pp 135-142 (1999).

[15] M. Seyama, I. Sugimoto and T. Miyagi, " Application of an array sensor based on plasma-deposited organic film coated quartz crystal resonators to monitoring indoor volatile compunds", *IEEE Sensors Journal*, 2(5), pp 422-427 (2002).

[16] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York (1995).

[17] O. Chapelle, P. Haffner and V. Vapnik, " Support vector machines for histogram-based image classification", *IEEE Transactions on Neural Networks*, 10(5), pp 1055-1064 (1999).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Towards a Computational Model of Affect for the Modulation of Mobile Robot Control Parameters

C.P. Lee-Johnson and D.A. Carnegie
School of Chemical and Physical Sciences,
Victoria University of Wellington, Wellington, New Zealand
christopher.lee-johnson@vuw.ac.nz, dale.carnegie@vuw.ac.nz

Abstract

Affective processes such as emotions and moods play an important part in human cognition, modulating our responses without requiring time-consuming deliberation. As the capabilities of software agents and robots move closer to those of humans, the potential value of simulating affective states in machines is becoming increasingly apparent. This paper describes an attempt to improve the adaptability of a mobile robot control architecture by incorporating a computational model of affect. The model evolves neural networks to modulate the operating parameters of the robot's planning and control systems based on appraisals of the robot's performance.

Keywords: mobile robot control, artificial affect, robotic emotion, neural network, genetic algorithm

1 Introduction

Modern cognitive theories acknowledge the strong influence that affective states such as emotions and moods have on human cognition. Affect provides functional advantages to tasks such as communication, memory retrieval, learning and perception [5]. As these tasks can now be performed by software agents and robots, incorporating "artificial affect" into machines has become a goal worth pursuing.

In addition to the potential performance improvements, artificial affect might improve the transparency of cognitive architectures. Humans think in affective terms, even to the point of imagining emotions for objects that do not possess them. Thus, an architecture that incorporates affective elements is more likely to be understood intuitively by a human observer.

However, applying affect to machines is not an easy task, as researchers are required to bridge the divide between computer science, neuroscience and psychology, fields which differ greatly in terminology and approach. Few widely-accepted theories exist, and authors cannot even agree on definitions for important terms [2].

The objective of this paper is to incorporate a model of affective states into the architecture of a mobile robot, and investigate how they influence its interaction with its environment. These states are not absolute arbitrators of its behaviours, but rather, modulators that adaptively adjust the control parameters of a hard-coded cognitive architecture.

This model is implemented on MARVIN (Mobile Autonomous Robotic Vehicle for Indoor Navigation), an autonomous security guard and public relations vehicle (figure 1). MARVIN possesses unique mobile torso and head units, allowing the robot to physically portray certain moods and emotions and to appear more intimidating or submissive depending on the actions of a potential intruder [1].

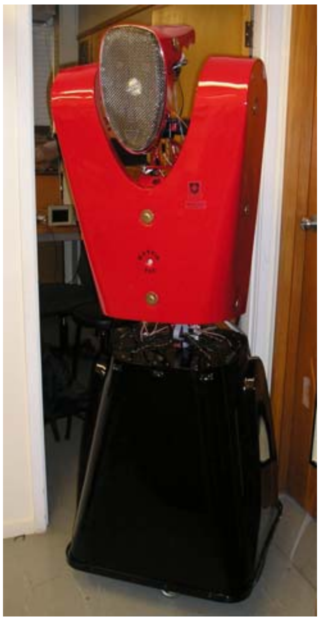


Figure 1: MARVIN, the mobile security robot.

2 Affective Mobile Robots

Many contradictions exist between the usage of words like "affect" and "emotion" by different authors. Some researchers use "emotion" as an umbrella term for all motivations, evaluations and/or reinforcements, while others define emotion as a subset of a more general category of affective phenomena [2]. This paper adopts the latter definition, with the term "affect" used to describe a general class of states that includes emotions, feelings, moods, motivations and attitudes.

A number of researchers have applied computational models of affect to robotics. There are two main practical applications for such a model.

First, affect can make a robot easier for a human to interact with [4][5]. In particular, affective signals such as facial expressions and tone of voice provide efficient channels of communication that operate in parallel with normal visual or verbal exchanges.

Second, affect can be a powerful mechanism for improving the survivability of a robot, by modulating its responses directly, without deliberation [3]. This is of particular interest to the mobile robotics community, whose creations are required to cope with sensor uncertainties and changing environments in real time.

Most affective models for mobile robots encountered in the literature thus far (e.g. [6][7][8][9][10][11][12]) incorporate only a small number of emotions that, although given human names such as "frustration" or "fear", attempt only shallow imitations of human affect. These models have largely been utilised only for action selection in simple behaviour-based control architectures.

In contrast, the control architecture described in this paper is a more complex multilayered system that incorporates both deliberative and reactive components. Although it does share some similarities with behaviour-based architectures, it is not explicitly behaviour-based. Instead of being utilised for action-selection, the affective model adaptively modulates the weights and thresholds of the planning and control systems, subtly influencing the robot's responses without explicitly adjusting its goals.

3 Planning and Control Architecture

MARVIN's navigation system incorporates both planning and reactive control in a multilayered architecture (figure 2). Each module possesses a number of parameters that can be tuned to improve performance in a given state.

A simple occupancy grid is utilised for path planning. Occupancy probabilities are updated in real time based on their proximity to measured obstacles and sensor coverage vectors. Changes to occupancy probabilities are damped to help prevent the robot from mapping dynamic obstacles such as humans. Damping factors can be varied to control the rate at which instantaneous sensor data influences the occupancy grid.

When a new target position is entered, the planner plots the shortest path from the current position to the target position by applying an A* algorithm [13] to the occupancy grid. The algorithm takes into account the occupancy probability of a cell when determining the cost of moving to that cell. If the time taken to travel along the planned path exceeds a threshold (indicating that the robot may have become stuck), the path is replanned, allowing the robot to utilise occupancy grid updates to improve its plan.

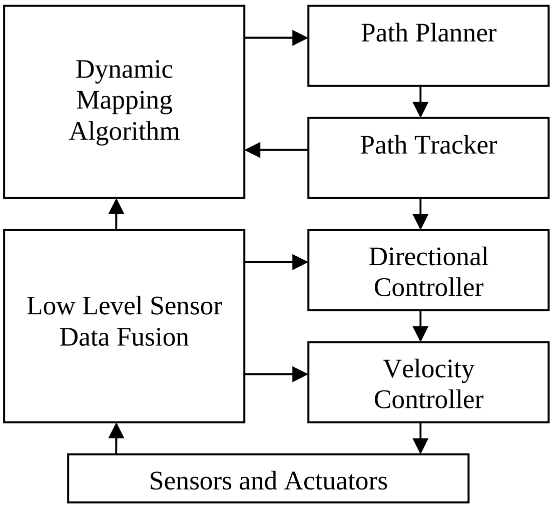


Figure 2: Planning and control system.

Each control cycle, the path tracking module obtains the direction of a node that is *N* nodes ahead of the closest point on the planned path. Adjusting *N* affects how closely the robot will attempt to adhere to the planned path.

Next, the planned direction is passed to a vector field histogram algorithm [14], which applies an optimisation to a discrete set of directions. A reactive direction is selected that maximises the distance to the nearest obstacle, while minimising the angular error with respect to the planned direction. The reactive direction takes into account any unmapped obstacles that may exist as the robot travels along its planned path. Important parameters include the granularity of the vector field discretisation and the relative weights of the parameters to be optimised.

Finally, a dynamic window algorithm [15] selects a linear and angular velocity couplet that drives the robot in the intended direction. An optimisation is applied to a discrete set of velocities that can be achieved given the robot's current velocity. The objective function to be optimised consists of a linear weighted combination of the following objectives:

- Speed up if it is safe to do so.
- Slow down when the robot gets close to its goal.
- Slow down if it gets close to an obstacle.
- Slow down if it is facing away from the reactive direction.
- Turn towards the reactive direction.

Key parameters include velocity search space granularity, absolute velocity limits, objective function weights and normalisation thresholds.

The navigation system can operate successfully with its parameters set to constant values. However, values that provide optimal performance in one state may yield dangerously unstable motion in another. On the other hand, utilising parameters that result in collision-free motion in all conceivable states means compromising the robot's performance in states where such a conservative approach is not required.

A superior solution is for the robot to recognise what state it is in and modulate its responses accordingly. This is essentially what the human affective system does, and what MARVIN's affective system is intended to model.

4 Affective Model

Given that MARVIN's control architecture bears little resemblance to the cognitive architecture of the human brain, there is little reason for it to possess the full range of human affective states, or for these states to have as broad an influence as they do in humans. Rather, this model is an attempt to capture some of the critical elements of human affect and utilise them to solve a problem of limited scope, with the intent to later broaden the model's functionality and apply it to other domains.

Human affect can be divided into two subsystems: appraisal and coping. Appraisal is the act of detecting your affective states. Coping is modulating actions to improve your physical welfare or, if insufficient improvements can be achieved, modulating appraisals to re-evaluate your priorities.

MARVIN's appraisal system is largely hard-coded, although, like the navigation system, its parameters can be modulated. Five affective dimensions are modelled, each the result of an appraisal of a single aspect of the robot's performance:

- **Caution** – How close is the robot to potentially dangerous obstacles?
- **Disorientation** – Do measured obstacles match its internal map?
- **Frustration** – Has the robot become stuck in a local minima?
- **Alertness** – What fraction of the available CPU resources is the current task consuming?
- **Urgency** – Is the robot heading towards its goal in a timely fashion?

Changes to affective variables are damped, allowing them to persist for a time after their initial stimuli have abated. This helps prevent oscillatory behaviour. Affective variables with high values indicate situations that are potentially detrimental to the completion of the robot's goal. The intent is therefore to adaptively modulate the robot's control parameters to minimise a linear weighted sum of the affective variables.

Unlike the appraisal system, the coping system possesses the ability to learn. It consists of multiple feed-forward neural networks that can modulate various planning, control and appraisal parameters, and whose inputs are the affective variables resulting from the appraisals (figure 3). This could be accomplished using a single network, but it is easier to train multiple small networks than one large network.

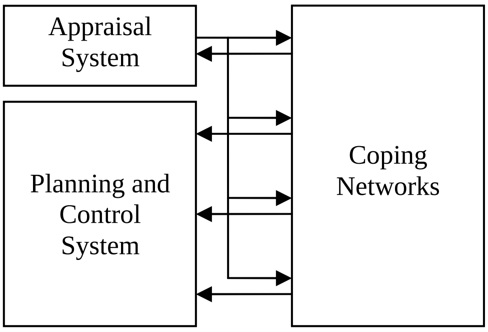


Figure 1: Affective model architecture.

First, the networks are trained independently using back-propagation to give outputs that are known to produce stable, collision-free (but non-optimal) behaviour. This yields a suitable starting point for the second training phase, which is implemented using a genetic algorithm.

The main advantage of a genetic algorithm over other training techniques is its versatility. A genetic algorithm can be tailored to solve almost any optimisation problem without requiring detailed knowledge of the system being optimised. Neural networks can be trained by evolving their weights and biases in order to optimise the genetic algorithm's fitness function. In this example, the fitness function measures the robot's performance while moving along multiple paths within a simulated environment.

Evolutionary learning can be applied to each neural network independently, or they can be evolved simultaneously. The latter method results in slower evolution, but it can yield higher quality systems with interdependencies that might not be possible in separately evolved networks.

5 System Implementation and Testing

The planning and control architecture and affective model are tested on a simulated version of MARVIN with the same sensors and drive system as the real robot. MARVIN features a two wheeled, differential drive system supported by casters at the front and rear. Exteroception is currently limited to a ring of 10 infrared distance measuring sensors with a maximum sensing range of 1.5 metres. This limited range means that the robot is heavily reliant on reactive control in order to safely avoid obstacles.

Training is performed within a simulated office block environment (figure 4). The environment is intended to be a significant navigational challenge for a robot. It contains doorways that are only a few centimetres wider than the robot itself. Obstacles are scattered throughout the environment that initially do not appear on MARVIN's internal map (although the robot is able to incorporate the obstacles into its map once it detects them).

The genetic algorithm fitness function simulates five paths between randomly selected waypoints within the rooms and at various points along the corridor. Randomness in the path selection means that the resulting fitness value is not always representative of the quality of the system, because the navigation "difficulty" is not equal during each iteration. However, it helps prevent the algorithm from converging on systems that perform well for the paths tested within the fitness function, but fail when presented with any other paths.

The fitness value is the weighted sum of affective appraisals performed throughout the simulated journey, combined with a parameter that measures the variability of the coping networks' outputs (the algorithm favours dynamic outputs). Failure states are represented by a fitness value of 1 or higher, and result from one or more of the following conditions:

- Collision with an obstacle.
- Failure to reach the goal within a given timeframe (i.e. becoming stuck).
- Code executing too slowly to function in real time.
- Evolved parameters or network outputs exceeding their allowable ranges.

For testing purposes, MARVIN's planning and control parameters are divided into four sets:

- Static parameters
- Dynamic parameters
- Directional control weights
- Velocity control weights

The first set of parameters can be evolved directly. The remaining three sets are the outputs of feed-forward neural networks whose weights and biases are evolved by the genetic algorithm. Each of these networks contains 5 inputs, 5 hidden neurons and 3-5 outputs.

Figure 5 shows the fitness values resulting from the simultaneous evolution of all the parameters and neural network weights over 120 generations. The following genetic algorithm parameters were utilised:

Population size:	30
Mutation function:	Adaptive Feasible
Crossover function:	Heuristic

This particular training run lasted about 32 hours. Training speed is expected to improve once the code has been optimised for performance, but this method is unlikely to ever be practical for online learning. However it does produce systems that can later be fine-tuned for real world operation. In this example, the best fitness value from each generation improved by about 30% over the course of the evolutionary training phase.

A second system comprised entirely of static parameters was evolved under the same conditions as the above example, in order to provide a performance comparison. Table 1 shows various performance parameters measured during of a sequence of paths through the simulated office block.

Table 1: Performance comparison between evolved systems with static and dynamic parameters.

	Static system	Dynamic system
Maximum velocity	0.82 m/s	1.42 m/s
Average velocity	0.41 m/s	0.41 m/s
Collision rate	35 %	34 %
Stuck rate	1.6 %	0.6 %
CPU usage	3 %	11 %

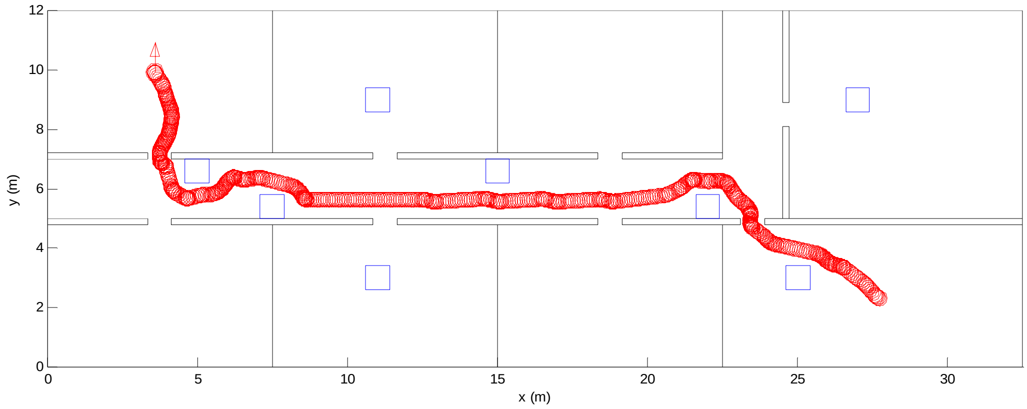


Figure 4: Robot's path through a simulated office block environment.

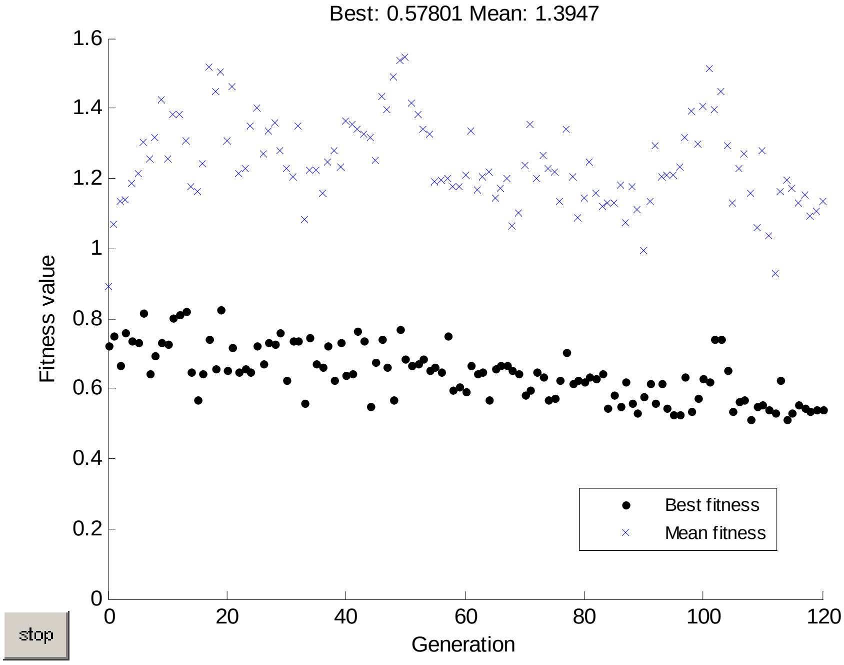


Figure 5: Fitness of evolved affective system over time (lower is better).

Overall, the dynamic system yields much higher peak velocities than the static system. It is also less likely to become stuck. There is no significant improvement in the collision rate. The dynamic system consumes more CPU resources than the static system, partly due to the inclusion of computationally intensive neural networks, but also because it evolved higher search space granularities.

The main factor limiting the reliability of the affective system in this test is the planning and control architecture itself. The test environment challenges the robot to exceed its original design constraints. In order to traverse narrow doorways, safety margins must be reduced to such a low level that some collisions are currently unavoidable. Thus, further improvements to the robot's reliability in this environment may require the inclusion of a specialised "doorway traversal" system.

6 Discussion

Preliminary results are encouraging, although the affective model has not yet been evolved to the point where it offers significant performance advantages over an equivalent system with static planning and control parameters. It is expected that the system's adaptability and performance can be greatly improved by adding features to the planning and control system and by expanding the genetic algorithm's fitness function to incorporate a broader range of test environments.

Because learning is implemented using a genetic algorithm, it is currently performed offline in simulation. While it is possible to evolve the system online, the method is ill-suited to this task. It is slow, and the random nature of evolution produces many unsuitable solutions that would result in collisions in the real world. Consequently, the current evolutionary learning algorithm will likely be supplemented or replaced with an online learning algorithm such as reinforcement learning.

As new features are added to the robot's cognitive architecture, increasingly complex affective-cognitive interactions will become possible. Once online learning is implemented, affective states can be utilised to influence the learning process. Adding long term memory capabilities will allow the robot to utilise *somatic markers* to favour previously travelled paths that correspond with its current affective state. If given object recognition capabilities, the robot could associate specific objects with affective states. For example, it may become cautious around humans if it has previously collided with one. Taking this idea further, if MARVIN were given the ability to recognise individual humans, it could learn to categorise individuals in affective terms.

7 Summary

Current research on human affect is extremely ad-hoc, with many competing theories and few agreements on important definitions. Nevertheless, affect is a powerful mechanism for facilitating adaptive behaviour. Models of affective states for software agents and robots might not only improve their performance, but also yield insights into the workings of human affect.

This paper describes the development of an affective model for mobile robots. Unlike previous models that largely utilise affect for action selection, this model modulates the operating parameters of a hard-coded planning and control system. The result is a robot whose actions are determined primarily by cognition, but which can utilise affect to adapt to changes in its own state or its environment.

8 Acknowledgements

The authors acknowledge the Tertiary Education Commission for providing the Top Achiever Doctoral Scholarship that funds this project. We would also like to acknowledge the other graduate students and staff in the Mechatronics Group, including Praneel Chand, Luke Cawley, Andrew Payne and Scott Forbes, who have provided assistance and support throughout the project.

9 References

[1] D.A. Carnegie, A. Prakash, C. Chitty and B. Guy, “A human-like semi autonomous mobile security robot”, *Proceedings 2nd International Conference on Autonomous Robots and Agents (ICARA2004)*, Palmerston North, New Zealand pp 64-69 (2004).

[2] A. Sloman, R. Chrisley and M. Scheutz, “The architectural basis of affective states and processes”, in J.M. Fellous and M.A. Arbib (eds), *Who Needs Emotions?: The Brain Meets the Robot*, Oxford University Press, Oxford, New York, pp 203-244 (2004).

[3] R. Arkin, “Moving up the food chain: motivation and emotion in behavior-based robots”, in J.M. Fellous and M.A. Arbib (eds), *Who Needs Emotions?: The Brain Meets the Robot*, Oxford University Press, Oxford, New York, pp 245-270 (2004).

[4] R.W. Picard, “What does it mean for a computer to ‘have’ emotions?”, in R. Trappl, P. Petta and S. Payr (eds), *Emotions in Humans and Artifacts*, MIT Press, Cambridge, MA, pp 213-236 (2003).

[5] C. Breazeal, “Function meets style: insights from emotion theory applied to HRI”, *IEEE Transactions on Systems, Man and Cybernetics*, 34(2), pp 187-194 (2004).

[6] T. Mochidai, A. Ishiguro, T. Aokitt and Y. Uchikawat, “Behavior arbitration for autonomous mobile robots using emotion mechanisms”, *Proceedings International Conference on Intelligent Robots and Systems (IROS 1995)*, IEEE Computer Society, Pittsburgh, PA, pp 3516-3521 (1995).

[7] J.D. Velasquez, “When robots weep: emotional memories and decision-making”, *Proceedings 15th National Conference on Artificial Intelligence*, Madison, Wisconsin, pp 70-75 (1998).

[8] S.C. Gadanho and J. Hallam, “Emotion-triggered learning in autonomous robot control”, *Cybernetics and Systems Special Issue: Grounding Emotions in Adaptive Systems*, 32(5), pp 531-559 (2001).

[9] F. Michaud, E. Robichaud and J. Audet, “Using motives and artificial emotions for prolonged activity of a group of autonomous robots”, *Proceedings 5th International Conference on Autonomous Agents*, Montreal, Canada, ACM Press, pp 188-189 (2001).

[10] M. Scheutz, “Affective action selection and behavior arbitration for autonomous robots”, *Proceedings International Conference on Artificial Intelligence (IC-AI 02)*, Las Vegas, CSREA Press, pp 334-340 (2002).

[11] R. Murphy, C. Lisetti, R. Tardif, L. Irish and A. Gage, “Emotion-based control of cooperating heterogeneous mobile robots,” *IEEE Transactions on Robotics and Automation*, 18(5), pp 744-757 (2002).

[12] M. Malfaz and M.A. Salichs, “Design of an architecture based on emotions for an autonomous robot”, *Architectures for Modeling Emotion: Cross-Disciplinary Foundations, Papers from the 2004 AAAI Spring Symposium*, The AAAI Press, Menlo Park, CA, pp 79-80 (2004).

[13] P. Judea, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, Reading, MA (1984).

[14] J. Borenstein and Y. Koren, “The vector field histogram: fast obstacle avoidance for mobile robots”, *IEEE Journal of Robotics and Automation*, 7(3), pp 278-288 (1991).

[15] D. Fox, W. Burgard and S. Thrun, “The dynamic window approach to collision avoidance”, *IEEE Robotics & Automation Magazine*, 4(1), pp 23-33, (1997).

Embedded Platform for Search and Rescue Applications

David A. Williamson and Dale A. Carnegie
School of Chemical and Physical Sciences,
Victoria University of Wellington, Wellington, New Zealand
david.williamson@vuw.ac.nz, dale.carnegie@vuw.ac.nz

Abstract

This paper describes the development of an embedded platform constituting the core of an Urban Search and Rescue system. The high level system constitutes a primary processor, camera, supervisory module and wireless gateway. The supervisory module facilitates acquisition and control of sensors and actuators. A distributed high efficiency power supply complements the low consumption of system components. Innovative design solutions have resulted in a compact, inexpensive, efficient and versatile embedded platform for semi-autonomous mobile robotic control, sensing and communications.

Keywords: USAR, search and rescue, robot, embedded, microcontroller, ARM7

1 Introduction

Robotic assisted urban search and rescue (USAR) aims to minimise human involvement in high risk emergency situations. These may include disaster zones caused by nature, terrorism or contamination. While a fast and reliable method of detecting people trapped under rubble could save thousands of lives, the efficiency of reported standalone SAR robots has been impeded by several factors. These include accessibility, efficiency and cost [1],[2].

A three-layer hierarchical system is being developed to address these issues. Successive layers within this system are termed grandmothers, mothers and daughters. A grandmother conserves the power of its smaller, more nimble payload of mothers by providing transportation to the edge of a disaster zone. Each mother, after penetrating the disaster zone, increases search efficiency by deploying, supervising and assisting a distributed network of no less than 10 matchbox-sized daughters.

The system in development expands on reported two-level systems [3],[4] by insisting the additional lowest level robots are disposable. Low cost design permits the use of alternate deployment methods to address otherwise inaccessible and irretrievable locations.

Fully autonomous victim location and rescue is not yet a reality. Human agents supply unduplicated decision making capabilities [5]. All reported systems require human-robot interaction to communicate with and coordinate the rescue of located victims. Full teleoperation, however, requires significant training and is notoriously difficult. Semiautonomous control of mobile robots mitigates the fatiguing aspects of teleoperation, saving human attention for non-automated tasks.

Autonomy is even more crucial in distributed systems where one human supervises multiple robots [6]. A solution to the mother requirements is being developed. The mother robots constitute the system's centres of intelligence. Grandmothers act as a base station, while daughters have limited functionality (primarily to locate a trapped person) and are highly dependent on instructions.

The mother embedded system constitutes a Single Board Computer (SBC), sensor/actuator interface module, camera, wireless router and a custom power supply. Subsequent sections offer a complete overview of the system and embedded hardware, followed by further detail into the low-level sensor/actuator interface module and power supply.

2 System Design

The mother must meet specific functional requirements within tight design restrictions to be successful. Functional requirements include high mobility, semi-autonomous operation, daughter supervision and manual override. Design restrictions include cost, efficiency, size, weight and minimum run-time. Figure 1 provides a high-level overview of the system being developed.

A Wi-Fi router connects local Ethernet-based system components, also providing a gateway to the grandmother and extended network. Core high-level components include an SBC, an IP camera and a supervisory module. Sensors and actuators are controlled by the supervisory module. Provision for enhanced image processing power via a DSP is made via an available Ethernet port.

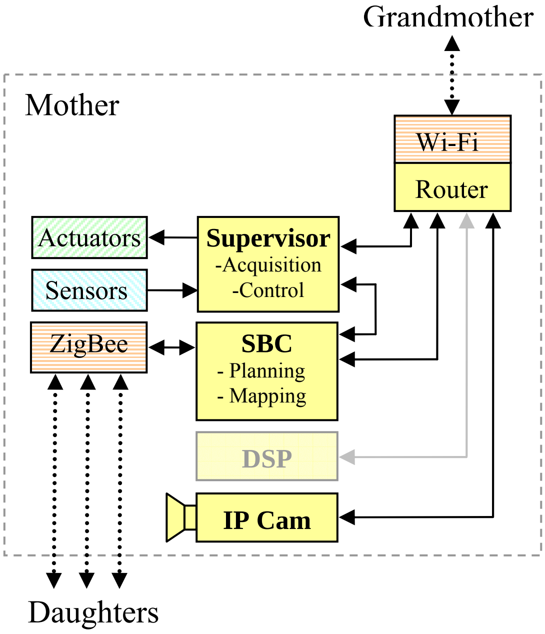


Figure 1: High-level system design.

2.1 Mechanical

The chassis design comprises symmetrical front and rear halves rotating on a common longitudinal axis. Each end conforms to the contours of rocky, uneven surfaces by twisting relative to the other. This suspension system eliminates the use of springs and is equally suited to inverted operation. Four local DC wheel drives offer optimum traction, with steering via differential skid. Mechanical simplicity adds robustness to drops, while vertical symmetry allows operation to resume from a wide range of orientations.

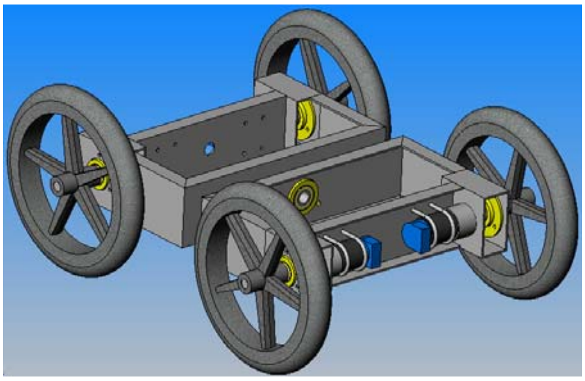


Figure 1: CAD chassis view.

The front chassis accommodates all system electronics. The rear chassis accommodates the payload of daughters and launching mechanism. Chassis construction primarily comprises welded aluminium, with dimensions of 400 mm wide, 500 mm long and 100 mm high. Weight, including chassis, wheels and motors, is 8.8 kg. This chassis is for proof of concept demonstration only, and will be re-evaluated for future development stages.

2.2 Communications

The mother robot must communicate with the grandmother. The popular IEEE WLAN communications standard 802.11g is utilised due to high bandwidth and ease of integration between products and operating systems. USAR personnel could potentially communicate with and relocate victims using 802.11g enabled PDA devices. Ranges around 500 metres are possible using high power transceivers at low data rates [7].

The mother must also communicate wirelessly with each of the daughters. The IEEE 802.15.4 based ZigBee communications standard for control and sensor networks is utilised. Support for peer-to-peer, star and mesh network topologies improves fault tolerance, while an alternate ultra-wideband (UWB) PHY under development, IEEE 802.15.4a, will add precision ranging capability. IEEE 802.15.4 is optimised for low power and low cost and operates up to a few hundred metres. The 260 kbps offered by current 2.4 GHz ISM modules is sufficient for low resolution images or audio.

2.3 Main Processor

The iEi WAFER-LX Single Board Computer is selected as the main processor for the mother and features an AMD LX-800 providing 800 MHz processing power at 6 W. Based on prior experience of the VUW Mechatronics Group [8] in constructing semi-autonomous mobile robots, we believe this processor will be sufficient to implement all of the mother's required high-level functionality.

The Damn Small Linux-Not (DSL-N) distribution of Linux is utilised as the operating system. DSL-N, at 80 Mb, is USB flash-disk bootable and can operate entirely from RAM. This adds robustness over both the moving parts of a HDD and the limited life of flash memory based Disk-On-Modules.

2.4 Camera

An IP camera serves an audio-visual stream directly to multiple remote viewers. The local SBC grabs and processes still images as required.

The AXIS 207 IP network camera selected features 30 fps full VGA MPEG-4 encoding, achieving up to ten times the compression of MJPEG. This increases effective WLAN operating range. Free access to the API and a Windows SDK simplified camera integration.

3 System Acquisition and Control

The mother SBC lacks the real time control and low level peripheral interfaces required by the system sensors and actuators. The system also requires a supervisor to control power distribution, administer override commands and reboot the SBC.

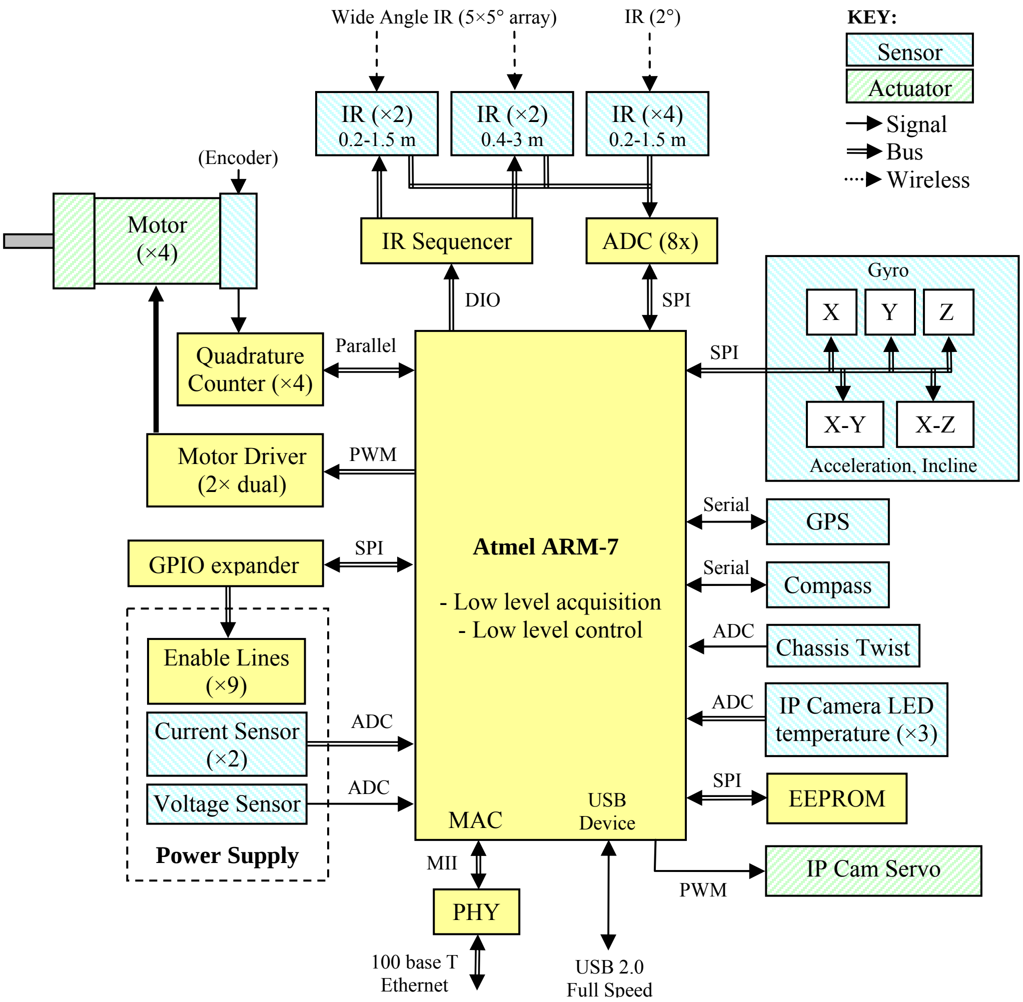


Figure 2: Acquisition and control diagram.

These requirements are met by an Atmel AT91SAM7X ARM7 microcontroller.

Figure 3 provides an overview of the sensors and actuators utilised, including their respective interfaces. The AT91 USB device port interfaces directly to the SBC. The Ethernet Medial Access Controller (MAC), complemented by a DP83848C physical layer transceiver (PHY), accepts remote commands via the WLAN.

3.1 IR Sequencer Module

The Sharp GP2Y300xK0F series of IR sensors offer an economical solution to wide angle ranging. Each sensor embeds five IR transmitters and one position sensitive device (PSD), offering 25 degrees total field of view. Only one transmitter per sensor may be enabled at a time.

The mother includes a sequencing module to reduce the GPIO requirements of multiple sensors to two lines. A clock signal both resets each PSD upon change of transmitter and increments a Johnson decade counter. Each counter output drives an IR transmitter. A reset signal enables modulo-n counter operation.

Possible IR interference between adjacent devices necessitates the option of sequential instead of concurrent sampling. Counter modulo-5 is implemented for parallel, and modulo-10 for series operation of two adjacent devices, respectively.

3.2 Inertial Measurement Unit (IMU)

A MEMs based six-degrees-of-freedom IMU is utilised. Two ADIS16201 dual-axis accelerometers provide both 14-bit acceleration and 12-bit inclination outputs. Three ADIS16080 gyros provide 12-bit angular rate outputs. Each sensor is interfaced via the SPI bus. Orthogonal PCB mounting, as shown in figure 4, provides inherent axial alignment.

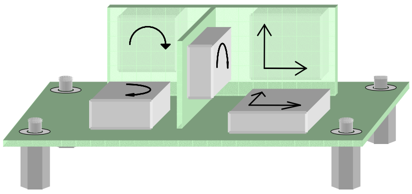


Figure 3: IMU orthogonal PCB layout.

3.3 Quadrature Decoder Module

An LSI7566 24-bit four axis quadrature counter directly interfaces the incremental encoders on the mother wheel drives. This solution relieves the AT91 of both hardware resource requirements and repetitive software routines.

4 Power Supply

Preliminary estimates for operating power consumption yield 20 W for electronic hardware and 80 W for motors. One hour travelling and two hours stationary operation requires a minimum 140 Wh of battery power. Lithium Polymer (Li-Po) battery packs are utilised for power density, weight and memory benefits over competing chemistries.

Conversion losses through pre-regulation are avoided by powering the motor drivers directly from the batteries. Li-Po packs operate safely between 16.8 and 12 V, well within the 6 to 24 V window accepted by both motors and drivers. Three 4000 mAh 4-cell series packs in parallel provide 178 Wh at 1.1 kg total weight.

Having chosen a primary power source, regulation for all other electronics is required. This supply must meet size, efficiency, cost and weight restrictions. Figure 5 provides an overview of the solution implemented.

A 15 W capacity supply meets the 6 W at 5 V Wafer-LX SBC requirements, allowing for USB peripheral and possible PC-104 bus current draw. This is achieved with an LM2677-adj switcher. Post-regulation by an LP3853 ensures an output well within the SBC ±5% tolerance. Provision for a 12 V regulator is included to accommodate possible PC-104 peripheral requirements.

The IP camera, Wi-Fi router, IR sensors and wheel encoders also operate on 5 V. Fault tolerance is improved by supplying these peripherals separately from the SBC. An LM2670 switcher feeds each device through separate high-side power distribution switches. Each power switch has an enable line, over current and thermal protection. A camera tilt servo shares the camera switch output.

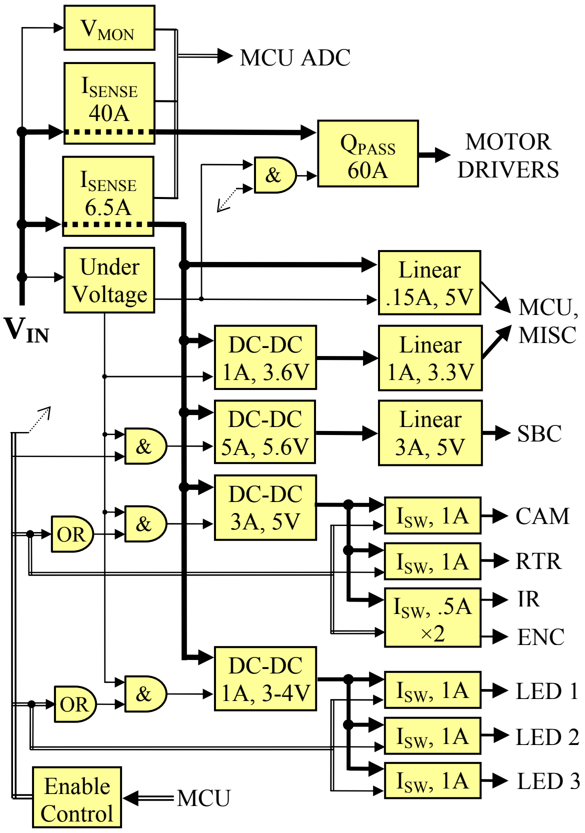


Figure 4: Mother power supply overview.

Three white Cree XR LEDs each provide 47 lum.W^{-1} illumination to improve IP camera performance in dark situations. An independent LM2670 adjustable switcher supplies each LED up to 700 mA through 1 A high side power switches. Each switch features an enable line for flexible luminescence and power control.

The AT91 based acquisition and control module primarily operates from 3.3 V, with 5 V required only for the IMU Gyros and level conversion logic. An LM2672-adj switcher is post regulated by an LP3855 to provide 3.3 V at 1 A with remote sense feedback. An LM9076 provides 5 V at 150 mA.

The LM267x family of DC-DC switching converters used provide a highly integrated solution. The 260 kHz internal clock frequency allows use of moderate to small components, while still achieving the high efficiencies required. The LP385x linear regulator maximum dropout voltages are closely matched to switcher outputs to maintain efficiency.

4.1 Power Supply Cross Coupling

Power supply cross coupling is a major contributor of noise in distributed switcher designs. Two common measures of reduction are implemented. Firstly, an LMC555 provides a synchronisation clock signal. The minimum 280 kHz synchronisation frequency implemented preserves efficiency.

Each LM267x also features a parallel damped LC input filter optimised for minimum peaking [9]. This common design is advantageous in that damping requirements are simpler to achieve than for a pure LC filter.

4.2 Current and Voltage Monitor

Two ZXCT1010 high side current sense monitors are used to monitor the consumption of both the motors and all other hardware. High side sensing eliminates disruptions to the ground plane. This scheme allows the main hardware to be monitored with higher precision.

The main hardware current sense is calibrated to a 1 V output at a maximum expected current of 6.5 A. A non-inverting amplifier of gain 3.3 scales the current sense output to a voltage matched to the MCU ADC reference. The MCU 10-bit ADC yields a resolution of 6.3 mA/LSB.

The motor current sense must accommodate a 9.6 A stall current on each of four drive motors. The components selected give a 1 V output for 40 A peak current. This output is also amplified by a gain of 3.3 to give an ADC resolution of 39 mA/LSB.

The AT91 also monitors battery voltage. The raw change between charged and discharged battery voltage is 4.8 V. A resistor divider bringing the 16.8 V battery charge within the 3.3 V maximum ADC value reduces voltage change by a factor of five to less than 1 V, or 16.1 mV/LSB. An improvement over this method is to reduce the battery DC offset measured by the ADC.

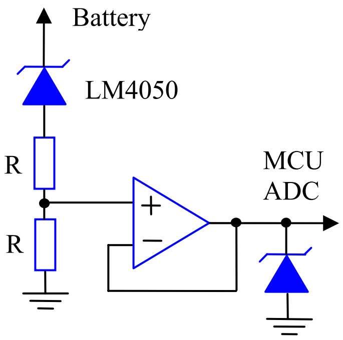


Figure 5: AT91 battery monitor circuit.

This is achieved using an LM4050 10 V precision reference configured as shown in figure 6. The observed change at the base of the zener is still 4.8 V, but only requires division by a factor of two before entering the ADC. The improved ADC resolution is 6.44 mV/LSB.

4.3 Power Management and Shutdown

The mother power supply provides a high level of control over power allocation. Eight peripherals, including the SBC, IP camera, router, IR sensors, wheel encoders and three LEDs, all feature MCU enabled control with manual override. The motor drivers can also be MCU controlled via a 2.5 mΩ BTS555 high-side power switch.

Figure 7 shows the enable logic for each 5 V peripheral and power LED. Enabling one or more power distribution switches automatically enables the associated supply, provided battery voltage is satisfactory. Manual override precedes MCU control but is still subject to battery voltage.

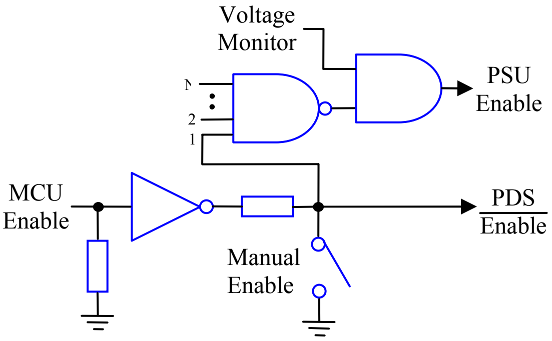


Figure 6: Power distribution enable logic.

The AT91 can estimate remaining battery capacity based on current and voltage measurements and, if necessary, prioritise peripheral usage to save power.

Li-Po cells must not be discharged below 3 V to preserve rated capacity. The ICL7665 dual-voltage monitor utilised features programmable hysteresis to account for loaded and floating cell voltage differences. A cut-off of 3.25 V with 0.25 V of hysteresis per cell is implemented.

The regulators supplying the AT91 are only disabled through a low battery voltage condition. Total power supply consumption in undervoltage shutdown is 15 mA.

5 Summary

This paper presents an innovative approach to the design of a compact, low-power, low-cost semi-autonomous robot. The authors believe the approach used in this design will be of substantial interest to researchers wishing to construct a similar scale device. The USAR application of this design has necessitated the integration of a number of sensors, communications devices and mechanical ruggedness which may be of particular interest to more specialist robotic developers.

The hardware described in this paper has been designed and is in the final stages of construction. By the time of the conference the system will have been completed and a quantitative description of its performance can be presented.

6 References

[1] A. Drenner, I. Burt, B. Kratochvil, B. J. Nelson, N. Papanikolopoulos and K. B. Yesin, "Communication and mobility enhancements to the scout robot", *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems,* Lausanne, Switzerland (2002).

[2] M.H. Yim, D.G. Duff and K.D. Roufas, "Modular reconfigurable robots, an approach to urban search and rescue", *Proceedings 1st International Workshop on Human Welfare Robotics Systems (HWRS2000)*, Taejon, Korea pp 19-20 (2000).

[3] R.R. Murphy, M. Ausmus, M. Bugajska, T. Ellis T. Johnson, N. Kelley, J. Kiefer and L. Pollock, "Marsupial-like mobile robot societies", *Proceedings 3rd International Conference on Autonomous Agents*, Seattle, pp 364-365 (1999).

[4] B. Jackson, I. Burt, B.E. Kratochvil and N. Papanikolopoulos, "A control system for teams of off-the-shelf scouts and megasScouts", *Proceedings 11th IEEE Mediterranean Conference on Control and Automation,* Rhodes, Greece (2003).

[5] R.R. Murphy and J.L. Burke, "Up from the rubble: lessons learned about HRI from search and rescue", *Proceedings 49th Annual Meetings of the Human Factors and Ergonomics Society*, Orlando (2005).

[6] P.E. Rybski, I. Burt, T. Dahlin, M. Gini, D.F. Hougen, D.G. Krantz, F. Nageotte, N. Papanikolopoulos and S.A. Stoeter, "System architecture for versatile autonomous and teleoperated control of multiple miniature robots," *Proceedings IEEE International Conference on Robotics and Automation (ICRA 2001)*, Seoul, Korea (2001).

[7] Buffalo Technology product datasheet, http://www.buffalo-technology.crom/documents/pdf/WHR-HP-G54_DS.pdf, visited 7/4/2006.

[8] D.A. Carnegie, "Towards a fleet of autonomous mobile mechatrons", *Proceedings 2nd International Federation of Automatic Control Conference on Mechatronics Systems*, Berkeley, CA, pp 673-678 (2002).

[9] M. Sclocchi, "Input filter design for switching power supplies", http://webench.national.com/appinfo/power/files/Input_filter_paper2.doc, visited 23/5/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Feasibility Study of Sensing Methods on Cooperative Localisation for Team Operation of Multiple Mobile Robots

Hiroyuki Takai[1], Jun Mitsuoka[1], Gen'ichi Yasuda[2] and Keihachiro Tachibana[3]
[1]Graduate School of Information Sciences, Hiroshima City University, Hiroshima, Japan
[2]Faculty of Engineering, Nagasaki Institute of Applied Sciences, Nagasaki, Japan
[3]Graduate School of Computer Science, Osaka Gakuin University, Suita, Osaka, Japan
takai@sys.im.hiroshima-cu.ac.jp, yasuda@csce.nias.ac.jp, tatibana@utc.osaka-gu.ac.jp

Abstract

Recently, multiple mobile robot systems that perform team operations have been researched. Since the robot team often has to work without collisions in a small workspace where there are many obstacles, they need a function to detect the exact position of teammates and obstacles. In this paper, we propose a cooperative localisation algorithm that involves two steps: 1) teammate position detection and 2) temporary landmark detection. A teammate position can be measured by using infrared incidence angle detectors, since all teammate robots send infrared signals for communication. A temporary landmark can be detected through the visual representation of its spatial features using an image sensor and a fan beam laser. These two algorithms were tested in experiments in an ideal workspace and successful results were obtained.

Keywords: localisation and navigation, sensor network, network robotics

1 Introduction

In recent years, multiple mobile robot systems that perform team operations have been researched [1]. These robot teams are expected to achieve complicated tasks such as repairing nuclear reactors or waste incineration facilities. Since these robots often have to work without collisions in a small workspace with many obstacles, they need a function that detects the exact position of teammates and obstacles. To avoid collisions with teammates or obstacles and to achieve tasks smoothly, team member robots need to share positional information.

However, global coordinate systems for mobile robots are rarely preprogrammed in the robot or installed within a given workspace. Moreover, since landmarks used as reference positions to decide common coordinate axes and a common origin are not always prepared in advance in a workspace, each robot needs to define a private coordinate system. Positional information that is determined by each robot is impossible to share among teammate robots directly, because each robot operates under its own coordinate system and also the coordinate axes and origins differ from one robot to another.

We examined a cooperative localisation method for a mobile robot team in an unknown unstructured workspace. This method involves two steps: 1) teammate position detection using an infrared incidence angle detection system and ultrasonic sonar, and 2) temporary landmark detection using an image sensor combined with a fan beam laser. In the first step, temporary common coordinate axes are defined between a few adjacent robots using an infrared incidence angle detection system and these robots detect and disseminate teammates' positional information mutually. In the second step, these adjacent robots detect a temporary landmark using a fan beam laser and the landmark's positional information is calculated and shared among the robots.

In this paper, a teammate robot position detection algorithm and a temporary landmark position detection algorithm are described.

2 Communication System for Team Operation of Mobile Robots

To achieve complicated tasks smoothly, the number of robots in a workspace is increased. These robots need to communicate mutually to disseminate each robot's role and task sequence so as to avoid collision or obstructing each other's work. Usually, radio waves or infrared rays are used as the wireless communication carrier.

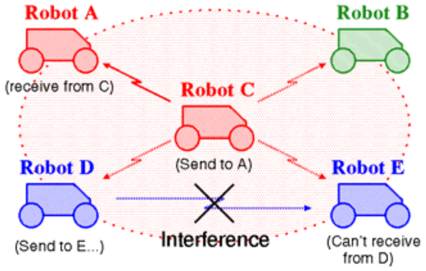


Figure 1: Interference of omni-directional signals.

As radio waves are omni-directional signals, signal interference often occurs if multiple robots use the same communication channel. Figure 1 shows interference of omni-directional signals.

On the other hand, since infrared rays have strong directivity and limited beam width, there is less signal interference. However, infrared communication systems often lose connection when robots move or rotate due to the directional nature of infrared rays. We designed an infrared communication system that can maintain connections when the robot moves or rotates.

2.1 Infrared Communication System for Mobile Robot Team

Since infrared rays have strong directivity and limited beam width, infrared communication systems often lose connection when these robots move or rotate. In order to maintain the connection, in the communication system, eight infrared transceivers are put on the circumference of the robot body facing outward to communicate in all directions (figure 2).

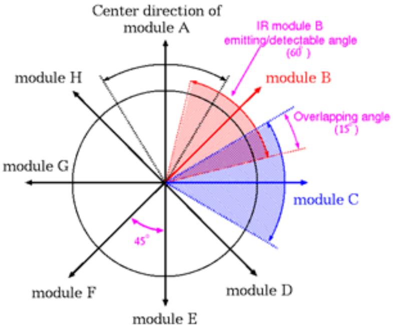


Figure 2: Arrangement of transceivers.

Each infrared transceiver is arranged so that both ends of the transceiver's reception area overlap with adjacent reception areas. These transceivers have a function that measures signal incidence angle, and for that reason the communication system can know the directions of the other robots and maintain the connections by tracking the other robots.

Yoshida et al. [2] and Arai et al. [3] have also proposed infrared inter-robot communication systems in order to avoid collisions in a small number of robots. They focused on communication with adjacent robots that are likely to collide. We thought about the creation of an inter-robot communication network and the occurrence of signal interference.

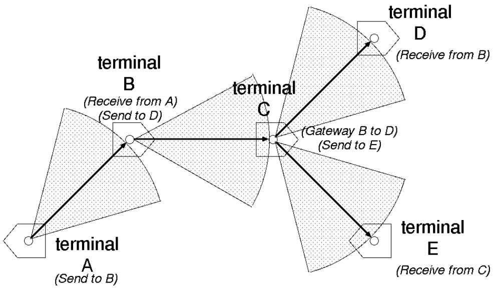


Figure 3: An inter-robot communication network.

The infrared inter-robot communication system that we designed is able to communicate in parallel with different partners in different directions without signal interference by using different transceivers simultaneously. As a result, space-division communication can take place. Since each robot can serve as a node in a communication network, they can create a communication network by relaying information among themselves. The created communication network is an ad hoc network, since all robots are independently mobile and may change positions depending on their tasks. Figure 3 shows an example of an inter-robot communication network.

When a transceiver on a robot receives signals from two or more teammate robots at the same time, the received signals become confused. If this happens, the robots in question will form a triangle. The interior angle of the triangle can be calculated from signal incidence angles of the infrared transceivers and each robot then informs other teammate robots of its own interior angle. Because the widest interior angle of the triangle is greater than or equal to 60 degrees, if the reception angle of each transceiver is restricted to 60 degrees or less, the robot that has the widest interior angle can communicate without signal interference with the other two. The robot that has the widest interior angle of the triangle becomes an arbiter that temporarily locally controls the other two robots thus reducing signal interference. The arbiter mediates both teammates by scheduling their transmissions. Figure 4 shows the selection of the arbiter. As the robots move and the triangle changes shape, the arbiter role is handed over to another robot.

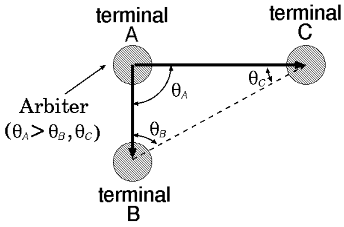


Figure 4: Selection of an arbiter.

2.2 Teammate Position Detection using Infrared Incidence Angle Detection

As previously mentioned, this infrared communication system enables the accurate detection of teammate direction by utilising the signal incidence angle measurement function. This means that it will be possible to calculate the position of teammates based on triangulation, if the distance between teammates can be measured.

Since usual ultrasonic sonar is used to measure distance to an object based on the time of the arrival of reflection signals, it is not suitable for measuring distance to a teammate. In order to measure the distance to a teammate, an ultrasonic sonar and an infrared communication system have to send their signals synchronously, because the distance can be measured based on the time difference of arrival of these signals.

Each robot can serve as a beacon transmitter for mobile robot navigation. This means that, if a certain robot serves as an origin, coordinate axes can be used together with the infrared communication between adjacent robots to infer the position of robots. When three teammates' positions are known in a workspace, a robot is able to compute its mutual position from measured interior angles and distances. These teammates' positions are arranged on coordinates from $P_1(x_1, y_1)$ to $P_3(x_3, y_3)$, and P_2 is the origin. When, the coordinate of the robot is $P(x, y)$ and the movement direction of the robot is θ these parameters are computed from the equation (1).

$$\left.\begin{aligned} x &= \overline{p_2 p} \cos \phi \\ y &= \overline{p_2 p} \sin \phi \\ \theta &= \phi - \theta_{01} - \theta_{12} + \pi \end{aligned}\right\} \qquad (1)$$

Parameters are computed as follows.

$$\phi = \tan^{-1} \frac{\overline{p_1 p_2} \sin(\theta_{12} + \alpha) \sin \theta_{23} - \overline{p_2 p_3} \sin \theta_{12} \sin \theta_{23}}{\overline{p_1 p_2} \cos(\theta_{12} + \alpha) \sin \theta_{23} + \overline{p_2 p_3} \sin \theta_{12} \cos \theta_{23}}$$

$$\overline{p_2 p} = \frac{\sin \theta_{23} \cos \phi + \cos \theta_{23} \sin \phi}{\sin \theta_{23}} \overline{p_2 p_3}$$

$$\alpha = \tan^{-1} \frac{y_1}{x_1}$$

Figure 5 shows mutual position computation using triangulation. The robot computes interior angles $(\theta_{01}, \theta_{12}, \theta_{23})$ from measured signal incidence angles and computes coordinate $P(x, y)$ by triangulation.

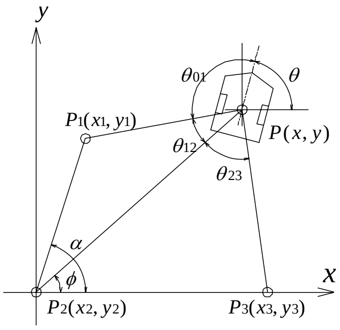


Figure 5: Mutual position computation.

Since all robots can move independently and may change their positions depending on their tasks, they have to compute their mutual positions continuously.

Takita et al. [4] have also proposed a mobile robot navigation method that detects the direction of an infrared light source mechanically. However, it cannot operate in an unstructured workspace because several infrared emitters that serve as landmarks have to be prepared in advance in the workspace.

As previously mentioned, the infrared communication system that we designed not only communicates among multiple partners but also is able to detect their mutual position. If a member of a robot team can find reference positions in a given workspace using onboard sensors, these robots are able to detect accurate positions in the workspace using this infrared communication system without landmark emitters prepared in advance.

3 Temporary Landmark Detection using Onboard Sensors

Although teammate robot position detection is extremely useful for local mutual cooperation among adjacent robots, it is not suitable for pointing to a specified position such as a stationary object in the workspace. Since there is no positional information on landmarks in an unknown workspace, it is difficult to define the global coordinate axis in order to point to a specific position. We considered a detection method of local temporary landmarks in order to define locally temporarily common coordinate axes. In order to detect objects in the workspace, an image sensor or ultrasonic sonar are used as an obstacle detector.

However a usual image sensor can capture precise images and can detect objects' directions accurately, it is quite difficult to measure depth or distance from these images. Since usual image sensor also has a slow capture interval that is approximately 30ms, it is not suitable for tracking fast moving objects.

Ultrasonic sonar has the opposite features to the usual image sensor. Since ultrasonic sonar can transmit signals in a broad area and can respond quicker than an image sensor, it is suitable for general obstacle detection. In addition, ultrasonic propagation velocity is approximately a million times slower than light velocity, making it suitable for time-based measurement of short distance up to a few meters.

In order to avoid collisions with obstacles and to plan behaviour in a small workspace, a robot team needs to know positions and features of obstacles accurately. Since it is difficult for usual image sensor to detect depth or spatial features, it needs to make other images or create shadow by using extra light sources. On the other hand, ultrasonic sonar also needs to rotate mechanically in order to detect surrounding obstacles by scanning, since sonar can only measure a distance to one frontal point.

3.1 Detection of Geometrical Feature of Landmark using a Fan Beam Laser

We examined a fan beam laser that can draw a straight line on a flat surface as an extra light source for an image sensor. The fan beam laser is also called a laser line generator or a slit laser. When the fan beam laser illuminates an object, it can create a visual representation of the object's geometrical features.

Torrie et al. [5] have also proposed an obstacle avoidance method using several fan beam lasers emitting in a vertical manner. They focused on a simple measurement method of cursory distances between obstacles in the surrounding environment for the purpose of a planetary investigation. We considered about selecting a temporary landmark in given workspace using the captured visual representation.

We examined a concept where a small number of teammate robots in close proximity can capture the visual representation of the same object. In order to confirm fundamental principles of a cooperative localisation method, we considered an ideal workspace, where a floor is flat and all objects stand perpendicularly. This ideal workspace is similar to most buildings where there are many walls and pillars on a flat floor. In addition, some of the walls and pillars create "corners/edges" to the workspace. As a result, these "corners/edges" can be used as temporary reference points in the workspace, because they are easy to detect using a fan beam laser emitting in a horizontal manner. Figure 6 shows a model of a "corner/edge" detection using a fan beam laser.

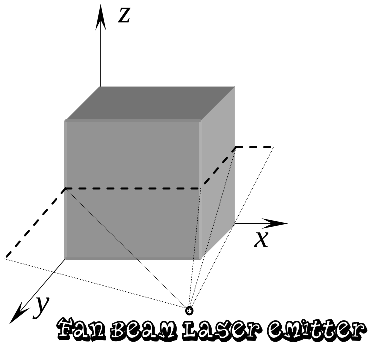


Figure 6: A model of a "corner/edge" detection.

Each robot there are in close proximity can detect the direction of a "corner/edge" by using an image processing algorithm from the captured visual representation. Since each robot has known mutual positional information from other teammate robots and can measure an interior angle between the "corner /edge" and an adjacent teammate, they can compute and share the positional information of the "corner/ edge" based on triangulation. The position of a visual representation that can easily be captured and computed from far distance can be used as a temporary landmark in the workspace.

3.2 Improvement in Distance Measurement Performance of Ultrasonic Sonar

As previously mentioned, since both incidence angle detection using an infrared communication and visual representation capture using an image sensor and a fan beam laser are not suitable for distance measurement, ultrasonic sonar measurement of distance has to be used as support. Conventional ultrasonic sonar measures a period in time when the envelope curve of received echo signal exceeds a threshold value from ultrasonic signal transmission. Since the threshold value of conventional sonar has weak theoretical background and is mostly a guess, distances measurement of conventional sonar is flawed. Even if the sonar measures a period of maximal amplitude of the received echo signal from a transmission, because it measures the distance of maximum reflective cross section, it does not represent true distance. Figure 7 shows the occurrence of the distance measurement error of ultrasonic sonar.

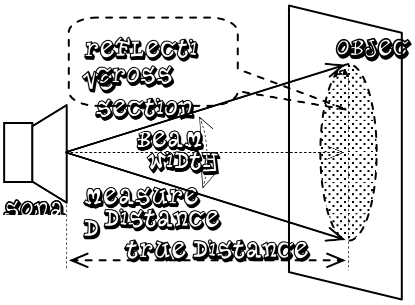


Figure 7: Occurrence of distance measurement error.

Since ultrasonic signals spread from the centre of a beam to a beam width as shown in figure 7, the amplitude of an echo is proportional to the temporal change in the size of a reflective cross section. We examined an algorithm that estimates the true value. Figure 8 shows a model of time period estimation algorithm.

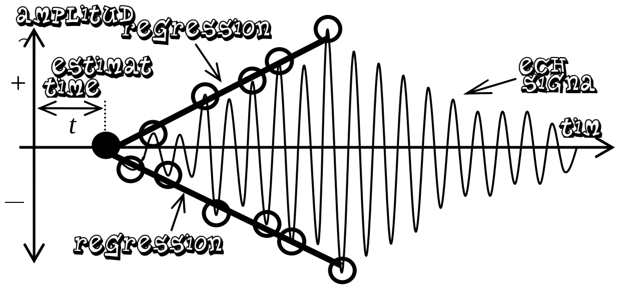


Figure 8: Model of time period estimation algorithm.

In figure 8, horizontal axis is time and vertical axis is amplitude. Several data sets of time and amplitude are measured and then regression lines p and q are computed based on the measured data sets using the least-square method. The time coordinate of intersection of regression lines p and q shows the estimated period t. The distance is transformed from the estimated time period t based on the acoustic velocity. In order to scan surrounding obstacles, ultrasonic sonar needs to rotate mechanically because sonar can only measure a distance of one frontal point.

4 Feasibility Confirmation on Multi-robot Cooperative Localisation

We tested the feasibility of multi-robot cooperative localisation by using onboard sensor systems. The cooperative localisation involves two steps: 1) teammate robot position detection and 2) temporary landmark position detection. The performances of onboard sensor systems that are used for each position detection algorithm are evaluated to examine the feasibility of cooperative localisation.

4.1 Performance Measurement of Teammate Robot Position Detection

A teammate robot's position is calculated from the incidence angles of infrared communication signals among them. The PSD photo diode (HAMAMATSU S6560) that can detect incidence angle of infrared rays was used as reception device for the infrared communication system (figure 9). This PSD photo diode has two current output electrodes and these ratios show the incidence angle.

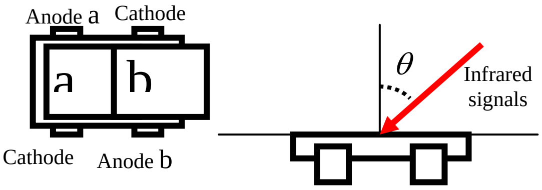


Figure 9: Schematic view of the PSD.

The electric current output *a* and *b* of the PSD are related to the incidence angle of the infrared signals θ as shown in equation (2).

$$\theta = (a-b)/(a+b) \qquad (2)$$

In an experiment of incidence angle detection, angle detection error was approximate 0.5 degrees.

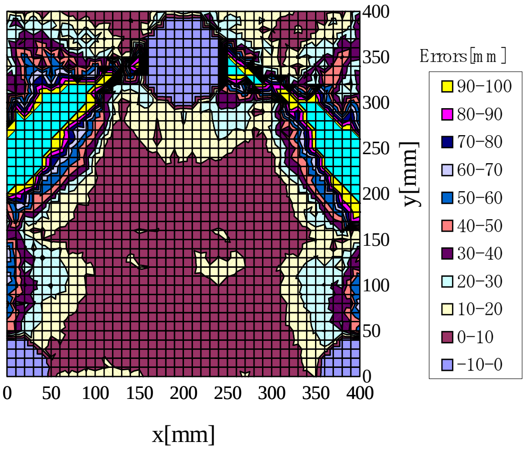


Figure 10: Accuracy of teammate position detection.

The accuracy of teammate robot positioning was confirmed by simulation. Three landmarks that were 100mm in diameters were placed on vertices of an equilateral triangle with 400mm side's length. Figure 10 demonstrates the accuracy of teammate position detection.

In figure 10, teammate robot positions can be calculated to a 90% or greater accuracy inside the triangle formed by the landmarks. Outside this triangle, there are some places where it is possible to compute positional information precisely, and some where it is impossible.

4.2 Performance Estimation of Temporary Landmark Position Detection

A temporary landmark position is calculated from a visual representation of the object's geometrical features that is created using a fan beam laser. A fan beam laser emitter (Coherent, LVM3 Laser-line micro 670nm 2.5mW 60degree) and image sensor (RF system Lab, PRO-5 2CCD colour wireless TV camera with lens) were used in the experiments. The laser emitter and the image sensor were attached to a motorised turntable (SIGMA KOKI, SGSP-160YAW) horizontally at 100mm vertical intervals.

The image sensor can capture 320 (horizontal) × 480 (vertical) pixel images; its horizontal view angle is approximately 23 degrees. The angle detection error of the image sensor capturing a visual representation was approximately 1.5 degrees in this experiment.

The temporary landmark position detection capability was examined. A motorised X-Y axis linear stage pair (SIGMA KOKI, SGSP46-500(X) and SGSP65- 1200(X)) was used for simulation of a mobile robot that moves around a cuboid obstacle set on the flat floor. The turntable on which the laser emitter and the image sensor were attached was put on the X-Y axis linear stage pair. The robot moved from $P_1(x_1,y_1)$ to $P_6(x_6,y_6)$ and the positions of the cuboid's edges $P_a(x_a,y_a)$, $P_b(x_b,y_b)$ and $P_c(x_c,y_c)$ were measured.

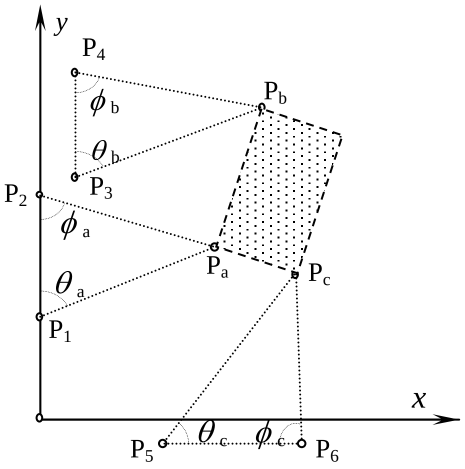


Figure 11: Experiment environment with an obstacle.

Figure 11 shows experiment environment with an obstacle. The accuracy of temporary landmark detection is shown in table 1, where 'x' and 'y' are arrangement coordinates. Measured angles θ and ϕ are shown in 'angle', detecting values 'x'' and 'y'' are the results of triangulation. The detection error was less than 3% in this experiment.

Table 1: Accuracy of temporary landmark detection (length: mm; angle: degree).

	Setting value		Measured Angle		Detecting value	
	x	y			x'	y'
P_a	500	500			498	510
P_1	0	300	θ_a	68.8		
P_2	0	650	ϕ_a	73.4		
P_b	635	900			633	890
P_3	100	700	θ_b	70.4		
P_4	100	1000	ϕ_b	78.3		
P_c	720	430			715	442
P_5	350	-75	θ_c	54.8		
P_6	750	-75	ϕ_c	86.1		

4.3 Confirmation of Distance Measurement Accuracy of Ultrasonic Sonar

We also confirmed the distance measurement accuracy of ultrasonic sonar. Ultrasonic speaker and microphone set (Murata MA40B5S/R, Carrier frequency: 40KHz) were used in the experiments. A resin flat panel target (1500mm tall × 1200mm wide) was installed perpendicularly at 1300mm distance. The received signal was amplified by a 40KHz tuned FET amplifier and captured by a 12bit A/D converter (Interface PCI-3525) in 200KHz sampling intervals.

Both the conventional method that measures time from the transmission to when maximum amplitude is received and the proposed method that calculate the least-square method from the received signal were compared. In addition, a horn (50mm length) was installed in front of the ultrasonic speaker or microphone. Table 2 shows the accuracy of distance measurement of the ultrasonic sonar.

Table 2: Accuracy of distance measurement (length: mm; target distance: 1300; horn length: 50).

Method	Conventional			Proposed		
Horn	W/O	MIC	Speaker	W/O	MIC	Speaker
Detecting value	1372	1387	1382	1287	1294	1294
Error	72	87	82	-13	-6	-6
Deviation	1.58	0.84	0.75	0.90	0.11	0.11

In table 2, the ultrasonic sonar was able to measure distance at approximately 1% or smaller error rate by the proposed method. Although the horn made deviation of distance measurement small, the error rate did not improve.

5 Conclusion and Future Work

We studied the feasibility of a multi-robot cooperative localisation algorithm. It was shown that a multi-robot cooperative localisation algorithm involves two steps: teammate robot position detection and temporary landmark position detection.

To detect teammate robot positions, we proposed an infrared communication system that can not only communicate mutually but also detect infrared incidence angles. By this method, it is possible to detect a teammate robot position to 90% accuracy or greater except where it is impossible to calculate positional information by triangulation.

To detect temporary landmark positions, we proposed a detection method based on a visual representation of the object's geometrical feature using an image sensor and a fan beam laser. By this method, temporary landmark position detection errors were less than 3% in experiments.

To improve the distance measurement accuracy of ultrasonic sonar, we proposed a method that calculates the least-square method from the received signal waveform. It was confirmed in the experiment that the distance measurement error of the proposed method is approximately 1% or smaller.

In future work, these position detection methods and the distance measurement method will be installed on mobile robots. Cooperative map creation in complicated situations using multiple mobile robots will be tried.

6 Acknowledgements

This work is supported in part by the Japan Society for the Promotion of Science Grant-in-Aid for Scientific Research (C) (No. 17500118).

7 References

[1] H. Asama, T. Fukuda, T. Arai and I. Endo (eds), *Distributed Autonomous Robotic Systems (DARS)*, Springer, Tokyo (1994).

[2] E. Yoshida and T. Arai, "Performance analysis of local communication by cooperating mobile robots", *IEICE Transactions of Communication*, Series EB, E83-B(5), pp 1048-1059 (2000).

[3] Y. Arai, T. Fujii, H. Asama, S. Suzuki, H. Kaetsu and I. Endo, "Collision avoidance in multi-robot environment based on local communication" (in Japanese), *Journal of Robotics Society Japan*, 19(1), pp 45-58 (2001).

[4] Y. Takita, Y. Hida and N. Shimoi, "Navigation and recognition of the unknown space using infrared-following servo system for moving robots" (in Japanese), *Transactions of the Japan Society of Mechanical Engineers*, Series C, 61(590), pp 209-216 (1995).

[5] M.W. Torrie, S. Veeramachaneni and B.A. Abbott, "Laser-based obstacle detection and avoidance system", *Proc. SPIE 3366, Robotic and Semi-Robotic Ground Vehicle Technology*, Orlando, FL, pp 2-7 (1998).

[6] H. Takai, G. Yasuda and K. Tachibana, "A space-division optical wireless communication system for fully distributed multiple autonomous mobile robots", *Telematics Applications in Automation and Robotics 2001 (TA2001)*, Weingarten, Germany, Pergamon Press, pp 333-338 (2001).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Development of an Autonomous Vehicle with Multiple Sensor Based Perception and Navigation System

S.S. Ahmed, S.M. Haq, Md. Manirul Islam, P.A. Raich, S.A. Polash, Mostafizur Rahman Mozumdar
Department of Computer Science,
American International University - Bangladesh, Dhaka, Bangladesh
manirul@aiub.edu

Abstract

In recent times, research and development of fully autonomous unmanned vehicle is an important area of robotics research. Such projects have multiple applications in the civilian and military domains. UAV (unmanned autonomous vehicles) research is still in its infancy in Bangladesh. Some projects with semi autonomous capabilities and remote controlled vehicles using remote video links have been developed with varying degrees of success. The objective of this project is to build a fully autonomous vehicle with vision based perception for object detection and tracking along with collision avoidance mechanism with the help of ultrasonic sonar's. The UAV built under this project can detect and chase a simple object without any human intervention. The UAV can also search for a target that has gone out of its field of vision. The project involves research and development in the fields of robotics, machine vision and electronics.

Keywords: UAV, machine vision, ultrasonic ping sensors, sobel edge detector, matlab, NeatVision

1 Introduction

UAV projects are not uncommon and a successful example would be the DARPA grand challenge winner 'Stanley' from Stanford University. Research into the field of unmanned autonomous vehicles is a complex research domain involving many subjects. For a vehicle to be fully autonomous it would require perception, cognition and control capabilities to feel, understand and manipulate its environment. It should be capable to navigate from one place to another and should be capable to take decisions on its own. We, the human beings have the natural ability in all of these. Even a small bug can outperform today's most smart robots in terms of intelligence and functionality.

To build a viable and working UAV we had to provide our vehicle with perception of its environment. We had to use sensors for this. To avoid deadlock or fatal malfunctioning we also had to provide redundancy in sensors. Our UAV uses various physical sensors to get useful information of its operational surroundings. Our project uses vision based physical sensor as its primary sensor. The primary sensor is augmented with ultrasonic ping sensor that provides collision avoidance information. A photo sensor is used to sense the illumination level in the environment. Navigation algorithms help the vehicle chase a specific target and seek a target when it is out of the field of vision of the camera. The control circuitry and logic of the vehicle issues and executes radio linked command signals to the vehicle to move it in any one of 8 directions. To keep the UAV design simple we had to assume that the target and the environment were of stark contrast. The illumination level would have to be maintained and the vehicle speed had to be optimised to allow for ample time for image processing and target acquisition.

2 Architecture

The objective of this project is to develop a fully autonomous unmanned vehicle which is capable to 'seek and chase' a ball on a given terrain without any human intervention. A UAV can have either physical or logical sensors or both. The choice of appropriate sensors is vital for such a project. We carefully analysed our objectives and the operational environment. The decision was to use physical sensors. As vision is the most prominent sensor for all living beings, we used a vision based perception system to 'see' and 'follow' the target. We had to ensure that our vehicle did not bump into walls or wander around in darkness. So we also had used ping sensors and photo sensors. Our UAV can cease its normal mode of operation if the illumination level drops below a certain acceptable limit. We have taken approach which we tend to call a 'split control' approach as the image processing and decision making tasks are delegated to a remote PC connected to the vehicle over a radio link; the collision avoidance and photometer computations are performed onboard the vehicle using a microcontroller. At this stage the two processing units are not connected any way and they perform their tasks independently.

The vehicle captures real time video using a wireless webcam and transmits the image to a PC. The PC is

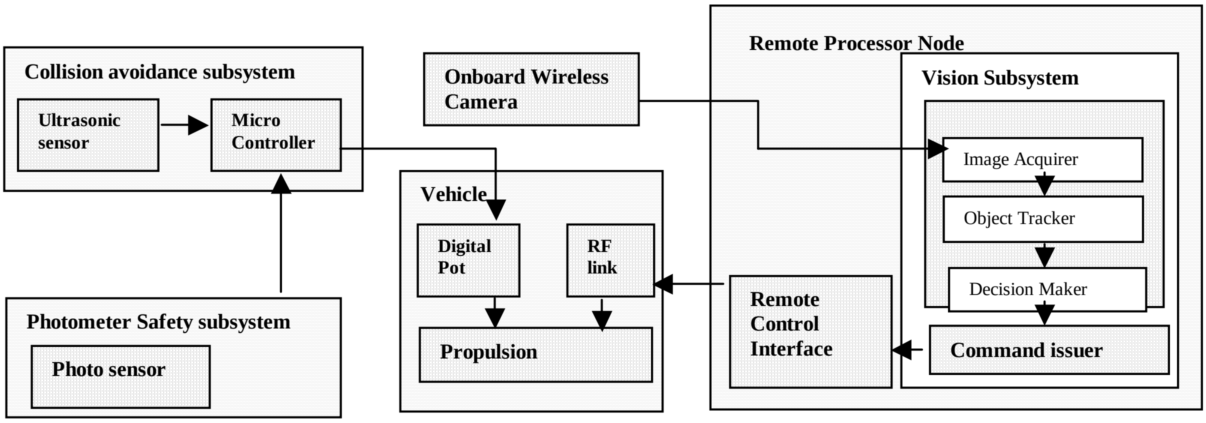


Figure 1: Block diagram of our unmanned autonomous vehicle.

responsible for acquiring the image and processing it to determine the target's position in relation to the vehicles current position. The PC then issues manoeuvring commands to the vehicle using a remote control connected to the PC via LPT1.

The collision avoidance subsystem is situated onboard and can take decisions independent of the vision based command and control subsystem. Finally the photometer is used to stop the vehicles motion when the room illumination drops below a predetermined safe level. The 'split control' approach helped us to avoid complex circuitry to send feed back telemetry to the remote processor node. In the following subsections we describe each of the sub system of the project.

2.1 Collision Avoidance Subsystem

This subsystem is situated onboard the vehicle and comprises a Parallax Basic Stamp-2 microcontroller module and a Parallax Ultrasonic Range Finder SRF04. The ultrasonic range finder transmits and receives a 40 KHz signal and can compute the distance of a target within the range of 3-300cm. The problem with ultrasonic range finders is that their accuracy depends a lot on variables such as the texture of the target, aspect, composition and temperature.

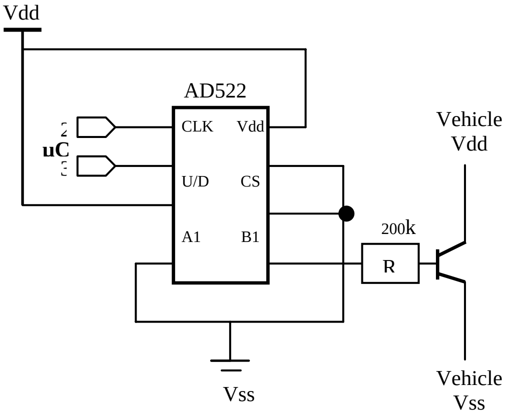


Figure 2: Collision avoidance circuit.

In our case the vehicle had to avoid collision with walls only. The range finding computations were performed with a BS2 microcontroller using pbasic code. The output from the microcontroller was fed to a digital potentiometer AD5220 which in-turn controls the electric supply to the vehicles propulsion system.

The application logic allows the microcontroller to sense for incoming walls. When one is detected the microcontroller sends PULSEOUT signals through its pin no 3 to adjust the vehicles speed by controlling the power flow to the vehicle motor. This subsystem takes decision independent of the remote processing node. Even if the vehicle is stopped because of an incoming wall, the main processing node will continue to analyse and compute incoming images and transmit command signals back to the vehicle. But it will not have any affect as the vehicle will not have any velocity. We had applied a 30 second delay before the digital pot would allow normal power supply to the motor. The vision algorithm ensures that with no change in the successive image frames the vehicle will initiate backward movements, thus removing the threat of any collision.

2.2 Photo Meter Safety Subsystem

The proper functioning of the vision subsystem is heavily dependant on the illumination level of the room. If the illumination level is too low then it is difficult to separate the target object from the background. If the illumination level is too high than the light reflection on the targets body causes the vision system to fail to detect edges properly.

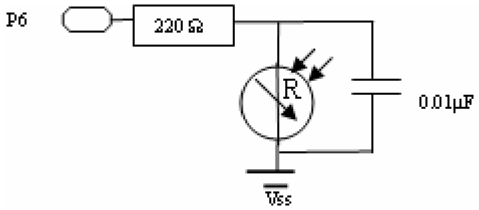


Figure 3: Photo sensor circuit.

To overcome this problem we had used a simple light sensor circuit using a photo resistor along with the microcontroller. A RCTIME function is implemented in circuitry and programme code. This was used to collect empirical data to form a range of suitable illumination values.

The application logic calculates the relative time required to drain the capacitor charge and transforms that value into a threshold limit to determine whether the vehicle is operating in a safe environment. If the limit is not within the predefined limit then the microcontroller will send command signals to the digital potentiometer to stop the vehicle.

2.3 The Vehicle

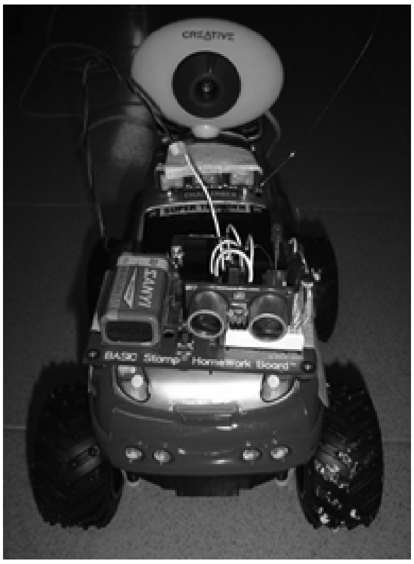


Figure 4: The vehicle.

An ordinary remote controlled vehicle is used to house the microcontroller unit, ultrasonic sensor and photo sensor. The digital potentiometer circuitry is connected to the vehicles power system to control the vehicles velocity. A Creative webcam is connected to the vehicle. This camera sends continuous video stream to the remote processing node. Command signals are sent back from PC to the vehicle. The vehicle is powered using four AA batteries. We used an ordinary Creative webcam is a CMOS camera, approximately has a 60 degree field of vision, provides continuous video stream and provides 640x480 pixels image at 30 frames per second. We chose this camera because of its cheap price and availability in the local market.

2.4 Remote Processing Node

The remote processing node comprises mainly of two subsystems:

(a) Remote control interface

(b) Vision subsystem

2.4.1 Remote Control Interface

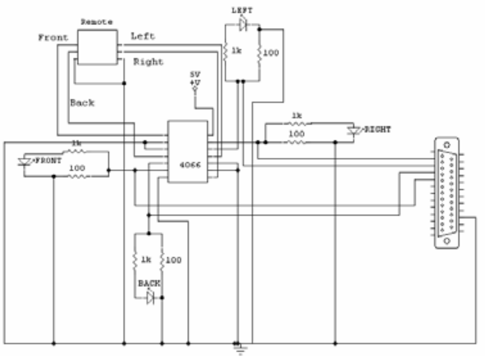


Figure 5: Basic circuit layout for remote control interface.

This unit is responsible for carrying command signals from the vision subsystem to the vehicle. The standard handheld remote controller of the vehicle was interfaced with the PC parallel port. Parallel communication requires as many wires as the number of bits in a word for its transmission. Word may be a combination of 4, 8, 16, 32 or 64 bits etc. As we allowed maximum of 8 moves, the command bits were each a 4bit word. So 4 data pins were used to transfer command data to the remote controller unit. The wire is connected from data port to a 4066IC. The 4066 is a quad bilateral switch intended for the transmission or multiplexing of analogue or digital signals. Basically this IC is used in our project for switching purposes and to decide which movement data to transmit to the vehicle. Each 4066 IC has 14 pins, four control pins, 4 pairs of switching pins, and 2 pins for power supply. Each control pin can turn one of the switch pairs on or off. Pin 14 is for the positive power supply and pin 7 for ground. The control pins are pin 5, 6, 12, 13. These pins work at "binary logic level" that means that they are either high or low. In other words if one applies +5 V to pin 13 then pin 1 and pin 2 will be shorted together, the same goes for the rest of the control pins. The data pins of computers parallel port work at exactly the same logic levels, +5 V for high, and +0 V for low. That means one can connect the data pins from the parallel port directly to the control pins of the 4066 IC. [1]. We had connected 4 LEDs to the circuit for debugging purpose.

If a data port is set to 1 then it generates +5V. So, the remote's right side signal generator pin gets a 5v. The remote device then transmits signal and the car receives the signal. In the same way computer sends right, front, and rear directions through setting corresponding data port to high. When the direction instruction is finished, the data ports need to be initialised to zero using different commands. 4066IC basically performs switching task. It makes high to the remote pin with corresponding data pin of computer's data port. To make circuit complete, connection is made with the ground port of the LPT1 port.

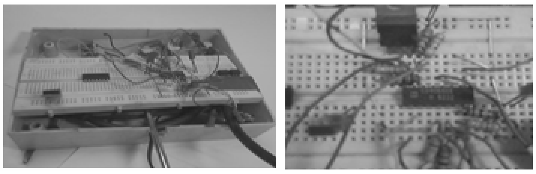


Figure 6: Car remote controller connected to LPT1.

2.4.2 Vision Subsystem

The vision subsystem is responsible for acquiring the image and interpreting it into useful data so that proper control decision can be taken. There are certain parameter variables that need to be within the required limit so that the vision system works perfectly. The ambient lighting condition should be met. As described above, the system has a built-in safeguard against poor illumination condition. The field of vision (FOV) of the camera is an important factor to consider. With wider field of vision, system gets more information to process at once. The Creative webcam provides a field of vision of approximately 60 degrees only. We have experienced problem with high relative speed of the vehicle, as the narrow FOV tends to loose the target with little movement of the vehicle or target. To offset with this problem we limit the forward motion command signal bursts to a very short duration. Before we dig dipper into the vision subsystem we shall look into the imaging software's used for this project.

2.5 Machine Vision Software

Initially we decided to use NeatVision-a java based image processing software. NeatVision is a visual programming environment for use in the design and development of machine vision algorithms and techniques. This software is developed by Dublin City University [2]. But it was difficult to find adequate tutorial materials for this tool. Moreover, we had to process image frames to obtain the centroid value of a ball and use that coordinate in the decision making part of the system. We were unable to find a way to extract this information from the NeatVision environment to be pipelined into the next phase of the program. So we chose to use Matlab-7 for our machine vision part of the project.

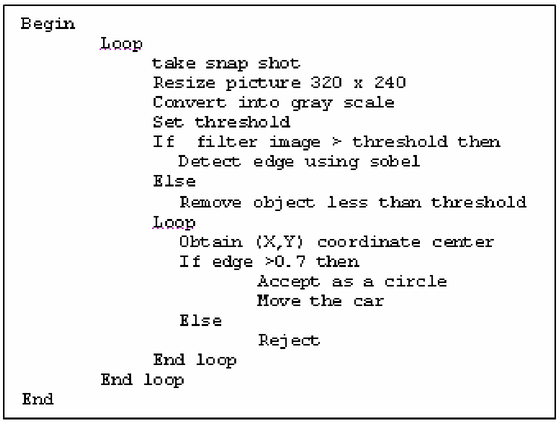
Matlab is a high-performance language for technical computing. It integrates computation, visualisation, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation [3]. Our interest in Matlab software was due to its extensive image processing library, easy to use interface, and effortless interface with Java programming language and being able to send command information from Matlab through the parallel port.

2.6 Image Acquisition Module

Image acquisition involves the capture of digital/analogue image signal and the generation of a 2 dimensional array of integers representing pixel brightness level. Initially, we used Matlab to capture image frames from the incoming video stream of an external source-in this case the web camera. But we were facing problem with the frame capture rate. We had to process approximately >10 frames per second to be able to send continuous command instructions to the vehicle. Without this rate the vehicle movement would appear to have a jerky motion. As we were trying to build our vision subsystem using the Java platform and NeatVision, we already had a small java program built using the Java media framework (JMF) for image acquisition. The program initialises the web camera and can take images at 20 frames per second. At first we create a capture device using the `CaptureDeviceInfo` class. We grabbed the image using the `FrameGrabbingControl` class. We then convert the frames into a buffered image and save them in a pre-determined folder for processing with Matlab-7 software.

2.7 Object Tracking Module

Reflection of light on the ball causes errors while filtering out the circle. This problem is reduced by prudent use of threshold values and controlling the intensity of light to an acceptable value.



```
Begin
      Loop
          take snap shot
          Resize picture 320 x 240
          Convert into gray scale
          Set threshold
          If  filter image > threshold then
            Detect edge using sobel
          Else
            Remove object less than threshold
          Loop
            Obtain (X,Y) coordinate center
            If edge >0.7 then
                  Accept as a circle
                  Move the car
            Else
                  Reject
          End loop
      End loop
End
```

Figure 7: Pseudo code for object detection and tracking.

Using Matlab we load a frame as a RGB24 image into the program and resize it into 320 x 240 pixel sizes to reduce the amount of information to process. We apply the Matlab7 function `imgfilter(A,H)` to filter the image of unwanted noise. After applying filters to remove noise from the image, the image is then converted to grey scale to further reduce the amount of data to process. We than apply a brightness threshold value to the image. Threshold is the minimum acceptable value, which in the user's judgment is necessary to satisfy the need.

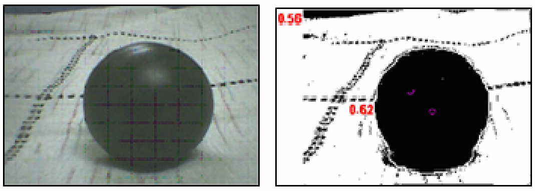


Figure 8: Centroid detection.

In our program we chose the threshold value to be 70%. So if a tracked object represents 70% of a circle we accept it to be a circle. We now remove all objects containing fewer than 30 pixels and fill in any hole so that region props can be used to estimate the area enclosed by each of the boundaries. We use Sobel edge detector to detect the circle from the image. Edge pixels can be calculated in terms of their relationship to their neighbouring pixels. An edge pixel has an associated magnitude and gradient. A Sobel edge detector uses a 3x3 mask to determine the edge gradient [4]. We set the threshold value to 0.70 to mark objects above the threshold with a black circle if metric value is greater than threshold value. The circles that failed to meet the given threshold value are filled with black so that no more irrelevant circle exists. Then we can find the circle's centre point from relevant method of Matlab. We compare the coordinate with the centre axis of the web camera and we can decide where the ball is. We use co-ordinate comparison in this case.

2.8 Decision Maker

The objective of the decision making part of the program is to keep the ball in the centre of the field of view of the camera. The program determines the difference between the centre of the ball and the frame edge. There can be two situations:

(a) The ball is in the field of view

(b) The ball is NOT in the field of view

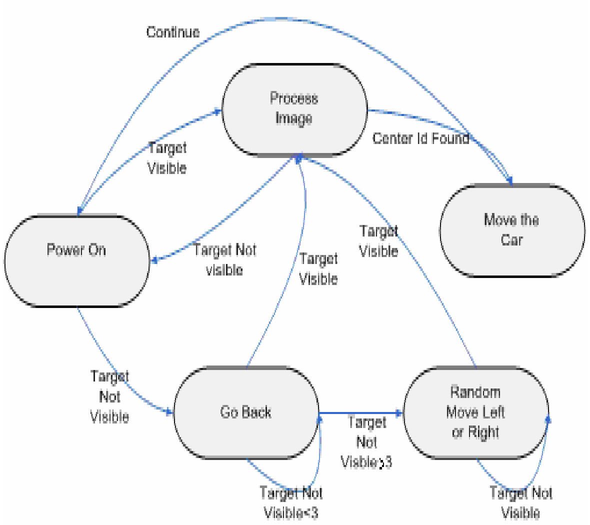


Figure 8: State diagram for decision-making.

2.8.1 Ball is in the Field of View

The web cam transmits video to the PC through USB cable. The computer takes snapshots on a given time interval. Matlab's built-in function takes snapshot from video file. Matlab then processes the snapshot with given instruction. It sends the output to the car using parallel port which is connected to the car's remote device. If the car needs to move to left side from its current position then second data pin set to 1. The wire is connected from data port to 4066IC. If a data port is set to 1 then it generates +5V. So, the remote's right side signal generator pin gets a +5v.

Table 1: Command signals.

Value	Forward	Left	Right	Backward
0001	**High**	Low	Low	Low
0010	Low	**High**	Low	Low
0011	**High**	**High**	Low	Low
0100	Low	Low	**High**	Low
0101	**High**	Low	**High**	Low
0110	Low	Low	Low	**High**
0111	Low	**High**	Low	**High**
1000	Low	Low	**High**	**High**

The remote device then transmits signal and the car receives the signal. In the same way computer sends right, front, and rear directions through setting corresponding data port to high. When the direction instruction is finished then the data ports need to be initialised to zero using a different command. We had implemented a safety system to check whether successive frames are changing over each 30 seconds window. Whenever a collision situation is detected, the digital pot shuts down the vehicles propulsion mechanism. So the view of each frame remains unchanged. If within the 30 second window the frames do not change then the 'seek' part of the program is initiated.

2.8.2 Ball is NOT in the Field of View

If the vehicle loses the ball from its FOV, it ceases its normal navigation mode and switches to 'seek' mode. In the 'seek' mode the vehicle attempts to increase its FOV by going backward. The UAV will repeat this step for 3 times and if still the ball is not within it's FOV it will randomly choose a side to move. This algorithm is repeated until the UAV catches the ball within its FOV. We had also experimented with saving the balls relative position in each successive frame. This was used to determine the proper side to move towards when the ball is lost from the FOV. We found that our camera provides a relatively narrow FOV and the target speed across the FOV is relatively faster than our frame processing speed. So, despite the later method of seeking the ball seemed to be more scientific, it was unable to provide a better result.

3 Test Results and Suggestions

The test results for our unmanned autonomous vehicle were satisfactory.

3.1 Collision Avoidance Subsystem

The collision avoidance subsystem performed extremely well and was capable of detecting obstacles (walls) almost every time and could stop the vehicles motion before a collision occurred. Some errors occurred when the aspect of the wall was not suitable for reflecting ultrasonic signals back to the car. We could offset this error by using a rotating sensor.

3.2 Photo Meter Safety Subsystem

The photo meter safety subsystem was very handy in determining the suitable illumination level for testing the UAV. But there were situations when localised illumination on the target had affected the subsystems ability to provide a suitable warning. Localised illumination on target also has the effect of making the vision subsystem unable to detect the circle.

3.3 Vision Subsystem

We had chosen the environment to be as suitable as possible for detecting the target from the background. The illumination level was carefully controlled and the environment was made to be of stark contrast to the target. The vision system was capable to perform its job satisfactorily most of the time but the 'seek' algorithm has room for improvement. Moreover the image-capture and processing rate has to be increased to achieve a smooth motion of the vehicle. We were able to achieve a satisfactory motion with processing 10-15 frames per second and the vehicle moving at approximately 0.05 meter per second. If we increased the speed of the vehicle we ran into danger of losing the target out of sight very frequently. The success rate of hitting the target increased with reducing the vehicles speed .The filtering and edge detection algorithms also has room for improvement.

4 Conclusion

We are currently working on the second phase of the UAV. This version would be able to detect a target object out of any kind of background. We shall also allow the system to be trained to identify objects of any pattern. The onboard microcontroller shall have a feedback system with the remote processing unit. This project has provided us with a strong ground to move further ahead.

A great variety of uses exist for self-driving or self-navigating vehicles. Technologies used for various subsystems, can also be used for individual projects.

Our use of machine vision techniques as the primary perception instrument will enable us to improve our system to a much higher level of perfection. Providing stereo vision capabilities to the system can eliminate the need to use physical range finding sensors.

This project has given us insight into many core aspects of this fascinating area of computer science. We hope that our future researches will help to enrich the knowledgebase of UAV development.

5 Acknowledgements

First of all we would like to thank Almighty God who gave us knowledge and strength to complete this paper. Also we would like to express our sincere appreciation and thanks to our friends, colleagues and our family for their spontaneous support and encouragement.

6 References

[1] T. Engdahl, "Get Power out of PC RS-232 Port", Helsinki University of Technology, http://www.tkk.fi/Misc/Electronics/circuits/ rspower.html, visited on 15/07/06.

[2] P.F. Whelan and D. Molloy, *Machine Vision Algorithms in Java: Techniques and Implementations* (2nd edition), Springer, London (2000).

[3] A. McAndrew, *An Introduction to Digital Image Processing with Matlab*, Thomson Course Technology, Boston (2004).

[4] Quantitative Imaging Group, "Derivative-Based Operations", http://www.ph.tn.tudelft.nl/Courses/ FIP/noframes/fip-Derivati.html, visited on 6/4/2006.

[5] Parallax Inc., "Home Page", http://www.parallax.com/, visited on 5/6/2006.

[6] Mikmo, "Details of the interface", www.mikmo.dk/cbinterfacedetails.html, visited on 5/6/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

A SpecC RTOS Model for Robot Obstacle Avoidance

John Collins and Akilesh Nukala
Engineering Research Institute,
Auckland University of Technology, Auckland, New Zealand
jcollins@aut.ac.nz, akinuk96@aut.ac.nz

Abstract

System level design languages (SLDLs) are being increasingly used to model embedded systems at high levels of abstraction. One such SLDL is SpecC. This paper shows how SpecC can be used to model a Real Time Operating System (RTOS) in a small application in which a robot avoids obstacles to reach its destination.

Keywords: SLDL, SpecC, RTOS, robot, obstacle avoidance

1 Introduction

In the embedded systems industry, design complexity continues to increase with mankind's desire for more and more sophisticated electronic devices. According to "Moore's Law", the number of transistors on a silicon chip doubles every two years. As a result, technology has developed to the extent of having millions of transistors on a chip.

Because of the growing size and complexity of embedded systems, there is pressure on software and hardware engineers to work at higher levels of abstraction during the design phase. However, engineers need to ensure that architectural decisions are consistent with the functional and non-functional (e.g. timing, power consumption) parts of the specification. This applies particularly when making decisions regarding the partitioning of an application between hardware and software, between several processor elements in hardware, or between tasks in software.

System level design languages (SLDLs) provide one means of raising the level of abstraction. SLDLs allow the hardware and software of embedded systems to be modelled at an early stage in the design process.

There are several SLDLs available in the market with SpecC [1] and SystemC [2] being among the most popular. These SLDLs are both based on the C language so they are immediately usable by most embedded systems engineers. SpecC has the additional advantage of having a well-defined methodology and refinement rules. For this reason, SpecC was chosen for this application.

The SpecC language [3, 4] and the supporting SpecC methodology [5] do not distinguish between hardware and software, allowing both to be modelled with the same language in a single model program. As a result, the SpecC methodology automatically allows for software/hardware co-design.

This paper focuses on the use of SpecC to model a real time operating system (RTOS). Models are developed for the main features of a commonly used RTOS (MicroC/OS-II).

These models are then used to construct a model of a small application in which a robot avoids obstacles to reach its destination.

2 Related Work

The current approaches for system level design include UML [6], which is primarily used for specification design in software, and VHDL and Verilog, which are primarily used for implementation of hardware design. Various system level design languages [1, 2] and methodologies have been proposed to address the issues involved in system level design. The current approaches mostly deal with synthesising the hardware part of the system.

In [7], a technique for modelling fixed priority pre-emptive multitasking systems based on concurrency and exception handling mechanisms provided by SpecC is shown. However, their model is limited in its support for inter-task communication. In [8], a high level model of a RTOS called SoCOS is introduced as a high level RTOS model supporting software generation. SoCOS requires its own simulation engine while our model in this paper is built on top of a SLDL that could be easily integrated into any system model. In [9], the RTOS was used for a mobile application. In [10], system level semantics is shown to cover the system design process for each model and they have demonstrated this using a JPEG encoder.

3 Overview of SpecC

3.1 SpecC Language

SpecC is a true super set of ANSI-C, so every C program is also a SpecC program. A SpecC program consists of a set of behaviours, channels and interface declarations. A *Behaviour* is a class consisting of a set of ports, a set of component instantiations, a set of private variables and functions, and a public main function. Through its ports, a behaviour can be connected to other behaviours or channels in order to communicate. A *channel* is a class that encapsulates communication consisting of a set of private variables and public functions or methods. An *interface* represents the link between a behaviour and a channel.

In SpecC, modelling of interrupts is provided by *try-interrupt* and *try-trap* constructs that operate similar to C++ exception handling. The interrupt service functions are modelled as separate behaviours in SpecC.

event is a SpecC data type that is used to model synchronisation. A SpecC behaviour is blocked while it waits for an event, and is unblocked when the event is notified. The *wait* and *notify* keywords implement this functionality.

One key feature of SpecC is the clear separation between communication and computation in system level descriptions used for both software and hardware. Communication is performed through channels and computation is performed in behaviours.

3.2 SpecC Methodology

The design flow of the SpecC methodology consists of four models. These are the specification model, architecture model, communication model and implementation model.

The specification model is the starting point for the SpecC design flow. This model is a single behaviour that contains the entire system functionality. The specification model describes how the system is going to respond to its inputs based on the system specification. The behaviour is purely functional without any timing or other information.

The architecture model is an intermediate model produced during architecture exploration. In hardware, architecture exploration selects a set of processing elements (PEs) and maps the computation behaviour of the specification onto the PEs. In software, architecture exploration maps the specification onto tasks for the RTOS. Hardware/software partitioning is explored and designed during this phase of the design process. The architecture model represents this mapping, thus exposing the communication between the components to be implemented by the following communication synthesis task. In the SpecC architecture model, the system is described as a set of interconnected behaviours representing processing elements and tasks. In this model, behaviours communicate directly through their ports.

The communication model is the final output of the system level design process. Communication between behaviours is modelled as channels between PEs and tasks. The communication model describes the system in terms of connections and communication protocols.

The implementation model describes hardware in terms of register transfers executed in each clock cycle for custom hardware or in terms of the instruction sequence for software.

4 RTOS in SpecC

4.1 Overview of RTOS

A task is an independent program function that handles a specific activity. Multitasking occurs when a number of tasks run in parallel.

Each task can be in one of the following states. In the *running* state, the task function code is actually being executed by the processor. Only one task can run on each processor. The *dormant* state corresponds to a task that is not available to be executed by the RTOS because it has not been started yet. A task is said to be *ready* when it is available to execute, but its priority is less than the running task. A task is in the *waiting* or *blocked* state when it requires the occurrence of an event to move it to the ready state. A task is said to be in the *interrupted* state when an interrupt has occurred and the processor is servicing the interrupt.

The MicroC/OS-II RTOS provides several techniques for inter-task communication: semaphore, message mailbox and message queue. It is also possible to use global variables for communication. The following sections describe the operation of these techniques for the MicroC/OS-II RTOS, and their implementation in SpecC.

4.2 Semaphore Channel

Semaphores are used to control access to a shared resource or as flags to allow two tasks to synchronise their activities. When used for shared resources, the semaphore is a counter that depicts the number of available resources. The semaphore is initialised with the number of available resources. If the semaphore value is greater than zero, the resource is available. A task calls the semaphore pend function to obtain a resource and calls the semaphore post function when the resource is no longer required. If the value of semaphore is positive, the pend function decrements the semaphore value and returns immediately so the calling task continues running. If the semaphore value is zero, the resource is not available and the calling task is blocked until the semaphore is posted by another task. The RTOS will then run the next highest priority task in the ready state. A task releases a semaphore by calling the semaphore post function. If a higher priority task is waiting for the semaphore then the waiting task is unblocked and the RTOS will resume execution of that task instead of returning to the calling task. If no task is waiting for the semaphore then the value of the semaphore is incremented and the calling task continues running.

When used as a flag, the semaphore is initialised to zero, and a task calls the post function to signal the flag and a task calls the pend function to wait for the flag.

The RTOS also has an accept function that allows a task to acquire the semaphore, without being blocked if the semaphore is not available.

The semaphore in SpecC is implemented as follows:

```
interface ISendSem // send interface
{
   void post(void);
};
interface IRecvSem // receive interface
{
   bool accept(void);
   void pend(void);
};
channel cSem(void) // channel definition
   implements ISendSem, IRecvSem
{
   event e;
   unsigned int n = 0;
   bool accept(void)
   {
           if (n > 0)
           {
              n--;
              return(true);
           }
           else
           {
              return(false);
           }
   }
   void pend(void)
   {
      if (n == 0)
         wait e;
      n--;
   }
   void post(void)
   {
           n++;
           notify e;
   }
};
```

The interface contains the declarations of the post, pend and accept functions, whereas the channel contains the function definitions.

4.3 Memory Management Channel

The RTOS memory manager provides a simplified memory management system avoiding the use of the C malloc and free functions because they have unpredictable timing requirements. The memory manager maintains a pool of fixed size memory blocks. Tasks call a memory get function to obtain a memory block and a memory put function to release the memory block. In SpecC, the memory manager channel is a counter whose value is the number of available memory blocks. The code is similar to a semaphore controlling access to a shared resource. The get function is similar to the semaphore pend function and the put function is similar to the semaphore post function.

4.4 Message Box Channel

A message box is used to pass a message to another task. This is different from a semaphore because data is transferred from one task to another. Usually the sending task places the message in memory obtained from the memory manager. The format of the message can be any data type. The address of the message is posted to the message box. The task that receives the message extracts the contents of the message, then frees the memory by calling the memory manager’s put function. Note that the message box can only contain one message. The accept function allows a task to check the message box without being blocked when the message box is empty. The message box channel in SpecC is implemented as follows:

```
interface ISendMbox // send interface
{
   bool send(void *message);
};
interface IRecvMbox // receive interface
{
   void accept(void *message);
   void receive(void *message);
};
channel cMbox() //channel definition
   implements ISendMbox ,IRecvMbox
{
   void *buf = 0;
   event e;
   void accept(void *message)
   {
      message=buf;
      buf = 0;
   }
   void receive(void *message)
   {
      if (buf == 0)
         wait(e);
      message=buf;
      buf = 0;
   }
   bool send(void *message)
   {
      if (buf == 0)
      {
         buf=message;
         notify(e);
         return true;
      }
      else
      {
         return false;
      }
   }
};
```

4.5 Queue Channel

A queue is also used to pass messages to another task. The difference between the message box and queue is that queue can store more than one message pointer at a time. The queue stores these messages in an array of pointers that can be read by the receiving task on a first in-first out basis.

5 SpecC Model of a Robot

This application is a SpecC model of the Hedgehog Robot constructed at Auckland University of Technology. It has a detachable CPU board, currently using an Atmel ATMega128L microcontroller.

5.1 Hardware

The robot has two independently driven wheels with a third undriven castor wheel for balance. The speed of each driven wheel is controlled by a pulse width modulation (pwm) signal. Digital outputs from the microcontroller control the wheel directions (forward or reverse).

Two independent optical wheel pulse generators (one for each driven wheel) are connected to external interrupts on the microcontroller, generating 24 pulses per wheel revolution. They are also connected to timer input capture inputs on the microcontroller so the time period between pulses can be measured.

A sonar system, mounted on the front of the robot, consists of a sonar transmitter with two ultrasonic speakers (left and right) operating at a frequency of 40Khz. The sound pulses sent from these are reflected back by objects in front of the robot and received by four ultrasonic microphones.

A serial port is available to transmit data from the robot to a PC.

5.2 Software

The application is to move a robot to a specified target position, avoiding any obstacles.

The program calculates the robot’s current position from the wheel pulses and adjusts the wheel speeds to move towards the target. If an obstacle is detected by the sonar inputs, a control algorithm is used to avoid the obstacle. When the robot has negotiated the obstacle it proceeds towards the original target position. The input capture time measurements are used in the calculation of the wheel pwm signals, because the two wheel motors do not have identical characteristics.

5.3 SpecC Testbench

The SpecC model must contain a *testbench* to simulate the external inputs of the system being modelled. The testbench consists of behaviours that perform this simulation. These behaviours use the system outputs to calculate values for the system inputs.

The sonar input behaviour contains information about the location of obstacles and uses the calculated location of the robot to periodically generate values for the sonar inputs. Similarly, the wheel pulse input behaviour uses the wheel pwm values to calculate the time between wheel pulses, and generates simulated interrupts at the appropriate times.

6 Results

6.1 Specification Model

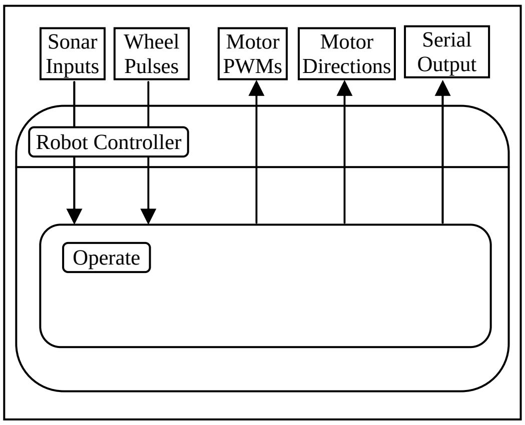


Figure 1: Specification model.

In the specification model, there is one behaviour that represents the entire robot control system. The inputs and outputs of this behaviour are the system inputs and outputs. For the hedgehog robot the inputs are the sensor values and the counter pulses from the left and right wheels, and the outputs are the motor speed and direction values and the serial output. The model also includes the testbench behaviours. A diagram of the robot controller specification model (without the testbench) is shown in figure 1.

6.2 Architecture Model

In the architecture model, the robot controller behaviour is divided into task behaviours. Each task is represented by a different individual sub-behaviour.

Values are passed between these sub-behaviours as global variables in the main system behaviour allowing communication between the task sub-behaviours.

A diagram of the architecture model is shown in figure 2.

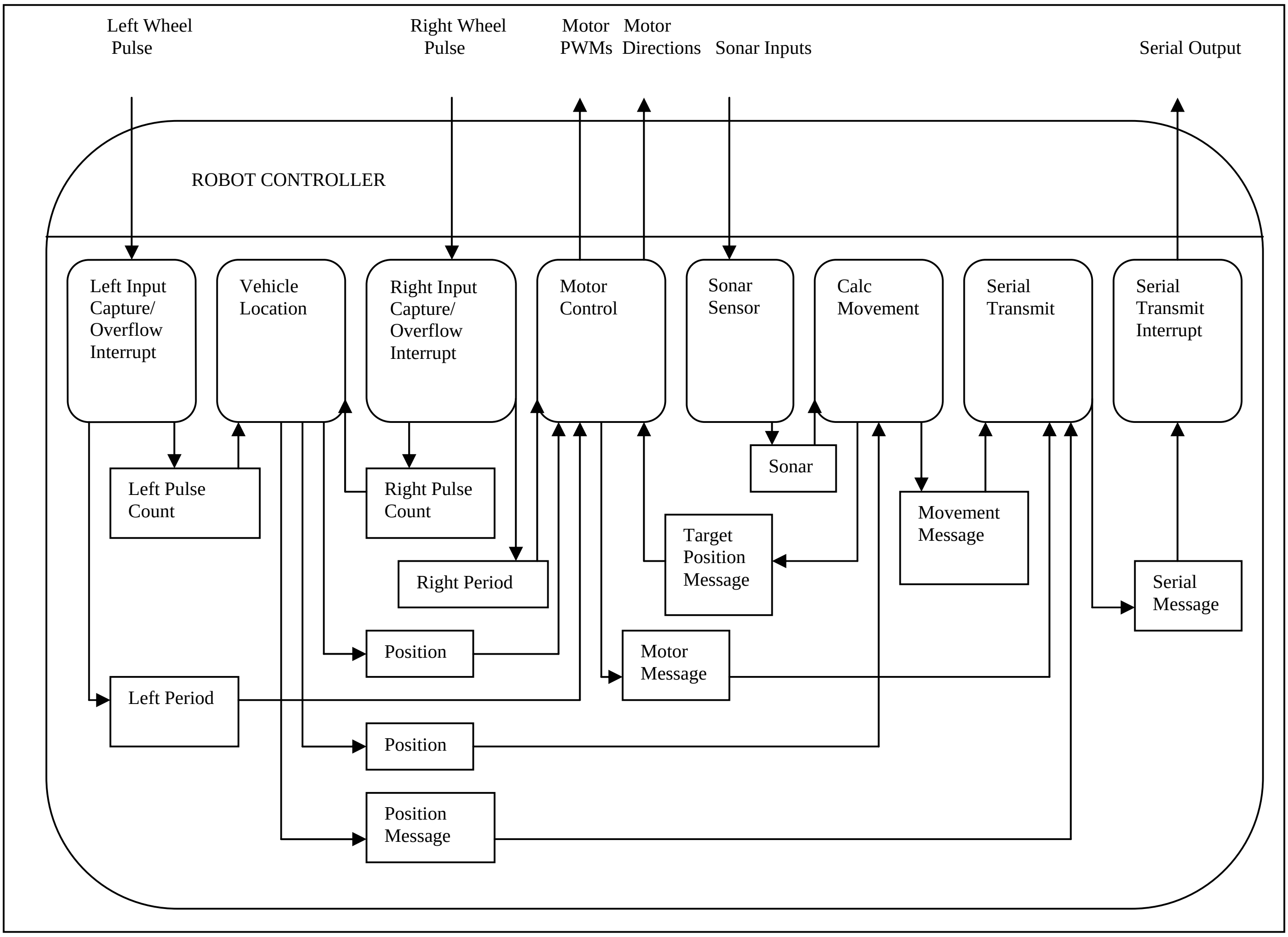


Figure 2: Architecture model.

6.3 Communication Model

In the communication model, channels perform inter-task communication replacing the global variables.

Communication between task behaviours is performed through channels representing the semaphores, message boxes, queues and memory manager of the RTOS.

6.4 Limitations of SpecC

In the try-interrupt construct, execution of the running behaviour is stopped immediately when a simulated interrupt occurs and the interrupt service function behaviour services the interrupt. After the interrupt service function behaviour has completed, control is transferred back to the originally running behaviour and execution is resumed exactly at the point at which it was interrupted. This construct does not allow for nested interrupts.

6.5 Testing the Model

The SpecC model has been tested by implementing the model in C code on the Hedgehog robot. The performance of the Hedgehog robot was then compared with the SpecC model performance.

Typical results are shown in figures 3 and 4. The robot was required to travel forward 3 metres, avoiding two obstacles as shown in the figures. The robot paths are shown in each case.

7 Conclusion and Future Work

We have shown how the SpecC SLDL can be used to model a RTOS and model a small real time application for robot obstacle avoidance. The SpecC models depict the real world application and allow the exploration of different architecture and communication methods.

Further work is required to complete the model of the MicroC/OS-II RTOS using SpecC. We have not modelled all the features of the RTOS, although most

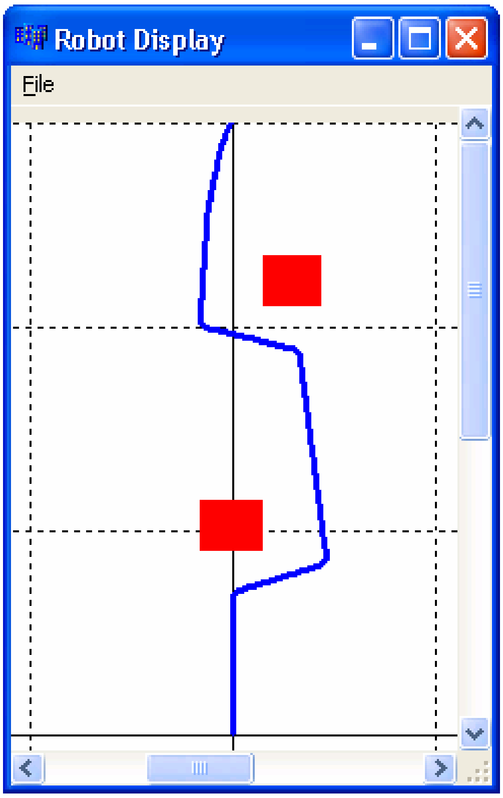


Figure 3: SpecC model performance.

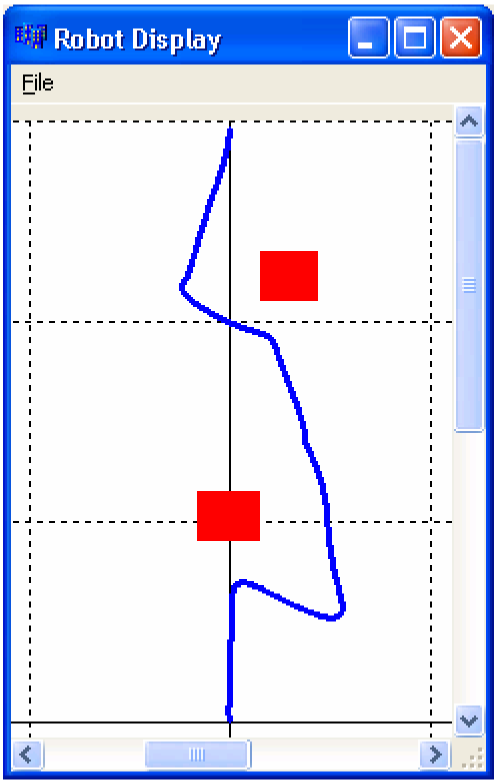


Figure 4: Actual robot performance.

could be modelled in the same style as we have presented here.

We have also not considered the important feature of timing, which would allow exploration of the time taken to run tasks and RTOS operations. An accurate timing model would allow investigation of the responsiveness of systems using the RTOS.

Differences between the SpecC model performance and the actual robot performance are caused by the dynamics of the robot that have not been included in the SpecC model. For example, at present the model assumes the motor speed changes as soon as the pwm output to the motor is changed. The model should also include some random variation in the timing of the wheel pulses as this is a significant issue in controlling the robot.

8 References

[1] SpecC Technology Open Consortium, "Home Page", http://www.specc.gr.jp/eng/index.html, visited on 31/10/2006.

[2] SystemC Community, "Home Page", http://www.systemc.org/, visited on 31/10/2006.

[3] R. Doemer, A. Gerstlauer and D. Gajski, *SpecC Language Reference Manual Version 1.0*, SpecC Technology Open Consortium, Kyoto (2001).

[4] M. Fujita and H. Nakamura, "The standard SpecC language", *Proc. 14th International Symposium on Systems Synthesis*, Montreal, pp 81-86 (2000).

[5] D. Gajski, J. Zhu, R. Domer, A. Gerstlauer and S. Zhao, *SpecC: Specification Language and Methodology*, Kluwer Academic Publishers, Norwell, MA (2000).

[6] IBM Unified Modeling Language, "UML Resource Center", http://www-306.ibm.com/software/rational/uml/, visited on 31/10/2006.

[7] H. Tomiyama, Y. Cao and K. Murakami, "Modeling fixed priority preemptive multitask systems in SpecC", *Proc. Workshop on Synthesis and System Integration of Mixed Information Technologies (SASIMI)*, Nara, Japan, pp 93-100 (2001).

[8] D. Desmet, D. Verkest and H.D. Man, "Operating system based software generation for system on chip", *Proc. 37th Design Automation Conference*, Los Angeles, pp 396-401 (2000).

[9] A. Gerstlauer, H. Yu and D.D. Gajski, "RTOS modeling for system level design", *Proc. IEEE Design Automation and Test in Europe Conference*, Munich, pp 130-135 (2003).

[10] A. Gerstlauer and D.D. Gajski, "System level abstraction semantics", *Proc. 15th International Symposium on Systems Synthesis*, Kyoto, pp 231-236 (2002).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Development of a General-purpose Expandable Arm for Small Mobile Robots

T. Hashimoto, T. Tomizawa, A. Ohya and S. Yuta
Intelligent Robot Laboratory
University of Tsukuba, Japan
hashi@roboken.cs.tsukuba.ac.jp, tomys@roboken.cs.tsukuba.ac.jp
ohya@roboken.cs.tsukuba.ac.jp, yuta@roboken.cs.tsukuba.ac.jp

Abstract

In this paper we present our developed expandable sliding arm which enables a small sized mobile robot to push and pull objects. The mobile robot arm can manipulate objects such as elevator buttons, light switches and mailbox drawers. The expandable arm has 3 degrees of freedom; a rotary degree of freedom at the base, a stretching degree of freedom and a rotary degree of freedom at the end-effector. The stretching section is composed of 4 links. The length of the contracted arm is 388mm and the expanded total length is 1218mm. Our mechanical arm is composed of linear motion guides in the stretching section, driving DC motors, an harmonic-drive as speed reducer and a solenoid to push switches and buttons at the end-effector.

Keywords: small mobile robot, expandable arm, physical services

1 Introduction

Recently, mobile robots working areas greatly expanded. For instance, home mobile robots with security and entertainment functions already entered our daily life. Small mobile robots are comparatively suitable for operation in tight spaces like home environments; this is why we believe that demand for such robots will rise in the future.

Many of actual home robots have determined functions such as taking pictures of indoor environments and transmitting them to user's mobile phones or terminals while being remotely operated. General home robots are used mostly as information service systems using communication technologies. Moreover, most systems don't target a task which involves physical interaction for modifying its surrounding environment. If robots can achieve real physical manipulation tasks, their utility value would be increased.

As concrete tasks of physical interaction we have grabbing, pushing, and pulling. Grasping is a considerable difficult task that depends in many factors such as grasping object characteristics like size, weight, shape, texture, and volume. Some research has been done about mobile manipulator grasping like in [1] and [2]. However, common grasping systems would be difficult to mount on a small mobile robot because usual devices are large scale. Operations such as pushing/pulling objects can be easily achieved as long as manipulator's position is accurately estimated. Moreover, if movable range and object loads are well defined and limited, it is possible to make a compact mechanism that can be equipped on a small sized mobile robot. Our goal of this research is the development of a general-purpose arm that can achieve the pushing/pulling operations for small mobile robots.

2 Arm Specifications

In this section we describe tasks to achieve with our developed manipulator, mobile robot platform, and general characteristics of expandable arm.

2.1 Concrete Tasks

2.1.1 Pushing Tasks

- Operating an elevator button

  Indoor navigation for small and medium mobile robots has being actively researched, with actual research, mobile robots can move freely at indoor one floor flat environments.

  If a robot could move from one floor to another using an elevator, its working area could be largely expanded. For this reason, robots which can move from one floor to another using an elevator have being developed like in [3]. In our research, a small-sized expandable arm for small sized mobile robots capable of pushing elevator buttons is proposed and described.

- Operating a light switch

  Our mobile robot is capable of operating the light switch of a wall for turning on/off the light. Checking and controlling light switches are very useful tasks for supporting human everyday life, especially for elder or handicapped people.

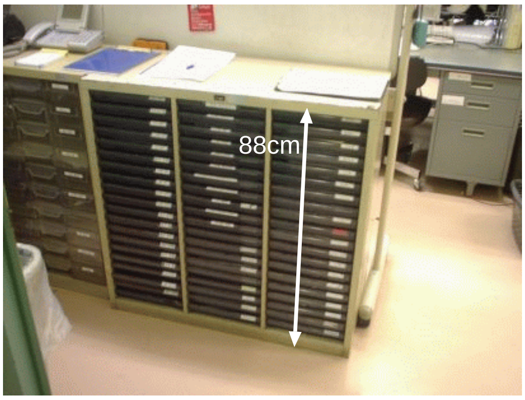


Figure 1: Mailbox.

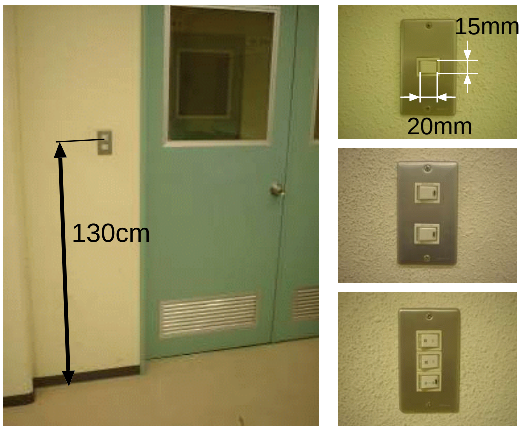


Figure 2: Light switches.

2.1.2 Pulling Task

A small robot can open a drawer by hooking and pulling it with the tip of our expandable mechanical arm. This drawer opening function could be applicable to a system which checks a mail box from a remote places.

2.2 Mobile Robot Platform

The developed arm was mounted on a mobile robot, and evaluation experiments were performed. In this research, YAMABICO is used as mobile robot platform. This robot's body size is 370mm × 322mm × 420mm (length × width × height), as it is a small sized robot, the arm must be small and lightweight.

For that purpose, structure of the hand and arm should be simplified, where heavy robotic parts must be allocated at the base of the arm. The arm was placed on robot's upper part. Moreover, the arm should be designed according to the size and power of the mobile robot platform.

2.3 The Movable Range of the Arm

In order to decide the movable range of the arm, the environments where target objects exist were investigated. First, the mail box currently used at our laboratory is shown in figure 1. The drawers of this mail box are located at a height of less than 88cm from the floor. Next, the switch panel of electric lights is shown in figure 2. Generally, light switch panels are set at a height of about 130cm from the floor and one to three switches are arranged on the panel. Figure 3 shows an example of the button of an elevator. These buttons are placed at a height of 110 to 150cm.

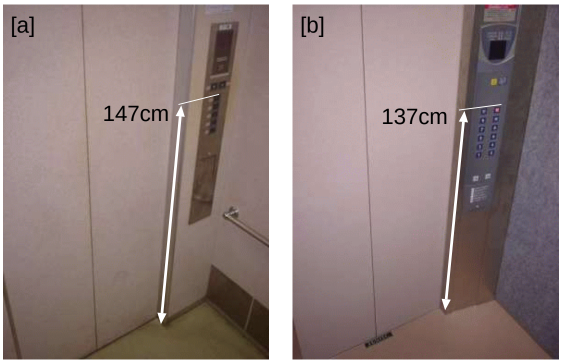


Figure 3: Elevator buttons (a) College of Engineering Systems and (b) Laboratory of Advanced Research B, University of Tsukuba.

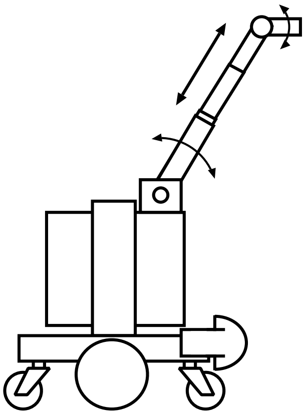


Figure 4: Image of the arm.

As mentioned above, our aim in this research is to develop an expandable arm which can operate objects which are located at a maximum height of 150cm from the floor.

2.4 Flexibility of the Arm

The YAMABICO robot platform used on our system is a non-holonomic system and has three degrees of freedom. Generally, when operating objects in three-dimensional space with a robotic hand, 6 degrees of freedom flexibility is needed. However, since only comparatively simple motions were targeted in this research, we decided that the flexibility of yaw posture angle was unnecessary. Therefore, flexibility of the arm was made only into 3 degrees of freedom

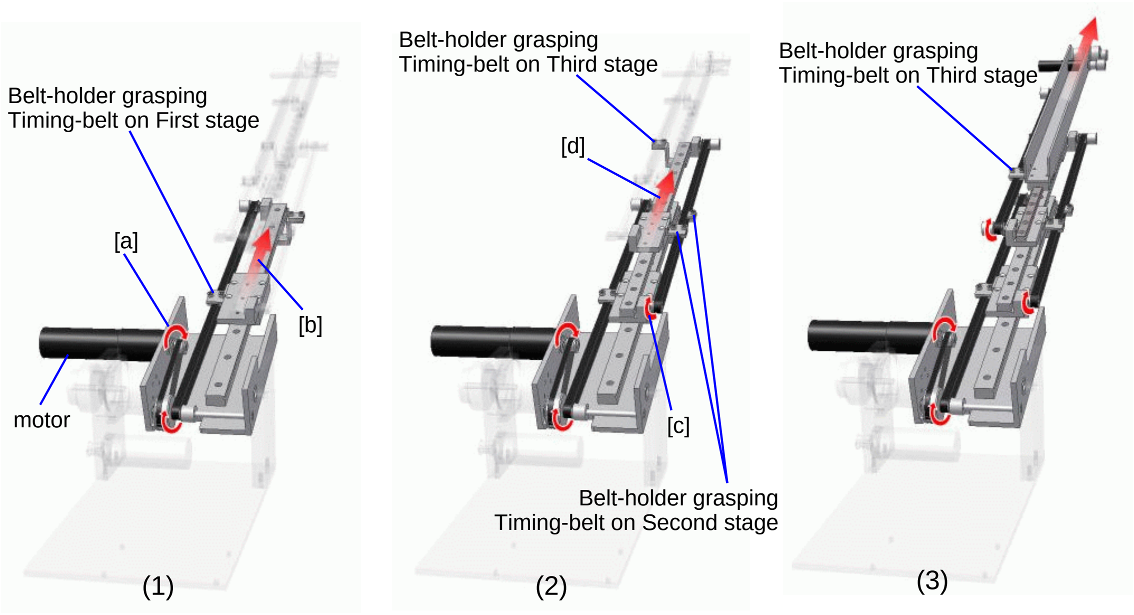


Figure 5: Structure of the stretching section.

flexibility. By these concept, the mechanism of an arm becomes simpler.

2.5 Maximum Load

The power required to operate each object was measured. The measured objects were: the button of an elevator, the switch of electric lights, and the drawer of mail boxes. By using a weight spring, we could arrive to the conclusion that about 500g power was required. [1] Therefore, the robot arm developed in this research was designed under the specification that it could output 500g power.

2.6 Structure of the Arm

Although mechanical arm must be of compact size, simultaneously it is required a large movable range to reach its surroundings. In order to obtain this functionality, we developed a sliding expandable arm which characteristics are mentioned in this section. The advantages of a flexible arm are shown below:

- Reachable range is wide.
- When an arm is not needed, it will slide in and become compact.
- Since there are few motors, arm weight is light.
- Simple calculation of inverse kinematics.

The appearance of the arm devised from the above conditions is shown in figure 4. An end-effector is connected at the tip of the arm. This device is used to push a switch, or to hook to the handle of a drawer. In addition to expansion and contraction capabilities, it has a rotational axis at the base and before end-effector, making it a 3DOF system.

2.7 Flexible Mechanism

The outline of flexible mechanism is shown in figure 5. Arm has a length of 40cm when contracted and a total length of 110cm when fully expanded. The arm consists of four slide-rails, the length of each rail is about 40cm. This flexible mechanism has only one motor as an actuator. When the motor pushes or pulls the 1st stage slider, the slider after the 2nd stage also are moved all together. The details of the mechanism which lengthens the arm are described below.

1. The belt is rotated by the motor (figure 5a), the 1st slider fixed to the belt moves along with the 1st rail (figure 5b).
2. The rail of 2nd stage is fixed on the 1st slider. The second belt is attached to the first sliding rail and after first slide rail is totally expanded, second belt starts to rotate as can seeing at figure 5c. This rotation of second belt is used to move the second sliding rail (figure 5d).
3. This mechanism is repeated with the second belt and third slide and third belt and fourth slide respectively until total extension of arm is achieved.

Since all stages move synchronously, this mechanism can be regarded as one actuator from the controller.

[1] Although the drawer was able to be pulled by about 200g force when it was empty, required power changes according to the weight of the object into which it is put in inside.

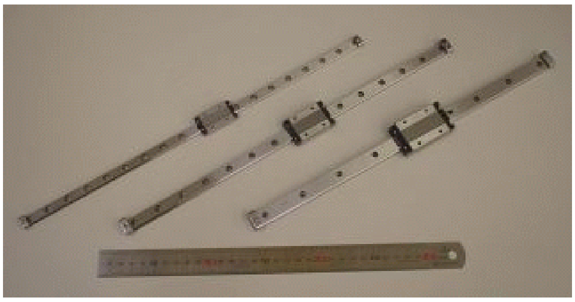


Figure 6: Linear motion guides (RSR9N, RSR12N, RSR15N).

Table 1: Specifications of linear motion guides.

	1st stage	2nd stage	3rd stage
Model number	RSR 15N	RSR 12N	RSR 9N
Moment rating	63.1Nm	28.9Nm	18.4Nm
Length of rail	350mm	370mm	375mm
Weight	417g	270g	147g

Table 2: Specifications of developed arm.

Length of contracted arm	388mm
Length of expanded arm	1218mm
Movable space of arm	225degrees
Movable space of end-effector	270degrees
Motor	3-DC motors
Max. expanding velocity	25cm/sec
Weight	2.8kg

3 Manufacture of the Arm

Here, the hardware of the developed arm according to specifications stated in Chapter 2 is described.

It is desirable that sliding mechanism has as reduced friction and back-lash as possible. For expandable arm explained in this paper, the slide rail shown in figure 6 was used. This slide rail despite being light and compact is rigid enough to support different loads with accuracy. According to load and stage each slide occupy, each rail was carefully selected. The specification of the slide rails used for each stage is shown in table 1.[2]

As a belt for power transfer, a timing belt (Made by Tsubakimoto Chain Co.) was used. A timing belt has high rigidity compared to normal belts, and fits with high precision which is adequate for position control. Moreover, a harmonic drive was adopted as the reducer of each actuator. This reducer is compact and can obtain high slowdown ratio and it has little back-lash compared with an ordinary gear.

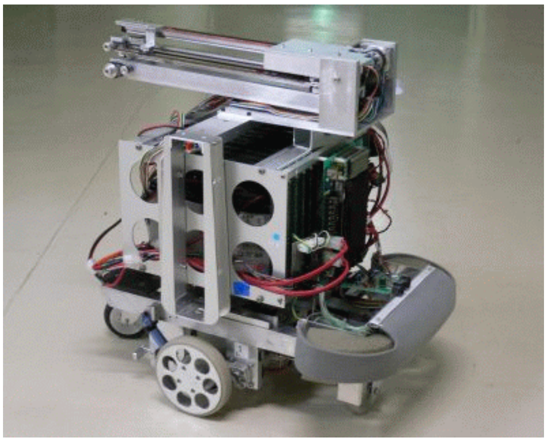


Figure 7: Contracted arm on mobile robot.

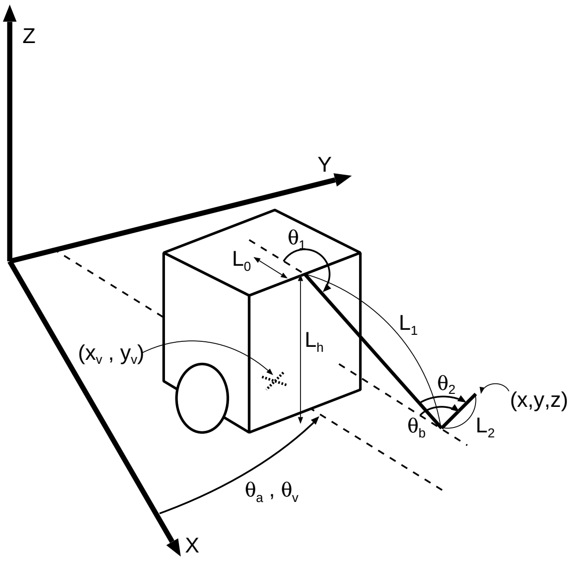


Figure 8: Mobile arm coordinate system.

The arm was built using the above parts and aluminum material. The flexible arm carried on the YAMABICO robot is shown in figure 7. And, the specification of the manufactured arm is shown in table 2.

A solenoid is placed on the tip of the arm as an end-effector and is used in order to push a switch. An end-effector can be stored in the 4th step, when not using it. The developed arm was mounted on a mobile robot with a height of 42cm. So the arm when fully extended can reach a height of 160cm.

4 Control of the Arm

The coordinates of mobile manipulator system are shown in Figure 8. Each joint angle(θ_1, θ_2 and length of the arm(L_1) are calculated with the following expressions:

$$\theta_1 = \arctan\frac{x + L_2\cos\theta_b}{z - L_2\sin\theta_b} + \frac{\pi}{2} \tag{1}$$

$$\theta_2 = \theta_b - \arctan\frac{x + L_2\cos\theta_b}{z - L_2\sin\theta_b} + \frac{\pi}{2} \tag{2}$$

[2] These slide rails are made by THK Co.,LTD.

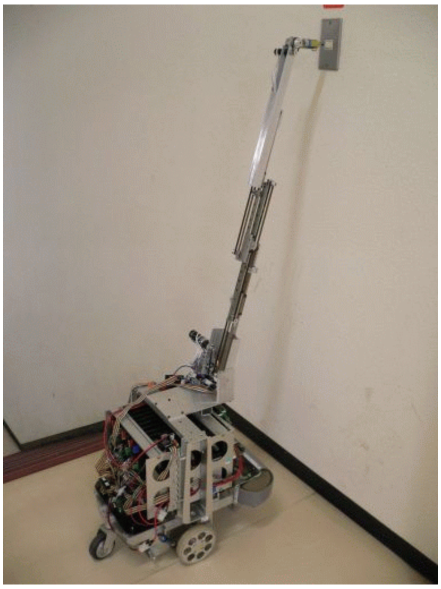


Figure 9: Pushing a light switch.

$$L_1 = \sqrt{(x+L_2\cos\theta_b)^2+(z-L_2\sin\theta_b)^2} \quad (3)$$

where, (x,z,θ_b) are position and posture of the hand.

Because of its own weight, when arm is expanded it slightly tends to slack. Therefore, controller estimates the value for slack correction. When it is programmed to reach a determined object, it will reach correct position despite arm slacking effects.

5 Experiments

Operational experiments were conducted using the previously explained expandable arm. Mobile robot has an environmental map and can move to arbitrary places by odometry autonomously. The map has not only the robot's path and obstacles but also the position and height of the buttons of an elevator or drawers.

For pushing light switches and elevator buttons, manipulator is expanded and solenoid at the tip of it performs pushing action as it can be seen in figure 9 and 10.

Figure 11 also shows a scene of pulling a drawer. Drawer opening action was realized with the arm contracted and the end-effector facing up, then mobile robot moves back pulling and opening it, and then content images can be taken after.

6 Conclusion

We have developed a general-purpose expandable arm for small mobile robots. This paper describes the design of the mechanism, and shows some examples of useful applications.

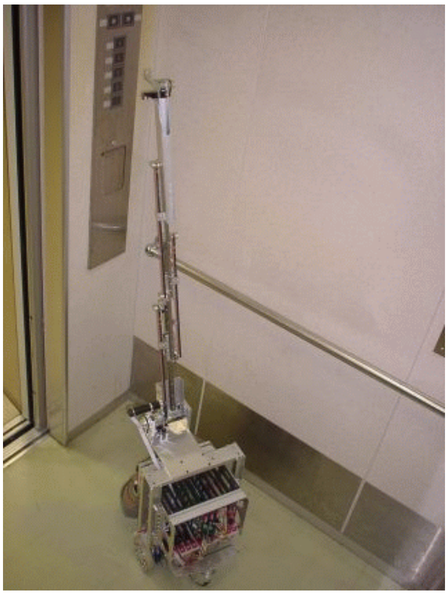


Figure 10: Pushing an elevator button.

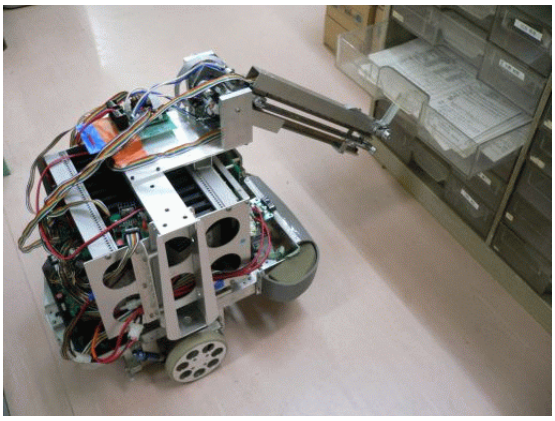


Figure 11: Pulling a mailbox drawer.

7 References

[1] T. Tomizawa, A. Ohya and S. Yuta, "Book browsing system using an autonomous mobile robot teleoperated via the internet", *Proceedings of 2002 International Conference on Robots and Systems (IROS 2002)*, Lausanne, pp1284-1289 (2002).

[2] D. Perrin, O. Masoud, C.E. Smith and N.P. Papanikolopoulos, "Unknown object grasping using statistical pressure models", *Proceedings of 2000 IEEE International Conference Robotics and Automation (ICRA 2000)*, San Francisco, pp1054-1059 (2000).

[3] K. Nagatani, S. Matsuura and Y. Tanaka, "A motion of taking elevators for autonomous mobile manipulator", *2003 JSME Conference on Robotics and Mechatronics*, Hakodate, 2A1-1F-B6 (2003).

AEDROMO: An Experimental and Didactic Environment with Mobile Robots

Humberto Ferasoli Filho[1], Renê Pegoraro[1], Marco Antonio Corbucci Caldeira[1]
and João Maurício Rosario[2]
[1]Computer Science Department, São Paulo State University at Bauru, Brazil
[2]Mechanical Project Department, University of Campinas, Brazil
ferasoli@fc.unesp.br, pegoraro@fc.unesp.br, caldeira@fc.unesp.br, rosario@fem.unicamp.br

Abstract

This work presents the design of an Experimental and Didactic Environment with Mobile Robots (AEDROMO), which is being developing in the Laboratory of Autonomous Mobile Robots (LARMA) of the Computer Science Department at São Paulo State University (UNESP)/campus of Bauru/Brazil. The objective of the AEDROMO is to provide a flexible structure of experiments for research, education or entertainment purposes. From the research point of view, it offers an interesting environment due to the multiplicity of study areas involved with the mobile robotics. On the other hand, it is also applicable as an educational support to basic courses in robotics, programming or other applications issues. This robotic environment is adaptable to solve problems in different levels, integrating educational goals with technological support. The control architecture of the AEDROMO follows the concepts of open architectures currently researched by the MANET (Manufacturing Automation Network) robotics group (www.manet.org.br). Finally, this work introduces the structure and presents results of the AEDROMO project.

Keywords: Autonomous mobile robots, robotics, control, computational vision, artificial intelligence, educational environment.

1 Introduction

Some initiatives of using robotics in education are described in literature, [1], [2], [3], [4] and [5]. Each one of them, normally, establishes a particular educational objective, focusing in mechanical design [6], [7], in programming [8], [9], in robotics itself [10] or as support tool in related subjects. In any of these applications, a goal always pursued is the low cost of the environment [2], [4] and its flexibility and modularity. The flexibility of an educational platform is establishes by the multiplicity of experiments supported in different applications.

The union of new technologies and the educational practice can offer an innovative alternative in the learning processes. Obviously, the use of the robotics in the educational context is not an original initiative; it has been already explored successfully in the past years and currently is a growing field. However, the AEDROMO is different because it presents a set of flexible modules of its open architecture, which can be customised to various applications. The AEDROMO is not a one-purpose product but an experimental environment.

2 Description of the Experimental and Didactic Environment with Mobile Robots (AEDROMO).

The AEDROMO is an experimental and didactic environment with mobile robots which in being developed by the Group of Systems Integration and Intelligent Devices (GISDI), of the Computer Science Department of the São Paulo State University (UNESP), Bauru campus. The flexibility of its use in different subjects and education levels and the low cost of the complete environment are the main objectives of this work. The experiments can be focused in research, education or entertainment. At this moment, the AEDROMO can be used in games between robots, such as a football game, or in a task to move objects from one area to other, with cooperation between the robots, or trails that the robot should follow or even to verify mathematical or physical concepts involved in the robot's movement.

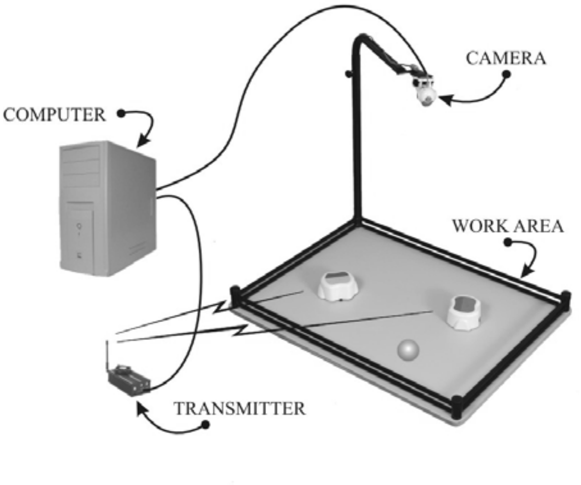


Figure 1: The AEDROMO environment.

The AEDROMO environment is shown in figure 1. Its control architecture is characterised by the closed loop (Sense - Plan - Act). The images captured by the satellite camera are sent by a cable to the computer. In the computer, the images are processed (Sense). As a result, it is established the individual positions of the objects (robots, balls, boxes) inserted in the work area. This information is used by the strategy module in accordance with the rules of the application (Plan). Then, commands are sent to the robots to act in the environment based on these rules (Act). The camera sees this movements and the process is repeated.

The AEDROMO components are:

- Vision system and work area.
- Computational system.
- Mobile Robots.
- Communication system.

2.1 Vision System and the Work Area

The Vision system defines the physical dimensions of the work area. The 4:3 ratio of the digital sensors dimensions is frequent in web cameras and for this reason it was kept to define the proportions of the work area. Thus, the dimensions of the work area was determined considering a camera with an aperture angle of 53º and located 1000 mm above the work level, resulting in a field with 800x600 mm. This height is enough so that the majority of the cameras can completely fit the work area. Figure 2 shows the dimensions of the work area.

The images captured by the web cam are processed and the information about the position of all elements in the work area is registered. Each element to be identified in the image is preset in terms of its minimum and maximum dimensions and colour (hue, saturation and intensity) ranges. Since an element is identified in the image, its information is placed in a table, which contains the coordinates of the centroid (average of all pixel positions within an element), the directions of the largest and smallest dimensions and the identification of the colour ranges. Each application uses this information in a different way. The images are captured in a resolution of 320x240 pixels and in a rate of 30 fps. The resolution of the image captured by the camera determines a pixel size of 2.5 mm in both directions of the work area, which can be considered accurate enough, and the rate of acquisition is adequate to process all environment information.

2.2 Computational System

Since the image is captured by the camera, until the movements are executed by the robot, several steps are processed by the computational system, as presented in figure 3. The strategy step is executed after the vision system analyses the image and gets the specific information about the objects. The strategy is the part of the computational system that is modified according to the application. For example, if a tennis game is executed, the strategy must locate one of the robots to intercept the trajectory of the ball "hurled" by the other robot and send it back to the opponent field. The strategy receives the information captured in the work area by the vision system and transforms it in a new position of the robot. Then the desired position of the robot is sent to the control system, which specifies the best command to be sent, by radio, to the robot.

The interesting characteristics of the AEDROMO are its modularity and open architecture of the software and the hardware. As a result, any one of the modules (vision, strategy and control system) can be modified without changing all the computational system. And more, the codes of a module can be used to implement a new module with more specific characteristics or for different purposes.

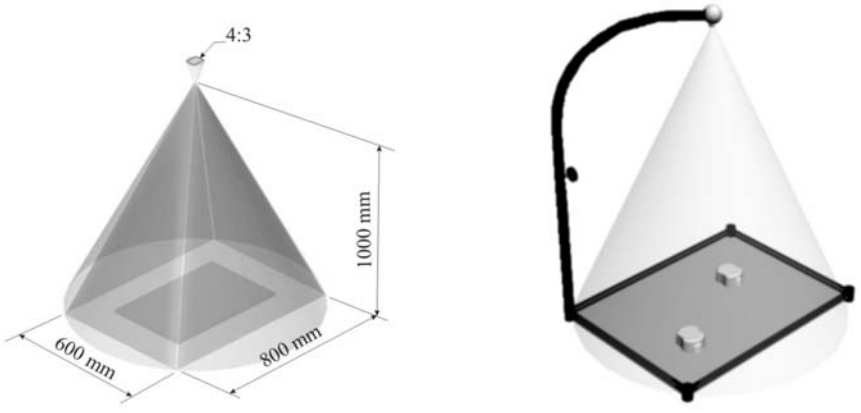


Figure 2: The AEDROMO dimensions.

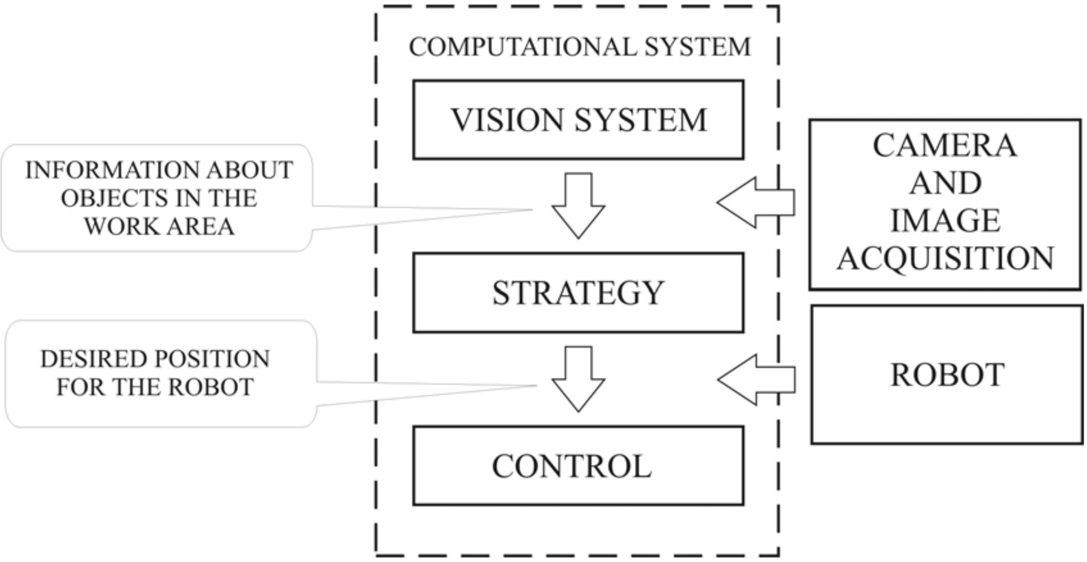

COMPUTATIONAL SYSTEM
VISION SYSTEM
INFORMATION ABOUT OBJECTS IN THE WORK AREA
STRATEGY
DESIRED POSITION FOR THE ROBOT
CONTROL
CAMERA AND IMAGE ACQUISITION
ROBOT

Figure 3: Block diagram of the AEDROMO software layers.

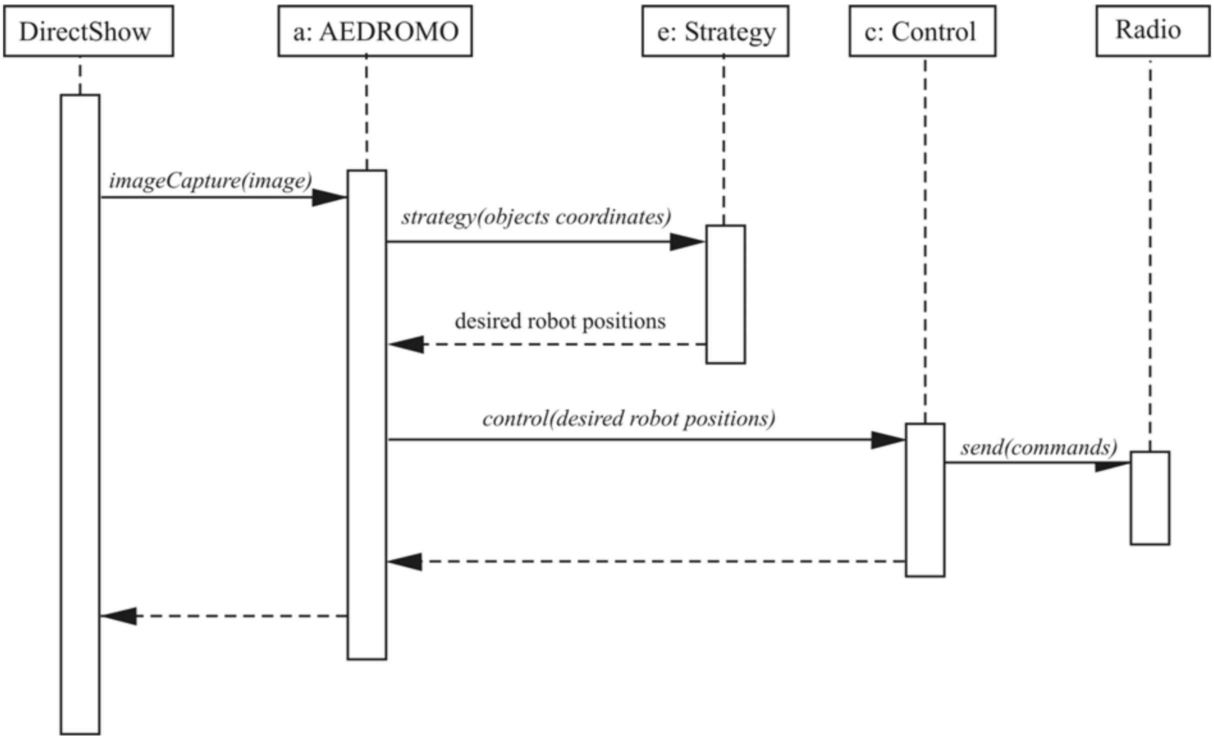


Figure 4: UML sequence diagram of the simplified function calls.

The software system is based on Windows XP and the Microsoft DirectShow carries through the images acquisition. The system was developed in C++ and uses architecture of classes and virtual functions allowing adaptation or development of just part of the system.

Figure 4 shows a UML sequence diagram representing a simplification of the function calls in the AEDROMO. The closed loop of the control architecture occurs in intervals of time of approximately 33ms and it is related to the execution of the call back function *imageCapture()* called by the DirectShow at the moment when a new digitised image is available. After the *imageCapture()*, the other layers are executed consecutively until the level of the robots control, where it finishes the loop (Sense-Plan-Act).

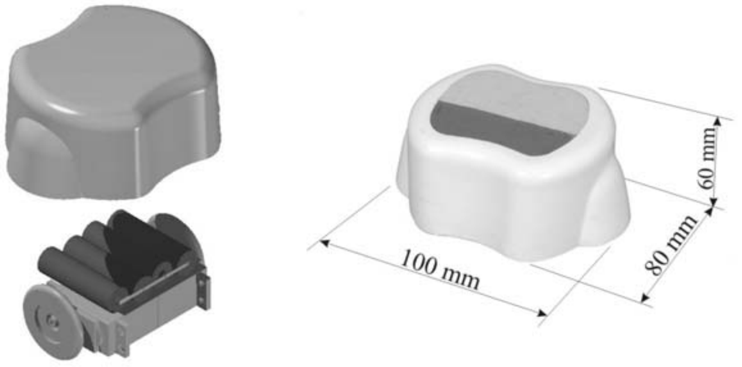


Figure 5: The mobile robot.

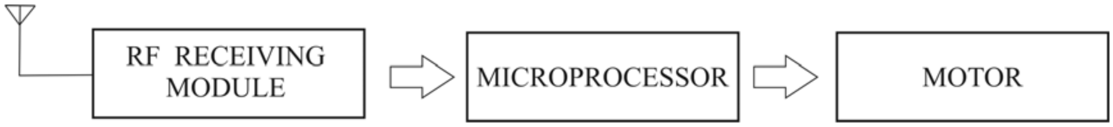


Figure 6: Diagram of the robot's onboard electronic system.

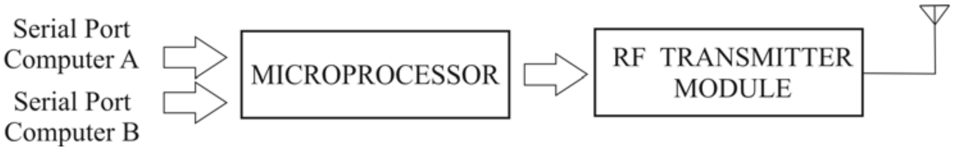


Figure 7: Diagram of the Transmission system.

2.3 Mobile Robots

The mobile robots used in the AEDROMO, also developed in the LARMA, use low cost components available in the market. The dimensions of the robots are 100x80x60 mm, as shown in figure 5. The size of the robots was specified according to the environment dimensions and to give them a low weight, that implies in a lower battery power and consumption. The motors are a Futaba servomotors or similar models. The RF receptor used is a Telecontrolli module.

The traction system used in the robots is the differential drive type, which has a motor for each traction wheel. The fact of the wheels axle is located in the middle of the base with independent control of each motor, allows a great mobility of the robots around the work area. Mobility is a basic characteristic to increase the options of use and application of the environment. The mobile robots receive information and act, in the environment, controlled by an on-board circuit as shown in the block diagram of figure 6.

The data captured by the RF (Radio Frequency) receiving module are processed by the microprocessor. The processed data is the communication protocol. The resultant information is translated in actions that will be carried through by the motors. The motors are controlled by a bridge (H-Bridge), which manages the direction of turn and the speed (PWM - pulse width modulation).

2.4 Communication System

The communication system uses a module of unidirectional and integrated transmission of Radio Frequency (RF). Furthermore, a circuit with a microprocessor is necessary to make the output signals of the computer serial port and the communication media. This circuit can receive and concentrate the signal of two different computers in the same transmission module, as shown in figure 7. The use of two computers occurs when the application implies in different strategies and/or different software modules implementations for each robot.

The RF module requires particular treatment to emit the information. The treatment is made by the microprocessor that receives the serial signals (standard RS-232), from one or two computers, and converts to the format required by the module RF. The transmission module radiates the signal converted in the microcontroller that is obtained by the receivers of each robot. Each robot has a particular address.

2.5 Applications of the AEDROMO

Computer Science students developed the first applications for the AEDROMO. These applications can be competitive, cooperative or individual tasks. In all of then, the challenge is to make the actions of the robots completely autonomous. In some situations, two computers had been used instead of only one.

1. **Collecting things:** In the environment, some balls of different colours are placed. The task is to collect balls of same colour (Things) and put then in a special area. The time to collect the balls is the objective of this work. There are two situations for this application. The first one is an individual task with one robot working in the area and the second is a competitive game, with two robots. Obviously the second situation gives much more dynamism and possibilities to the exercise.

2. **Maze:** In the work area several obstacles are placed, creating alternative routes to a special region. The objective is to reach this area returning to the origin in the least time or by the shortest way. In this case, only one robot is used. Other situation can be implemented with the two robots, where one robot has to capture the other

3. **Tennis:** the environment is divided in two halves and each robot occupies one area. A ball is then set free and the objective is to push the ball to the other side of the line, without exceeding it, in order to reach the back of the opponent's area. The adversary has to defend its field and return the ball to the other side.

4. **Dragging blocks:** In the environment, different blocks are placed. The work of the robots will be to join these blocks in a specific area. The robots can drag the blocks with greater or minor difficulty due to the block's different weight. Initiatives of cooperative works can be explored in this situation.

Since the objective is to see the strategies involved, the robots speed is about 12 cm/s and the balls are filled up to the middle with sand to prevent greater velocities.

Many other exercises can be created and implemented in this environment. Because of the modularity of the system, in each new implementation only the strategy system is modified

3 Results

The main results obtained until the moment, with Computer Sciences students in this environment are:

1. Practical aid in programming and others related subjects, which makes possible the verification of theoretical concepts;

2. Encouragement to group work and to propose challenging, competitive and pleasant exercises;

3. Stimulation of creativity in problem solving;

4. Interest in discovering the functioning of things and stimulating new interests

5. Incentive to the awareness of the students on the theoretical concepts due to the verification of the practical results.

4 Conclusions

The project can be characterised in two hierarchic parts, the hardware and the software. The hardware that includes mechanical devices, simple electric-electronic components and the electronic circuit for connection with the serial interface. The software, until the moment, consists of specific systems for the experiments and basic languages of high level. The low cost of the environment, around US$300 (computer not included), turns it in an accessible and interesting educational tool. The work continues in developing applications for the basic education, where different objectives will be accomplished. The circuits will be designed as constructive blocks corresponding to the learning level of the students. A set of basic experiments will be proposed with the objective of stimulating and challenging the student to solve new problems and verify learned concepts.

Competitions can be suggested as a motivating element for the students. The interesting point of these competitions is not only restricted to the knowledge acquired and used by the students when programming the robots, but also in the social relationship that the competitions can provide and in the occupation of the students in other periods, without the necessary teacher presence.

5 References

[1] K. Schilling, H. Roth and O.J. Rösch, "Mobile mini-robots for engineering education", *Proceedings 4th UICEE Annual Conference on Engineering Education*, Bangkok (2001).

[2] C. Cardeira and J.S. Costa, "A low cost mobile robot for engineering education", *Proceedings 32nd Annual Conference of IEEE (IECON 2005)*, Raleigh, North Carolina (2005).

[3] B. Surgenor, K. Firth and P. Wild, "Lesson learned a mobile robot based mechatronics course", *Proceedings of the 2005 American Society for Engineering Education Annual Conference and Exposition* (2005).

[4] J.B. Weinberg, "Robotics in education: low-cost platforms for teaching integrated systems", *IEEE Robotics and Automation Magazine* (2003).

[5] M. Cooper, D. Keating, W. Harwin and K. Dautenhahn, "Robots in the classroom: tools for accessible education", *Proceedings 5th European Conference for the Advancement of Assistive Technology (AAATE)*, Düsseldorf (1999).

[6] B. Potsaid and J.T. Wen, “Edubot: a reconfigurable kit for control education”, *Proceedings of the 2000 IEEE International Conference on Control Application*, Anchorage, Alaska (2000).

[7] F.T. Mrad and G. Deeb, “Extending the utility of the RHINO educational robot”, *IEEE Transactions on Education* (1997).

[8] J.S. Kay, “Teaching robotics from a computer science perspective”, *Journal of Computing Sciences in Colleges* (2003).

[9] S. Atkins, L. Hafer and P. Leung, “Robots in the laboratory”, *ACM SIGCSE Bulletin* (1988).

[10] L. Almeida, J. Azevedo, C. Cardeira, P. Costa, P. Fonseca, P. Lima, F. Ribeiro and V. Santos, “Mobile robot competitions: fostering advances in research, development and education in robotics”, *Proceedings Controlo 2000*, Guimarães, Portugal (2000).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Novel Technique of Configuring Ubiquitous Wireless Devices for Automatic Remote Surveillance

Anton G. Pereira, Liyanage C. De Silva, Amal Punchihewa
Institute of Information Sciences and Technology,
Massey University, Palmerston North, New Zealand
agpereira@theiet.org, l.desilva@massey.ac.nz, g.a.punchihewa@massey.ac.nz

Abstract

In this paper, we propose and describe the implementation details of a prototype system for remote monitoring using ubiquitous wireless mobile devices such cellular phones. With the involvement of an existing wireless communication network and a mobile device, the remote monitoring can be accomplished by configuring the mobile devices without any additional infrastructure improvements. The device can be easily deployable as a stationary or a movable surveillance device such as a robot at the convenience of the user. The proposed system captures images at a predefined rate and saves it into a database. The information is then transmitted to a remote server for further information extraction using image processing techniques. The final product has potential marketability as would assist law enforcement officers with their duty and companies providing security and surveillance services to their customers.

Keywords: J2ME, mobile communication, neural networks, surveillance, wireless devices

1 Introduction

In the 21st century, where technological advancement is steadily increasing with developments in the area of surveillance and monitoring, it is no surprise that eventually most areas around the country will be monitored to ensure the safety of the public especially in densely populated areas. Generally this sort of technology is implemented to ensure the safety of individuals and/or their private property. Within a few decades, the mobile phone, today's most ubiquitous devices has become a pervasive low cost personal item owned by most individuals from being a rare and expensive piece of technology owned by businesses decades ago. The main goal for mobile communication has always been able to keep people connected, exchange information, and communicate using Internet and entertainment at high speed data rates.

With the standard voice function of a telephone, the mobile phone can support additional services such as SMS for text messaging, MMS for multimedia data and is capable of opening HTTP connections to the Internet. With the deployment of the 3G cellular networks real-time video calling and exchange of images and audio files between individuals is possible with the significant increase in efficiency and improved performance of mobile wireless networks. With phones today integrated with cameras, cameraphones and videophones have become very common for image and video capture.

In this project it was necessary for us to employ a cost effective method to be able to configure the wireless mobile device and this was achieved using a common technology available in majority of mobile phones in the market today know as Java 2 Micro Edition (J2ME). Currently, most security and surveillance devices required some sort of hard wiring through solid structure in order to maintain communication between devices. This not only adds complexity but also tends to be costly and time consuming to the user.

What we propose to accomplish here is a ubiquitous wireless device, reconfigured to be able to capture images in the area it is placed in for surveillance. This reconfiguration is achieved through the use of J2ME technology. We initially develop a system capable of human detection. The captured images are all transmitted to a host server for further image processing to extract information regarding the person captured in the image, etc.

2 Previous Research Work

J2ME is a version of Java 2 being designed to cater for consumer electronics and embedded devices that have low resources, display and memory constraints. This technology consists of a virtual machine designed for small devices and a set of runtime APIs. The virtual machine is known as K-Virtual Machine (KVM) where the K refers to the virtual machine running on kilobyte memory as opposed to the megabytes memory of CPUs. This enables the application to run effectively on 16-bit processors [1].

The reason for the development of J2ME is simply to reduce complexity. The mobile phone industry in general lacks a dominant operating system or platform unlike that of PCs. This means that if any third party developers had the intention of developing applications on a mobile phone, they'd have had to learn right down to the finite detail of each phone's platform before developing code to operate on it. With the release of J2ME specification in September 2000, software developers were now able to write and run code independent of the mobile device itself. Being a subset of Java, J2ME ensures lower development costs as code written once can be deployed across most J2ME enabled phones.

Zarka et al. have developed a server application on a PC using Java 2 and a mobile phone being the client with the application designed using J2ME [2]. This system is designed to enable a doctor to remotely access the profile of a patient by sending a request to a central mobile phone connected to the PC and the information will then be downloaded onto the doctor's wireless device where updates can be done in regard to the patient's health status and also previous records of the patient can be retrieved. The information is relayed back to the central mobile device at the discretion of the doctor where the information is then stored or updated in the central database.

In paper [3], Watanabe et al. have devised a system that captures an image of Japanese text in a scene and translates it into English. The authors had developed the application on the mobile phone that enables the phone to manually capture an image. The image is then again transferred manually via Internet to a remote server for image processing involving edge detection, character recognition and translation to English. The resulting information will then be transmitted to the mobile phone user via the wireless Internet or mobile phone network.

Divya et al. have developed a vector graphic based digital map system that can be displayed on J2ME enabled devices. The digital map information is encoded in the vector graphics file format utilising GPS co-ordinates as references and is stored in a database on the Map Server. The information can later be retrieved by acquiring the relevant section of the map, translating the detail into XML format and then transferring the information to the mobile device to be viewed by the user.

In paper [5], Jones details a prototype system of how users can retrieve information regarding flight booking. The system enables users place a booking for available flight seats and then this information is transmitted via HTTP connection to the host servers where confirmation of the flight booking is made. The system then sends an SMS to the user to confirm the bookings.

These papers related to the wireless device being configured to manipulate the on board image capturing device and display some technique of remote processing accomplished. It is intended that this system be adopted for traffic monitoring involving number plate recognition or surveillance at a park where face recognition techniques are involved. In this paper, we'll discuss the latter implementation at the remote processing server.

3 Design Approach

In this section two design aspects of the system is discussed. The first being the wireless application ported on the mobile phone and the second is in regard to remote processing server. The graphical representation of the general system operation can be seen in figure 1. When the application is executed on the mobile phone installed with the MIDlet, images are automatically captured. The application will open a socket connection to the web server via HTTP which will then established a connection between the mobile phone and the mobile phone service provider's base station.

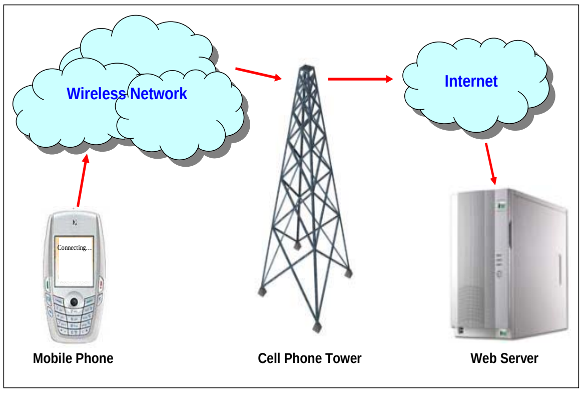


Figure 1: Graphical representation of system.

The image will be transmitted as packet switch high speed mobile data, (GPRS) to the cell base station. With this methodology in application, faster data rates can be achieved at a cheaper rate. The information is then passed on through the internet to the web server where the image is stored and further processing of the image is carried out. A sample application is programmed to run on the mobile phone for image capture and transmission. Concurrent to this is the remote processing server running with an application implemented to accept connections from a sending source to store incoming images into the database and scripts to invoke the necessary processing tools to extract information from the image. This system's operational block diagram is shown in Figure 2.

3.1 Wireless Mobile Application Design

The application, also known as a MIDlet, when initially executed enables the user to either edit settings for the camera or to begin the surveillance with the camera. If the settings are not edited prior to execution, default settings are applied. In the settings menu, the application is devised to take two parameters being the initial time delay prior to surveillance period and the image capture frequency both measured in seconds. The time delay is implemented to allocate time for the user to place the mobile device within the premises that is being monitored and vacate after. When the settings are adjusted, it would replace the default settings and save into the Record Store Management (RMS) for the application in the current session and for sessions thereafter.

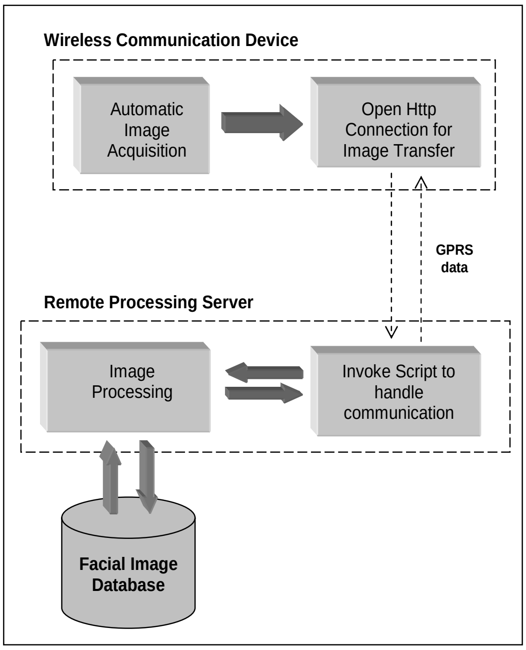

Wireless Communication Device
Automatic Image Acquisition
Open Http Connection for Image Transfer
GPRS data
Remote Processing Server
Image Processing
Invoke Script to handle communication
Facial Image Database

Figure 2: System block diagram.

Record stores in wireless devices are small databases that contain small amounts of data called records. Each MIDlet that is created is able to have a small portion of phone memory allocated for saving data and this data can be accessed by the MIDlet. This storage element can be a device such as a battery backed RAM or flash ROM.

The following stage involves the on board camera utilisation. In J2ME, in order to manipulate the hardware, the MMAPI, defined by JSR 135, is necessary. MMAPI provides a generic but flexible foundation for multimedia processing for devices with advanced sound and multimedia capabilities. This API enables MIDlets to play and record audio and video files. For notations and definitions not explicitly mentioned here, the reader is referred to [7].

Our algorithm creates the player and waits in the unrealised stage to receive information from the manager regarding the data it is going to handle. When the required resources are free to be handled, in this case being the input video stream from the physical camera viewfinder, the player enters the realised stage to begin receiving the video stream for images to be captured.

The final step of the automatic image acquisition stage is to save the capture images into the record store as we had done previously with the user preferences to avoid any loss of information. The timestamp is also saved into the record store. Effectively here, the record store acts as a buffer, while awaiting resources to be free for http data transfer. For successful operation of the code, threads are implemented to avoid potential deadlock. The deadlock could likely occur at the command action handler method. This would tend to slow processes down.

For the communication of information to the server, a new thread will start once an image is saved in the RMS by opening the socket in order to transmit the image bytes as GPRS data. For this we use the HTTP class and send the data using the Output Stream class. Once the file has been transmitted a flag is set to true indicating that the file has been transmitted and is now scheduled for deletion. While the application is still running the next available file is read and transmitted again. The programme flow chart for this implementation is shown in Figure 3.

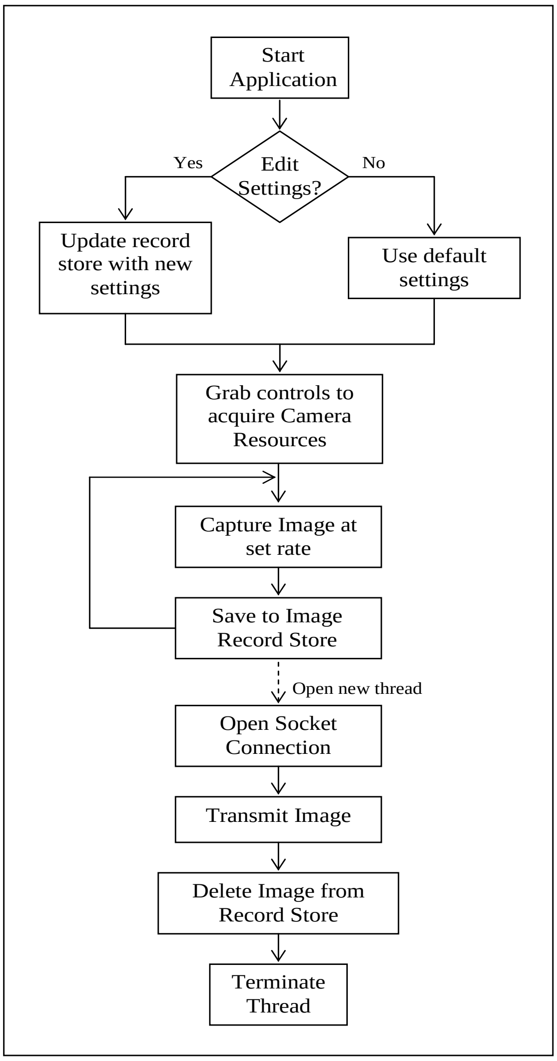

Start Application
Edit Settings?
Yes
No
Update record store with new settings
Use default settings
Grab controls to acquire Camera Resources
Capture Image at set rate
Save to Image Record Store
Open new thread
Open Socket Connection
Transmit Image
Delete Image from Record Store
Terminate Thread

Figure 3: Wireless mobile application flowchart.

3.2 Remote Server Design

An XML script was designed to receive connections from the application for image transfer to the server for processing. Data is stored using Input Connection. Once the image file bytes are received and stored in the server database, the script will invoke a resident programme to perform image analysis.

Figure 4 shows the two camera phones to be communicating information images of two separate locations to the database where the images are stored and processed further at the server.

4 Implementation and Results

J2ME applications were initially developed and tested on the Sun J2ME Development Kit. Eventually the deployable JAR files were tested on a physical Sony Ericsson's K750i model.

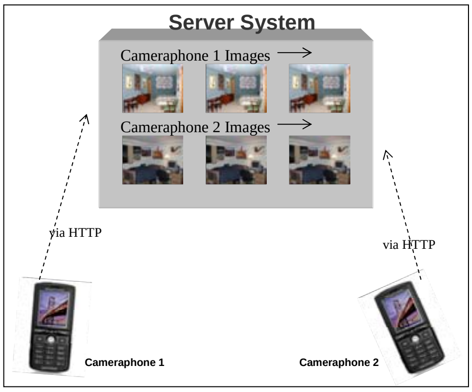


Figure 4: Server system implementation.

The K750i phone has a Java-enabled CPU of about 29.1MHz with memory read and write operations at 17463 kb/s and 16697 kb/s respectively. Since the record store is used often in our algorithm, timing operations of processes involving are noted. The K750i has a flash memory available for the RMS at 1502 kb with speeds for memory operations as shown in Table 1.

Table 1: RMS speed operation to memory.

Operation on RMS	Time taken (ms)
Create	62
Add	345
Set	388
Get	81
Delete	421
Close	71
Destroy	314

Figure 5: K750i and its (inset) 2.0 mega pixel camera located on the back plane.

In the algorithm, the images are encoded as jpeg with the options for image sizes; 160×120, 640×480, 1280×960 and 1632×1224. Images that are captured by the MIDlet are of the 160×120 size. Once this image is transmitted to the server, the server assesses whether there is an object of interest within its field of view using multi scale algorithm to analyse low resolution images. If there is object detected, the XML script will transmit a flag to the phone to set a flag in the J2ME algorithm to then capture the consecutive images with a size of 1280×960 for feature extraction. When the image is captured with a face to be recognised, the image is then sent to a neural network to identify the individual. The server application is still a work in progress.

Figure 6: Wireless device positioned for surveillance.

The MIDlet named 'Survcamm' when initially installed on the wireless device requires permission for it to be able to carry out certain operations autonomously namely for internet access, messaging, multimedia and reading and writing user data.

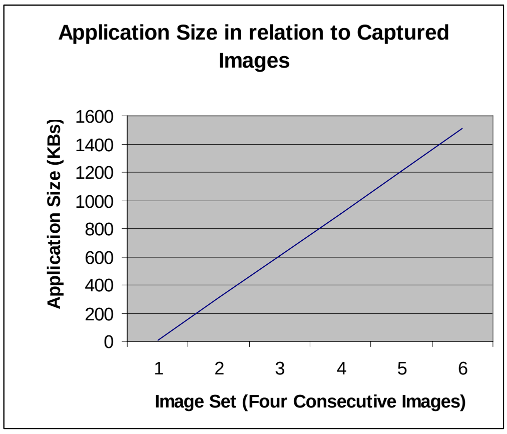


Figure 7: Relationship between image size and application properties.

When images are captured, it is noticed from the properties how the size of the application increases linearly, as noted in Figure 7.

5 Discussion

In this paper, the smart surveillance system using mobile device has been presented. Ideally it would be preferable that all complex calculations be performed on the mobile device including the image processing in regard to face recognition but due to the wireless device processing limitations this issue is not trivial, hence our solution to combine the J2ME application with the web server to handle most of the complexity in the project.

With most security and surveillance devices today requiring some sort of hard wiring through the solid structure or even creating a bigger mess by running all over the place in order to maintain communication between devices such as sensors and an alarm has become common. This product design is bound to reduce complexity of the infrastructure installation and it is be cheaper and less time consuming to the user.

The final product will become useful to organisations providing better services for their clients or to the public at large. There is much potential in the end product and it is attractive as we propose that the service be brought to one of the most ubiquitous products being the mobile phone. Deployment of the final system will be at the ease of the customer as we have devised a new and useful function for the mobile device.

6 Acknowledgements

The authors would like to thank the Mary Earle Product Development Committee for providing funds and resources for the initial research and development of this work.

7 References

[1] J. Keogh, *J2ME: The Complete Reference*, McGraw Hill (2003).

[2] N. Zarka, M.F. Hinnawi, A. Dardari and M.A. Tayyan, "'Patient Keeper' medical application on mobile phone", *Proceedings of the International Conference on Information and Communication Technologies: From Theory to Applications, 2004*, Thailand, pp 37-38 (2004).

[3] Y. Watanabe, K. Sono, K. Yokomizo and Y. Okada, "Translation camera on mobile phone", *Proceedings International Conference on Multimedia and Expo (ICME '03)*, 2, pp 177-80 (2003).

[4] M. Divya, T.A. Phan and N. Vun, "A J2ME based digital map system," *Proceedings of the 2004 IEEE International Symposium on Consumer Electronics*, pp 583-586 (2004).

[5] S. Jones, "Next generation airline information: using consumer devices to keep passengers informed and as an additional channel to market", *Proceedings of the 2004 IEEE International Conference on Mobile Data Management*, pp 94-99 (2004).

[6] J. Knudsen, *Wireless Java: Developing with J2ME* (2nd edition), Apress (2003).

[7] Java Community Process, "JSR 135: Mobile Media API", http://jcp.org/jsr/detail/135.jsp.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Feasibility of Velocity Estimation for All Terrain Ground Vehicles using an Optical Flow Algorithm

Savan Chhaniyara, Pished Bunnun, Yahya H. Zweiri, Lakmal D. Seneviratne and Kaspar Althoefer
Department of Mechanical Engineering,
King's College London, UK
savan.chhaniyara@kcl.ac.uk, yahya.zweiri@kcl.ac.uk, lakmal.seneviratne@kcl.ac.uk,
k.althoefer@kcl.ac.uk

Abstract

This paper presents a novel velocity estimation method for all terrain ground vehicles. The technique is based on a camera that scans the ground and estimates the velocity by using an optical flow algorithm. The method is tested and validated for different types of soil such as fine sand, coarse sand, gravel and mixed coarse sand and gravel. Measured velocities are compared with the velocities predicted by the optical flow algorithm, showing promising potential applications to all terrain vehicles, including wheeled/tracked and manned/unmanned vehicles.

Keywords: optical flow, motion estimation, slip parameters, visual odometry

1 Introduction

Slip parameters are very important to achieve accurate localisation and improve navigation of ground vehicles. One of the important parameter for ground vehicle slip estimation is the vehicle velocity [1]. In this paper, a novel technique to estimate ground vehicle velocity is presented.

Over the past decade, several researchers have made significant advances in utilising various vision and image processing techniques for navigation, obstacle avoidance and control of ground vehicles. Most of the researchers use image processing techniques for generating 3D space from 2D images, detecting obstacles or extracting motion from the image sequence [2]-[7]. Generally, the vehicle motion is determined by comparing features, lines, or textures between the current and previous images in a video stream. However, in order to achieve position control, velocity signals have to be integrated, thereby resulting in a dead-reckoning error which grows rapidly as a function of time [8].

Researchers are presently investigating a number of navigation strategies based on technologies such as GPS, inertial sensors, RF beacons, laser positioning systems, and optical navigation system [9]-[17]. These techniques have some advantages and disadvantages over others, but still there is no commercially proven navigation module available in the market. GPS is one widely used technique for commercial and military vehicles, but suffers from precision which is in meters at best. GPS has also limited use for indoor applications and it is not a standalone system. Inertial sensors can work stand alone but are affected by integration errors [9]-[11]. In [10]-[11], laser positioning is used for obstacle avoidance rather than localisation purposes. [12]-[15] used a vision systems to detect obstacles and to control the steering position of ground vehicles.

In this paper, a camera system similar to that in [16, 17] is used. In [16], a mobile robot tracks its position by matching a current camera image with an existing map while moving over a flat ground. The mobile robot generates an image map around the moving area and stores it in the memory, prior to use. In [17], research on visual odometry for an outdoor mobile robot is proposed. Visual odometry is able to estimate both translational and rotational velocities based on a pseudo-optical flow technique. The system is tested with various terrains such as grass, gravel, concrete, tarmac and dirt, but a flat ground is assumed. Based on simulation results, up to 300 degrees per second for rotation and 350 millimetres per second for translation can be estimated.

In this paper, a vehicle velocity estimation algorithm is presented while the vehicle moves on a rough ground. The advantage of this technique is the estimation of vehicle velocity as an instantaneous value which does not suffer from accumulative errors.

In the next Section, the camera velocity estimation algorithm is presented. Experimental results are presented in Section 3 and conclusions are shown in Section 4.

2 Optical Flow and Velocity Estimation Algorithms

There are two phases for estimating the vehicle velocity. In the first phase, the optical flow algorithm is used to estimate an optical flow from an image pair. The optical flow is then used to estimate the vehicle velocity.

2.1 Optical Flow Algorithm

Horn [18] did pioneering work on the development of optical flow techniques based on computing spatiotemporal differences from an image sequence. Since then, many methods and algorithms for determining optical flow were developed [19].

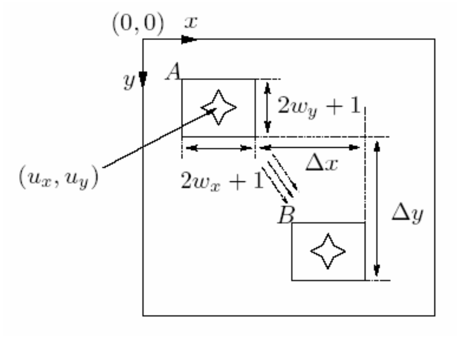


Figure 1: Optical flow.

According to the result of Barron et al. [19], Kanade-Lucas optical flow algorithm [20] is used because it is robust, accurate, insensitive to noise and non-uniform light intensity sources, and suitable for real-time computation.

In this method, let I_i be the greyscale image at time t_i and I_{i+1} be the greyscale image at time t_{i+1}. During this time interval, let the image be translated by $d=(\Delta x,\Delta y)$. A Let be a feature window in I_i and B be the same feature window in I_{i+1}. The objective is to find d by minimising a residual function $\varepsilon(d)$ in (1) :

$$\varepsilon(d)=\iint_w (I_i(p)-I_{i+1}(p+d))^2\,dxdy \qquad (1)$$

where

1. $P=(x,y)$ is the pixel coordinates of a generic image point. The upper left corner pixel coordinate is and the lower right corner pixel coordinate is (n_x-1,n_y-1) , where n_x and n_y are the width and height of the image respectively.
2. I_i and I_{i+1} are the greyscale values of the first image and the second image respectively.
3. W is the feature window area, of size equal to $(2w_x+1,2w_y+1)$.
4. $d=(\Delta x,\Delta y)$ is the optical flow.

In practice, the solution of minimising (1) can be obtained by using an iterative algorithm like the Newton-Raphson method. The algorithm is implemented as follows [21]:

1. Select a good feature window to track in the image $I_i(p)$.
2. Let $u=(ux,uy)$ be the centre of the feature window in the image $I_i(p)$, figure 1.
3. Window width and length, w_x and w_y , are user defined, with typical values of 2 to 7 pixels.
4. Let $d=(\Delta x,\Delta y)$, where $(\Delta x,\Delta y)$ is an initial guess.
5. Calculate

$$G=\sum_{x=u_x-w_x}^{u_x+w_x}\sum_{u_y-w_y}^{u_y+w_y} g(p)g^T(p) \qquad (2)$$

$$b=\sum_{x=u_x-w_x}^{u_x+w_x}\sum_{u_y-w_y}^{u_y+w_y}\left(I_i(p)-I_{i-1}(p+d)\right)g(p) \qquad (3)$$

where

$$g(p)=\begin{bmatrix}\frac{\partial}{\partial x}(I_i(p)+I_{I+1}(p+d))\\ \frac{\partial}{\partial y}(I_i(p)+I_{i+1}(p+d))\end{bmatrix} \qquad (4)$$

6. Solve linear equation $Gd_\delta=b$.
7. Update $d=d+d_\delta$.
8. Calculate

$$\varepsilon(d)=\sum_{x=u_x-w_x}^{u_x+w_x}\sum_{y=u_y-w_y}^{u_y+w_y}\left(I_{i+1}(p+p)-I_i(p)\right)^2 \qquad (5)$$

9. If $\varepsilon(d)<\varepsilon$, where ε is a predefined threshold, go to 5.
10. The optical flow is d .

2.2 Camera Model

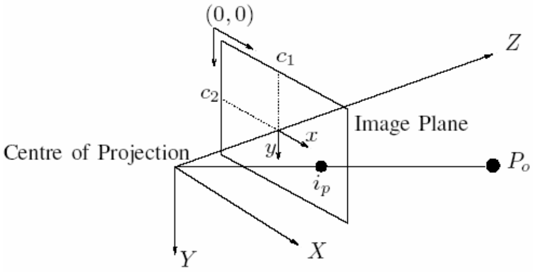


Figure 2: The standard coordinate system of a camera.

A camera model giving the relationship between an object position and its pixel position is presented. There are two groups of parameters. The first group, called extrinsic parameters, are not constant and depend on the camera orientation. The second group, called intrinsic or internal parameters, are independent of the camera orientation.

Intrinsic parameters are the important parameters since estimates of the camera velocity or vehicle velocity is needed while the camera orientation is fixed as is the case in this paper. The intrinsic parameters are determined by using the method described in [22]. Figure 2 shows the standard coordinate system of a camera.

The intrinsic parameters are listed below.

1. Focal length: $f_c = \left[f_1 \quad f_2 \right]'$
2. Principal point: $c_c = \left[c_1 \quad c_2 \right]'$
3. Skew coefficient: α_c
4. Distortion: $k_c = \left[k_1 \quad k_2 \quad k_3 \quad k_4 \quad k_5 \right]'$

Suppose $P_o(X_c, Y_c, Z_c)$ is the coordinate of P_o in the camera reference frame. The point is projected onto an image plane normal to the Z or optical axis. The normalised (pinhole) image projection $i_n(x_n, y_n)$ is defined by (6). From (6), after including lens distortion k_c the new normalised point $i_d(x_d, y_d)$ is defined as [22]:

$$i_n(x_n, y_n) = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} \tag{6}$$

$$i_d = \begin{bmatrix} x_{d1} \\ x_{d2} \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) i_n + t_d \tag{7}$$

where t_d is the tangential distortion vector

$$t_d = \begin{bmatrix} 2k_3 x_n y_n + k_4 (r^2 + 2x_n^2) \\ k_3 (r^2 + 2y_n^2) + 2k_4 x_n y_n \end{bmatrix} \tag{8}$$

and $r^2 = x_n^2 + y_n^2$. Finally, the projected point $i_p(x_p, y_p)$ of $P_o(X_c, Y_c, Z_c)$ on the image plane is

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_1 (x_{d1} + \alpha_c x_{d2}) + c_1 \\ f_2 x_{d2} + c_2 \end{bmatrix} \tag{9}$$

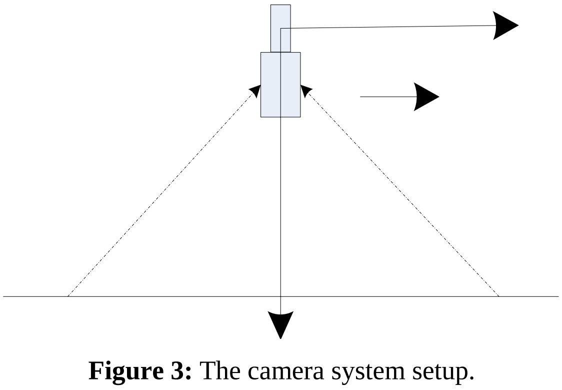


Figure 3: The camera system setup.

2.3 Camera Velocity Estimation

The camera system setup is prepared as shown in figure 3. The optical axis of the camera is perpendicular to the ground plane. If the image can be assumed to have negligible distortion and skew coefficient, then $k_c = \left[0 \quad 0 \quad 0 \quad 0 \quad 0 \right]'$, and $\alpha_c = 0$. By substituting (6), (7) and (8) into (9), gives:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_1 \dfrac{X_c}{Z_c} + c_1 \\ f_2 \dfrac{Y_c}{Z_c} + c_2 \end{bmatrix} \tag{10}$$

Differentiate (10) with respect to time; the velocity of the camera in $X - Y$ plane is given as:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \dfrac{f_1}{Z_c^2} (Z_c V_x - X_c V_z) \\ \dfrac{f_2}{Z_c^2} (Z_c V_y - Y_c V_z) \end{bmatrix} \quad \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \dfrac{Z_c v_x}{f_1} + \dfrac{X_c V_z}{Z_c} \\ \dfrac{Z_c v_y}{f_2} + \dfrac{Y_c V_z}{Z_c} \end{bmatrix} \tag{11}$$

Rearrange (10) to get $\frac{X_c}{Z_c}$ and $\frac{Y_c}{Z_c}$ and substituting in (11), gives:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \dfrac{Z_c v_x}{f_1} + \dfrac{1}{f_1} (x_p - c_1) V_z \\ \dfrac{Z_c v_y}{f_2} + \dfrac{1}{f_2} (y_p - c_2) V_z \end{bmatrix} \tag{12}$$

where

- V_x, V_y and V_z are the camera velocities in X, Y and Z directions respectively.
- Z_c is the distance between the feature point on the ground and the centre of projection of the camera.
- v_x and v_y are the image velocity, $v_x = \Delta x F$ and $v_y = \Delta y F$, where F is the camera's frame rate and $(\Delta x, \Delta y)$ is the optical flow.
- (x_p, y_p) is the centre point of the feature window in the first image.
- Focal length: $f_c = \left[f_1 \quad f_2 \right]'$. Obtained from camera calibration.
- Principal point: $c_c = \left[c_1 \quad c_2 \right]'$. Obtained from camera calibration.

If the ground is assumed flat, i.e. Z_c is constant, (12) becomes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \dfrac{Z_c v_x}{f_1} \\ \dfrac{Z_c v_y}{f_2} \end{bmatrix} \tag{13}$$

2.4 Velocity Estimation Procedure

For 1D velocity estimation, every feature window in the same image would move with the same speed. If only one feature window is considered, estimated velocity may fluctuate. Therefore, 20 feature windows are selected and the average optical flow **d** is used to estimate the velocity for the image pair. Then, the estimated velocity is filtered using a low pass IIR-filter, to further eliminate noise. The velocity estimation procedure is as follows:

1. Do camera calibration to get the intrinsic camera parameters (f_c and c_c).
2. Take two consecutive images I_1 and I_{i+1}.
3. Apply the optical flow algorithm to I_1 and I_{i+1} to get $\mathbf{d}$, the average optical flow from $n(=20)$ feature windows.
4. Calculate the velocity using (13).
5. The estimated velocity is obtained after the ripple is filtered using a low pass IIR-filter.
6. Repeat for each consecutive image.

3 Experimentation

Experiments are performed to check the robustness, precision and feasibility of the proposed approach using different soil types, namely coarse sand, fine sand, gravel, and mixed coarse sand and gravel. Two sets of experiments are performed. First, the camera intrinsic parameters are determined and then the velocity estimation performed.

3.1 Test Rig

The velocity measurement test rig is shown in figure 4 and a schematic of the test rig is shown in figure 5. The test rig consists of a DC motor driven carriage attached with a camera. During experimentation, the motor drive a chain attached to the carriage and causes the camera to move at different speeds. The measured velocity is calculated from a high resolution encoder attached to the motor. The camera is a Silicon Video 1281 CMOS Cameras having a maximum frame rate 6,460 fps at $50X50$ pixels^2 image size.

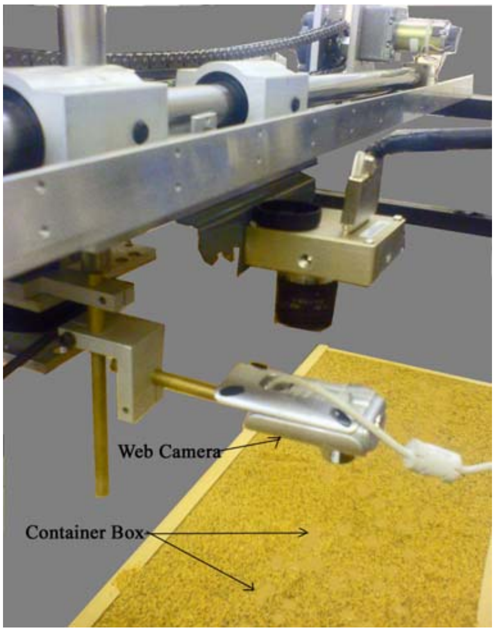


Figure 4: The velocity measurement test rig.

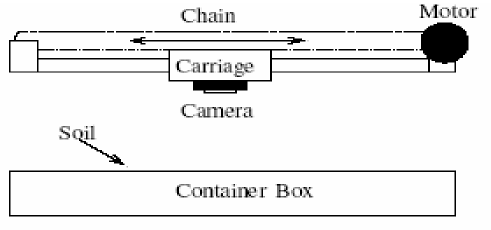


Figure 5: Schematic of velocity measurement test rig.

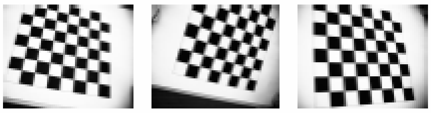


Figure 6: Chessboard images for the camera calibration.

3.2 Camera Calibration

The intrinsic parameters of the camera are determined using the Matlab camera calibration toolbox [22] and chessboard images like those shown in figure 6. The intrinsic parameters obtained are:

$$f_c = [1731.525 \quad 1729.896] \pm [14.150 \quad 14.098] \tag{14}$$

$$c_c = [598.587 \quad 503.540] \pm [17.324 \quad 16.506] \tag{15}$$

$$\alpha_c = 0.0 \pm 0.0 \tag{16}$$

$$k_c = [-0.131 \quad 0.351 \quad 0.004 \quad -0.002 \quad 0.0] \pm \\ [0.045 \quad 0.325 \quad 0.002 \quad 0.003 \quad 0.0] \tag{17}$$

3.3 Velocity Estimation

Experiments are carried out by moving the camera with various constant velocities from 0 to 50 mm/sec over 4 different types of ground (coarse sand, fine sand, gravel, and mixed coarse sand and gravel). The camera captures images at 30 fps and the image size is 1280×1024 pixels^2. The images are undistorted by using the Matlab camera calibration toolbox and cropped into 640×512 pixels^2 to minimise computational time, and then the optical flow algorithm, (13), is applied. The number of feature windows used is 20 and $w_x = w_y = 7$ is chosen for the optical flow algorithm. A result from the optical flow algorithm for fine sand is shown in figure 7. Camera velocity is estimated using (13) and the noise is filtered out by a stable 3rd-order IIR digital filter. The results shown in tables 1-4 are average estimated velocities at steady state. Velocity v is the measured using the encoder and $\hat{v}$ is the estimated velocity from the optical flow algorithm. Figure 8 compares the predicted and measured velocities. The results (tables 1-4 and Fig. 8) show that there is very good agreement between the predicted and measured velocities, with a maximum error of ± 2.017 mm/sec, a maximum %error of 10.939%, and a mean %error of 1.489%.

Table 1: Average estimated velocities over coarse sand.

v **(mm/s)**	$\hat{v}$ **(mm/s)**	$e = \lvert v - \hat{v} \rvert$ **(mm/s)**	**%Error**
2.806	2.612	0.194	6.903
7.047	6.969	0.078	1.110
11.111	10.994	0.117	1.052
15.565	15.553	0.012	0.078
19.672	19.399	0.273	1.389
24.194	24.282	0.088	0.366
28.726	28.797	0.071	0.247
33.074	33.104	0.029	0.090
37.499	37.563	0.063	0.169
41.926	41.911	0.015	0.035
46.259	46.048	0.211	0.456
50.447	50.274	0.173	0.344

Table 2: Average estimated velocities over fine sand.

v **(mm/s)**	$\hat{v}$ **(mm/s)**	$e = \lvert v - \hat{v} \rvert$ **(mm/s)**	**%Error**
2.577	2.479	0.098	3.821
6.748	6.769	0.022	0.322
10.809	10.879	0.069	0.647
15.226	15.362	0.136	0.895
19.676	19.915	0.239	1.214
23.796	24.077	0.281	1.179
28.268	28.503	0.234	0.831
32.518	32.599	0.082	0.251
36.984	37.016	0.031	0.084
41.269	40.972	0.297	0.719
45.720	45.269	0.450	0.986
49.984	47.967	2.017	4.036

Table 3: Average estimated velocities over gravel.

v **(mm/s)**	$\hat{v}$ **(mm/s)**	$e = \lvert v - \hat{v} \rvert$ **(mm/s)**	**%Error**
2.612	2.325	0.286	10.939
6.802	6.539	0.263	3.870
11.075	10.883	0.192	1.733
15.319	15.097	0.223	1.454
19.672	19.399	0.273	1.389
24.127	23.854	0.273	1.136
28.583	28.163	0.420	1.471
32.791	32.380	0.411	1.253
37.158	36.640	0.518	1.395
41.483	40.811	0.672	1.620
45.992	45.188	0.804	1.748
50.211	49.176	1.0335	2.062

Table: 4 Average estimated velocities over coarse sand and gravel.

v **(mm/s)**	$\hat{v}$ **(mm/s)**	$e = \lvert v - \hat{v} \rvert$ **(mm/s)**	**%Error**
2.744	2.315	0.159	6.458
6.498	6.299	0.199	3.056
10.839	10.685	0.155	1.426
15.278	15.139	0.139	0.909
19.594	19.496	0.098	0.498
23.997	24.003	0.006	0.023
28.117	28.146	0.029	0.103
32.652	32.796	0.144	0.442
36.983	36.945	0.039	0.0105
41.426	41.275	0.153	0.369
45.904	45.858	0.045	0.099
50.079	49.504	0.575	1.149

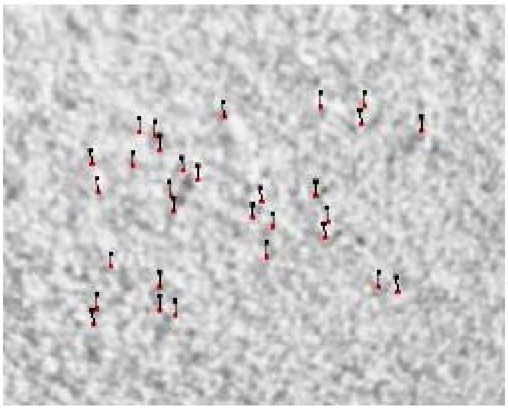


Figure 7: The optical flow result.

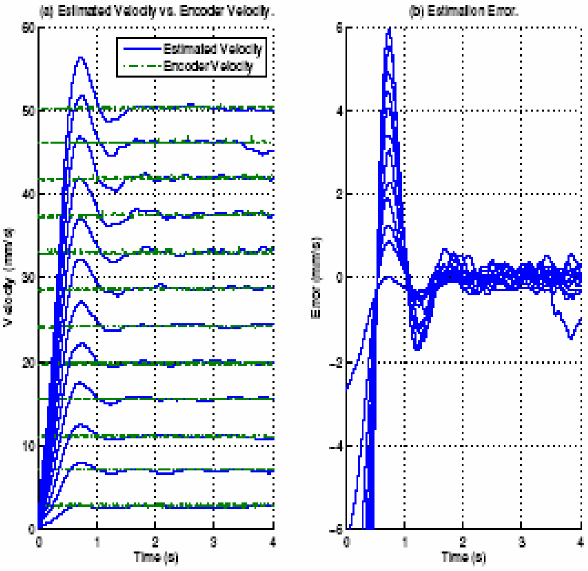


Figure 8: The estimated velocities over coarse sand at 640 x 512 pixels^2.

4 Conclusion and Future Work

An optical flow algorithm for velocity estimation is proposed and its feasibility is evaluated by using a specially designed test rig. It is shown that the vehicle speed can be predicted to a relatively high accuracy (±2.017 mm/sec).

Four different ground samples are used to validate the algorithm, and for all the tests, the system accurately measures the carriage velocity, with a mean maximum %error of 1.489%.

The velocity estimation experiment is performed on a flat test bed. Measured velocities are compared with the velocities predicted by the optical flow algorithm showing very good agreement. It is concluded that the approach has potential for accurate velocity estimation. The following issues need to be addressed in the future:

1. The optimal feature window sizes for real-time computations.
2. The optimal image size for real-time computations.
3. The effect of Z_c on velocity estimation.

Experiments are underway for different real world conditions for developing a robust stand alone velocity measurement technique.

5 References

[1] Z.B. Song, Y.H. Zweiri, L.D. Seneviratne and K. Althoefer, “Non-linear observer for estimating slips of wheeled vehicles”, *Proc. 12th Annual IEEE Conference on Mechatronics and Machine Vision in Practice*, Manila (2005).

[2] A. Davison, “Real-time simultaneous localization and mapping with a single camera”, *Proc. IEEE International Conference on Computer Vision*, Nice, pp 1403-1410 (2003).

[3] Y. Takaoka, Y. Kida, S. Kagami, H. Mizoguchi and T. Kanade, “3D map building for a humanoid robot by using visual odometry”, *Proc. IEEE International Conference on Systems, Man and Cybernetics*, The Hague, pp 4444-4449 (2004).

[4] E. Marchand, P. Bouthemy, F. Chaumette and V. Moreau, “Robust real-time visual tracking using a 2D-3D model-based approach”, *Proc. IEEE International Conference on Computer Vision*, Kerkyra, Greece, pp 262-268 (1999).

[5] B. Jung and G. S. Sukhatme, “Detecting moving objects using a single camera on a mobile robot in an outdoor environment”, *Proc. 8th Conference on Intelligent Autonomous Systems*, Amsterdam, pp 980-987 (2004).

[6] M. Betke, E. Haritaoglu and L. Davis, “Multiple vehicle detection and tracking in hard real time”, *Proc. IEEE Symposium on Intelligent Vehicles*, Tokyo, pp 351-356 (1996).

[7] J. M. Ferryman, S. J. Maybank and A. D. Worrall, “Visual surveillance for moving vehicles”, in *Journal of Computer Vision*, 37(2), pp 187-197 (2000).

[8] S. Singh and B. Digney, *Autonomous Cross-country Navigation using Stereo Vision*, Technical Report CMU-RI-TR-99-03, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1999).

[9] P. Wu, *Optical Navigation System*, BSc Thesis, University of Queensland (2001)

[10] M. H. Bruch, G.A. Gilbreath, J.W. Muelhauser and J.Q. Lum, *Accurate Waypoint Navigation using Non-differential GPS*, Technical Report, Space and Naval Warfare Systems Centre, San Diego (2002).

[11] S. Clark, “Autonomous land vehicle navigation using millimetre wave radar”, *Proc. IEEE ICRA 1998*, Leuven, Belgium, pp 3697-3702 (1998).

[12] D. Dickmanns et al., “The seeing passenger car Vamorsp”, *Proc. IEEE Symposium on Intelligent Vehicles*, Paris, pp 68-73 (1994).

[13] B. Ulmer, “Vita ii-active collision avoidance in real traffic”, *Proc. IEEE Symposium on Intelligent Vehicles*, Paris, pp 1-6 (1994).

[14] A. Suissa, U. Franke, S. Mehring, and S. Hahn, “The Daimler-Benz steering assistant: a spin-off from autonomous driving”, *Proc. IEEE Symposium on Intelligent Vehicles*, Paris, pp 120-124 (1994).

[15] T. Kanade, C. Thorpe, M. Herbert, and S. Shafer, “Vision and navigation for the Carnegie-Mellon Navlab”, *IEEE Trans. Pattern Anal. Machine Intell.*, 10(3), pp 361-372 (1988).

[16] A. Kelly, “Pose tracking for mobile robot localization from large scale appearance mosaics”, *International Journal of Robotics Research*, 19(11) pp 1104-1125 (2000).

[17] D. Fernandez and A. Price, “Visual odometry for an outdoor mobile robot”, *Proc. IEEE RAM 2004*, Singapore, pp 816-821 (2004).

[18] B.K.P. Horn and B.G. Schunck, “Determining optical flow”, *Artificial Intelligence*, 17, pp 185-203 (1981).

[19] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, “Performance of optical flow techniques”, *Int. J. Comput. Vis.*, 12(1), pp 43-77 (1994).

[20] B.D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision”, *Proc. Imaging Understanding Workshop*, pp 121-130 (1981).

[21] S. Birchfield, “Kanade-Lucas-Tomasi feature tracker”, http://www.ces.clemson.edu/~stb/klt visited on 10/05/2006.

[22] California Institute of Technology, “Camera calibration toolbox for Matlab”, http://www.vision.caltech.edu/bouguetj, visited on 15/5/06.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

The Computation Method for Batch Belief Revision of Autonomous Agent

Yang Gao[1], Hao Wang[1], Ruili Wang[2]
National Laboratory for Novel Software Technology
Nanjing University[1], Nanjing, China
Institute of Information Sciences and Technology
Massey University[2], Palmerston North, New Zealand
gaoy@nju.edu.cn, leafwanghao@gmail.com, r.wang@massey.ac.nz

Abstract

In the classical AGM theory, new knowledge is a single sentence. However, in the practical agent systems, new knowledge often comes from several other agents and world which forms a sentence set. Though iterated belief revision is used to deal with the sequential incoming sentences, a practical agent system need maintain its belief set batch during a fix time slot where the agent couldn't and needn't distinguish the sequence of these arriving knowledge correctly. In this paper, a practicable computational method using the link list data structure is brought forward. An example is used to compare with the SATEN system, an iterated belief revision system. The experimental results show that the computational method in this paper, can deal with the multiple sentences batch efficiently.

Keywords: belief revision, agent, computation method, batch belief revision

1 Introduction

Suppose that an intelligent agent has the following pieces of information:

α : All European swans are white.

β : The bird caught in the trap is a swan.

γ : The bird caught in the trap comes from Sweden.

δ : Sweden is a part of Europe.

Then the agent receives several new pieces of information at a given time:

ε : The bird caught in the trap is black.

ζ : Sweden isn't in Europe.

Now, how will the intelligent agent revise its knowledge? Should the agent revise $\alpha - \delta$ by sentence ε before dealing with ζ or vice versa add ζ before ε? Maybe the agent revises its knowledge by ε and ζ simultaneously or by $\varepsilon \wedge \zeta$? Are the above four processes equivalent? And how to realise the revision process in the intelligent agent?

When new information conflicts with old knowledge base system, belief revision theory is used to preserve consistency. The most well-known belief revision system is built and developed by Alchourrón, Gärdenfors and Makinson in 1980s, which is usually abbreviated as AGM [1]. But in AGM theory, the incoming knowledge is single sentence. When agents face multiple sentences, iterated revision operation has to be performed. Though iterated belief revision can be used to deal with multi-sentences, agent processes these sentences one by one. Essentially iterated revision is different from the batch processing. In a practical agent system, agent must process the incoming knowledge in a fixed time slot. During this period, agent always processes a batch of knowledge without considering the temporal nuance between each belief. Many researchers focus on the theory and computational method of multi-sentence belief revision recently [2],[3], however they usually adopt the numeric method (probability theory) to determine the faith of each belief and to solve the collision among beliefs [4],[5]. In this paper, we design a computation method for batch belief revision process of autonomous agent.

The rest of this paper is organised as follows. In section 2, we firstly introduce the computation method for batch belief revision. Three practicable computational procedures based on link list structure are presented. In section 3, we compare the computation method with SATEN system [6]. And make a conclusion in the last section.

2 The Computation Method for Batch Belief Revision

2.1 Data Structure

In practice, we must solve two problems in computation of belief revision. The first question is how to express the epistemic entrenchment according to the partition of the belief set. The second one is how to change the epistemic entrenchment after belief revision. we solve the first problem by means of constructing the nice-ordering partition from total-ordering partition, and tackle the second one depending on the correlation between new knowledge's entrenchments and conflict knowledge.

Above of all, it is necessary to determine the data structure of belief revision. We define $DS = (D, S)$. Here D is the current belief set of intelligent agent, and S is the relationship in D. We call S a logical structure of D. Commonly, the logical structure includes four types: set structure, linear structure, tree structure and net structure. Among these structures, list and tree are often applied in practice. List can be used to express the total-order relationship in a data set. And tree is used to reflect the partial-order relationship. In contrast, net is only applied to realise the local-order relationship.

Then we need to maintain the map from the logical structure to the physical storage structure, when realise the logical structure of agent beliefs in computer. In other words, define $PS : (D, S) \to M$. This map can be divided into sequential, linked, indexed and hashing. So we can get the total $16(4 \times 4)$ possible data structures. In this paper, we describe a nice-order relationship in the multi-sentences belief revision of intelligent agent, which is based on the total-order partition. So we choose list as logical structure and linked map as physical storage. Furthermore, we design the linked list as the data structure of agent beliefs and degrees of beliefs in multi-sentences belief revision.

2.2 Algorithms

Now, all operations of agent beliefs including revision, contraction and expansion and degree's adjustment of beliefs are based on the linked list. The detail algorithm will be discussed as below.

Step A. Agent sorts all beliefs in the belief set in order to get total-ordering partition. In detail, give an initial value for each belief based on agent's subjective judgment. This initial value must be less than 1 and not less than 0. And let the tautology's entrenchment degree equal to 1. But we don't restrict rigidly that the degree of every sentence must be different.

Step B. Then agent use algorithm 1 to adjust the entrenchment so that belief set's entrenchment is satisfied with logical constraint. In other words, if $A_1, A_2, \ldots, A_n \vdash B$, then B's entrenchment degree isn't less than the minimum of $A_1, A_2, \ldots, A_n$'s degrees. So we get the nice-ordering partition in the belief set.

Step C. Lastly, agent use algorithm 2 to add new knowledge into the belief set, which has already been nice-ordering partitioned. And if necessary, remove some knowledge whose entrenchments are the lowest.

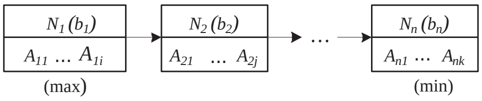


Figure 1: Initial belief entrenchment linked list

In algorithm 1, agent adjust the entrenchment in order to let its belief set satisfied with the logical constraint. At first, design a linked list arranged for all beliefs according to their value of the entrenchments. In this linked list, set beliefs, whose entrenchments are equivalent in the same link node. And the entrenchment degree of link node is the entrenchment of the beliefs. The node, which has higher entrenchment, is in front of the nodes that have lower entrenchment. If there are N-nodes entrenchment degree in belief set, then

$$\begin{aligned} b(A_{11}) &= b(A_{12}) = \ldots = b(A_{1i}) = b_1 \\ b(A_{21}) &= b(A_{22}) = \ldots = b(A_{2j}) = b_2 \\ &\cdots\cdots \\ b(A_{n1}) &= b(A_{n2}) = \ldots = b(A_{nk}) = b_n \end{aligned}$$

where $b_1 > b_2 > \ldots > b_n$. This belief set can be described in figure 1.

Next, process this link list step by step following algorithm 1.

Step 1-1. If the link list has only one node, don't do any process, and algorithm 1 ends.

Step 1-2. Compose a new set $[B] = \{A_{11}, A_{12}, \ldots, A_{1i}\}$ using all beliefs in the node N_1. And analyse every belief such as $A_{n1}, A_{n2}, \ldots, A_{nk}$ in the node N_n. If a belief in N_n can be logically derived from $[B]$, then add this belief into the node N_1 and delete it from N_n.

Step 1-3. Compose a new set $[B]$ using all beliefs in the node N_1 and N_2 again, then do step 1-2 repeatedly. Add the beliefs of N_n, which are the logical consequence of $[B]$, into the node N_2.

......

Step 1-4. Compose a new set $[B]$ using all beliefs in the node $N_1, N_2, \ldots, N_{n-1}$ again, then do step 1-2 repeatedly. Add the beliefs of N_n, which are the logical consequence of $[B]$, into the node N_{n-1}.

Step 1-5. For the sub-link list containing the frontal $n-1$ nodes, recursively run algorithm 1 for it.

Briefly, the algorithm 1 considers every belief from the lowest degree node to the highest degree node carefully. According to the definition 1, we get the nice-ordering partition in the belief set.

Now use algorithm 2 to add new knowledge, and remove some old knowledge if necessary. Suppose that the new knowledge set is $[F]$, and the result belief set is $[B_N]$. The old belief set has already been arranged in the link list by algorithm 1.

Step 2-1. $[B_N] = [F]$.

Step 2-2. Go through the link list from the head to the end. Revise the beliefs $[B]$ in every node in the link list using the **get** function. Let $[B_N] = [B_N] \cup$ **get**$([B],[F])$.

Table 1: A pseudo code of adding new example into the belief set (algorithm 2)

```
get$([B],[F])\{$
  if ($[B]+[F]$ is consistent)
    return $[B]$
  else {
    if ($[B]$ includes one and only one belief)
        return Ø
    else {
        for every $B_i \in [B]$
            if ($[B]-B_i$ is consistent with $[F]$)
                $[A]_i = [B]-B_i$
            else
                $[A]_i$ = get($[B]-B_i,[F]$)
            }
    return $\cap[A]_i$
    }
  }
```

In algorithm 2, hold most knowledge from the old belief set on condition that these beliefs are consistent with new knowledge. When there exists multiple possible sets, algorithm returns the intersection of these multi belief sets.

Now use algorithm 3 to determine the entrenchment degrees of new knowledge. Suppose that new knowledge is $[F] = \{F_1, F_2, \ldots, F_m\}$, the initial entrenchments are set to 1. The old belief set $[K]$ has already been designed in the link list by means of algorithm 1. Hereinafter, determine the entrenchment of new knowledge.

Step 3-1. Compose a set $[B] = \{A_{n1}, A_{n2}, \ldots, A_{nk}\}$ using all beliefs in the node N_n. Consider every sentence whose initial entrenchment is equal to 1 in $[F]$, and let its entrenchment set be $1-b_n$ if the negation could be derived from $[B]$ logically.

Step 3-2. Compose a new set $[B]$ using all beliefs in the node N_n and N_{n-1} again, then do step 3-1 repeatedly. The new entrenchment is $1-b_{n-1}$.

……

Step 3-3. Compose a new set $[B]$ using all beliefs in the node $N_1, N_2, \ldots, N_n$ again, then do step 1-2 repeatedly. The new entrenchment is $1-b_1$.

Step 3-4. After step 3-1 to step 3-3, if there are new beliefs whose entrenchment is 1, adjust its entrenchment to a random value between b_1 and 1.

Obviously, the entrenchment degree of new knowledge is relative to the entrenchment degree of inconsistency knowledge. We figure that the entrenchment of new knowledge is higher when the entrenchments of inconsistency are lower. If new knowledge is consistent with old beliefs, its entrenchment's degree is higher than any other old beliefs. In addition, the logical coherence couldn't be met after adding new knowledge to the belief set using algorithm 3. So agent must run algorithm 1 again in order to get nice-ordering partition in new belief set. Thus in multi-agent system, when agent receive multi new knowledge, agent should revise its belief set following this detailed steps below.

Step 1. According to algorithm 1, adjust the entrenchment degree of every belief in the initial belief set in order to get a nice-ordering partition in the belief set.

Step 2. When new knowledge arrives, run algorithm 3 so that we can determine the entrenchment of every belief in the new belief set.

Step 3. Run algorithm 2 to add new knowledge into the new belief set. If necessary, remove some old beliefs.

Step 4. Go to step 1.

3 Experiment and Analysis

The below table is a simple example used for explaining agent revision progress in batch belief revision system in practice.

Assume the initial belief set of the agent is $\{a, e, b, c, a|b, d, a\&e\}$. Now agent receive a new sentence set $\{f, \neg c|\neg d\}$. Obviously, in the batch belief revision, we first use algorithm 1 to estimate and adjust ordering logically, then use algorithm 3 to determine the entrenchment of new knowledge. And finally add new knowledge and remove inconsistent beliefs using algorithm 2. Table 2 is the result of experiment.

Table 2: A example using batch belief revision

Initial belief set and entrenchment degree		**The result of algorithm 1**		**After add new beliefs, the result of algorithms 3 and 2**	
0.8	a, e	0.8	$a, e, a\&e$	0.85	f
0.6	b	0.6	$b, a\|b$	0.8	$a, e, a\&e$
0.4	$c, a\|b$	0.4	c	0.6	$b, a\|b, \neg c\|\neg d$
0.3	$d, a\&e$	0.3	d	0.4	c

We contrast this example with the Maxi-adjustment in SATEN system [6]. Table 3 gives the results when add sentence f first or add sentence $\neg c|\neg d$ first.

Although there is not much revision to eliminate conflict in this example, we can still find the difference between multi-sentences and SATEN. First, in SATEN, the entrenchment of new belief is always the highest. But in multi-sentences belief revision, the

Table 3: The experimental result in SATEN using maxi-adjustment policy

Initial belief set and entrenchment degree		**Add f before $\neg c \mid \neg d$ in SATEN**		**Add $-c \mid -d$ before f in SATEN**	
		0.95	$\neg c \mid \neg d$	0.95	f
		0.9	f	0.9	$-c \mid -d$
0.8	a,e	0.8	a,e	0.8	a,e
0.6	b	0.6	b	0.6	b
0.4	$c, a \mid b$	0.4	$c, a \mid b$	0.4	$c, a \mid b$
0.3	$d, a\&e$	0.3	$a\&e$	0.3	$a\&e$

entrenchment of new belief can still be determined by logical restriction. Second, the algorithm of SATEN is based on single sentence iterated belief revision. Different revision sequence will get different results. But multi-sentences belief revision is based on link list structure, which can dealt with multi new knowledge pretty good. Third, as an algorithm of single sentence revision, SATEN can't be used on multi-sentences revision. But multi-sentences belief revision may do revision with single new sentence. Certainly it will be more complex, because the multi-sentences belief revision involves three algorithms for each operation and has to adjust ordering each time. SATEN performs these three algorithms separately, one algorithm each time. It is very simple and fast when deals with single sentence. But it has to judge which algorithm to be used when calculate. Final, it is more rational for time-sharing agent system to use multi-sentences belief revision, because during each period of time it may receive several knowledge. But for real-time system, using single sentence revision is more rational. When a piece of knowledge comes, it deals with it. The requirement of the algorithm is simple and fast. In the discussion before, we aimed at proposition logic. When the algorithm is used in first-order logic, it is infeasible because of the undecidability of first-order logic. But Reiter pointed out that although the decision procedure for logical consequence isn't guaranteed to terminate, the domain of real problem is limited and the decision procedure is decidable. But even if in the decidable instance, the time complexity is still exponential. To simplify our analysis, we use Dixon's analysis method [7]. It can be seen from algorithm above that the operations of remove or add beliefs and adjust ordering are trivial when contrasted with the decision procedure. Therefore, here we only use the time of the decision procedure to present the algorithm complexity. Assume the algorithm complexity of once decision procedure is M, then the complexity of performing the multi-sentences belief revision once is 3M. The complexity is related to the size of the knowledge base. Obviously, the bigger the set is, the higher M is.

4 Conclusion

In this paper, we present a feasible computation method of batch belief revision, which can process sentence set efficiently. Thus in intelligent agent system, we can model agent using batch belief revision process when agent receives multi sentences. In the future works, we will make an effort to apply the batch belief revision into actual robot domain.

5 Acknowledgement

The paper is supported by the Natural Science Foundation of China (No.60475026), and the National Grand Fundamental Research 973 Program of China (No.2002CB312002).

6 References

[1] C.E. Alchourró, P. Gädenfors and D. Makinson, ”On the logic of theory change: partial meet contraction and revision functions”, *Journal of Symbolic Logic*, 50(2), pp 510-530 (1985).

[2] A. Fuhrmann and S.O. Hansson, ”A survey of multiple contractions”, *Journal of Logics, Language, and Information*, 3, pp 39-76 (1994).

[3] H. Rott, ”Modelings for belief change: base contractions, multiple contractions, and epistemic entrenchmen”, *Lecture Notes in Artificial Intelligence*, 633, pp 139-153 (1992).

[4] G. Kern-Isberner, ”Conditionals in nonmonotonic reasoning and belief revision: considering conditionals as agents”, *Lecture Notes in Artificial Intelligence*, 2087, pp 190 (2001).

[5] G. Kern-Isberner, ”Revising and updating probabilistic beliefs”, *Frontiers in Belief Revision*, Kluwer Academic Publishers, Dordrecht, pp 329-344 (1999).

[6] SATEN, ”Information extraction and revision engine”, http://magic.it.uts.edu.au/systems/saten.html, visited on 1/12/2004.

[7] S.E. Dixon and W.R. Wobcke, ”The implementation of a first-order logic AGM belief revision system”, *Proceedings of 5th IEEE International Conference on Tools with AI*, Boston MA, pp 40-47 (1993).

Audio Based Activity Classification for Security Applications

Mahmood Jadaan, Liyanage DeSilva and Amal Punchihewa
Institute of Information Sciences and Technology,
Massey University, Palmerston North, New Zealand
mjadaan@gmail.com, l.desilva@massey.ac.nz, g.a.punchihewa@masssey.ac.nz

Abstract

In this paper we present an audio based voice classification system, such a system can be very crucial for many applications; for example, it can be used to create a number of new and growing services in the area of health care and safety for elderly and ill patients. There has been a considerable amount of research undertaken on the topic of security applications; however, much of this research is based on video alone or video combined with audio. The research reported in this paper is rather unique since it involves the use of audio only. Our research makes use of different software tools such as Praat™ and MATLAB^{TM} in order to distinguish between vocal and non-vocal sounds as well as identifying whether a vocal sound is present in a given sound wave.

Keywords: Intensity, pulses, spectrum, pitch, formants, non-dispersive medium, dispersive medium, ultrasonic, subsonic

1 Introduction

There has recently been considerable interest in audio based activity classification for security applications, and this interest has been growing over the years mainly due to the increased use of information technology. In researching this topic area, it has been found that there has been plenty of research done in the use of captured and stored images, video, and digital audio, but the principal focus of the research reported in this paper will be on audio based activity.

This research direction involves developing an audio based security system rather than a video based security system; since, even though a video based security system would make it much easier to identify where the sound is coming from, an audio based security system is much cheaper to make and also, for privacy reasons, video is not always desirable. It has also been found that there hasn't been much research done on this topic area in the past.

The main objective of this project was to be able to experimentally distinguish between vocal and non-vocal sounds by experimenting with a variety of different sounds, for example:

- For vocal sounds we can use normal speech, singing, yelling, and even coughing sounds etc
- For non-vocal sounds we can use different sounds like breaking of glass, footsteps, clapping, opening and closing of doors, etc.

This project has involved developing a system that is able to identify audible sounds and can classify the difference between them according to what caused these sounds. It has also been important to be able to distinguish between the vocal and non-vocal sounds, as this will be essential in security and health related applications. Figure 1 shows some types audio signal together with a few simple examples of each type:

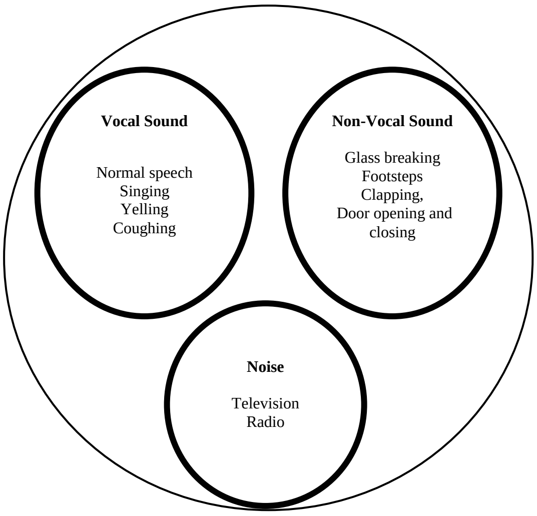


Figure 1: Types of audio signal with examples.

2 Research and Results

Some research was performed on the basic characteristics of sound waves, and it was found that these characteristics can vary significantly between different types of sound waves. These characteristics were used to help simplify the process of identifying vocal and non-vocal sounds.

The following key characteristics were identified:

- Pulses in a sound wave
- Sound wave intensity
- Spectrum of a sound wave
- The pitch of a sound wave
- A formant in a sound wave

These characteristics will now be discussed in more detail.

2.1 Pulses in a Sound Wave

Usually a pulse wave is a kind of non-sinusoidal waveform that is similar to a square wave, but does not have the symmetrical shape associated with a perfect square wave. It is a term common to synthesiser programming, and is a typical waveform available on many synthesisers. However, in signal processing, the term *pulse* has the following interpretations:

- In telephony - which is the transmission of sound between distant stations - pulse dialling is a way of dialling a telephone number using interrupted electrical pulses.
- A rapid, transient change in the amplitude of a signal from a baseline value to a higher or lower value, followed by a rapid return to the baseline value.
- A rapid change in some characteristic of a signal, e.g., phase or frequency, from a baseline value to a higher or lower value, followed by a rapid return to the baseline value [1].

The last two meanings of the word *pulse* are the two appropriate interpretations for use in this project.

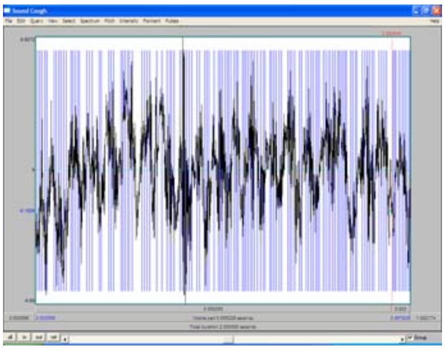


Figure 2: Pulses shown in a vocal sound wave of a person coughing.

Figure 2 shows a waveform of a recorded cough which is a vocal sound; the vertical lines seen in the figure represent the pulses on the waveform, so from the above wave we can see that pulses occur when there is a sudden change in frequency, phase, or amplitude.

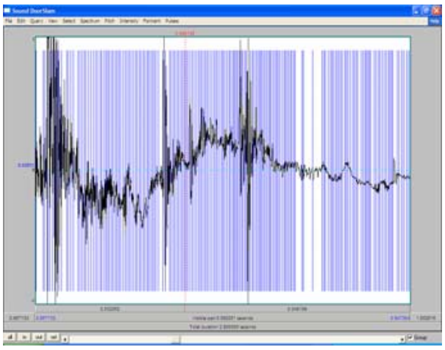


Figure 3: Pulses shown in a non-vocal sound wave of a door slam.

Figure 3 shows a waveform recorded for a door slam which is a non-vocal sound; the vertical lines seen in the figure represent the pulses in the waveform.

Comparing the pulses from the two waveform results we can see that there's not enough variation in these pulses to be able to distinguish between the two types of sound so that it will be hard to use pulses in waveforms to tell the difference between vocal and non-vocal sounds.

2.2 Sound Wave Intensity

The term *intensity* is used exclusively for the measurement of sound in watts per unit area. To describe the strength of a sound in terms other than strict intensity, one can use "magnitude" "strength", "amplitude", or "level" instead.

Sound intensity is not the same physical quantity as sound pressure. Hearing is directly sensitive to sound pressure which is related to sound intensity. In stereo the level differences have been called "intensity" differences, but sound intensity is a specifically defined quantity and cannot be sensed by a simple microphone, nor would it be valuable in music recording if it could.

The sound intensity, **I**, (acoustic intensity) is defined as the sound power $\mathbf{P_{ac}}$ per unit area **A**. The usual context is the noise measurement of sound intensity in the air at a listener's location. The SI units are W/m^2 (watts per square meter) [2].

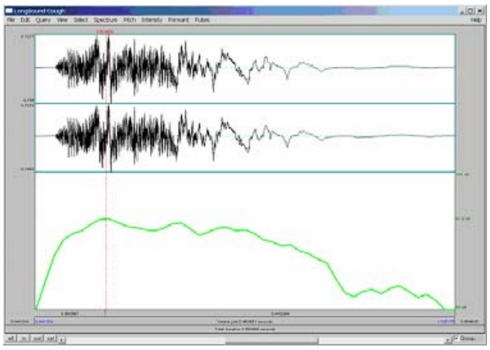


Figure 4: Intensity shown in a vocal sound wave for a person coughing.

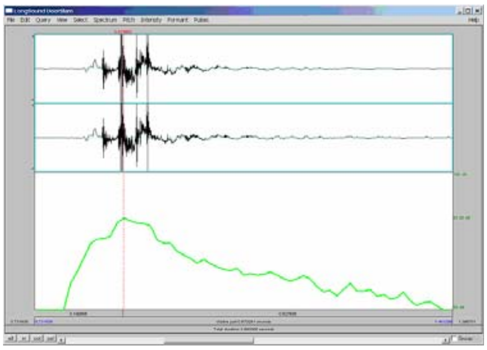


Figure 5: Intensity shown in a non-vocal sound wave of a door slam.

Figure 4 shows a wave form for a recorded vocal sound of a person's cough; and below the waveform of the figure is a plot of its *intensity*.

Figure 5 shows the wave form of a recorded non-vocal sound using a door slam; and below this waveform is a plot of its intensity. Once again, we see that there's not enough information in the intensity of a sound wave to enable us to distinguish between vocal and non-vocal sounds. Although it should be noted that the intensity in a sound wave can have other uses; for example, it can be used in the process of locating a sound source in a given area.

2.3 Spectrum of a Sound Wave

A spectrogram displays time on the X axis and frequency on the Y axis. The darkness of the display indicates how much energy or how big the amplitude is, as seen at that frequency at that time. A spectrogram is created by performing a Fourier transform on small time-slices of the signal to determine the amplitudes of the component sine waves at each time-slice.

A sound spectrum is a representation of a sound, usually a short sample of a sound in terms of the amount of vibration at each individual frequency. It is usually presented as a graph of either power or pressure as a function of frequency. The power or pressure is measured in decibels and the frequency is measured in vibrations per second (or Hertz, abbreviation Hz) or thousands of vibrations per second (kilohertz, abbreviation kHz). Sound spectrum can be thought of as a form of "sound recipe", viz: take this amount of frequency, add this amount of frequency etc until we have put together the whole, complicated sound.

So, a source of sound can have many different frequencies mixed together. Each frequency stimulates a different length receptor in our ears. When only one length is predominantly stimulated, we hear a note. A steady hissing sound or a sudden crash stimulates all the receptors, so we say that it contains some amounts of all frequencies in the audible range. Things in our environment that we refer to as noise are usually made up of many different frequencies. When the sound spectrum is flat, it is called *white noise*. This term carries over into other types of spectrums apart from sound [3].

Figure 6 is the wave form of a recorded vocal sound using a person's cough; and below the waveform is a plot of its spectrum.

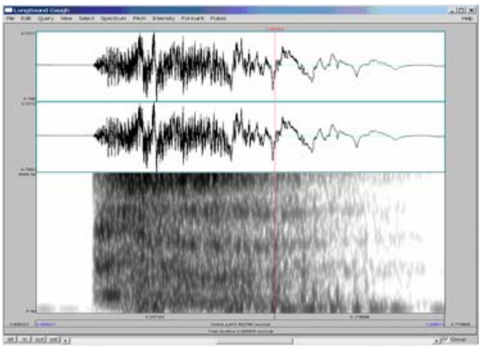


Figure 6: Spectrum plot shown in a vocal sound wave of a person coughing.

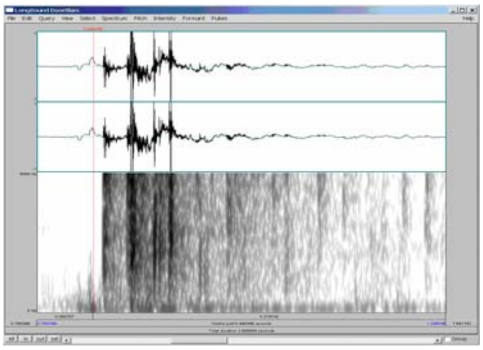


Figure 7: Spectrum plot shown in a non-vocal sound wave of a door slam.

Figure 7 shows a waveform for a recorded non-vocal sound of a door slam; and below the waveform is a plot of its spectrum. In the two previous spectrum plots we can see that the vocal sound frequencies have a constant depth in colour, while, on the other hand, the non-vocal sound frequencies devoid of any pattern on the plot; therefore, *these two plots could be used in the process of distinguishing between vocal and non-vocal sounds.*

2.4 The Pitch of a Sound Wave

The frequency of sound is the rate at which the waves pass a given point. It is also the rate at which a guitar string or a loud speaker vibrates. Frequency is also called the *pitch* of a sound. A high pitched sound corresponds to a high frequency and a low pitched sound corresponds to a low frequency.

The relationship between velocity, wavelength and frequency is:

Velocity = Wavelength x Frequency

Since the velocity of sound is approximately the same for all wavelengths, frequency is often used to better describe the effects of the different wavelengths. The pitch of a sound that we experience is determined by its wavelength or its frequency. The shorter the wavelength, the higher the frequency becomes, and the higher the pitch that we hear [4].

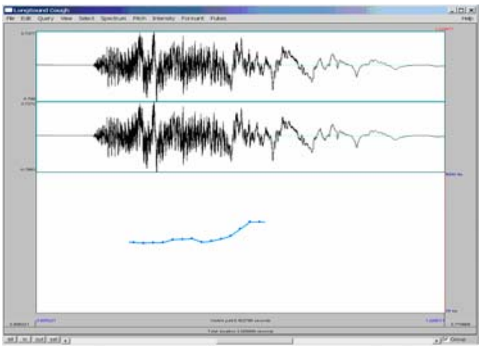


Figure 8: Pitch shown in a vocal sound wave of a person coughing.

The figure above is a wave form of a recorded vocal sound of a persons cough; and below the waveform is a plot of the pitch that was generated by this sound. From this plot we can also see that the pitch in a vocal sound is continuous.

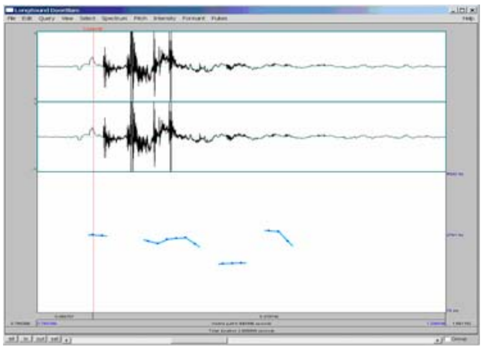


Figure 9: Pitch shown in a non-vocal sound wave of a door slam.

Figure 9 is the waveform of a recorded non-vocal sound using a door slam; and below the waveform is a plot of the pitch that was generated by this sound. Thus, by observing these two plots of pitch it can be seen that for vocal sounds, pitch is continuous, while on the other hand, for non-vocal sounds pitch is discontinuous, therefore these two plots can also be used in the process of distinguishing between vocal and non-vocal sounds.

2.5 Formant in a Sound Wave

A *formant* is a peak in an acoustic frequency spectrum which results from the resonant frequencies of any acoustical system. It is a very important term, normally measured from a spectrogram plot. It is most commonly invoked in phonetics or acoustics involving the resonant frequencies of vocal tracts or musical instruments. However, it is equally valid to talk about the formant frequencies of the air in a room.

Formants are the distinguishing or meaningful frequency components of human speech and of singing. By definition, the information that humans require to distinguish between vowels can be represented in a purely quantitative way by the frequency content of the vowel sounds. Formants are the characteristic partials that identify vowels to the listener. Most of these formants are produced by tube and chamber resonance, but a few whistle tones derive from periodic collapse of Venturi effect low-pressure zones. The formant with the lowest frequency is called f_1, the second f_2, and the third f_3. Most often the two first formants, f_1 and f_2, are enough to disambiguate the vowel which, in other words, means that we can establish a single grammatical or semantic interpretation for them. These two formants are primarily determined by the position of the tongue. f_1 has a higher frequency when the tongue is lowered, and f_2 has a higher frequency when the tongue is forward. Generally, formants move about in a range of approximately 1000 Hz for a male adult, with 1000 Hz per formant. Vowels will almost always have four or more distinguishable formants; sometimes there are more than six [5].

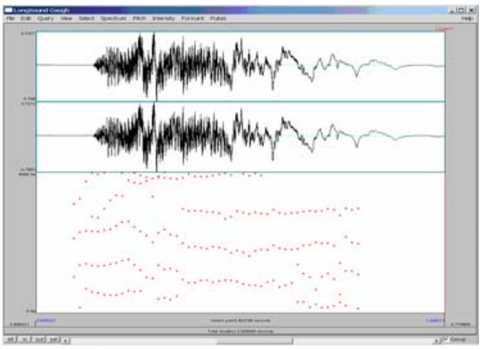


Figure 10: Formants shown in a vocal sound wave of a person coughing.

Figure 10 is the waveform of a recorded vocal sound of a person's cough; and below the waveform is a plot of the formants that exist in this sound. From this plot we can also see that the formant in a vocal sound has a kind of pattern that is a continuous line.

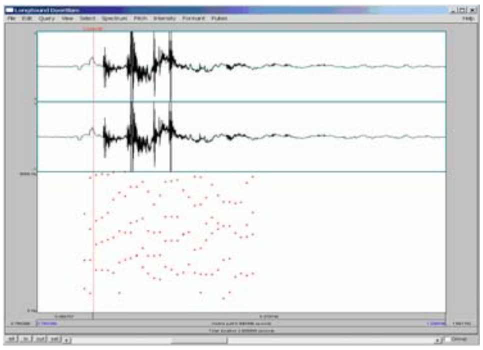


Figure 11: Formants shown in a non-vocal sound wave of a door slam.

Figure 11 shows a waveform for a recorded non-vocal sound of a door slam; and below the waveform is a plot of the formants that exist in this sound. Thus if we look at the two formant plots we can see that the vocal sounds formant has a kind of pattern similar to a continuous line; while, in the other hand, the non-vocal sounds formant is all over the place and there are hardly any patterns. Therefore, these two plots are also useful and can also be used in the process of distinguishing between vocal and non-vocal sounds.

So, after finding out more about these characteristics, and how they can be used in the different processes of identifying between the unique properties of sound waves, two programs were developed using MATLAB™ programming software.

An initial program was developed to find the five formant values and calculate the value of the pitch, and a second program was developed to find out if a vocal sound was present in a given (.wav) file.

The MATLAB results for the first program are shown in figure 12:

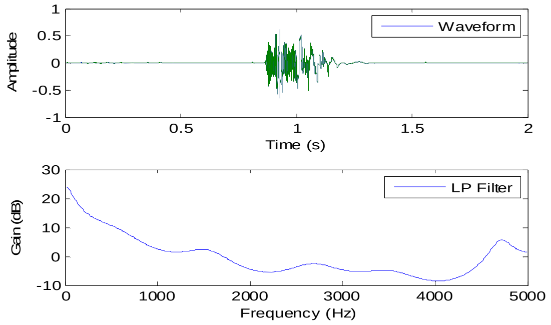


Figure 12: Plot of the amplitude vs. time of a vocal sound wave (coughing sound), and a plot of the gain (dB) vs. frequency that is used to locate the formants.

Formant 1 Frequency:	549.1
Formant 2 Frequency:	1550.5
Formant 3 Frequency:	2689.5
Formant 4 Frequency:	3517.3
Formant 5 Frequency:	4706.1

The results shown in figure 12 are for a vocal sound (coughing) given as a plot in the time domain, this plot is the top one (amplitude vs. time in seconds), the second plot is a plot of the gain in dB vs. the frequency in Hz. It is seen in the second plot that the peaks give the formant values, and normally the first formant value is equal to the pitch value, therefore the pitch value in this case is equal to 549.1Hz. The formants were found using the linear prediction coding (LPC) method of analysis. The LPC method is normally used in analysing a speech signal by estimating the formants, removing their effects from the speech signal, and estimating the intensity.

The results shown in figure 13 are for a non-vocal sound (the door slam sound) shown as a plot in the time domain, (the upper plot) for the case of amplitude vs. time in seconds.

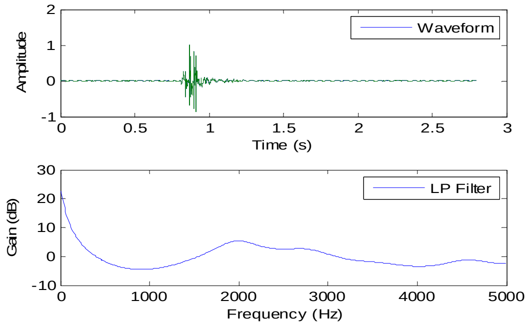


Figure 13: Plot of the amplitude vs. time of a non-vocal sound wave (door slam), and a plot of the gain (dB) vs. frequency, which is used to locate the formants.

Formant 1 Frequency:	1460.0
Formant 2 Frequency:	1975.1
Formant 3 Frequency:	2777.3
Formant 4 Frequency:	3605.5
Formant 5 Frequency:	4562.1

The lower plot is a plot of the gain in dB versus the frequency in Hz. It is seen in the second plot that the peaks give the formant values. The first formant value is usually equal to the pitch value; therefore, the pitch value in this case is equal to 1460.0Hz. After this was done the program was further enhanced in order to tell if the pitch of the sound was continuous or discontinuous, which enables us to distinguish between vocal and non-vocal sounds; since, as mentioned earlier for a vocal sound this will result in a continuous pitch and a non-vocal sound will result in a discontinuous pitch.

The second program developed, was created in order to find out whether a human speech sound (a voiced sound) was present or not. The first thing that was done was to take the absolute value of the original plot of the speech signal and plot it on one side of the x-axis; as shown in figure 14:

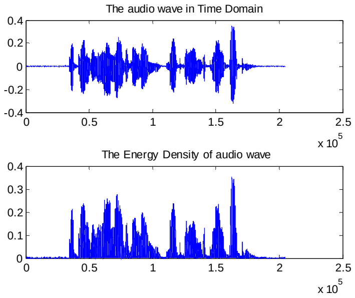


Figure 14: Plot of a vocal sound wave in the time domain (female speech sound), and plot of the absolute value taken of that original speech signal plot.

The next step taken was to chop up the speech signal in to heaps of small samples, and then take the average of each sample, this would give us the intensity of the plot, and this is shown in figure 15:

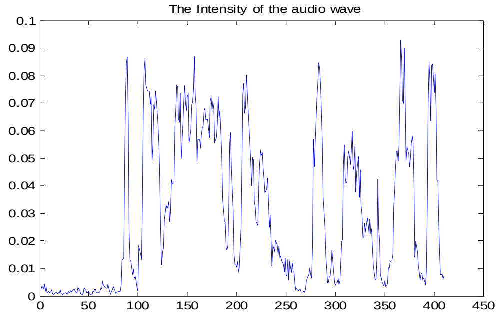


Figure 15: Plot of the intensity of the speech signal.

After this is done, we can see that above a certain threshold, in this case 0.005, a speech signal is present and, therefore, figure 16 is produced using a MATLAB program that shows where the vocal sound is present.

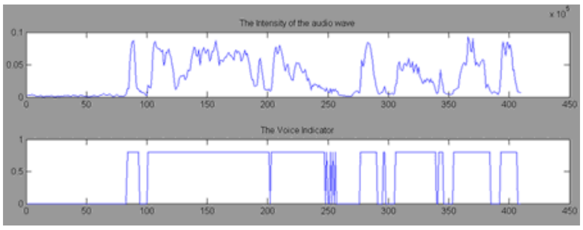


Figure 16: Plot of the intensity of the speech signal, and plot of where the speech signal is present.

3 Discussion and Conclusions

Work is still in progress to analyse various samples of audio data that have been taken from different users, the MATLAB software has been used to produce a program that is able to distinguish between vocal and the non-vocal sounds, this was done by researching the five potential properties for distinguishing between these types of sound, viz: intensity, pulses, spectrum, pitch and formant. Each of these properties had some important characteristics that could potentially aid in analysing the different observed waveforms. It was seen that the first two properties (intensity and pulse shape) didn't have enough information to help identify differences between vocal and non-vocal sounds. However the intensity factor can be used in the process of locating the source of the sounds in a room, meanwhile the other three properties (pitch, formants and spectrum) showed useful patterns that provided superior information to assist in identifying the differences between vocal and non-vocal sounds, and also in the process of showing whether a vocal sound was present in a sound wave.

4 Future Work and Recommendations

For future work it would be useful to be able to distinguish both vocal and non-vocal sounds within an actual sound using a single program. This means that it would be possible to identify between coughing and sneezing sounds which are the vocal sound and also distinguish between clapping, footsteps or door closing/opening sounds that are the non-vocal sounds.

Throughout this paper it has been assumed that there are no extraneous sounds present. However, in most applications this will not be the case and hence it will be necessary to perform proper noise elimination. This can have many other applications; for example, it would be useful to work on developing a program that is able to extract lyrics from songs and then those lyrics can be output as text.

5 References

[1] Wikipedia, “Pulse in a sound wave” http://en.wikipedia.org/wiki/Pulse_wave, visited on 18/9/2006.

[2] Wikipedia, “A sound wave’s intensity”, http://en.wikipedia.org/wiki/Sound_intensity, visited on 18/9/2006.

[3] University of New South Wales School of Physics, “Spectrum in a sound wave”, http://www.phys.unsw.edu.au/~jw/sound. spectrum.html, visited on 20/9/2006.

[4] School for Champions, “A sound wave’s pitch”, http://school-for-champions.com/science/sound.htm, visited on 20/9/2006.

[5] Wikipedia, “Formant in a sound wave”, http://en.wikipedia.org/wiki/formant, visited on 20/9/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

On the Service Template-based URC Service Planning

Sangseung Kang and Joo-chan Sohn
Intelligent Robot Research Division,
Electronics and Telecommunications Research Institute, Daejeon, Korea
kss@etri.re.kr

Abstract

The purpose of semantic web services is to support effective automation of various web services related activities including service discovery, composition, execution, and monitoring. In this paper, we propose a URC service planning system for intelligent robot services in the semantic web services environments. In the proposed system, generated service plans are transformed into executable service plans, and the system provides real robot services through the web service runtime engine.

Keywords: service plan, service template, ubiquitous robot companion, semantic web services

1 Introduction

Semantic web languages, such as the OWL (Web Ontology Language) [1], provide the foundation for such sufficient and rich descriptions. The OWL-S [2], which are OWL service languages, is also a set of ontologies for description of the web service property and capability. Semantic web concepts are used to define intelligent web services [3]. The joint concept of semantic web and web service for intelligent web services are usually referred to semantic web services. Semantic web services target support for effective automation of various web services relating activities such as service discovery, composition, execution, and monitoring [4-6]. Particularly, problem in the service composition is important; web services fall into diverse categories and compose such a diverse set of services that may require various technologies.

We propose a URC service planning system for intelligent services in the semantic web service environments, requirements in considerations for the URC service planning, design a new architecture of the proposed system, and implement of the system. A service plan is a successive process in the system that includes the necessary activities for the composition and the connections of real web services and other components. Generated service plan is transformed into the executable service plan, and the system provides real robot services through the web service runtime engine in the semantic web service environments.

We implement a test bed system that provides real web services. The relevant information is described in the OWL-S, which consists of profile, process, and grounding. Then the system generates a plan instantly from the OWL-S profile, and composes a service plan for the web service calling from the OWL-S grounding and the WSDL [7] information. In the implemented system, executable service plans are written by the BPEL4WS (Business Process Execution Language for Web Services) [8].

This paper is organised as follows. In section 2 we present requirements and considerations for the URC service planning. Section 3 introduces our approach with service templates. Section 4 describes the structure of the proposed system and the features of components. Conclusions are given in section 5.

2 Requirements and Considerations

Current web service techniques are limited to services, and each of them is independently operated. There have problems where extensions of syntax are possible as a solution, but the extensions of semantic aren’t applied. Semantic web service technology is presented by solution of the above problems that is deduced in the development process of these web services. It is our attempt to solve limitations of web services by applying the semantic web and agent technologies to web services. Semantic web services help achieve semantic interoperability include following; knowledge representation and reasoning, automated planning to enable service composition, and the use of shared ontologies [5-6].

URC is an acronym for Ubiquitous Robot Companion and a robot which provides various necessary services *whenever* and *wherever*. URC is newly defined by fusion of robot technologies and information telecommunication technologies [9]. The concept of URC is feasible to use rich semantic web services in the networked environments.

The URC service planning system provides the functions that generate executable service plans from user commands. Functional and non-functional requirements for the service planning for intelligent web services are as follows:

- The system should be able to provide the function that preprocesses the input data from the context processing system for user commands.
- They should be able to express web services, and provide a query interface that sends a query to the

service registry for service discovery and receives the result for the query.

- Depending on the result of service discovery, the system should be able to make atomic service models, and generate the service plan instance using composing function for the selected service models.
- The generated service plan should be able to transform into the executable service plan, and provide real web services through the web service runtime engine.
- The system should be able to support an access interface to the service registry for the effective service discovery.
- They should be able to consider a service definition language that is machine-readable for the automatic composition of service processes.
- The system should be able to support interoperability among various systems, and see about dynamic configuration of complex services.
- They must consider the effectiveness and practical use that can be used generally in various services.

3 Service Templates

The URC service planning technology provides various functions that automatically select web services, and dynamically create and compose web service plans based on service templates. For that, we define a description language of the service template. The description language is base on the web ontology language.

A service template is a reusable building block for the service composition, and an object described meta semantic which represents each service. Service templates are classified according to type of basic and aggregate. The basic template describes service semantic of the minimum unit; the aggregate template is represented by a set of basic templates or the other aggregate templates. A service template instance describes binding information for concrete service objects of service templates that describe meta semantic. The service template registry provides semantic based indexing services for service templates. The semantics of the service template are various information including input, output, precondition, effect, and category. Class Template collects the two types of templates such as BasicTemplate and AggregateTemplate. Each of these is described below.

```
<owl:Class rdf:ID="Template">
  <rdfs:label>Template</rdfs:label>
  <rdfs:comment>Definition of Template
  </rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="BasicTemplate">
  <rdfs:label>BasicTemplate</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Template" />
  <rdfs:comment>Definition of Basic Template
  </rdfs:comment>
</owl:Class>

<owl:Class rdf:ID="AggregateTemplate">
  <rdfs:label>AggregateTemplate</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Template" />
  <rdfs:comment>Definition of Aggregate Template
  </rdfs:comment>
</owl:Class>
```

Class Template has related properties hasParameter, hasInput, hasOutput, hasPrecondition, and hasEffect, which range over classes Parameter, Input, ConditionalOutput, Precondition, and ConditionalEffect, respectively. Inputs specify the information that the process requires for its execution. The result of the execution of the process is the generation of a set of outputs. Effects describe conditions in the world, while outputs describe information. Preconditions specify conditions that should be satisfied for a process to execute correctly.

```
<rdf:Property rdf:ID="hasParameter">
  <rdfs:domain rdf:resource="#TemplateProfile"/>
  <rdfs:range rdf:resource="&rdfs;#Class"/>
</rdf:Property>
<owl:ObjectProperty rdf:ID="hasInput">
  <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasOutput">
  <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPrecondition">
  <rdfs:subPropertyOf rdf:resource="#hasStatus"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasEffect">
  <rdfs:subPropertyOf rdf:resource="#hasStatus"/>
</owl:ObjectProperty>
```

Below is an example code fragment that shows a service template in our service template authoring environment. The template is base on the service plan of the URC service composition system. There is generated a service plan instance for URC execution from a service template.

```
<tmpl:AggregateTemplate rdf:ID="TestTemplate">
<rdfs:label>TestTemplate</rdfs:label>
  <tmpl:representedBy>
    <tmpl:TemplateProfile rdf:ID="TestTemplateProfile">
      <tmpl:hasInput rdf:resource=
        "&HumanRelation;#Person" />
      <tmpl:hasEffect rdf:resource=
        "&Behaviour;#DoService" />
    </tmpl:TemplateProfile>
  </tmpl:representedBy>
  <tmpl:constructedBy>
```

```
    <tmpl:Sequence>
      <tmpl:hasComponents rdf:parseType=
        "Collection">
        <tmpl:BasicTemplate rdf:about=
          "&A1Template;#A1Template"/>
        <tmpl:BasicTemplate rdf:about=
          "&A2Template; #A2Template"/>
      </tmpl:hasComponents>
    </tmpl:Sequence>
  </tmpl:constructedBy>
</tmpl:AggregateTemplate>
```

4 URC Service Planning

The framework configuration and functional flow of the proposed service planning system are illustrated in figure 1. The proposed system consists of three functional subsystems such as template search and selection subsystem, service discovery and selection subsystem, service plan composition subsystem. The template search and selection subsystem consist of these modules: composition preprocessing, template selection, service model operation. The service discovery and selection subsystem consists of these modules: service discovery, service selection, atomic service model construction. The service plan composition subsystem consists of plan instance construction module and plan assembly module.

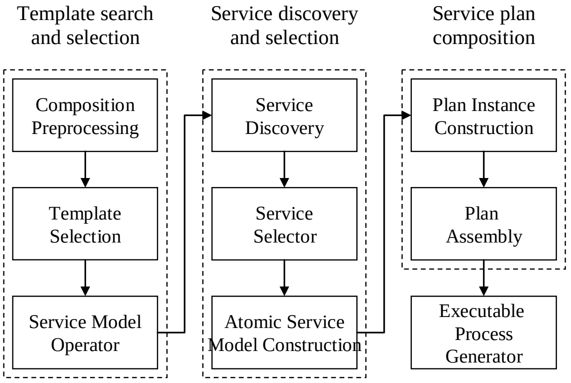

Template search and selection
Service discovery and selection
Service plan composition
Composition Preprocessing
Template Selection
Service Model Operator
Service Discovery
Service Selector
Atomic Service Model Construction
Plan Instance Construction
Plan Assembly
Executable Process Generator

Figure 1: Functional flow diagram of the system.

The specifications of each modules of the service planning system for intelligent web service are as follows: The composition preprocessing module needs to be parsed in fixed form from input data. This component performs analysis and preprocessing for IOPE (input, output, precondition, effect) information from the context processor. The template selection module searches the service template repository base on the analysed IOPE information, and selects a relevant service template. The service model operation module analyses the atomic service models from the service selection module and the template object model from the template selection module, and supports the generation of the service plan instance. The service discovery module sends a query to the service registry for searching of the relevant service model. The service registry returns the result that consists of OWL-S and WSDL contents. The service selection module extracts an optimal service model from the result that gets through the semantic service discovery. The atomic service model construction module controls the atomic service models that extracts from the service selection. The plan instance construction module generates a new service plan instance base on the selected service template, and initialises the generated service plan. The plan assembly module supports composition of extracted atomic services and verification of the composed service plan.

Generated service plans are transformed into the executable service plans in the executable process generation module and the real web services are provided through the web service runtime engine.

Below is an example code fragment that shows a generated plan instance in the implemented system. The instance is the result of the service plan composition subsystem. It is also transformed by the executable process generation module.

```
<inst:TemplateInstance
rdf:ID="TestTemplateInstance">
<inst:instantiates rdf:resource=
  "&TestTemplate;#TestTemplate"/>
  <inst:hasGroundings rdf:parseType="Collection">
    <inst:Grounding>
      <inst:forTemplate rdf:resource=
        "&A1Template;#A1Template"/>
      <inst:byService rdf:resource=
        "&E1Service; #E1ServiceGrounding"/>
    </inst:Grounding>
    <inst:Grounding>
      <inst:forTemplate rdf:resource=
        "&A2Template;#A2Template"/>
      <inst:byService rdf:resource=
        "&E2Service; #E2ServiceGrounding"/>
    </inst:Grounding>
  </inst:hasGroundings>
</inst:TemplateInstance>
```

The semantic information of service from service registry is written by OWL-S [2]. OWL-S makes these functionalities possible: the service profile for advertising and discovering services; the process model, which gives a detailed description of a service operation; and the grounding, which provides details on how to interoperate with a service, via messages. The Profile of a service provides a concise description of the service to a registry, and uses IOPE for the expression of semantic information of the service.

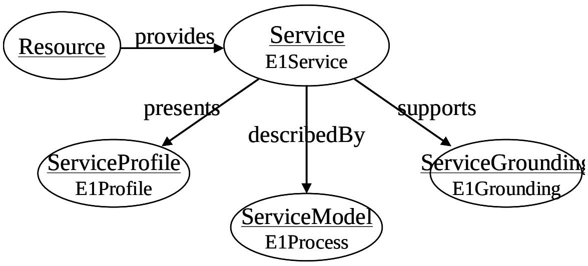


Figure 2: OWL-S service diagram.

We implement a test bed system that provides real web services. The relevant information is described in the OWL-S that consists of profile, process, and grounding. The system generates a plan instance from the OWL-S profile, and composes a service plan for calling the web service from the OWL-S grounding and the WSDL information [7]. In the implemented system, executable service plans are written by the BPEL4WS (Business Process Execution Language for Web Services) [8].

Below is an example code fragment that shows a WSDL document, which represent basic robot services as web services in the implemented system. The document describes message, port type, operation, binding information, service name, and so on. It is distributed by Axis as SOAP engine [10].

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://129.254.70.234/axis/WakeUpService"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://129.254.70.234/axis/WakeUpService"
xmlns:intf="http://129.254.70.234/axis/WakeUpService"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <wsdl:message name="ApproachLocationRequest">
 …
 <wsdl:portType name="WakeUpService">
   <wsdl:operation name="E3ApproachLocation">
     <wsdl:input name="ApproachLocationRequest"
       message="impl:ApproachLocationRequest"/>
     <wsdl:output name=
       "ApproachLocationResponse"
        message="impl:ApproachLocationResponse"/>
   </wsdl:operation>
   …
 </wsdl:portType>
 <wsdl:binding name=
   "WakeUpServiceSoapBinding"
    type="impl:WakeUpService">
   <wsdlsoap:binding style="rpc" transport=
     "http://schemas.xmlsoap.org/soap/http"/>
   <wsdl:operation name=
     "ERSP30ApproachLocation">
     <wsdlsoap:operation soapAction=
       "WakeUpService#ERSP30ApproachLocation"/>
   …
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="WakeUpServiceService">
    <wsdl:port name="WakeUpService"
       binding="impl:WakeUpServiceSoapBinding">
      <wsdlsoap:address location=
        "http://129.254.70.234/axis/WakeUpService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

In our test bed system, a BPEL4WS document is generated from input of service plan instance. It runs the distributed web services by INVOKE for performing real robot services. An implemented system environment of our test bed is represented in figure 3.

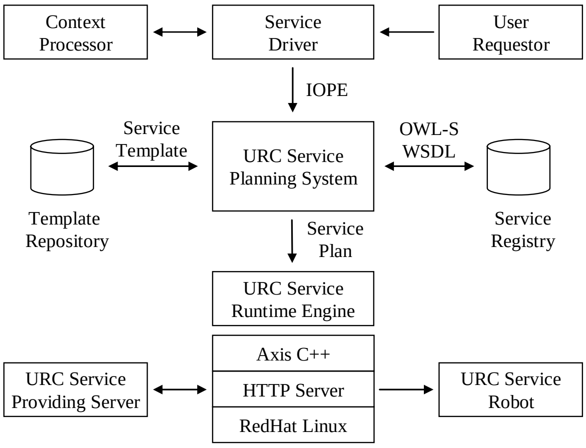


Figure 3: Our test bed system environment.

5 Conclusions

Semantic web services is proposing a new environment that provides intellectual and autonomous services of service agents base on semantic information of service that is described in ontology language. In this paper, we present a service template-based URC service planning system for intelligent web services in the semantic web services environments. The proposed system provides various functions that automatically select web services, and dynamically create and compose web service plans from selected web services. The generated service plans are transformed into the executable service plans, and the system provides real web services through the web service runtime engine.

6 References

[1] D.L. McGuinness and F. van Harmelen, “OWL Web Ontology Language Overview”, http://www.w3.org/TR/owl-features/, (2003).

[2] The OWL Services Coalition, “OWL-S: Semantic Markup for Web Services”, http://www.daml.org/services/owl-s/1.0/owl-s.html

[3] T. Berners-Lee, J. Hendler and O. Lassila, "The semantic web", *Scientific American*, 284(5), pp 34-43 (2001).

[4] W3C, "Web Services", http://www.w3.org/2002/ws/, visited on 2/6/2006.

[5] S. McIlraith, T.C. Son and H. Zeng, "Semantic web service", *IEEE Intelligent Systems*, 16(2), pp 46-53 (2001).

[6] R. Lara, H. Lausen, S. Arroyo, J. de Bruijn and D. Fensel, "Semantic web services: description requirements and current technologies", *Workshop on Semantic Web Services for Enterprise Application Integration and e-Commerce (SWSEE03)*, Pittsburgh, PA (2003).

[7] D. Martin, M. Burstein, O. Lassila, M. Paolucci, T. Payne and S. McIlraith, "Describing web services using OWL-S and WSDL", http://www.daml.org/services/owl-s/1.0/owl-s-wsdl.html

[8] IBM, "Business Process Execution Language for Web Services Version 1.1", http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

[9] Y. Cho and S. Oh, "Fusion of IT and RT: URC (ubiquitous robotic companion) program", *JRSJ*, 23(5), pp 22-25 (2005).

[10] Apache, "AXIS", http://ws.apache.org/axis/, visited on 2/6/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Tactile Sensing by the Sole of the Foot Part II: Calibration and Real-time Processing

Abhinav Kalamdani[1], Chris Messom[2], Mel Siegel[1]
[1] The Robotics Institute,
Carnegie Mellon University, Pittsburgh, USA
[2] IIMS, Massey University, Auckland, New Zealand.
mws@cmu.edu, c.h.messom@massey.ac.nz

Abstract

This paper introduces prototype experimental apparatus and the calibration and real-time signal processing required to investigate stability in standing, walking and running of humanoid robots using pressure sensing at the sole-of-the-foot contact. The system can provide very good spatial or temporal resolution and these can be traded off against each other dynamically to accommodate the instantaneous requirement, for example, sparsely sampling the whole sole during static balancing vs. densely sampling the impact region during walking or running. Dynamic variation in sampling policy during different phases of the gait is foreseen so as to optimise utilisation of the total sampling bandwidth available. Periodic signals like walking and running would be sampled repetitively, achieving by accumulation both high spatial and high temporal resolution.

Keywords: Tactile sensing, force/ pressure sensing, humanoid robots, real-time processing

1 Introduction

We aim to characterise and understand the role played by tactile sensing by the sole of the foot in stabilising bipedal standing, walking and running [1-3]. By measuring and understanding the spatial and temporal pressure patterns that are generated during the various phases of human and simulated-human standing and locomotion, we expect to learn how better to control these activities when they are attempted by humanoid robots [4-8] equipped with human-like sole-of-the-foot sensing capability. The literature on human balance control indicates that success will depend on incorporating the pressure sensing by the sole into the actuation algorithm [1-2]. Dynamic adaptation of the sensing and control system so as to customise what is measured spatially and temporally, and how actuation should be adjusted in response, appears to be crucial to bipedal balance for standing, walking, and running, possibly even more so for the former than the latter.

Our prototype apparatus, shown in figure 1a, and our initial experiments, illustrated in figure 1b, were introduced in [9]. These experiments are briefly summarised in the introductory sections of this paper. In the later sections of this paper we describe calibration procedures and issues, new experiments that demonstrate dynamic reallocation of sampling between the spatial and temporal domains, and signal accumulation and averaging over multiple cycles to achieve high resolution in both domains.

2 Apparatus

The apparatus allows static and dynamic loads to be applied to the system, as illustrated in the previously cited figures 1a and 1b. The sensor per se, shown in figure 2, is an x-y-addressed force-sensitive resistor array sold by TekScan[1] for medical diagnosis of foot problems. The sensors are inexpensive, but TekScan's monitoring hardware and software are both prohibitively expensive and generally ill-suited to the experimental scenarios we contemplate [10-17]. We therefore designed and built our own monitor. It uses analog current multiplexers controlled by a single board computer (SBC) to select which of the 960 tactile elements in the sensor array is monitored by the SBC's analog-to-digital converter at any instant.

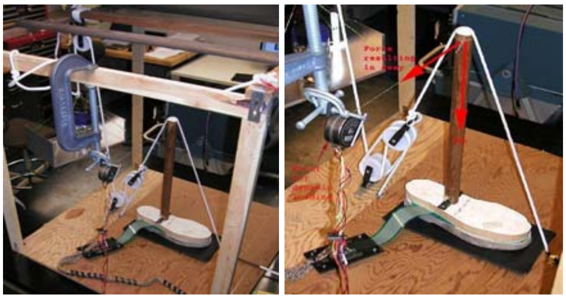


Figure 1: (a) (left) Overview of the apparatus; (b) (right) detail of the foot, leg, static and destabilising loading mechanisms.

Figures 3-6 show the physical and electrical features of the sensor. Individual tactile pressure-sensing elements are addressed via a left-right and front-back grid system, most easily seen in figure 3. The grid is centre-fed so the sensor can be trimmed to fit a shoe. The columns are split, allowing in principle twofold parallel access, but we have not implemented it.

On the SBC's low-level side it controls the analog current multiplexers - selecting the tactile element

[1] TekScan Corporation, South Boston MA, http://www.tekscan.com

desired at each instant - and on its high-level side it communicates with a PC via a serial link. The PC generates the scan sequence that the controller will execute - single tactile element, full raster scan of all tactile elements, low resolution raster scan of all tactile elements, foveal pattern scanning, etc. - so as to trade off the available spatial and temporal resolutions, whose product is limited by the sampling rate of the controller's analog-to-digital controller.

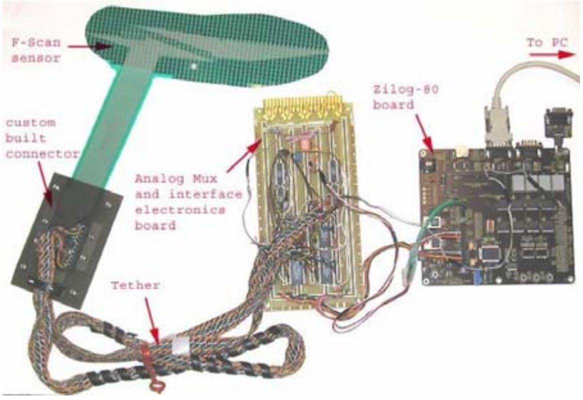


Figure 2: Sensor array, multiplexer, single-board controller.

The PC also supports the user interface, including implementation of the sampling policy either directly or by downloading a programme (which may include dynamic adaptation features), whatever analysis is applied to the sensor data before display, and a false-colour-mapped display of the pressure map.

In this prototype design, destabilising disturbances are generated by a stepper motor (figure 1b) - controlled by the PC's parallel port - that oscillates the tension in a string attached to the knee. The next step in our ongoing programme will be to reverse the sign: to analyse the dynamical behaviour of the pressure map so as to recognise and characterise disturbances and to generate control signals whereby, e.g., a pair of motors will restore stability after a disturbance.

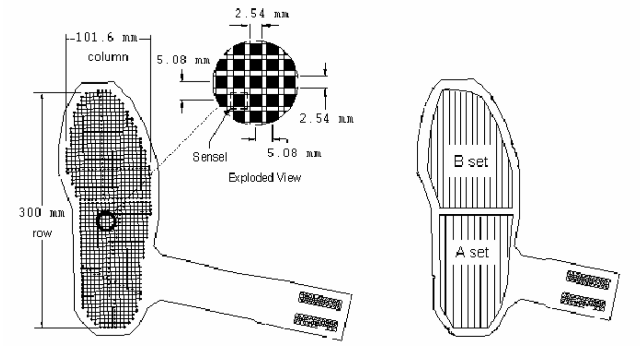


Figure 3: Dimensions and layout of tactile elements. Note that parallel scanning of two elements could be implemented in principle using two ADCs.

3 Calibration

The system is calibrated end-to-end by applying known forces to the whole sensor surface or to smaller groups of tactile elements and measuring the response over a range of input values. This end-to-end calibration allows us to model element-to-element variations due to manufacturing variations, but it does not account for effects such as hysteresis, which is significant in these sorts of pressure-sensitive devices. Nor does it account per se for problems like drift and temperature sensitivity, so the full calibration procedure addresses these separately.

The conductance vs. pressure of each tactile element is approximately linear in the mid- and high-ends of the operating range, but deviates significantly at the low-end, as illustrated in figure 4. The first stage amplifier receiving the current through the addressed tactile element is quasi-linear in the logarithm of the resistance of the addressed element, resulting in the convenient output characteristic seen in figure 5.

Figure 6 illustrate the extent of observed hysteresis and figure 7 illustrates the variation between two randomly selected tactile elements. The hysteresis is sufficiently large that there is no good reason to

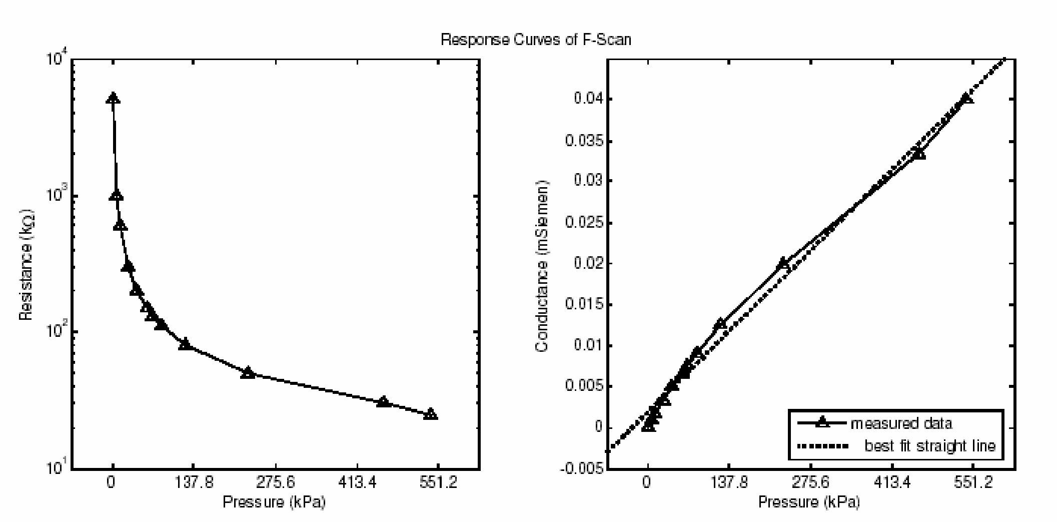


Figure 4: Response curves of a single sensor element.

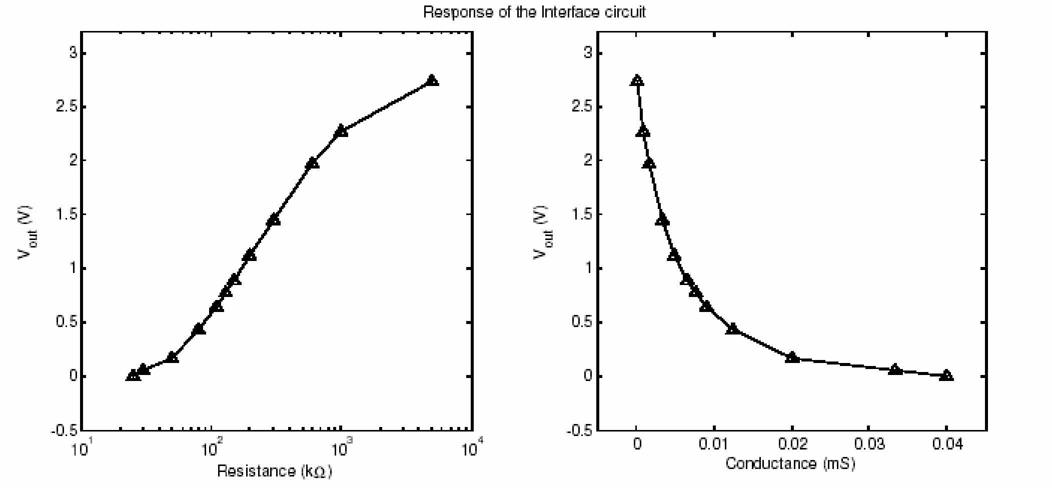


Figure 5: Response curves of interface circuit.

laboriously calibrate individual tactile elements. Figure 8 shows a rational function fit to the end-to-end calibration; we use this function to adequately relate the digital output of the SBC's analog-to-digital converter to the force applied to a tactile element. An absolute error might be as large as 30 kPa, but relative to the anticipated standing and ambulating pressures of a human or a near-human size hominoid robot, this is no more than 20%, which, after spatial and temporal averaging, is more than adequate for the intended applications.

4 Signal Processing

Figure 9 shows the impulse response of the tactile elements to a dropped rubber ball. Sampling of single tactile element can be completed in 0.3 ms, which allows the capture of fast signals, e.g., the impact of the heel strike in walking or running, or push-off with the toe over small regions of the sensor. Inasmuch as the appropriate small area of heel or toe region can be found iteratively over several cycles, this temporal capability seems quite adequate, though future experiments with humans or a running robot may make it necessary to rethink this.

To expand on this point, recall that when studying a periodic signal, such as a walking or running gait, high temporal and spatial resolution can be achieved by sampling and summing over several step periods. Potential difficulties with this approach include the variable period of actual walking and running, e.g., in response to terrain variations, or just fatigue in the human case, or its robotic equivalent as batteries run down. Variation in period can be largely corrected by marking the precise heel strike and toe push-off points and re-scaling time relative to these points in each cycle. Then multiple foot plant periods might be combined to provide an average high spatial and temporal resolution picture. Deviations of individual cycles from this average, i.e. residuals, would be indicative of effects like terrain-induced cycle-to-cycle gait variation and monotonic trends like fatigue.

4.1 Heel Strike and Toe Push-Off

Walking and running gaits include - by definition - a phase where one foot is in flight. During this phase the load on the sensor may be zero and may possibly even be negative - which would be seen as zero by the sensor we are using - depending on the accelerations and the nature of the foot-shoe contact. At the instant when the foot strikes the ground, the transient load increases significantly at the point of contact before being distributed over the foot. At lift-off at the end of the cycle the load may again increase as it is supported by a smaller area - the front of the foot - before finally it is precipitously reduced to zero as the foot leaves the ground.

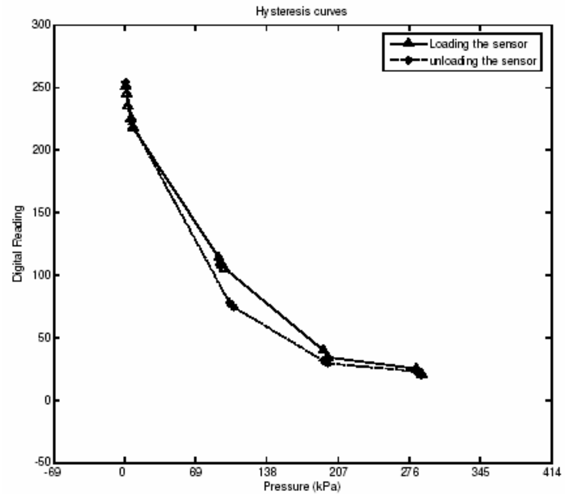


Figure 6: Hysteresis variation in tactile element.

To produce the high spatial and temporal resolution of the foot during the stance phase requires the start and end time of the stance to be located precisely. The sample time when sampling a single tactile element is about 0.3 ms, which is fast enough for most contemplated applications. But to achieve this sampling rate the point of contact must be known, so that sampling can focus on a small predetermined area. For idealised walking and running gaits the heel strike is the first instance of contact. This means that a known small number of tactile elements are loaded initially before the load is distributed to the foot. If fewer than 10 of these can be identified, then a scan with a total cycle time under 3 ms can be made, providing the required accuracy for finding the start of the stance phase. This "3 ms criterion" is obtained experimentally from the ball-bounce experiments illustrated by figure 9. It corresponds well in order of magnitude to estimates based on the human mass and spring stiffness of a typical running shoe. (It is well-known - and easily derived using only freshman physics - that the contact time of an idealised "bounce" is independent of impact speed. It is given quantitatively by half the oscillation period of the equivalent mass-spring system, i.e. half the square root of the ratio of spring constant to the mass.)

In a similar manner, the end of the stance phase can be precisely determined by locating the toe-off instant. But sampling to locate the end of the stance period is more challenging than for heel strike, as the sampling of the whole of the foot will be in progress - with consequent decreased temporal resolution - whilst the foot begins to lift off the ground. More favourably, during heel strike all the full sampling bandwidth can be dedicated to the heel-strike region, as the load then is essentially zero over the entire foot.

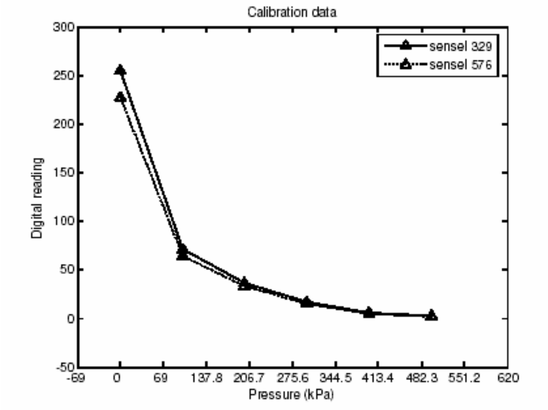


Figure 7: Variation in element-to-element calibration.

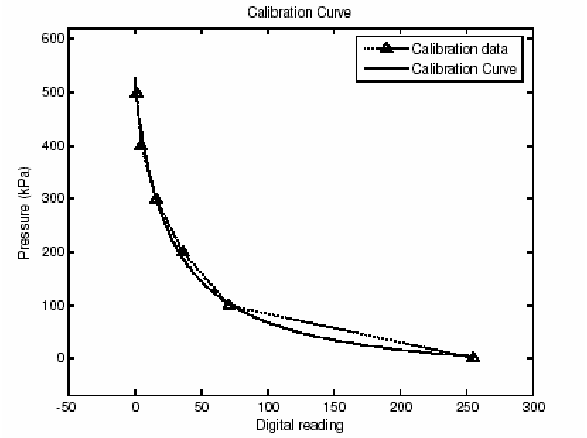


Figure 8: Calibration curve fit to a rational function (ax +b)/(x+c).

4.2 Sampling During Stance

Having determined the start- and end-points of the stance period, all stance data can be re-scaled as described above and the residuals examined to understand cycle-to-cycle differences due to e.g. terrain or fatigue.

As noted above, building a high spatial and temporal resolution view of the stance requires that the different regions of the foot be sampled for different relative times over different stance periods. A naïve approach would raster scan the array, adjusting the scan phase to be out of phase with the stance phases. But this is not optimal. During several phases of the stance large parts of the array are unloaded, as is seen in figures 10 and 11. These phases in which little interesting is happening can be exploited in the scanning strategy. If a low temporal and spatial resolution load pattern is known already, this can be used as a mask to direct sampling attention to the active regions only. This is shown in figure 12 for heel strike. If no initial load mask exists yet, it can be constructed in the first few strides of the gait. As the higher temporal and spatial resolution sampling is completed, a higher temporal and spatial resolution mask can also be applied to further reduce spending bandwidth sampling uninteresting regions.

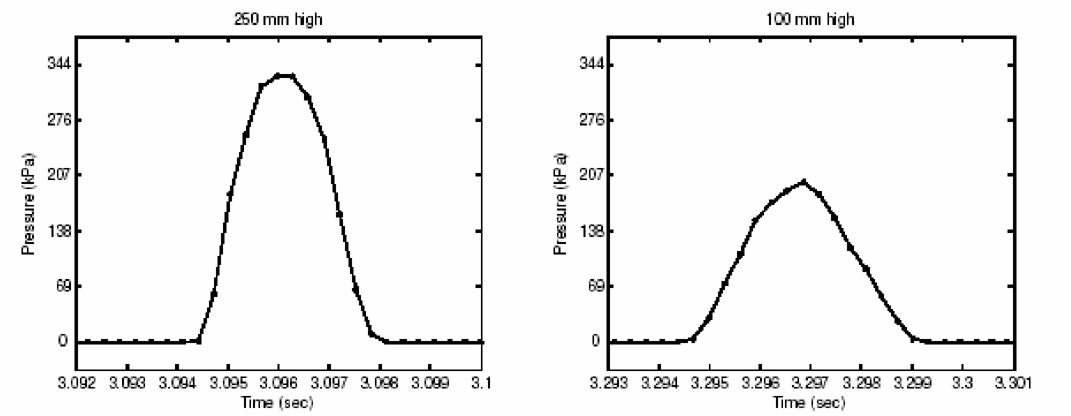


Figure 9: Impulse response to a rubber ball dropped from two different heights.

Variation in the duration of stance cycle will cause uncertainty that must be overcome by applying a mask with a conservative temporal extent. For example, if the stance period varies by 100 ms from cycle to cycle, then as the stance is near completion the uncertainty in the standardised time can be as large as 300 samples. The unloaded masks must be adjusted to ensure that requisite sample points are not overlooked under these circumstances. Depending on the statistical distribution of the actual stance periods, an aggressive mask - not too rapidly adjusted for the stance period delay - might still be applied on the assumption that any missed points will probably be seen during stance periods that are closer to the mean.

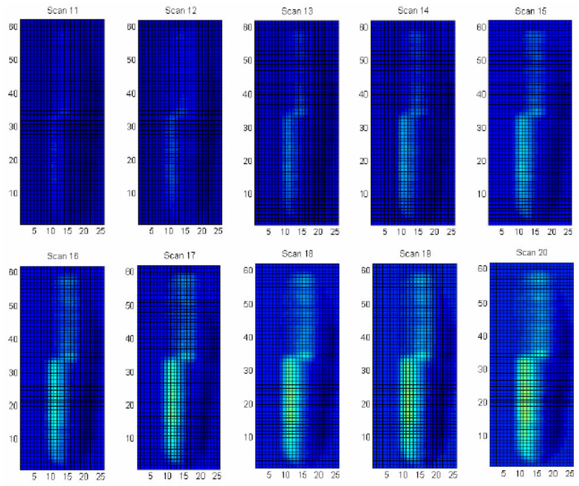


Figure 10: High resolution view of a lightly loaded foot with pressure changes during lateral sway. Note the large unloaded regions (dark blue).

As the high resolution sampling of the stance develops, current samples can be correlated with this reference so that an estimate of the current standardised time can be given. With the hardware limitations of the first-generation prototype it is unlikely that these strategies could be implemented well enough to actually improve performance significantly, but we are certain that given only time and money we could improve bandwidth at least up to the underlying limitations of the TekScan sensor without excessively straining our electronics acumen.

4.3 Force Image Processing

At present the first few moments of the pressure distributions seem to be entirely adequate for conveying the essential features of the data statically and dynamically. In future an image processing approach might be investigated for extracting higher spatial frequency information from the signals, e.g. walking and running surface tilt, unevenness and texture. Identifying these surface features might allow the control strategy to be adjusted for different surfaces. It might also help to identify the danger signals of incipient slip, incomplete foot support, etc.

An additional challenge that is introduced, if image processing techniques are adopted, is that each “pixel” is sampled at a different instant and the scan patterns are not generally progressive, so additional signal processing must be applied to obtain an inferred-time interpolated image before analysis. An advantage of a time-interpolated image will be that there is no inherent image frame time at which the image is taken, so the inferred-time interpolated image can be synthesised for any instant. The quality of the time-interpolated image will not depend on the instant selected, as the sampling occurs continuously and asynchronously, in contrast to the rigidly raster-scanned and framed signals from e.g. video cameras.

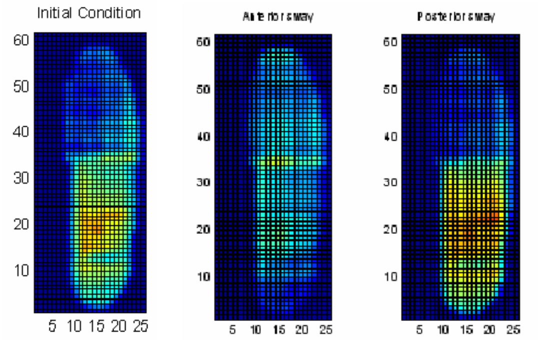


Figure 11: High resolution view of a heavily loaded foot with pressure changes during anterior and posterior sway. Posterior sway resembles a heel strike scenario. Note the large unloaded or lightly loaded regions (dark blue).

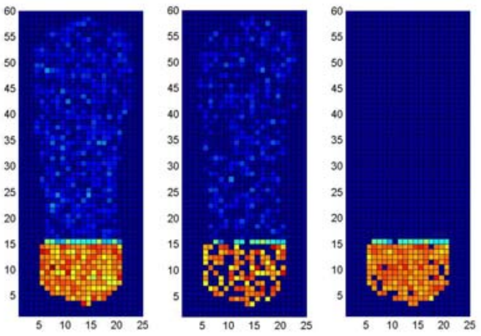


Figure 12: (a) (left) Simulated pressure profile during heel strike; (b) (centre) sampled force profile using random scanning; (c) (right) sampled force profile sampling only loaded region (right). Sampling only the loaded region during heel strike allows a high resolution force image to be captured quickly during a small number of steps.

5 Conclusions

This paper introduced a prototype apparatus for investigating the pressure distribution underfoot. The sensor system can be used to investigate balancing, walking and running. We expect future experiments will yield insight into the actuation and control mechanisms involved in human balancing, walking and running, and their implications for the design and control of humanoid bipedal walking robots.

The measurement system bandwidth dictates that either high spatial or high temporal resolution pressure images can be obtained at any instant. However, spatial and temporal resolution can be traded off dynamically, even on a timescale that allows adaptation inside the duration of a single step, to accommodate the requirements of any particular application. The quasi-periodic nature of walking and running allows both high spatial and high temporal resolution to be obtained by sampling over multiple cycles, thus overcoming the trade-off limitation.

Limitations of the 960 tactile element commercial transducer on top of which our sensor system is built include substantial element-to-element variation and substantial hysteresis. Their combined distortion can be as high as 20%. But software compensation is straightforward and, with clever real-time adapting sampling schemes implemented, there are more than enough elements to characterise the dynamics of the pressure distribution over the sole of the foot during standing, walking and running on any not-too-pathological surface.

6 References

[1] A. Kavounoudias, R. Roll and J.P. Roll, “The plantar sole is a ‘dynamometric map’ for human balance control”, *Cognitive Neuroscience Neuroreport*, 9(14), pp 3247-3252 (1998).

[2] C. Maurer, T. Mergner, B. Bolha and F. Hlavacka, “Human balance control during cutaneous stimulation of the plantar soles”, *Neurosci. Lett.*, 302(1), pp 45-8 (2001).

[3] D.C. Gravelle, C.A. Laughton, N.T. Dhruv, K.D. Katdare, J.B. Niemi, L.A. Lipsitz and J.J. Collins, “Noise-enhanced balance control in older adults”, *Somatosensory Systems, Pain Neuroreport*, 13(15), pp 1853-1856 (2002).

[4] A. Goswami, “Postural stability of biped robots and the foot-rotation indicator point”, *Int. Jour. of Rob. Res.*, 18(6), pp 523-533 (1999).

[5] S.H. Collins, M. Wisse and A. Ruina, “A three-dimensional passive-dynamic walking robot with two legs and knees”, *Int. Jour. Rob. Res.* 20(7), pp 607-615 (2001).

[6] P. Sardain and G. Bessonnet, “Forces acting on a biped robot, center of pressure-zero moment point”, *IEEE Trans. on Sys. Man and Cyb. - Part A: Sys. and Humans*, 34(5), pp 630-637 (2004).

[7] P. Sardain and G. Bessonnet, “Gait analysis of a human walker wearing robot feet as shoes”, *Proc. 2001 IEEE Int. Conf. on Rob. and Auto.*, Seoul, Korea, pp 2285-2292 (2001).

[8] T. Tanaka, H. Takeda, T. Izumi, S. Ino and T. Ifukube, “Effects on the location of the center of gravity and the foot pressure contribution to standing balance associated with ageing”, *Ergonomics*, 42(7), pp 997-1010 (1999).

[9] A. Kalamdani, C.H. Messom and M. Siegel, “Tactile sensing by the sole of the foot: part I: apparatus and initial experiments toward obtaining dynamic pressure maps useful for stabilizing standing, walking and running of humanoid robots”, *IEEE Int. Workshop on Haptic Audio Visual Environments and their Applications,* Ottawa, Canada (2006).

[10] F. Vecchi, C. Freschi, S. Micera, A.M. Sabatini, P. Dario and R. Sacchetti, “Experimental evaluation of two commercial force sensors for applications in biomechanics and motor control”, *5th Ann. Conf. of Int. FES*, Aalborg, Denmark, (2000).

[11] H. Zhu, N. Maalej, J.G. Webster, W.J. Tompkins, P.B.Y. Rita and J.J. Wertsch, “An umbilical data-acquisition system for measuring pressures between the foot and shoe”, *IEEE Trans. on Biomed. Eng.*, 37(9), pp 908-911 (1990).

[12] S.J. Morris, *A Shoe-Integrated Sensor System for Wireless Gait Analysis and Real-time Therapeutic Feedback*, DSc thesis, MIT (2004).

[13] Z.P. Luo, L.J. Berglund and K.N. An, “Validation of F-Scan pressure sensor system: a technical note”, *Jour. of Rehab. Res. and Dev.*, 35(2), pp186-191 (1998).

[14] T. Sumiya, Y. Suzuki, T. Kasahara and H. Ogata, “Sensing stability and dynamic response of the F-Scan in-shoe sensing system: a technical note”, *Jour. of Rehab. Res. and Dev.*, 35(2), pp 192-200 (1998).

[15] J. Woodburn and P.S. Helliwell, “Observations on the F-Scan in-shoe pressure measuring system”, *Clin. Biomech.*, 12(3), pp S16-S16(1997).

[16] E. Morin, S. Reid, J.M. Eklund, J. Stevenson and J.T. Bryant, “Comparison of 3 measures of ground reaction force: force plate, F-scan and multiple individual force sensors”, *Proceedings 4th World Congress of Biomechanics*, Calgary (2002).

[17] M. Lowe, A. King, E. Lovett, T. Papakostas, “Flexible tactile sensor technology: bringing haptics to life”, *Sensor Review*, 24(1), pp 33-36 (2004).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Modelling and FPGA-based Implementation of Graph Colouring Algorithms

Valery Sklyarov, Iouliia Skliarova, Bruno Pimentel
Department of Electronics, Telecommunications and Informatics, IEETA
University of Aveiro, Aveiro, Portugal
skl@det.ua.pt, iouliia@det.ua.pt, pimentel@ieeta.pt

Abstract

The paper discusses an effective matrix-based exact algorithm for graph colouring that is well-suited for implementation in FPGA. It is shown that the algorithm can also be used as a base for a number of approximate algorithms. Software models (described in C++) and hardware circuits (synthesised from Handel-C specifications) for the exact algorithm are discussed, analysed and compared. Characteristics and capabilities of the approximate algorithms are also examined. It is shown that such algorithms can be used for a wide range of practical applications, including robotics and embedded systems.

Keywords: graph colouring, Handel-C specification, FPGA implementation

1 Introduction

Graph colouring algorithms are widely used for solving different engineering problems in robotics and embedded systems. One of the most common examples is register allocation [1], where the compiler assigns intermediate computation values to storage units in an embedded processor. Another example can be found in [2] for memory-constrained networked embedded systems. One of the tasks, considered in [2] was scheduling the activities of so called ping nodes (actuators), which can be formulated as a distributed graph colouring problem. A colour corresponds to a specific time slot in which a ping node vibrates. Two adjacent nodes in the graph, each representing an actuator, cannot have the same colour since the vibrations from these actuators would then interfere with each other. The number of colours is therefore the length (in distinct time slots) of a schedule. The objective is to find a shortest schedule such that the ping nodes do not interfere with one another, in order to minimise damage detection and response times. A key open problem in mobile robotics is to find a way to program a mobile robot to explore an environment and visually determine its position (coordinates) [3]. A promising approach is to have the robot first determine an optimal set of visual landmarks for navigation, and then use the landmarks to find its position. The problem of finding the set of landmarks can be formulated as a graph colouring problem. Another potential scope is microprogramming for application-specific embedded microprocessors (ASEMP). Let M = {m_1,m_2,…m_N} be a set of micro-operations for an ASEMP. Consider a graph G, which has N vertices. Two vertices $m_i \in M$ and $m_j \in M$ are connected with an edge if and only if m_i and m_j can be executed in ASEMP at the same time. Colouring the graph G enables us to code the micro-instructions (composed of micro-operations from M) using the minimal number of fields. A similar problem has to be solved for resource distribution in parallel systems.

There are many more general problems. For example, it is known that any geographical map can be painted in four colours in such a way that any two countries having common boundary will be painted in different colours. This follows from the theorem of Haken-Appel [4,5] that any planar graph can be painted in the maximum of four colours. Many other practical applications of graph colouring problem can be found in [6,7].

The considered above different tasks can be formulated mathematically and solved through applying the same model and methods. This shows the importance of the graph colouring problem. It should be noted that the graph colouring is a very computationally complex task (this is NP-hard problem [5]). Often it is necessary to solve this problem in run-time (for example, for register allocation [1]), which increases essentially the total execution time for the relevant applications. However, acceleration can be achieved with the aid of graph-colouring targeted hardware [8,9]. This makes possible the number of the required clock cycles for the respective applications to be decreased significantly.

The paper suggests algorithms for solving the graph colouring problem in FPGA. To compare the results two different models have been explored:

- C++ projects running on general purpose computers;
- Handel-C based projects for FPGA.

It is known [10] that exhaustive search (complete enumeration) for the considered problem requires N^K iterations, where N is the number of vertices in a graph and K is the number of colours. Thus, this might involve significant computational resources. In order to find out a compromise between the execution time and the required accuracy two types of algorithms have been implemented and analysed. They are an exact algorithm producing the optimal solution and approximate algorithms enabling us to stop the execution as soon as an appropriate for the considered problem solution has been found.

2 Matrix Specification

Binary and ternary matrices are very well suited for processing them in FPGAs [11]. This section demonstrates that any graph, that has to be coloured, can be converted to matrix specification in such a way that solving the problem over the matrix is equivalent to solving the problem over the graph. The graph [10] in figure 1 will be used as an example to explain all the steps for the conversion.

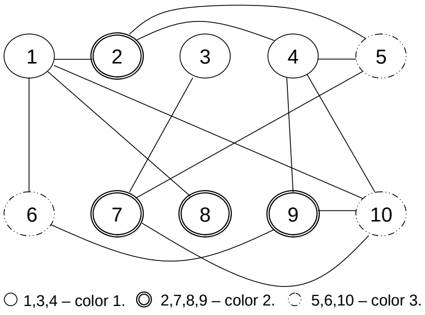


Figure 1: Graph G and its optimal colouring.

For solving the graph-colouring problem it is required to paint graph nodes in such a way that: any two nodes connected with an edge are painted in different colours and the number of used colours is minimal.

There are several optimal solutions for the considered graph (with the minimum number of colours, 3) and one of them is shown in figure 1: {{1,3,4},{2,7,8,9},{5,6,10}}.

Let us consider a matrix, which has the same number of rows as the number N of vertices of the graph G. If and only if two vertices m_i and m_j are connected with an edge in G then the matrix rows i and j must be orthogonal. Orthogonality of two ternary vectors is defined as follows [10]:

$(m_i \text{ ort } m_j) \Rightarrow \{m_i\} \cap \{m_j\} = \varnothing$,

where $\{m_i\}$ ($\{m_j\}$) is a set of binary vectors that correspond to the ternary vector m_i (m_j) by replacing the don't care values (-) with all possible combinations of 0s and 1s. For example, the following two ternary vectors 0-11- and -1110 are not orthogonal, because {00110,00111,01110,01111} ∩ {01110,11110} = {01110} ≠ ∅.

If and only if two ternary vectors are not orthogonal (let us designate this as *m_i ~~ort~~ m_j*) they do intersect in the Boolean space:

(m_i ~~ort~~ m_j) ⇔ (m_i ins m_j).

Table 1 depicts matrix μ(G) that was built for the graph G in figure 1. This is a symmetric matrix with respect to the main diagonal that is why just a lower triangle of the matrix is sufficient. The second (upper) part might contain all don't care values because the first (lower) part provides all the necessary and sufficient conditions.

Table 1: Matrix μ(G) with rows 1, 2,…,10 and columns I, II,…,VIII.

	I	II	III	IV	V	VI	VII	VIII
1	0	-	-	-	-	-	-	-
2	1	0	-	-	-	-	-	-
3	-	-	0	-	-	-	-	-
4	-	1	-	0	-	-	-	-
5	-	1	-	1	0	-	-	-
6	1	-	-	-	-	0	-	-
7	-	-	1	-	1	-	0	-
8	1	-	-	-	-	-	-	-
9	-	-	-	1	-	1	-	0
10	1	-	-	1	-	-	1	1

The matrix was built using the following rules:

Insert the first row corresponding to the vertex 1 (see figure 1), which has the first element zero and all the remaining elements equal to *don't care* (note that at the beginning we do not know the exact number of remaining don't cares, but this does not give rise to any problem);

Provide orthogonality of the 1st vertex with the vertices connected to the 1st vertex by edges. For our example they are 2, 6, 8 and 10 (see figure 1). It is done by recording the value "1" in the first column (I) of the rows 2,6,8 and 10 of table 1. All empty elements of the first column are assigned to *don't care* ("-"). Correctness of these operations is evident from the definition of orthogonality;

If the vertex 2 has connections with succeeding vertices, i.e. with the vertices 3,4,5,… then use column 2 to indicate orthogonality; otherwise, skip vertex 2 and consider the next one. For our example the vertex 2 is connected with the succeeding vertices 4 and 5. Thus the column II contains "1"s in the rows 4 and 5 and "0" in the row 2.

Use the same rules for all the remaining vertices. For our example it permits to fill in table 1.

Obviously, the maximum number of columns for the considered matrix μ(G) cannot exceed the value N-1, where N is the number of vertices in the graph G. The method of conversion considered above makes possible to formulate the graph colouring problem over a matrix μ(G). Then it is necessary to find out the minimal number K of such subsets of rows that any subset does not contain mutually orthogonal rows. In other words, any two of the rows belonging to the same subset must intersect in the Boolean space. The number K will be the minimal number of colours for graph G and rows from each subset will correspond to the vertices of graph G that can be painted in the same colour.

For example, the following sets give a feasible solution: {{1,3,4}, {2,6,7,8},{5,9},{10}}. However K = 4 > 3 and this is a non-optimal solution. Let as consider any subset, for example the first one {1,3,4}:

```
0 - - - - - - -
- - 0 - - - - -
- 1 - 0 - - - -
```

Any two vectors of this subset do intersect and the result of intersection for all three vectors is: 0 1 0 0 - - - - . It is important to note that the considered above set cannot be expanded. If we add any of the remaining rows from table 1 the requirements will be violated. In other words, the vector 0 1 0 0 - - - - is orthogonal to all the remaining vectors (i.e. to rows 2,5,6,7,8,9,10) of table 1.

3 The Exact Algorithm

We have already mentioned that exhaustive search algorithms for the considered problem are very time consuming. They are not efficient and sometimes even cannot be used at all due to time limits. The considered exact backtracking search algorithm is based on the following four steps that are common for combinatorial search methods [10] and that are repeated sequentially until the solution is found.

1. *Reduction*. The matrix is reduced as much as possible applying some pre-established rules.
2. *Splitting*. The problem is decomposed in less complicated sub-problems.
3. *Termination*. The current step is terminated as soon as a new incomplete solution is of the same quality (i.e. contains the same number of colours) as any previous solution that has already been found.
4. *Search for the optimal result*. The steps 1-3 have to be repeated until all possible solutions have been implicitly examined.

Some of the reduction and splitting rules can be directly borrowed from the method of condensation, proposed in [10].

The algorithm will be demonstrated on the example of graph G in figure 1 and the respective matrix μ(G) in table 1. The following reduction rules will be used:

1. If, after selecting a new colour, the matrix μ(G) contains a column *i* without values 0 (1) then this column *i* can be deleted;
2. If, at any intermediate step of the algorithm, the matrix μ(G) contains a row *j* with just don’t cares (“-“) then this row *j* can be deleted from the matrix and included in the constructed subset;
3. All the rows that have already been included in the constructed subsets are removed from the matrix μ(G).

Figure 2 depicts a part of the search tree, which will be built sequentially in accordance with the following steps:

1. Choose a new colour (i.e. create a new initially empty subset);
2. Apply the reduction rules;
3. Consider the topmost row m_i in the matrix;
4. Include the row m_i in the constructed subset and delete it from the matrix;
5. Find out all other rows intersecting (i.e. not orthogonal) with the vector m_i;
6. Select the first not tried yet row m_j from point 5, include it in the constructed subset and then delete it from the matrix;
7. Assign $m_i = m_i$ *ins* m_j and repeat the steps 5-7 if this is possible. If this is not possible go to the step 8;
8. If the intermediate matrix is not empty repeat the steps 1-7. Otherwise, store the solution found and then backtrack to the nearest branching point (set at the step 6 and try to find a better solution by repeating the steps 6-8.

Let us consider how to discover an exact solution for the graph depicted in figure 1 and the relevant matrix μ(G) shown in table 1. The subsequent list contains all the required operations. The first level of headings, such as 1, 2,… indicate the numbers of subsets of vertices (rows) that are being constructed. The second level of headings, such as 1.1, 1.2,…,2.1… indicate the sequence of operations needed to construct any subset of vertices (rows) that have to be painted in the same colour.

The list is given below.

1. Executing operations for discovering the first subset of vertices (rows of table 1) that have to be painted in the first colour.

1.1. The reduction rules cannot be applied.

1.2. Selecting the first row (1) of table 1. This row forms the root of the search tree or the level 0 (see figure 2).

1.3. Removing the row 1 from the matrix μ(G) and including this row in the first subset $S_1 = \{1\}$.

1.4. Building the first group of nodes of the tree that correspond to such rows of table 1 that are compatible with the row 1. The row m_j is compatible with the row 1 if and only if the condition m_1 *ins* m_j is satisfied. Since all the rows 3, 4, 5, 7, 9 are compatible with the row 1 they form the first level of the tree in figure 2.

1.5. Selecting the first vertex (row) from the first level. Let us agree to order vertices of the same level from the left (the smallest number) to the right (the greatest number). Thus, the row 3 is selected, included in S_1 ($S_1 = \{1,3\}$) and then deleted from the matrix $\mu(G)$.

1.6. Building the second group of nodes of the tree that correspond to such rows of table 1 that are compatible with the vector (*1 ins 3 = 0------- ins --0----- = 0-0-----*). Since the rows 4, 5, 9 are compatible with the vector (*1 ins 3*) they form the second level of the tree in figure 2 and $S_1 = \{1,3,4\}$. The row 4 is deleted from the matrix.

1.7. Selecting the first vertex (row) from the second level and repeating the same operations (forming the 3d, the 4th, etc. levels) while this is possible. For our example, the vector (*1 ins 3 ins 4*) is not compatible with the remaining rows of table 1, thus, the first subset is $S_1 = \{1,3,4\}$. Note that later this subset might be changed.

1.8. The resulting matrix $\mu(G)\backslash S_1$ (where all the rows included in S_1 have been deleted) is shown in table 2.

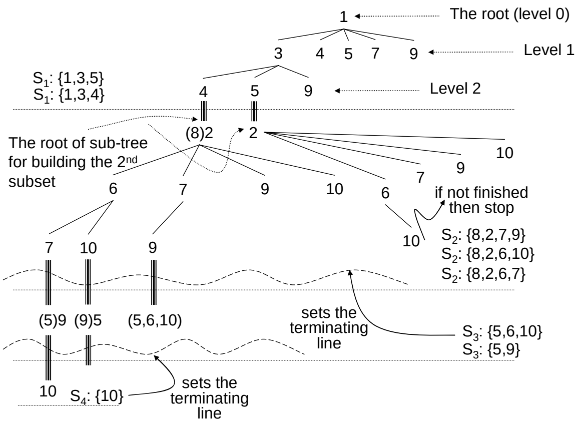


Figure 2: Execution steps of the exact algorithm.

2. Executing operations for discovering the second subset of vertices (rows of table 1) that have to be painted in the second colour.

2.1. Applying the reduction rules. The rule 1 can be applied to columns I, III, IV and then the rule 2 can be applied to row 8. The resulting matrix is shown in table 3.

2.2. Executing operations that correspond to points 1.2-1.8 above. As a result, the subset S_2 will be built ($S_2 = \{8,2,6,7\}$ – see figure 2). The resulting matrix $\mu(G)\backslash(S_1 \cup S_2)$ is shown in table 4.

3. Executing operations for discovering the third subset of vertices (rows of table 1) that have to be painted in the third colour. Finally, $S_3 = \{5,9\}$ – see figure 2.

4. Executing operations for discovering the fourth subset of vertices (rows of table 1) that have to be painted in the fourth colour. Finally $S_4 = \{10\}$ – see figure 2. After this step the resulting matrix does not contain any row and we have got the first solution (S_1, S_2, S_3, S_4) and K=4 colours.

Table 2: The matrix $\mu(G)\backslash S_1$.

	I	II	III	IV	V	VI	VII	VIII
2	1	0	-	-	-	-	-	-
5	-	1	-	1	0	-	-	-
6	1	-	-	-	-	0	-	-
7	-	-	1	-	1	-	0	-
8	1	-	-	-	-	-	-	-
9	-	-	-	1	-	1	-	0
10	1	-	-	1	-	-	1	1

Table 3: The matrix $\mu(G)\backslash S_1$ after applying the reduction rules.

	II	V	VI	VII	VIII
2	0	-	-	-	-
5	1	0	-	-	-
6	-	-	0	-	-
7	-	1	-	0	-
9	-	-	1	-	0
10	-	-	-	1	1

Table 4: The matrix $\mu(G)\backslash (S_1 \cup S_2)$.

	II	V	VI	VII	VIII
5	1	0	-	-	-
9	-	-	1	-	0
10	-	-	-	1	1

Now we will use the value K as a constraint for future steps. In other words, if the reduced matrix is not empty and at least K-1 colours have already been used we have to stop forward propagation steps and backtrack to the nearest branching point in order to try to find out a better solution. Thus, the first solution (S_1, S_2, S_3, S_4) for our example sets the terminating line shown in figure 2. For our example the nearest branching point is 6 (see figure 2). Now we can construct a new set $S_2 = \{8,2,6,10\}$ for the level 2 (see figure 2). For the next step the reduced matrix contains three rows (5,7,9). After constructing a new set $S_3 = \{9,5\}$ for level 3 the matrix is not empty. Thus, not less than 4 colours will be needed and we have to: 1) stop forward propagation; 2) backtrack to the row 2; and 3) select the first row (this is 7 – see figure 2) after the row 6 that has already been considered. Using this tree pruning method together with the reduction rules enables the total number of visited nodes in the search tree to be decreased significantly.

The subsequent steps are shown in the list below:

2. Executing operations for discovering the second subset of vertices (rows of table 1) that have to be painted in the second colour. Finally, $S_2 = \{8,2,7,9\}$ (see figure 2).

3. Executing operations for discovering the third subset of vertices (rows of table 1) that have to be painted in the third colour. Finally $S_3 = \{5,6,10\}$

and the reduced matrix is empty. Thus, (S_1, S_2, S_3) is a better solution and K=3 colour.

Now the constraint for the future steps is changed to 3. Thus, the new terminating line in figure 2 is being set up. It is easy to check that the subsequent steps do not permit to improve the solution.

4 Derived Approximate Algorithms

The considered algorithm possesses the following primary advantage. Beginning from the first iteration we have got a solution of the problem. As a rule, this solution is not far away from the optimal solution and therefore might be appropriate for some practical applications. It is important that any new iteration does not make worse the first result and it can only improve it. Thus, either the number of iterations might be limited or any intermediate complete result can be checked for adequacy. This permits to suggest a number of approximate algorithms derived from the exact algorithm. It is very interesting to estimate how the optimal result depends on the number of iterations; how far is the first result from the optimal result, etc. The details of relevant experiments for real-world and randomly generated graphs will be reported in section 6.

5 General Description of FPGA-targeted Projects

Initially, the exact algorithm for graph colouring was described in C++ language and carefully debugged and tested in PC. Then C++ code was converted to a Handel-C project using the rules established by *Celoxica* [12]. The basic blocks of the hardware project are shown in figure 3.

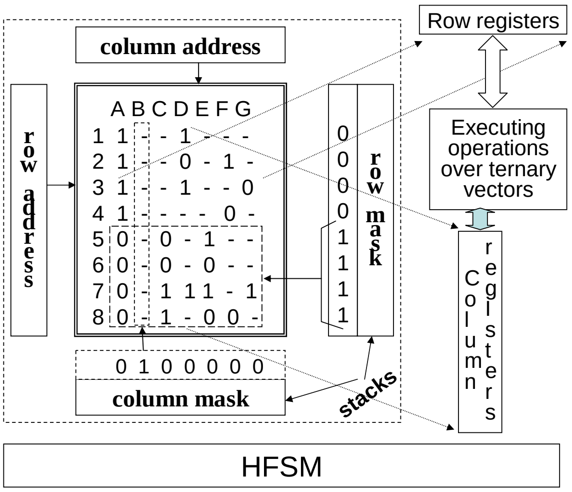


Figure 3: General structure of the hardware project for solving the graph colouring problem.

The initial and all the intermediate matrices are kept in the same RAM (constructed from embedded in FPGA memory blocks). Mask registers (for columns and for rows) make possible some rows and columns to be excluded from the matrix in order to select a minor (i.e. to specify the reduced matrix). The column/row address registers provide dual access allowing either a row or a column to be read (to a column/row register). The sequence of the required operations is generated by a hierarchical finite state machine [13].

6 Experiments and Comparison

The synthesis and implementation of circuits from the specification in Handel-C was done in Celoxica DK4 design suite [12] and implementation was finalised in Xilinx ISE 8.1 for xc2v1000-4fg456 FPGA (Virtex-II family of Xilinx [14]) available on RC200 prototyping board [12]. Note that each circuit includes not only components that are needed for problem analysis, but also auxiliary blocks for entering input data and visualising the results.

The results of experiments are presented in table 5 for the exact algorithm and in table 6 for approximate algorithms.

Table 5: The results for the exact algorithm.

P(K)	N	L	N_s	F_h	N_{clk}^h	N_{clk}^s
1(4)	32	53	3481	26	587103122	$7.3568*10^9$
2(3)	18	21	4028		166755	$3.99*10^6$
3(4)	28	50	4013		102509429	$1.77*10^9$
4(3)	18	12	4028		24122	$0.426*10^6$
5(8)	32	258	3481		97356959	$2.627*10^9$
6(7)	32	241	3481		122103423	$2.534*10^9$

Table 6: The results for approximate algorithms.

P(K)	N	L	N_{clk}^h	N_{clk}^s	K_a	R
1(4)	32	53	24770	$0.41 * 10^6$	4	1
2(3)	18	21	4885	$0.12 * 10^6$	3	
3(4)	28	50	17612	$0.31 * 10^6$	4	
4(3)	18	12	3861	$0.067 * 10^6$	3	
5(8)	32	258	20388	$0.506 * 10^6$	10	
6(7)	32	241	21578	$0.53 * 10^6$	10	
5(8)	32	258	25877	$1.04 * 10^6$	9	2
6(7)	32	241	24803	$0.625 * 10^6$	9	
5(8)	32	258	2680182	$1.11 * 10^9$	8	3
6(7)	32	241	3318341	$72.67 * 10^6$	8	
6(7)	32	241	106231355	$2.24 * 10^9$	7	4

Rows/columns of the tables contain the following data: P(K) – the number of problem instance, which can be optimally painted with K colours, N – the number of graph vertices; L – the number of graph edges; N_s – the number of occupied FPGA slices; F_h – the resulting circuit clock frequency in MHz; N_{clk}^s – the number of clock cycles required for solving a given problem instance in software; N_{clk}^h – the number of clock cycles required for solving a given problem instance in hardware; K_a – the number of colours obtained by the approximate algorithm, R – the number of respective iterations for K_a colours.

The first problem (1) is the colouring of central and western European map. The next three problems (2-4) are encodings of micro-operations in microinstructions. The last four examples (5,6) were generated randomly.

We can summarise the results of experiments and analysis of alternative realisations as follows:

1. In average the FPGA-based implementations require one order of magnitude less clock cycles comparing to software implementations.
2. Approximate algorithms make possible to obtain either optimal or near-optimal results after just a few iterations. Thus, they are very well-suited for many practical applications. Besides, they enable us to reduce very significantly the execution time (up to four orders of magnitude, which can easily be calculated comparing data from tables 5 and 6).
3. The analysis of execution time has shown that approximate algorithms are very well suited for run-time solution of the graph colouring problem.
4. Preliminary tests of the considered algorithms in hardware description language based projects (VHDL, in particular) show that the number of FPGA slices can be significantly reduced.
5. The considered projects are easily scalable making possible to customise the required complexity and to link the projects with embedded systems for robotics working in real-time environment.
6. The used architecture (see figure 3) can easily be retargeted to solving some other combinatorial problems, such as matrix covering or Boolean satisfiability [11].

7 Conclusion

The first important contribution of the paper is the presentation of the known graph colouring problem in such a way that makes possible easy implementation of projects for solving this problem in commercially-available FPGAs. The second contribution is the analysis which permits to conclude that FPGA-based projects are more advantageous than relevant software projects executing on general-purpose computers (in terms of the number of clock cycles). The remaining significant issues considered in the paper are the exact algorithm for graph colouring and the derived approximate algorithms, which can be used for solving numerous engineering problems in the scope of robotics and embedded systems.

8 Acknowledgements

The authors would like to acknowledge Ivor Horton for his very useful comments and suggestions.

9 References

[1] G. Goossens, J. van Praet, D. Lanneer, W. Geurts, A. Kifli, C. Liem and P.G. Paulin, “Embedded software in real-time signal processing systems: design technologies”, *Proceedings of the IEEE*, 85(3), pp 436-454 (1997).

[2] V. Subramonian, H.M. Huang, G. Xing, C. Gill, C. Lu and R. Cytron, “Middleware specialization for memory-constrained networked embedded systems”, http://www.cs.wustl.edu/~venkita/publications/rtsj_norb.pdf, visited on 25/09/2006.

[3] J. Ezick, “Robotics”, http://dimacs.rutgers.edu/REU/1996/ezick.html, visited on 25/09/2006.

[4] K.H. Rosen (ed.), *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, USA (2000).

[5] R. Thomas, “The four color theorem”, http://www.math.gatech.edu/%7Ethomas/FC/fourcolor.html, visited on 25/09/2006.

[6] J. Culberson, “Graph coloring page”, http://www.cs.ualberta.ca/~joe/Coloring/index.html, visited on 25/09/2006.

[7] Y.L. Wu and M. Marek-Sadowska, “Graph based analysis of FPGA routing”, *Proceedings of the European Design Automation Conference*, Germany, pp 104-109 (1993).

[8] T.K. Lee, P.H.W. Leong, K.H. Lee, K.T. Chan, S.K. Hui, H.K. Yeung, M.F. Lo and J.H.M. Lee, “An FPGA implementation of GENNET for solving graph coloring problems”, *Proc. of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, USA, pp 284-285 (1998).

[9] L.M. Pochet, M. Linderman, R. Kohler and S. Drager, “An FPGA based graph coloring accelerator”, http://klabs.org/richcontent/MAPLDCon00/Papers/Session_A/A2_Pochet_P.pdf, visited on 25/09/2006.

[10] A.D. Zakrevskij, *Logical Synthesis of Cascade Networks*, Science, Moscow (1981).

[11] I. Skliarova, *Reconfigurable Architectures for Combinatorial Optimization Problems*, PhD Thesis, University of Aveiro, Portugal (2004).

[12] Celoxica, “Celoxica Products”, http://www.celoxica.com, visited on 25/09/2006.

[13] V. Sklyarov, “FPGA-based implementation of recursive algorithms,” *Microprocessors and Microsystems, Special Issue on FPGAs: Applications and Designs*, 28(5-6), pp 197–211 (2004).

[14] Xilinx, “Xilinx Products”, http://www.xilinx.com, visited on 25/09/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Using Camera Tilt to Assist with Localisation

Trevor Taylor, Shlomo Geva, Wageeh W. Boles
Queensland University of Technology
Brisbane, Australia
t.taylor@qut.edu.au, s.geva@qut.edu.au, w.boles@qut.edu.au

Abstract

Digital cameras are often used on robots these days. One of the common limitations of these cameras is a relatively small field of view. Consequently, the camera is usually tilted downwards in order to see the floor immediately in front of the robot. With the camera tilted, vertical edges no longer appear vertical in the image. This feature can however be used to advantage to discriminate amongst straight line edges extracted from the image when searching for landmarks. It might also be used to estimate angles of rotation and distances moved between successive images in order to assist with localisation.

Keywords: computer vision, monocular camera, camera tilt, perspective, visual localisation

1 Introduction

The price of digital cameras has plummeted in recent years due to the popularity of web cameras and cameras in mobile phones. This has provided a useful sensor for robots.

These cheap cameras often have poor optics and a limited field of view (FOV) – as little as 40–60°. This is a problem if the robot needs to use the camera to locate obstacles in its path because the camera needs to be tilted downwards. One side-effect is that vertical edges no longer appear vertical in the image.

This paper addresses the issue of camera tilt and attempts to draw some positive benefit from it. The term “tilt” in this context means angling the camera downwards towards the floor so that the principal axis of the camera is below the horizontal. This might also be referred to as pitch (as opposed to yaw or roll) and sometimes the camera azimuth angle.

There are alternatives to using a single camera with a small FOV in order to avoid camera tilt, such as stereo vision, wide-angle lenses or even panoramic or omni-directional cameras. However, these solutions are more expensive and introduce the problems of complex calibration and/or dewarping of the image. Therefore, we persist with a single camera.

In this paper, the term “vertical” is used to refer to any edge of a real world object that is vertical, e.g. a door frame or the corner where two walls meet. This is distinct from a “horizontal” which refers to an edge that is horizontal in the real world, e.g. the skirting board where a wall meets the floor. These vertical and horizontal edges do not usually appear as such in a camera image, especially when the camera is tilted.

Note that for a robot moving in two dimensions on a horizontal ground plane, i.e. the floor, the location of a vertical can be specified using only two coordinates.

1.1 Related Work

Our objective is for a robot to navigate in an indoor environment using only monocular vision and build a map as it goes. Several algorithms have emerged in recent years for simultaneous localisation and mapping (SLAM) using vision.

In this paper we do not present a complete visual SLAM algorithm, however we outline the theoretical underpinnings of a new approach to localisation that takes advantage of camera tilt.

The SIFT (scale-invariant feature transform) algorithm has been used for visual SLAM by several researchers, including the original developers [1]. Because SIFT features have distinctive signatures, no attempt is made to extract lines from images.

In [2], the approach is also to identify significant features in images and then determine location using an image retrieval procedure. To make this robust, they incorporate Monte Carlo localisation. Although they rely on an algorithm for calculating the signature of selected image features, they also note that features such as lines are useful in general.

Conversely, a structure from motion (SFM) approach has also been used. This relies on developing a model of the real world by solving sets of equations based on point correspondences derived from multiple images. Davison used this approach, and more recently has enhanced his original work by using a wide-angle lens [3]. The key advantage claimed is that the wider field of view allows landmarks to remain in view for longer and therefore contribute to more matches. However, it is pointed out that a special model is required for the camera because it does not conform to the conventional perspective projection.

Eade and Drummond [4] apply the particle filter approach of FastSLAM to visual SLAM using a single camera. As in the previous methods, the selected features are based solely on how distinctive they are, not on their significance in the real world.

Visual odometry has been demonstrated using a commercial web camera [5]. This relies basically on optic flow. The intention in this case is obstacle avoidance and reliable odometry, but not the repeatable identification of landmarks. Therefore the algorithm does not extract lines.

We take a geometric approach to processing images to identify vertical edges. These edges are important because they represent significant features in the environment. This approach follows the seminal work of Faugeras [6] on 3D vision. However, in his work he used matrix notation and dealt with the general case. We have taken specific cases with simple solutions that seem to have gone unnoticed.

1.2 Background

By convention, the coordinate system used for a camera has its origin at the focal point of the camera. Figure 1 shows the side view of a robot with a camera mounted on top of it. The *X*-axis is into the page and the negative *Z*-axis is along the camera's principal axis (for a right-hand coordinate system).

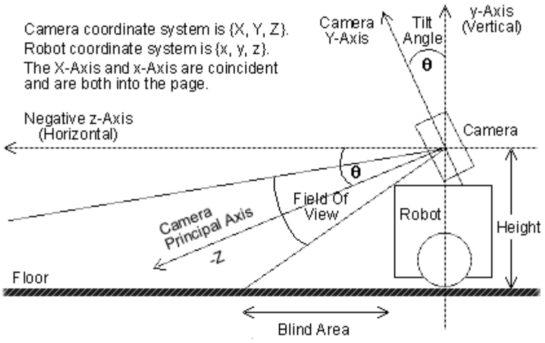

Camera coordinate system is {X, Y, Z}.
Robot coordinate system is {x, y, z}.
The X-Axis and x-Axis are coincident and are both into the page.
Camera Y-Axis
Tilt Angle
θ
y-Axis (Vertical)
Negative z-Axis (Horizontal)
Camera
θ
Field Of View
Robot
Height
Camera Principal Axis
-Z
Floor
Blind Area

Figure 1: Camera coordinate system.

The diagram shows that the camera is tilted. It also shows the vertical field of view. (The horizontal FOV can be calculated based on the aspect ratio of the camera.) Notice that there is a blind area in front of the robot. The choice of camera tilt angle is therefore a compromise between visible range and being unable to see small obstacles in the immediate vicinity.

The robot coordinate system {*x, y, z*} has its origin at the same point as the camera coordinate system {*X, Y, Z*}. It would make more sense to have the origin of the robot coordinates on the floor, but only the *x* and *z* coordinates are of interest because the robot moves in two dimensions. Therefore, the height off the floor (in the *y* direction) is not important except to the extent that it affects the size of the blind area.

Ignore the camera tilt for now. Start with the standard pin-hole camera equations. (Refer to a textbook on computer vision, such as [7], p20.)

$$u = \frac{fX}{Z}, \quad v = \frac{fY}{Z} \tag{1}$$

where (*u, v*) are the coordinates of a pixel in the image (assuming the image origin is at the centre) of a real-world point (*X, Y, Z*) expressed in the camera coordinate system, and *f* is the camera focal length.

By using an edge detector and a line detector, e.g. the Canny Edge Detector and Probabilistic Hough Transform in the Intel OpenCV Library [8], it is possible to obtain a set of lines from an image.

The slope of any given line in an image (which is a 2D space) can be obtained from two points on the line, (u_1, v_1) and (u_2, v_2), as follows:

$$m = \frac{v_2 - v_1}{u_2 - u_1} \tag{2}$$

where *m* is the slope of the line.

If the actual angle is required, then this can be calculated as arctan(*m*). Note that the nature of the tangent function is such that *m* approaches infinity as the line approaches vertical, which can create computational problems. An alternative would be to use the inverse slope, which would move the singularity to lines of zero slope – well outside our range of interest.

Substitute into equation (2) after using equation (1) to determine the image coordinates for two real-world points, p_1 and p_2. After a small amount of simplification the result is:

$$m = \frac{Y_2 Z_1 - Y_1 Z_2}{X_2 Z_1 - X_1 Z_2} \tag{3}$$

This equation describes the slope of a line between two points as it appears in the image.

It is worth noting at this stage that the focal length has disappeared from the equation. Also, it is irrelevant which point is chosen as p_1.

2 Finding Vertical Edges

Vertical edges are important as landmarks, especially in man-made environments. These features can be useful for localisation of a robot after small motions.

For a camera that is not tilted, finding vertical edges is trivial because the edges are vertical in the image. However, for a tilted camera this is no longer the case. Furthermore, the slope of the "vertical" edge in the image is dependent on where the edge is physically located with respect to the camera.

2.1 The Effect of Camera Tilt

Consider the case when the camera is tilted from the horizontal as shown in figure 1. This is a rotation about the *X*-axis of the camera by an angle of *θ*.

Henceforth it is assumed that this angle is known. If necessary, it can be obtained from the extrinsic parameters of the camera via a calibration procedure.

It might be tempting to think that a transformation of the image could undo the effects of the camera tilt. However, when a camera moves the images are not related by a projective transformation unless all of the points in the image are coplanar [9]. (This principle is exploited in stereo vision to detect depth disparities.) In other words, it is impossible to correct for projective distortion across a whole image without knowing the locations of all the objects in the image.

Before images can be processed, especially where straight lines are involved, the effects of camera distortion must be removed, i.e. the camera needs to be calibrated, e.g. using [10]. This is particularly important for cheap lenses which often suffer from some sort of radial or barrel distortion.

Figure 2 shows objects viewed from a tilted camera. The left-hand image is the raw image from the camera, and the right-hand one is after correction for lens distortion. It is clear that the vertical edges are not vertical, nor parallel to each other. In fact, if all the "verticals" were extrapolated they would meet at a vanishing point which is well outside the image.

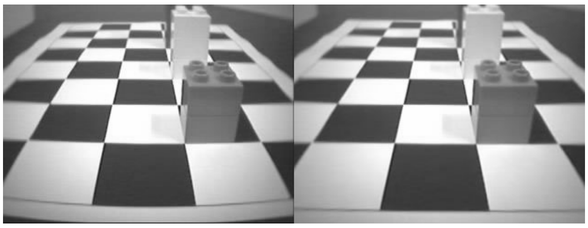


Figure 2: Variations in slope of vertical edges.

We would like to use the information in the image to assist in identifying the location of obstacles. It is more convenient for the robot to use its own coordinate system, rather than camera coordinates, because the robot's *x* and *z* coordinates relate directly to distances that can be measured across the ground.

The effect of the camera tilt is that the *x* coordinates of all points in robot space remain the same in camera space, i.e. $x = X$. However, the *y* and *z* coordinates must be transformed. The new coordinates can easily be calculated as explained below.

For notational convenience, let $c = \cos(\theta)$ and $s = \sin(\theta)$. The rotation about the *X*-axis is:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix} \tag{4}$$

In terms of robot coordinates, the rotated coordinates of two points can be expressed in camera space as:

$$\begin{aligned} p_1 &= (x_1,\ y_1 c - z_1 s,\ y_1 s + z_1 c) \\ p_2 &= (x_2,\ y_2 c - z_2 s,\ y_2 s + z_2 c) \end{aligned} \tag{5}$$

If the two points were selected so that they were both on a vertical edge in the real world, then in the robot coordinate system $x_1 = x_2$ and $z_1 = z_2$, i.e. only the *y* coordinate varies. Therefore no subscript is used on *x* or *z* in the following equations.

Using a similar approach to that used to obtain equation (3), substitute these new coordinates from (5) into equations (1) and (2) and simplify:

$$m = \frac{z(y_2(c^2+s^2) - y_1(c^2+s^2))}{xs(y_1 - y_2)} \tag{6}$$

Applying the trigonometric identity $c^2 + s^2 = 1$, the final result is:

$$m = \frac{-z}{xs} \tag{7}$$

This is a very simple result, but it should not be surprising given what we know from our personal experience about how perspective works.

Some conclusions can be drawn from equation (7). Firstly, when $x = 0$ the slope is infinite, i.e. a vertical edge will appear vertical at the centre of the image. This singularity can be avoided by inverting the calculations and using $1/m$ instead. However, the results are then very close to zero near the centre of the image, and are still difficult to compare.

Also, the sign of the slope changes as the *x* coordinate changes from positive to negative. This means that the vertical edges appear to slope further and further away from vertical the further they are from the centre. This feature can be used as a heuristic to filter out line segments as possible verticals.

And finally, since sin(0) = 0, the slope also becomes infinite if the camera tilt angle is zero, regardless of the values of *x* or *z*. In other words, the lines associated with vertical edges are all vertical in the image if the camera is not tilted.

Figure 3 shows a simulation where the camera has a 60° horizontal FOV and is tilted downwards at 30°. Vertical lines have been drawn from the corners of the checkerboard pattern on the floor. The slopes change significantly across the image, as expected.

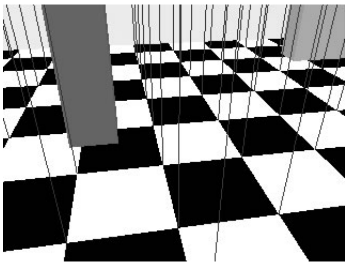


Figure 3: Verticals as viewed by a tilted camera.

The maximum possible slope for a vertical edge is determined by the FOV of the camera and its height above the floor. This sets a limit on the ratio of *z* to *x* that is possible for a given camera. For the example shown above, the “worst case” slope is around 66°. Over much of the image the slope will be close to vertical. For instance, most of the verticals in figure 3 have a slope in the range of 70° to 90°.

Another aspect of equation (7) that is apparent in figure 3 is that the equation describes families of lines. For any set of vertical edges with the same ratio of *z* to *x*, the slope will be the same. In fact, these lines will be coincident in the image.

Notice that the leftmost edge of the large object on the left in figure 3 almost coincides with one of the vertical lines, even though the respective vertical edges originate from different places on the floor.

It is easy to see that horizontals, e.g. the checkerboard pattern, would be difficult to confuse with the verticals even though the verticals do not appear vertical in the image.

If the point of intersection of the vertical edge with the floor can be determined, then this can be used to obtain the *x* and *z* coordinates of the feature using inverse perspective mapping via a simple table lookup [11]. For these coordinates, the slope of the vertical edge can be calculated, thereby confirming whether or not the intersection point is correct.

3 Incremental Localisation

A common problem in computer vision is establishing correspondences between points in images. This is often done on the basis of small patches in stereo vision for instance. In our case of using vertical edges, making the correspondence between two edges in successive frames for a moving camera is relatively straightforward.

In the discussion that follows we consider two different types of moves that can be performed by a robot with two independently driven wheels. These two motions - translations and rotations - allow the robot to follow any possible 2D path.

3.1 Translations

Faugeras [6] points out (p283) that it is not possible to determine camera displacement based on the correspondence between lines in two images. However, we add a further constraint in that we only allow our robot to move in a straight line. This reduces the number of degrees of freedom so that there is an analytical solution to the problem.

For a forward motion, i.e. translation along the negative *z*-axis, the appearance of a given vertical edge will change. It is theoretically possible to use the change in line slope to measure the distance moved, or alternatively confirm the distance moved.

Assume the initial feature coordinates of interest are x_1 and z_1. (The *y* coordinate is irrelevant for a vertical edge because the edge runs parallel to the *y*-axis.) After the move, the new coordinates will be x_2 and z_2.

The slope of the line can be measured in the image before and after the move. The two slopes are:

$$m_1 = \frac{-z_1}{x_1 s}, \quad m_2 = \frac{-z_2}{x_2 s} \tag{8}$$

The *x* coordinate of the edge is not affected by the move, so $x_1 = x_2$. Expressing the equations (8) in terms of *x* and then setting them equal to each other results in:

$$z_1 = \frac{m_1}{m_2} z_2 \tag{9}$$

If the distance travelled can be measured accurately, e.g. using odometry, then the difference between z_1 and z_2 is known, say $d = z_1 - z_2$.

Substituting into equation (9) and solving for z_1 gives:

$$z_1 = \frac{m_1}{(m_1 - m_2)} d \tag{10}$$

A simple rearrangement of equation (10) allows *d* to be calculated if z_1 is known (or z_2 for that matter):

$$d = \left(1 - \frac{m_2}{m_1}\right) z_1 \tag{11}$$

At first glance, this looks like a simple way to determine the range of objects based solely on odometry, or conversely to use range information to verify odometry data.

However, in practice odometry is notoriously unreliable. Furthermore, the difference between the two slopes will often be very small and might not be measurable to within an acceptable level of error given the resolution of the imaging device.

Following a similar derivation, it is possible to use a motion in x direction. A sideways motion can have a much larger impact on the slope of vertical edges for nearby objects than forward motion. However, this type of motion is only possible for holonomic robots, not for wheeled robots.

3.2 Rotations

Consider the situation where the camera rotates on the spot, i.e. around the *y*-axis. This might occur for a camera that can pan, or for a two-wheeled robot that can turn on the spot if the camera is located in the centre of the robot.

Again, for a given vertical edge, let the feature coordinates be x_1 and z_1 and after rotation the new coordinates will be x_2 and z_2.

(These coordinates are related by the radial distance from the camera, which must remain constant. We do not need to use this information, but it is a constraint that should be checked.)

In the robot's view of the world, the angles between the *x*-axis and vectors drawn to the vertical edge are given by:

$$\tan(\alpha) = \frac{z_1}{x_1}, \quad \tan(\beta) = \frac{z_2}{x_2} \qquad (12)$$

The difference between these two angles is the desired angle of rotation, i.e. we want to know $\psi = \beta - \alpha$.

Applying equation (7) to the equations (12), the resulting difference between the two angles, i.e. the amount of rotation, is:

$$\psi = \arctan(m_1 s) - \arctan(m_2 s) \qquad (13)$$

(When calculating this difference, allowance must be made for a possible change of sign if the vertical edge moves across the centre of the image.)

4 Examples

The following two examples are for rotations.

The top two images in figure 4 show two different views with a rotation of 20° between them. The bottom two images show the vertical edges detected using the algorithm in this paper as thick lines.

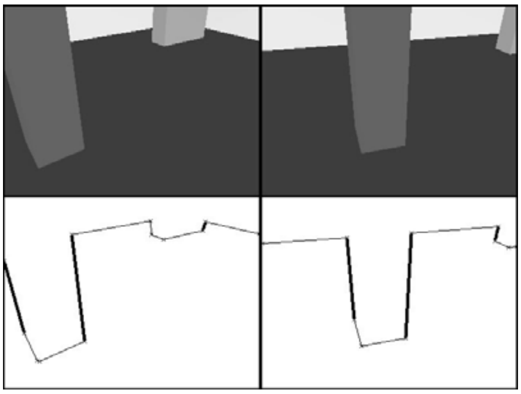


Figure 4: Rotation example 1 using simulator.

There are two significant verticals visible in both images. Calculating the slopes of these lines allows the angle of rotation to be calculated according to equation (13). The results are shown in table 1.

Table 1: Slopes and rotation angles for example 1.

Slope m_1*	Slope m_2	Angle ψ
106.82°	95.16°	20.9°
97.54°	87.54°	19.7°

* Slopes have been converted to angles in degrees because it is easier to verify them visually. This is not part of the calculation.

These estimated rotation angles are very close to the actual value, at least to within a 1 pixel quantisation error in the location of the edges in the image.

The next example shows images from an X80 robot. The robot was given three commands to rotate right by 10°. Segmentation of the floor is work in progress, so the edges of the door jamb have been marked manually. The images are shown in figure 5.

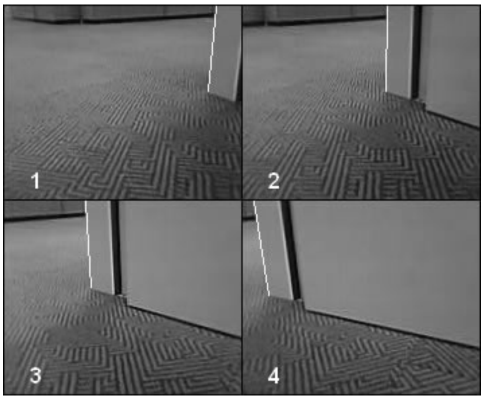


Figure 5: Rotation example 2 from X80 robot.

The camera has a resolution of 176x144 pixels. It was calibrated using the software from [10]. This software provides the camera extrinsic parameters. From these, the tilt angle was calculated to be 18.97°, and the transformation between pixel coordinates and real world coordinates was also determined.

On the X80 robot, the camera is not located at the centre of the robot. Therefore the offset from the centre was included when calculating the actual turn angles based on the camera's extrinsic parameters.

The results are shown in table 2 below. This gives the angle calculated based on the camera extrinsic parameters, δ, and the angle calculated according to equation (13) above, ψ.

Note that in the second image the edge is very close to vertical. An error of one pixel at either end of the edge could result in the calculation of *m* becoming undefined. The code needs to handle this as 90°.

Table 2: Calculated rotation angles for example 2.

Image	Angle δ	Angle ψ
2	10.96°	10.72°
3	10.33°	8.09°
4	10.39°	14.92°

When the robot had completed all the rotations, it had moved slightly to the left. This might account for the poor estimate of the last angle using equation (13) which assumes no movement. The angle calculated using extrinsic parameters, however, is a relative change and is not affected in the same way. It is hard to say which of the results is actually correct.

4.1 Practical Considerations

Different image processing tools produce different results. When the images are processed using canny edge detection followed by Hough lines in OpenCV and Matlab, the results are not identical, even when using ostensibly the same parameters.

Also, the Hough transform, although widely used in computer vision, is a parametric algorithm and it must be tuned to obtain the best results. In particular, bin sizes must be specified for the radial distances and angles of the lines. Even small changes in the parameter for the bin size for line angles can result in missing a line because it does not line up precisely with one of the bin angles. Conversely, using a bin size for the radial distance that is too small can result in a myriad of extraneous lines.

Accurately detecting the slope of the lines is essential for our algorithms to produce accurate results for translations and rotations. However, the approach outlined here is not affected by errors in the absolute location of an edge in the image. Only the slope of an edge is used, which is obtained from the differences between pixel coordinates.

We have not performed a rigorous analysis of variance, but it is possible to place a reasonable limit on the range of errors simply using trial and error.

Detecting edges accurately in a digital image is difficult because of the quantisation effects. Although there are algorithms that will locate an edge with sub-pixel accuracy, it is reasonable to assume that there might be an error of ±1 pixel in determining the image coordinates of any edge pixel.

In the worst case the errors at either end of the line will combine in the same direction. By varying the coordinates at each end of a line by one pixel, it was found that the resulting change in the calculated rotation angle ranged from ½ to 2 degrees. This represents an error of around 5-10% for a 20° rotation. This is not much better than the accuracy of the wheel encoders on the X80, so it is of questionable value in this case.

The images in figure 5 were obtained by deliberately positioning the robot so that the door edge would stay in view for three rotations. In general, most edges disappear after only two, so tracking edges for more than one rotation is not always possible, especially if the rotation is more than 10 degrees.

Finally, the resolution of the images used in these examples is typical for cheap web cameras. However, consumer digital cameras now have resolutions up to 8 megapixels. This is over 100 times the resolution we have used. Even assuming only a ten-fold improvement in the accuracy of measuring the slope of lines, it is clear that more accurate results could be obtained with better cameras.

5 Conclusion

This paper has outlined some simple geometric relationships that can be used to recover vertical edges from an image when the camera is tilted. By comparing the slopes of vertical edges in successive images, it is possible to calculate distances moved or angles of rotation. This information can be used to assist with localisation. We are working to apply this to visual SLAM.

6 References

[1] S. Se, D.G. Lowe and J.J. Little, “Vision-based global localization and mapping for mobile robots”, *IEEE Transactions on Robotics*, 21(3), pp 364-375 (2005).

[2] J. Wolf, W. Burgard and H. Burkhardt, “Robust vision-based localization by combining an image retrieval system with Monte Carlo localization”, *IEEE Transactions on Robotics*, 21(2), pp 208-216 (2005).

[3] A.J. Davison, Y.G. Cid and N. Kita, “Real-time 3D SLAM with wide-angle vision”, *Proceedings 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon (2004).

[4] E. Eade and T. Drummond, “Scalable monocular SLAM”, *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, 1, pp 469-476 (2006).

[5] J. Campbell, R. Sukthankar, I. Nourbakhsh and A. Pahwa, “A robust visual odometry and precipice detection system using consumer-grade monocular vision”, *Proceedings IEEE International Conference on Robotics and Automation*, Barcelona, pp 3421-3427 (2005).

[6] O. Faugeras, *Three-Dimensional Computer Vision*, MIT Press, Cambridge, MA (1993).

[7] B.K.P. Horn, *Robot Vision*, MIT Press, Cambridge, MA (1986).

[8] Intel Corporation, “Open Source Computer Vision Library” (OpenCV), http://www.intel.com/technology/computing/opencv/, visited 2/8/2005.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision* (2nd edition), Cambridge University Press, Cambridge, UK (2003).

[10] J-Y. Bouguet, “Camera calibration toolbox for Matlab”, http://www.vision.caltech.edu/bouguetj/calib_doc/, visited on 19/3/2006.

[11] T. Taylor, S. Geva and W.W. Boles, “Monocular vision as a range sensor,” *Proceedings International Conference on Computational Intelligence for Modelling Control and Automation*, Gold Coast, Australia, CD-ROM (2004).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Vision-based Interpolation of 3D Laser Scans

Henrik Andreasson[1], Rudolph Triebel[2] and Achim Lilienthal[1]
[1]Department of Technology
Örebro University, Örebro, Sweden
[2]Department of Computer Science
University of Freiburg, Freiburg, Germany
henrik.andreasson@tech.oru.se, triebel@informatik.uni-freiburg.de, achim.lilienthal@tech.oru.se

Abstract

3D range sensors, particularly 3D laser range scanners, enjoy a rising popularity and are used nowadays for many different applications. The resolution 3D range sensors provide in the image plane is typically much lower than the resolution of a modern colour camera. In this paper we focus on methods to derive a high-resolution depth image from a low-resolution 3D range sensor and a colour image. The main idea is to use colour similarity as an indication of depth similarity, based on the observation that depth discontinuities in the scene often correspond to colour or brightness changes in the camera image. We present five interpolation methods and compare them with an independently proposed method based on Markov Random Fields. The algorithms proposed in this paper are non-iterative and include a parameter-free vision-based interpolation method. In contrast to previous work, we present ground truth evaluation with real world data and analyse both indoor and outdoor data. Further, we suggest and evaluate four methods to determine a confidence measure for the accuracy of interpolated range values.

Keywords: 3D range sensor, laser range scanner, vision-based depth interpolation, 3D vision

1 Introduction

3D range sensors are getting more and more common and are found in many different areas. A large research area deals with acquiring accurate and very dense 3D models, potential application domains include documenting cultural heritage [1], excavation sites and mapping of underground mines [2]. A lot of work has been done in which textural information obtained from a camera is added to the 3D data. For example, Sequeira et al. [3] present a system that creates textured 3D models of indoor environments using a 3D laser range sensor and a camera. Früh and Zakhor [4] generate photo-realistic 3D reconstructions from urban scenes by combining aerial images with textured 3D data acquired with a laser range scanner and a camera mounted on a vehicle.

In most of the approaches that use a range scanner and a camera, the vision sensor is not actively used during the creation of the model. Instead vision data is only used in the last step to add texture to the extracted model. An exception is the work by Haala and Alshawabkeh [5], in which the camera is used to add line features detected in the images into the created model.

To add a feature obtained with a camera to the point cloud obtained with a laser range scanner, it is required to find the mapping of the 3D laser points onto pixel coordinates in the image. If the focus instead lies on using the camera as an active source of information which is considered in this paper, the fusing part in addition addresses the question of how to estimate a 3D position for each (sub) pixel in the image. The resolution that the range sensor can provide is much lower than obtained with a modern colour camera. This can be seen by comparing figure 1, created by assigning the intensity value of the projected laser point to its closest neighbors, with the corresponding colour image in figure 1.

To our knowledge the only approach that uses colour information from a camera image to obtain a high-resolution 3D point model from a low-resolution 3D range scan is the algorithm by Diebel et al. [6], where both colour information and the raw depth information are used. Their method is also compared with the methods suggested in this paper and is further described in section 3.

2 Suggested Vision-based Interpolation Approaches

The main idea is to interpolate low-resolution range data provided by a 3D laser range scanner under the assumption that depth discontinuities in the scene often correspond to colour or brightness changes in the camera image of the scene.

For the problem under consideration, a set of N laser range measurements $\mathbf{r}_1..\mathbf{r}_N$ is given where each measurement $\mathbf{r}_i = (\theta_i, \pi_i, r_i)$ contains a tilt angle θ_i, a pan angle π_i and a range reading $\mathbf{r}_i$ corresponding to 3D Euclidean coordinates (x_i, y_i, z_i).

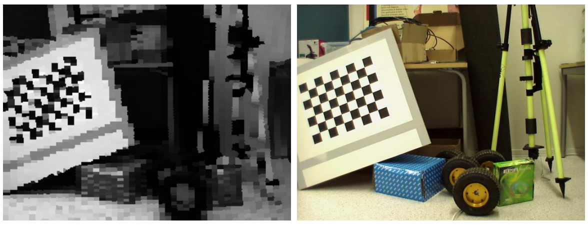


Figure 1: Left: Image intensities plotted with the resolution of the 3D scanner. The laser range readings were projected onto the right image and the closest pixel regions were set to the intensity of the projected pixel for better visualisation. Right: Calibration board used for finding the external parameters of the camera, with a chess board texture and reflective tape (gray border) to locate the board in 3D using the remission / intensity values from the laser scanner.

The image data consists of a set of image pixels $\mathbf{P}_j = (X_j, Y_j, C_j)$, where X_j, Y_j are the pixel coordinates and $C_j = (C_j^1, C_j^2, C_j^3)$ is a three-channel colour value. By projecting a laser range measurement $\mathbf{r}_i$ onto the image plane, a projected laser range reading $\mathbf{R}_i = (X_i, Y_i, r_i, (C_i^1, C_i^2, C_i^3))$ is obtained, which associates a range reading $\mathbf{r}_i$ with the coordinates and the colour of an image pixel. An image showing the projected intensities can be seen in figure 1, where the closest pixel regions are set to the intensity of the projected pixel for better visualisation.

The interpolation problem can now be stated for a given pixel $\mathbf{P}_j$ and a set of projected laser range readings $\mathbf{R}$, as to estimate the interpolated range reading r_j^* as accurately as possible. Hence we denote an interpolated point $\mathbf{R}_j^* = (X_j, Y_j, r_j^*, C_j^1, C_j^2, C_j^3)$.

Five different interpolation techniques are described in this section and compared with the MRF approach described in section 3.

2.1 Nearest Range Reading (NR)

Given a pixel $\mathbf{P}_j$, the interpolated range reading r_j^* is assigned to the laser range reading r_i corresponding to the projected laser range reading $\mathbf{R}_i$ which has the highest likelihood p given as

$$p(\mathbf{P}_j, \mathbf{R}_i) \propto e^{-\frac{(X_j-X_i)^2+(Y_j-Y_i)^2}{\sigma^2}} \qquad (1)$$

where σ is the point distribution variance. Hence, the range reading of the closest point (regarding pixel distance) will be selected.

2.2 Nearest Range Reading Considering Colour (NRC)

This method is an extension of the NR method using colour information in addition. Given a pixel $\mathbf{P}_j$, the interpolated range reading r_j^* is assigned to the range value r_i of the projected laser range reading $\mathbf{R}_i$ which has the highest likelihood p given as

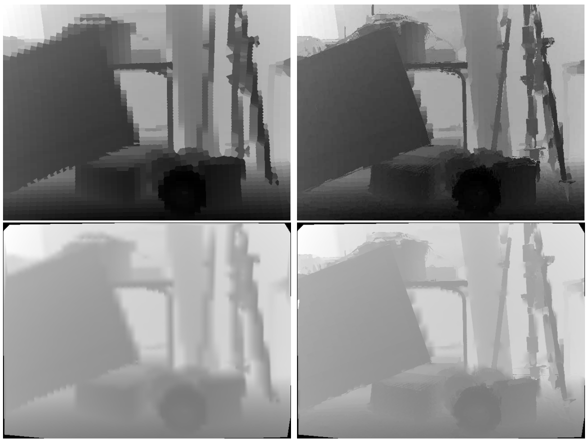


Figure 2: Top left: Depth image generated with the NR method. Top right: Depth image generated with the NRC method, small details are now visible. Note that a depth image generated from a similar viewpoint as the laser range scanner makes it very difficult to see flaws of the interpolation algorithm. Bottom left : MLI. Bottom right : LIC.

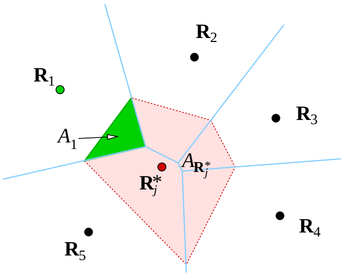


Figure 3: Natural neighbors $\mathbf{R}_1..\mathbf{R}_5$ of $\mathbf{R}_i^*$. The interpolated weight of each natural neighbor $\mathbf{R}_i$ is proportional to the size of the area which contains the points Voronoi cell and the cell generated by $\mathbf{R}_j^*$. I.e. the nearest neighbor $\mathbf{R}_1$ will have influence based upon the area of A_1.

$$p(\mathbf{P}_j, \mathbf{R}_i) \propto e^{-\frac{(X_j-X_i)^2+(Y_j-Y_i)^2}{\sigma_p^2}-\frac{||C_j-C_i||^2}{\sigma_c^2}} \qquad (2)$$

where σ_p and σ_c is the variance for the pixel point and the colour respectively.

2.3 Multi-Linear Interpolation (MLI)

Given a set of projected laser range readings $\mathbf{R}_1..\mathbf{R}_N$, a Voronoi diagram V is created by using their corresponding pixel coordinates $[X,Y]_{1..N}$. The natural neighbors NN to an interpolated point $\mathbf{R}_j^*$ are the points in V, which Voronoi cell would be affected if $\mathbf{R}_j^*$ is added to the Voronoi diagram, see figure 3. By inserting $\mathbf{R}_j^*$ we can obtain the areas $A_{1..n}$ of the intersection between the Voronoi cell due to $\mathbf{R}_j^*$ and the Voronoi cell of $\mathbf{R}_i$ before inserting $\mathbf{R}_j^*$ and the area $A_{\mathbf{R}_j^*}$

as a normalisation factor. The weight of the natural neighbor $\mathbf{R}_i$ is calculated as

$$w_i(\mathbf{R}_j^*) = \frac{A_i}{A_{\mathbf{R}_j^*}}. \tag{3}$$

The interpolated range reading r_j^* is then calculated as

$$r_j^* = \sum_{i \in NN(\mathbf{R_j^*})} w_i r_i. \tag{4}$$

This interpolation approach is linear [7]. One disadvantage is that nearest neighborhood can only be calculated within the convex hull of the scan-points projected to the image. However, this is not considered as a problem since the convex hull encloses almost the whole image, see figure 2.

2.4 Multi-Linear Interpolation Considering Colour (LIC)

To fuse colour information with the MLI approach introduced in the previous subsection, the areas $A_{\mathbf{R}_i}$ and $A_{\mathbf{R}_j^*}$ are combined with colour weights $w_{1..n}^c$ for each natural neighbor based on spatial distance in colour space.

Similar as in section 2.2, a colour variance σ_c is used:

$$w_i^c(\mathbf{R}_j^*) = e^{-\frac{||C_i - C_j||^2}{\sigma_c^2}}. \tag{5}$$

The colour based interpolated range reading estimation is then done with

$$r_j^* = \sum_{i \in NN(\mathbf{R}_j)} \frac{w_i w_i^c}{W^c} \tag{6}$$

where $W^c = \sum_{i=1}^{n} w_i^c$ is used as a normalisation factor.

2.5 Parameter-Free Multi-Linear Interpolation Considering Colour (PLIC)

One major drawback of the methods presented so far and the approach presented in the related work section is that they depend on parameters such as σ_c, for example. To avoid the need to specify colour variances, the intersection area $A_{\mathbf{R}_i}$ defined in Section 2.3 is used to compute a colour variance estimate for each nearest neighbor point $\mathbf{R}_i$ as

$$\sigma_c = \frac{1}{n_i - 1} \sum_{j \in A_i} ||\mu_i - C_j||^2 \tag{7}$$

where $\mu_i = \frac{1}{n_i} \sum_{j \in A_i} C_j$ and n_i is the number of pixel points within the region A_i.

This results in an adaptive adjustment of the weight of each point. In case of a large variance of the local surface texture, colour similarity will have less impact on the weight w_i.

3 Related Work

To our knowledge, the only work using vision for interpolation of 3D laser data is [6] where a Markov Random Field (MRF) framework is used. The method works by iteratively minimising two constraints: ψ stating that the raw laser data and the estimated depth should be similar and ϕ stating that the depth estimates close to each other with a similar colour should also have similar depths.

$$\psi = \sum_{i \in N} k(r_i^* - r_i)^2 \tag{8}$$

where k is a constant and the sum runs over the set of N positions which contain a laser range reading r_i and r_i^* is the interpolated range reading for position i. The second constraint is given as

$$\phi = \sum_i \sum_{j \in NN(i)} e^{(-c||C_i - C_j||^2)} (r_i^* - r_j^*)^2 \tag{9}$$

where c is a constant, C is the pixel colour and $NN(i)$ are the neighborhood pixel around position i.

The function to be minimised is the sum $\psi + \phi$.

4 Evaluation

Experimental evaluation is a crucial point and has been done using both simulated and real data. All data sets D were divided into two equally sized parts D_1 and D_2. One dataset, D_1, is used for interpolation and D_2 is used as the ground truth where each laser range measurement is projected to image coordinates. Hence for each ground truth point $\mathbf{R}_i$ we have the pixel positions $[X,Y]_i$ and the range r_i. The pixel position $[X,Y]_i$ is used as input to the interpolation algorithm and the range r_i is used as the ground truth. The performance of the interpolation algorithms is analysed based on the difference between the interpolated range and the range from the ground truth.

5 Experimental Setup

5.1 Hardware

The scanner used is a 2D SICK LMS-200 mounted together with a 1 MegaPixel (1280x960) colour CCD camera on a pan-tilt unit from Amtec where the displacement between the optical axis is approx 0.2 m. The scanner is located on our outdoor robot, see figure 4, a P3-AT from ActivMedia. The SICK scanner has a larger spot size compared to many other laser scanners and often gives wrong range estimates close to edges where the laser spot covers multiple objects at different distances, see figure 4. Of course, this flaw of the sensor will be reflected in the ground truth as well. The angular resolution of the laser scanner is 0.5

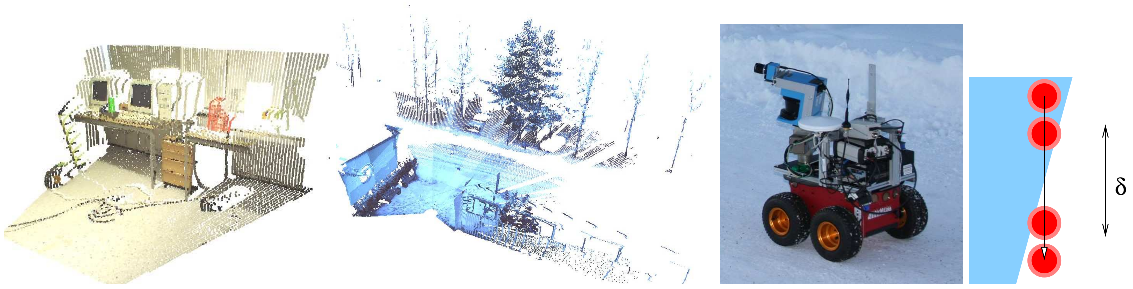


Figure 4: Left: The third indoor evaluation scan, *Indoor*$_3$. Middle left: Scans taken in winter time with some snow. (The figure consist of 7 fused scans. Only 3 of them are used in the evaluation presented in this paper). Middle right: Our outdoor robot with the SICK LMS scanner and a colour CCD camera mounted on a pan tile unit from Amtec, that were used in the experiments. Right: When the laser range finder spot covers an area which contain different depths (blue and white areas), the range reading returned might be unreliable and vary anywhere between the closest to the furthest range (shown as the region δ).

degrees. Half of the readings were used as ground truth, so the resolution for the points used for interpolation is 1 degree.

6 Results - Interpolation

The most common colour spaces were also compared to evaluate if better illuminance/shading invariance could be useful. The colour spaces compared were standard RGB, Normalised RGB, HSV and YUV. Since a consistent improvement could not be observed for neither of the colour spaces tested, only results based on standard RGB normalised to [0,1] are presented in this paper.

In all experiments the colour variance $\sigma_c = 0.05$ and the pixel distance variance $\sigma_d = 10mm$ where used, which were found empirically. The parameters used within the MRF approach described in section 3, where obtained by extensive empirical testing and were set to $k = 2$ and $C = 10$. The optimisation method used for this method was the conjugate gradient method described in [8]. In all experiments the full resolution (1280x960) of the camera image was used.

All the interpolation algorithms described in this paper were tested on real data consisting of three indoor scans and two outdoor scans. The outdoor scans were taken in winter time with snow, which presents the additional challenge that most of the points in the scene have very similar colours.

The results are summarised in table 1 and table 2, which show the mean error with respect to the ground truth $\bar{e}$, and the percentage of outliers o_t for different thresholds t. The percentage of outliers is the percentage of points for which the interpolated range value deviates from the ground truth value by more than a threshold t (specified in meters in table 1 and table 2).

Table 1: Results from *Indoor*$_1$ $-$ *Indoor*$_3$ data sets.

	NR	*NRC*	*MLI*	*LIC*	*PLIC*	*MRF*
$\bar{e}$	0.092	0.087	0.076	**0.072**	0.073	0.074
$o_{0.1}$	0.139	**0.119**	0.155	0.123	0.126	0.141
$o_{0.2}$	0.096	0.084	0.091	**0.079**	0.080	0.083
$o_{0.5}$	0.050	0.048	0.029	0.036	0.037	**0.030**
$o_{1.0}$	**0.008**	0.022	0.011	0.012	0.012	0.011
$o_{3.0}$	0.003	0.004	0.003	0.003	**0.002**	0.003

For the indoor data sets, which comprise many planar structures, the lowest mean error was found with the multi-linear interpolation methods, particularly LIC and PLIC, and MRF interpolation. LIC and PLIC produced less (but larger) outliers.

With the outdoor data the results obtained were more diverse. For the data set *Outdoor*$_1$, which contains some planar structures, a similar result as in the case of the indoor data was observed. For data sets with a very small portion of planar structures such as *Outdoor*$_2$, the mean error was generally much higher and the MRF method performed slightly better compared to the multi-linear interpolation methods. This is likely due to the absence of planar surfaces and the strong similarity of the colours in the image recorded at winter time. It is noteworthy that in this case, the nearest neighbor interpolation method *without* considering colour (NR) performed as good as MRF. The interpolation accuracy of the parameter-free PLIC method was always better or comparable to the parameterised method LIC.

7 Confidence Measure

The interpolated range reading r_j^* may be a good estimate of the actual range or it might deviate substantially from the true value. Therefore a confidence measure for the correctness of the interpolated range reading estimate is desirable, allowing to detect and handle errornous measures appropriately.

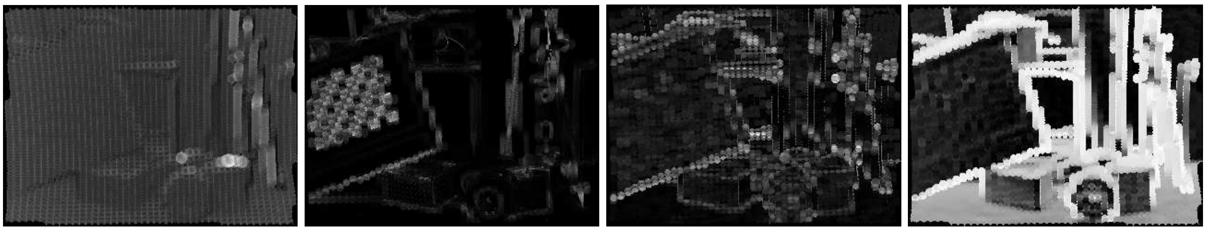


Figure 5: Visualisation of the confidence measures suggested. From left to right: NLR showing the distance to the closest point, NLRC using colour distance, PS showing the plane factor of the neighborhood of the interpolated point and AON showing the angle difference between the normal of the extracted local plane and the camera axis. The parameter free method (PLIC) were used.

Table 2: Results from $Outdoor_1$ and $Outdoor_2$.

	NR	*NRC*	*MLI*	*LIC*	*PLIC*	*MRF*
$\bar{e}$	0.067	0.068	0.056	0.059	**0.054**	**0.054**
$o_{0.1}$	0.147	0.160	0.156	0.146	**0.138**	0.150
$o_{0.2}$	0.076	0.080	0.078	0.073	**0.068**	0.076
$o_{0.5}$	0.032	0.032	0.016	0.020	**0.015**	0.016
$o_{1.0}$	0.005	0.002	0.001	**0.001**	0.002	**0.001**
$o_{3.0}$	**0.000**	**0.000**	0.001	0.001	**0.000**	**0.000**
$\bar{e}$	0.219	0.294	0.235	0.322	0.275	**0.218**
$o_{0.1}$	0.196	0.240	0.242	0.269	0.264	**0.187**
$o_{0.2}$	**0.096**	0.152	0.140	0.168	0.160	0.098
$o_{0.5}$	**0.047**	0.088	0.077	0.094	0.083	0.051
$o_{1.0}$	0.036	0.057	0.043	0.059	0.049	**0.030**
$o_{3.0}$	**0.016**	0.023	0.019	0.028	0.022	0.017

In this paper we suggest and evaluate four different confidence measures.

7.1 Proximity to the Nearest Laser Range Reading (NLR)

This confidence measure is based on the distance between the pixel position of the interpolated point $\mathbf{R}_j^*$ to the nearest projected laser range reading $\mathbf{R}_i$. The idea is that if the interpolated pixel point is close to a point where a range measurement is available the interpolation is considered more trustworthy.

$$NLR(\mathbf{R}_j^*, \mathbf{R}_i) = e^{-\sqrt{(X_j^*-X_i)^2+(Y_j^*-Y_i)^2}} \qquad (10)$$

7.2 Proximity to the Nearest Laser Range Reading Considering Colour (NLRC)

This confidence measure is based on the distance between the colour of the pixel of the nearest projected laser range reading $\mathbf{R}_i$ and $\mathbf{R}_i^*$. This confidence measure works better since it takes into account that confidence in the interpolated value should decrease if the two points have different colour.

$$NLRC(\mathbf{R}_j^*, \mathbf{R}_i) = e^{-||C_j-C_i||} \qquad (11)$$

7.3 Degree of Planar Structure (PS)

Our confidence in the range interpolation also depends on how well a planar surface can be fitted to the local neighbors $NN(\mathbf{R}_j^*)$ of the interpolation point $\mathbf{R}_j^*$ since planar surfaces support a linear interpolation technique very well. The neighbors are either determined from the grid defined by the projected laser range readings or the nearest neighbors found in the Voronoi tessellation. The parameters of the planar surface are obtained from the 3D covariance matrix of $NN(j)$ where the two main eigenvectors are extracted which span a planar surface S_j with the normal vector $\mathbf{n}_j$. The confidence measure is then calculated from the average distance of the local neighbors to the fitted plane as

$$PS(\mathbf{R}_j^*) = e^{-\frac{1}{NN}\Sigma_{i \in NN(\mathbf{R}_j^*)} ||\mathbf{r}_i \cdot \mathbf{n}_j - d_j||} \qquad (12)$$

where d_j is the distance of the plane S_j to the origin and $\mathbf{r}_i = (x_i, y_i, z_i)$ is the 3D position of point i.

7.4 Angle Between the Optical Axis and the Fitted Plane Normal (AON)

This confidence measure considers the orientation of the planar surface S_j described in the previous section relative to the optical axis of the camera $\mathbf{z}_{cam}$. If the angle between the normal vector $\mathbf{n}_j$ and the optical axis is small, the confidence should be high since we expect only one reflexion from the laser scanner and the displacement between the laser and the camera will have a negligible impact.

$$AON(\mathbf{R}_j^*) = \mathbf{z}_{cam} \cdot \mathbf{n}_j \qquad (13)$$

8 Result - Confidence Measure

With the exception of the NLR method, a distinct negative correlation was found for all the confidence measures suggested in this section. Due to the experimental setup where the evaluation points were taken from the laser scanner in an evenly spaced grid the parallax errors caused by the displacement correspond to a low distance between $\mathbf{R}_j^*$ and $\mathbf{R}_i$ which gave the proposed

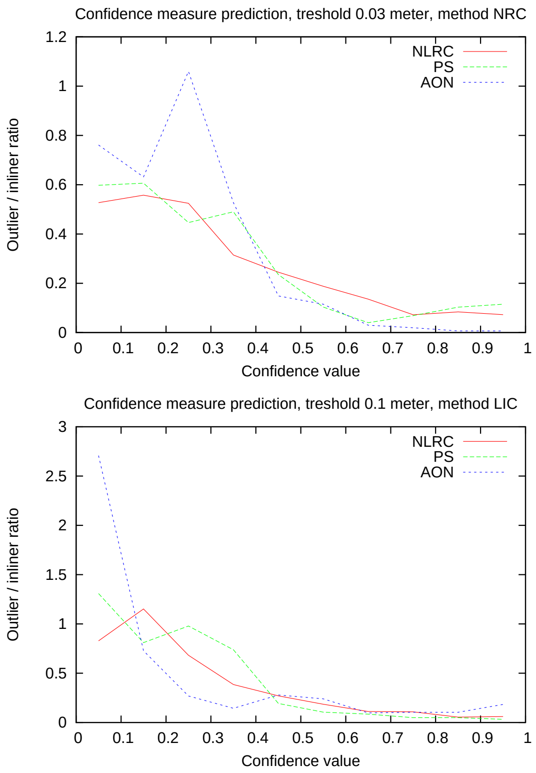


Figure 6: Suitability of the confidence measures introduced in this paper. The graphs show the number of outlier / number of inlier ratio, depending on the confidence in the interpolated points. All points with a depth error > 0.03 meter are considered outliers in the upper image and in the lower graph the threshold was 0.1 meter. Top: $Indoor_1$ data set with method NRC. Bottom : $Outdoor_2$ with method LIC.

NLR method to give high confidence correlated with parallax errors.

Figure 6 shows the inlier/outlier ratio depending on the confidence calculated with the NLRC, PS, and AON method. Interpolated range values were classified as outliers if the deviation from the ground truth value was larger than approximately a third of the mean error obtained with the particular interpolation method. The same general trend of a clear negative correlation, however, was observed with all interpolation methods and for all data sets.

9 Conclusions

This paper is concerned with methods to derive a high-resolution depth image from a low-resolution 3D range sensor and a colour image. We suggest five interpolation methods and compare them with an alternative method proposed by Diebel and Thrun [6]. In contrast to previous work, we present ground truth evaluation with real world data and analyse both indoor and outdoor data. The results of this evaluation do not allow to single out one particular interpolation method that provides a distinctly superior interpolation accuracy, indicating that the best interpolation method depends on the content of the scene. Altogether, the MRF method proposed in [6] and the PLIC method proposed in this paper provided the best interpolation performance. While providing basically the same level of interpolation accuracy as the MRF approach, the PLIC method has the advantage that it is a parameter-free and non-iterative method, i.e. that a certain processing time can be guaranteed. We further suggest and evaluate four methods to determine a confidence measure for the accuracy of interpolated range values. The confidence value calculated showed a distinct negative correlation with the occurrence of outliers. This was observed independent of the scene content and the interpolation method applied.

10 References

[1] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, "The Digital Michelangelo Project: 3D scanning of large statues", *Proceedings SIGGRAPH*, pp 131–144 (2000).

[2] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer and W. Whittaker, "A system for volumetric robotic mapping of abandoned mines", *Proceedings ICRA*, Taipei, Taiwan, pp 4270–4275 (2003).

[3] V. Sequeira, J. Goncalves and M.I. Ribeiro, "3d reconstruction of indoor environments", *Proceedings ICIP*, Lausanne, Switzerland, pp 405–408 (1996).

[4] C. Früh and A. Zakhor, "3D model generation for cities using aerial photographs and ground level laser scans", *Proceedings CVPR*, Hawaii, USA, pp 31–38 (2001).

[5] N. Haala and Y. Alshawabkeh, "Application of photogrammetric techniques for heritage documentation", *Proceedings 2nd Int. Conf. on Science & Technology in Archaeology & Conservation*, Amman, Jordan (2003).

[6] J. Diebel and S. Thrun, "An application of markov random fields to range sensing", *Proceedings NIPS*, Vancover, Canada, pp 291–298 (2005).

[7] R. Sibson, "A brief description of natural neighbour interpolation", in V. Barnet (ed.), *Interpolating Multivariate Data*, John Wiley & Sons, Chichester, pp 21–36 (1981).

[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge (UK) and New York (1992).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Real-Time Coastline Detection Using an FPGA Based Vision Processor

Andrew Price, David Ashiri and Jacob Pyke
Autonomous Aircraft Research Group
Department of Electrical and Computer Systems Engineering
Monash University, Melbourne, Australia
andrew.price@eng.monash.edu.au

Abstract

Unmanned Aerial Vehicles (UAVs) are becoming a useful and cheap tool for complex day to day missions, which can vary from search and rescue missions to coastline tracking. By using a UAV it would be possible to save a substantial amount of money on sending a helicopter to execute the same mission. An example for an application would be the identification of oil spills close to shore and the ability to measure the extent of the damage. This paper discusses and uses traditional image processing techniques in order to track coastlines. This research was completed by Ashiri and Pyke as part of an undergraduate thesis project at Monash University.

Keywords: UAV, hardware, FPGA, vision system, real time, onboard image processing, coastline detection and tracking

1 Introduction

The intended outcome of this research was to develop an algorithm for the detection and tracking of coastlines by a low cost UAV, which can be translated to an FPGA using VHDL with near zero modification. Consequently, the implementation had to meet several practical constraints and thus made use of traditional image processing algorithms which have been already proven feasible in FPGA hardware [1].

2 System Design

2.1 Overview

The general concept which was taken from the previous research done was to base the system on a serial stream of pixels from a CMOS camera, which could in turn perform the image processing at pixel rate - in other words, at the same rate as the pixels get clocked out of the camera (~15 MHz).

It was determined that two subsequent frames captured by the UAV's camera at 25 frames per second are unlikely to differ greatly, due to the altitude of the UAV reducing the relative movement of the ground below with respect to the camera's field of vision. Therefore, it was a reasonable assumption to rely on a histogram constructed from a previous frame to make decisions on the current frame as it is being clocked through the system. Furthermore, it would be possible to implement both the histogram creation and the coast detection algorithm to operate concurrently in the hardware.

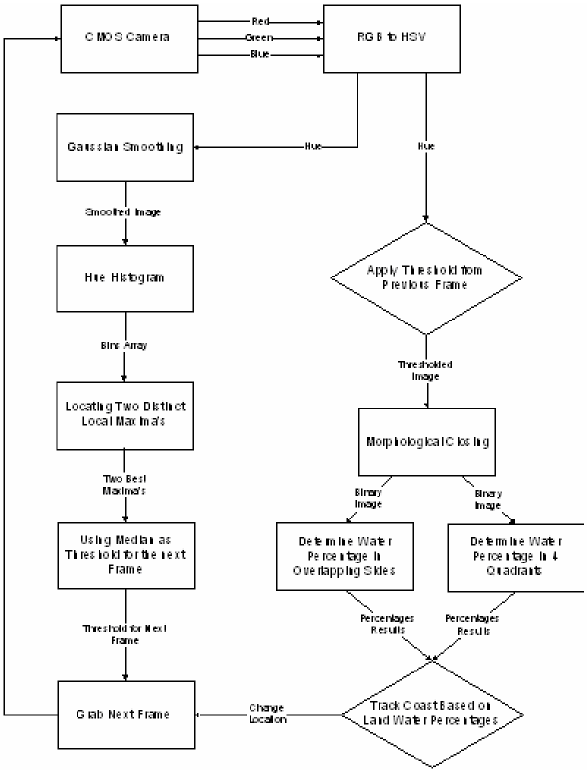


Figure 1: System block diagram.

2.2 Dealing with Colour

It is unreasonable to expect, in the case of detecting a coastline, that the major feature in the image which can be measured is two distinct bodies of colour [2],[3]. Water is not necessarily blue, nor are there typically golden beaches and wide brown land. Other factors must be considered when attempting to make use of a contrast in colour. The colour of the water can vary significantly due to other factors such as variable lighting conditions, water depth, cloud cover and the underlying colour of the seabed. Hence it is not practical to rely on the RGB colour space alone.

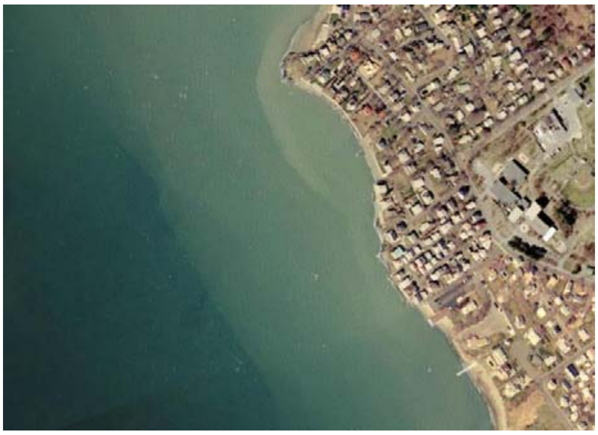


Figure 2: Colour: an unreliable measure.

In order to take into account these factors, the first stage of the system is to convert the pixels to a more useful colour space: HSV. Using the Hue channel from this colour space, it was possible to account for the variations in water colour in order to better segment the captured frames. This was implemented solely with integer arithmetic, thus providing portability to a hardware design in VHDL.

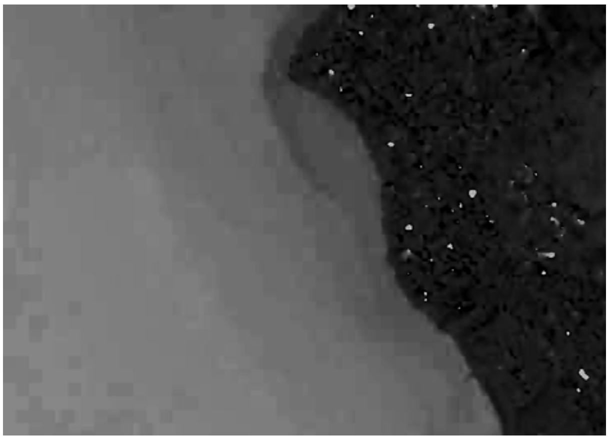
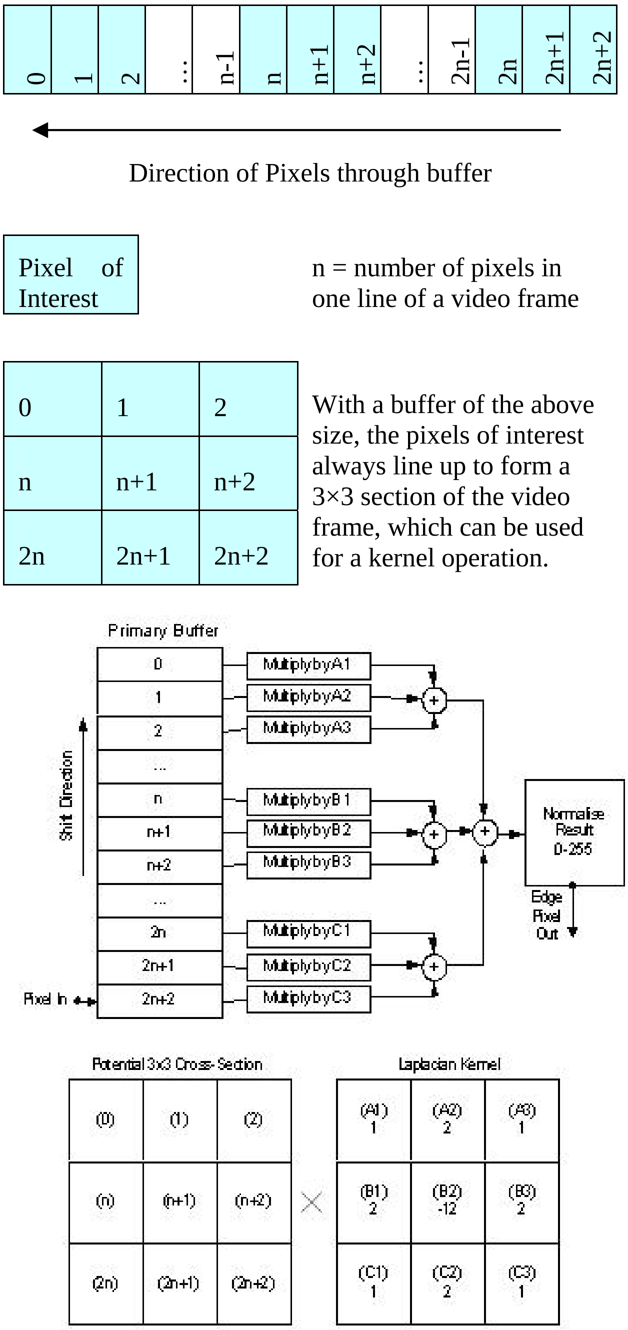


Figure 3: Hue channel from HSV image (scaled to 8bit depth).

Figure 3 demonstrates that despite the colour space conversion from RGB to HSV, there were still a significant number of regions on land with similar hue characteristics to the water. To reduce these effects, a Gaussian smoothing kernel was applied to the Hue channel of the image. This was implemented using a serial implementation of a matrix kernel filter operation. Hue has the problem of being circular, a value typically between 0 and 360. Hence it is necessary to actually use a modified HSV space where the hue is calculated beyond one complete circle [6].

A serial pixel stream approach to image filtering.

Serial pixel stream processing of an image has an excellent advantage in image processing since many of the traditional image processing tasks, such as kernel or matrix filtering, colour space conversions and feature determination, can be achieved without the need for a frame buffer.



Direction of Pixels through buffer

n = number of pixels in one line of a video frame

With a buffer of the above size, the pixels of interest always line up to form a 3×3 section of the video frame, which can be used for a kernel operation.

Figure 4: Serial implementation of a matrix kernel filter.

As we are dealing with an onboard vision processing system for a UAV under 2kg, smaller and lighter is definitely better. The matrix kernel is formed by using three buffers of one line each and extracting particular pixels from within each line. In this way, rather than pass the kernel filter over the image, the image is passed over the kernel. It is only necessary to store three lines of the image, greatly reducing the memory requirements and making it possible to implement the system on a low cost FPGA without the need for external memory.

The Gaussian smoothed image shows a greater differentiation between land and sea, though not yet sufficient to reliably distinguish the two.

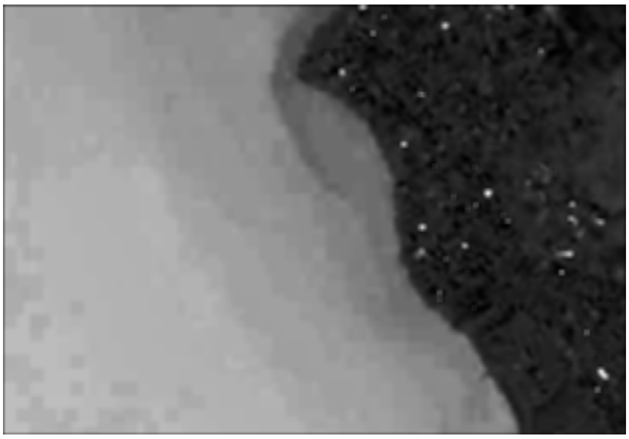


Figure 5: Gaussian smoothed image.

A histogram of the Gaussian image (figure 6) demonstrates two distinct bodies of hue, one centred around 30° and another more spread out between 140° and 190°, the latter of which represents the variation in hue present in the water. By means of a discrete sampling of the histogram into 12 bins, it was possible to identify the two local maxima and subsequently calculate a suitable threshold for segmenting the image. The number of bins is arbitrary and in some cases a power of 2 is desirable for reduced complexity in storage. For the example shown in figure 6, the threshold was calculated to be 135°. The threshold was determined manually for this example; however, as images vary, the process of thresholding needs to be automated.

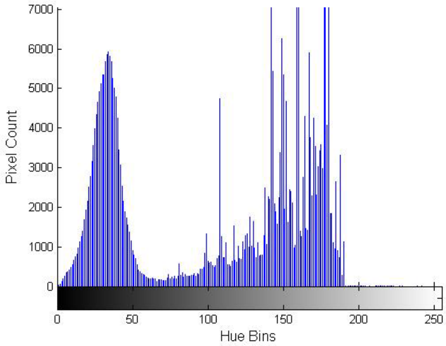


Figure 6: Histogram of smoothed image.

The results of this thresholding process for the sampled image can be seen in figure 7.

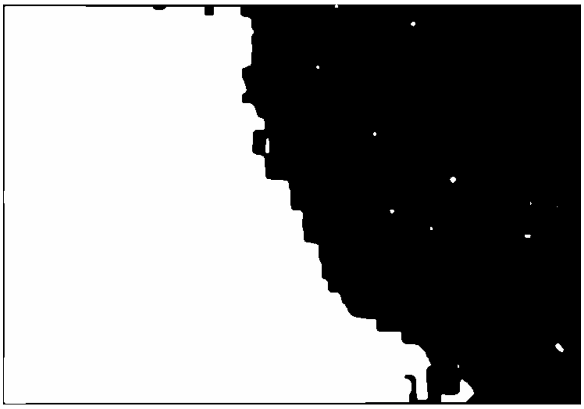


Figure 7: Image segmented based on threshold from discrete histogram.

It is evident that the steps taken have segmented the frame into the two correct regions of water and land. Nevertheless, there are slight gaps in parts of the water segment (white). To help fill out these gaps, a simple morphological closing operation is applied to the image, whereby dilation, followed by erosion, is performed using a similar inline serial implementation of a matrix kernel filter, as shown in figure 4.

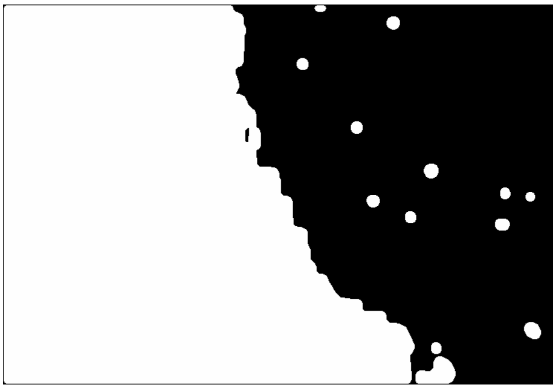


Figure 8: Results of morphological closing.

As can be seen by the result of the morphological closing in figure 7, the small fragments of white in the land (black) region were made more prominent. However, as will be discussed in the following section, these had minimal effect on the decision produced by the coast tracking algorithm.

2.3 Constructing a coastline tracking system

The first step necessary to construct the tracking system was to collect a number of statistics from the thresholded image [4],[5]. To do this, the image was divided into four equal sized quadrants. For each quadrant the ratio of water to land (white to black in the threshold image) was calculated. For each pair of adjacent quadrants, left to right and top to bottom, the same ratio was calculated. Finally, the above ratio was calculated for the image as a whole. This was done serially as the pixels were clocked through the system. It should be noted that as the number of operations performed on the serial pixel stream increases, the delay of the frame reaching either a final buffer or a display also increases; however, the delay is in the order of hundreds of pixels compared with upwards of a hundred thousand pixels for the entire image.

For the purposes of this research the UAV coast tracking system was set to maintain the coast on the right hand side of its field of vision. In other words, the sea would always be to the port wing and the land to the starboard. One thing that is common to many image processing systems is that data does not arrive in pristine condition. Decisions frequently have to be made on the evidence available.

The first decision made by the coastline tracking system is based upon the ratio of water to land present in the entire frame. If the water occupies greater than 75% of the image, the algorithm chooses a course of action based on the quadrant with the lowest water to land ratio, i.e. the most likely candidate for a coast.

If the water occupies 60%-75% of the total image, the amount of land visible is now sufficient to more reliably set a course for the land using the entire image. If the water to land ratio is found to be a minimum on the left or right half of the image, a sharp course either left or right is chosen. Alternatively, if the ratio shows the land to be most prominent in the top half, a course of ±22.5° is taken, depending on which of the top two quadrants best meets the criteria to turn the UAV to form a parallel course to the land (in this example with the land to the right). If the land is found on the bottom half of the frame, then this means the opposite case, in which the UAV is approaching the coast from inland and thus the course is set to veer sharply to the right to align the UAV with the coast on the right hand side. If the coast is found on the left hand side, the UAV sets a course 180° about and offset by a small angle calculated from the ratio of land in the two left quadrants of the frame with respect to the current position of the UAV. Alternatively, if the land is on the right hand side as desired, the course of action is changed to make small corrections to better follow the coast and maintain a consistent ratio of water to land.

The algorithm we have produced is expected to operate within a water to land ratio of 40%-60%. This is following the decision that the system should track the coast when the land and water both cover approximately half of the camera's field of vision.

3 Experimental Results

Figure 9 shows a frame processed using our system. The green highlight arrow is a product of the algorithm and shows the course which the UAV should most likely steer to maintain approximately 50% water and 50% land in the image.

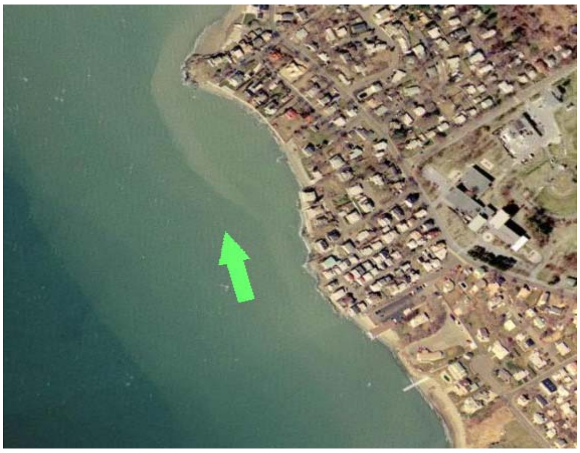


Figure 9: Telling the UAV where to go.

It is fair to suggest however that not all images are the same. Waves, whitecaps and different coloured sand on beaches all have an effect. Figure 10 shows a more complex image.

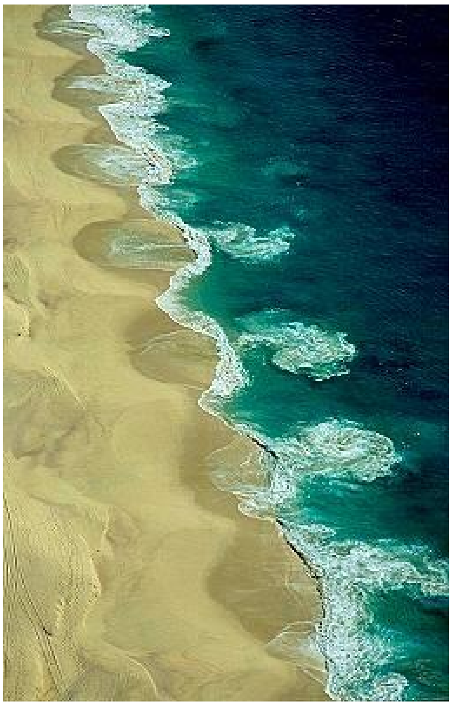


Figure 10: A not so simple example.

Figure 10 shows differences in sand colour as well as substantial turbulence in the water. In addition, the water is shown as the darker of the two regions, a position reversed from the previous example. It is necessary to make a determination based on other information and in some cases a priori information may need to be used. Measures of the types of noise present in both land region and water regions can sometimes be used. For example, the transient distribution of whitecaps or the regular nature of streets and houses may need to be introduced to make an initial determination of what is land and what is sea. The image processing techniques presented in this paper are quite effective on images of the type presented in figure 10. Figure 11 shows the results of the segmentation and the directional arrow our algorithm has predicted in order to maintain the UAV above the coastline.

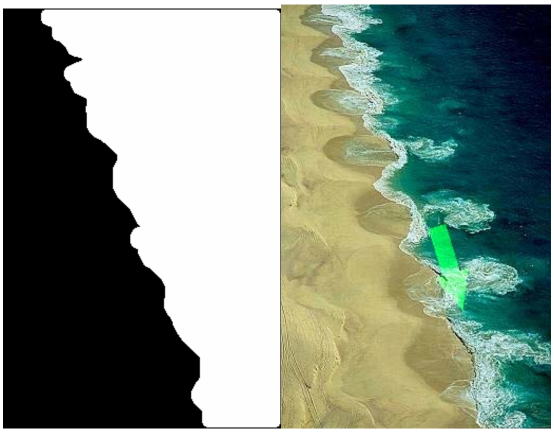


Figure 11: Correctly identifying the appropriate course.

The difficulty arises when the coastline is completely messed up by the inconvenient presence of natural features such as rivers, as shown in figure 12.

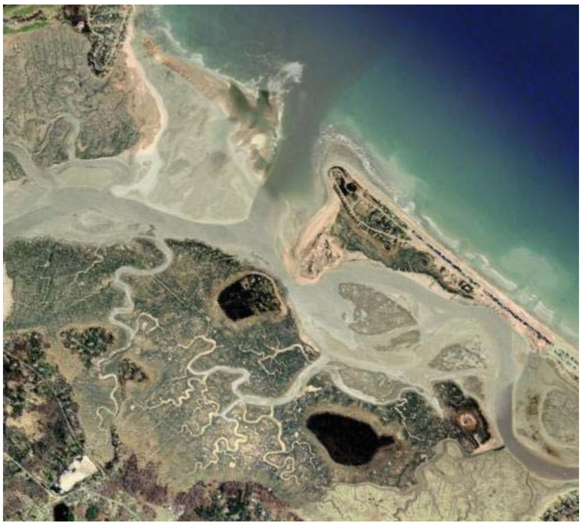


Figure 12: A very messy image.

This research, though extensively part of an undergraduate thesis project, was carried out under the conditions that the techniques developed had to work outside pristine laboratory conditions. In short, experiments were conducted using less than ideal examples, which demonstrate a level of robustness in the technique. While the presence of waves and whitecaps may disturb water close to shore, a measure was taken that larger bodies of water are likely to be more consistent. Figure 12 demonstrates that the land is likely to have more disturbance or noise than the sea region. This can be used as one measure to distinguish what is actually land and what is water.

The results of processing this image are shown in figures 13 and 14.

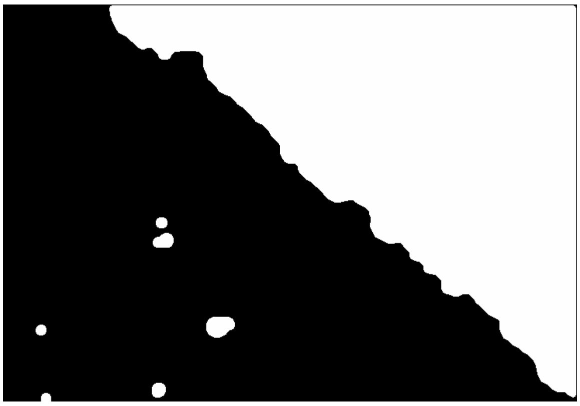


Figure 13: Morphological closing showing noise on the land side.

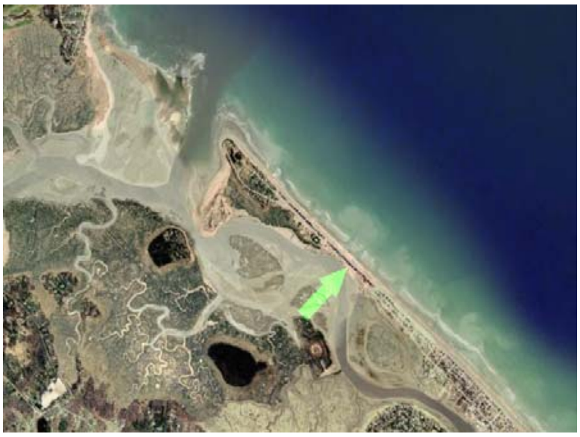


Figure 14: More land than sea, head out.

In this case, the UAV has determined a greater presence of land in the image and is making a course out to sea. One should note that as more frames are recorded the turning angle of the UAV will be adjusted so that the UAV corrects its course and tracks down along the coast.

4 Conclusions

The coastline tracking system we have developed operates using a very simple low cost image processing hardware, without the need for buffering entire frames. It has been shown to correctly identify the coastline and, based on a series of measures and intelligent decisions, determine the correct course of action in a number of both favourable and unfavourable image scenarios. While the system is not perfect and does require considerable development, we have shown its feasibility in this initial study. Initially the work was conducted using general purpose FPGA development hardware; however, a more highly integrated image capture and process system would enable more rigorous testing. This is being undertaken as a next generation exercise using the Monash Minivision and Miniflight processor systems.

Our system operates with a delay equivalent to approximately three lines of pixels. That is to say, once the stream of pixels constituting one image has passed through the colour space converter, the equivalent delay of approximately 3 image lines is needed to extract all of the results necessary to determine the course of action. As a consequence, the inter-frame delay time is almost entirely available for more substantial image processing tasks such as land based features or the identification of ships at sea.

We have demonstrated that our system works under adverse conditions with highly cluttered images containing rivers and substantial noise. Further development work is necessary in the expert system to broaden the system capabilities and improve its survivability.

5 Acknowledgements

The authors wish to acknowledge the assistance of Mr Terry Cornall from the Department of Electrical and Computer Systems Engineering, Monash University, for his assistance in development of initial hardware experiments.

The processing hardware used in this research was developed by the Department of Electrical and Computer Systems Engineering, Monash University.

6 References

[1] A. Price, J. Pyke, D. Ashiri and T. Cornall, “Real time object detection for an unmanned aerial vehicle using an FPGA based vision system”, *Proceedings of ICRA’06*, Florida (2006).

[2] K Nordberg, P. Doherty, G. Farneback, P. Forssen, G. Granlund, A. Moe and J. Wiklund, “Vision for a UAV helicopter,” *Proceedings of IROS’02, Workshop on Aerial Robotics*, Switzerland (2002).

[3] B. Draper, W. Najjar, W. Bohm, J. Hammes, R. Rinker, C. Ross and J. Bins, “Compiling and optimizing image processing algorithms for FPGAs”, *IEEE International Workshop on Computer Architecture for Machine Perception (CAMP)*, 11-13 September, Padova, Italy, pp 222-232 (2000).

[4] S. Jorg, L. Langwald and M. Nickl, “FPGA based real-time visual servoing,” *ICPR*, 1(1), pp 749-753 (2004).

[5] H.B. Christophersen, W.J. Pickell, A.A. Koller, S.K. Kannan and E.N. Johnson, “Small adaptive flight control systems for UAVs using FPGA/DSP technology,” *Proceedings of the AIAA “Unmanned Unlimited” Technical Conference, Workshop and Exhibit*, Chicago (2004).

[6] A.R. Price, *Mobility and Vision for Mobile Robots in Non-Deterministic Competitive Environments*, PhD Thesis, Deakin University, (1999).

Energy Optimisation of Vision Guided Manipulator for Optimal Dynamic Performance

M.H. Jamaluddin, M.A. Said and Marizan Sulaiman
Faculty of Electrical Engineering,
Kolej Universiti Teknikal Kebangsaan Malaysia, Malacca, Malaysia
herman@kutkm.edu.my, azmisaid@kutkm.edu.my, marizan@kutkm.edu.my

Abstract

This paper presents a step on how to optimise the energy and performance of an industrial robot. The project consist of three major phases; (1) theoretical, simulation and practical of forward and inverse kinematics for Fanuc LR Mate 200iB robot to determine their D-H parameters, (2) optimisation of robot movements, and (3) implementation of practical tasks. The optimisation process involves the control of two parameters known as position of joint angle (ө) and speed of the motor (ω) of the three (3) main axes of the robot. The energy is measured regarding 3 types of categorised movements known as reference, fixed and optimised which do the same repetitive task. Its forward and inverse kinematics problems will be simulated using Robotic tools of Matlab 7.0 and Roboguide V2.3.2. The simulation results will be compared with the real practical movements. This paper will describe the techniques, calculations and methods on how the second phase was done.

Keywords: manipulators, optimal control, path-finding, dynamic performance.

1 Introduction

In this project, one of the sensors that are currently being used in controlling robot mechanism is vision. Optimal-based control strategies will be developed to perform manipulative actions. This vision guided robot could perform the tasks with improved efficiency using the optimal-based control strategies. The advantages of these optimal control strategies can be seen during performing the tasks with fewer movements of angles, better joint's motor speed and improved time motion that result in more efficient usage of energy.

To find the optimal dynamic performance of a robot, it is important to know the architecture of its kinematics first. From forward or inverse kinematics analysis of that particular robot, we will obtain an exact position and orientation for each of its movements. This will help to find the most suitable position and orientation to achieve optimal performance.

The outline of the paper is as follows. In section 2 a typical step on searching process of the manipulator's inverse kinematics with given end point is explained in detail. Simulation process is described in section 3 followed by energy measurement in section 4 and experimental result in section 5. Conclusions are stated in section 6.

2 Inverse Kinematics

Inverse Kinematics does the reverse of forward kinematics. Given the end point of the structure, what angles do the joints need to achieve that end point? It can be difficult, and there are usually many or infinitely many solutions. Most of the real systems are under constrained, so for a given goal position, there could be infinite solutions (i.e. many different joint configurations could lead to the same endpoint). The field of robotics has developed many inverse kinematics systems which, due to their constraints, have closed-form solutions.

One of the techniques that had been used to find the exact angle of each joint to reach the goal target is by using the Jacobian method which described briefly in [1].

2.1 Inverse Kinematics Analysis of Fanuc

After considering the Jacobian method trough the theoretical calculation based, the analysis which is using the Matlab Robotics tools were done. To find the inverse kinematics in Matlab, the function that had been used is `q = ikine(robot, T)`. This function returns the joint coordinates for the manipulator described by the object *robot* whose end-effectors homogeneous transform is given by T.

In this project, the robot needs to pick an object at position of [600, 0, -320] and place the object at position of [0, 620, -300].

The complete lists of function regarding this analysis of picking and placing the object using Matlab are as such:

```
%Entering the D-H parameter information for FANUC

L1=link([pi/2 150 0 0 0])
L2=link([0 250 0 0 0])
L3=link([pi/2 75 0 0 0])
L4=link([pi/2 0 0 290 0])
L5=link([4.7124 0 0 0 0])
L6=link([0 0 0 205 0])

%Searching for FANUC Position and Orientation
Matrix
```

```
fanuc = robot({L1 L2 L3 L4 L5 L6})

%Generate the transform corresponding to a
particular joint coordinate
q=[0 0 0 0 0 0]

%Forward Kinematic
T=fkine(fanuc,q)

Td=transl(-35,0,615)*T

%Pick Position
T1=transl(160,0,-440)*Td

%Inverse Kinematic
q1=ikine(fanuc,T1)

%Place Position
T2=transl(-600,620,20)*T1
%Inverse Kinematic
q2=ikine(fanuc,T2)
```

The outputs for that executed function are as below:

```
L1 =
  1.570796   150.000000 0.000000   0.000000 R
         (std)
L2 =

  0.000000   250.000000 0.000000   0.000000 R
         (std)
L3 =

  1.570796   75.000000  0.000000   0.000000 R
         (std)
L4 =

  1.570796   0.000000   0.000000 290.0000   R
         (std)
L5 =

  4.712400       0.000000    0.000000      0.00000
         R       (std)
L6 =
  0.000000       0.000000  0.000000 205.000000
         R       (std)

fanuc =
noname (6 axis, RRRRRR)
                 grav = [0.00 0.00 9.81]
standard D&H parameters

  alpha      A        theta      D          R/P
1.570796 150.000000  0.000000  0.000000    R(std)
0.000000 250.000000  0.000000  0.000000    R(std)
1.570796 75.000000   0.000000  0.000000    R(std)
1.570796 0.000000    0.000000   290.0000   R(std)
4.712400 0.000000    0.000000   0.000000   R(std)
0.000000 0.000000    0.000000   205.0000   R(std)

q =
     0     0     0     0     0     0

T =
    1.0000         0         0  475.0000
         0   -1.0000    0.0000    0.0023
         0   -0.0000   -1.0000 -495.0000
         0         0         0    1.0000

Td =
    1.0000         0         0  440.0000
         0   -1.0000    0.0000    0.0023
         0   -0.0000   -1.0000  120.0000
         0         0         0    1.0000

T1 =
    1.0000         0         0  600.0000
         0   -1.0000    0.0000    0.0023
         0   -0.0000   -1.0000 -320.0000
         0         0         0    1.0000

qi =

     -0.0000     0.3736     0.1848    -0.0000     0.5583
-0.0000
T2 =
    1.0000         0         0         0
         0   -1.0000    0.0000  620.0023
         0   -0.0000   -1.0000 -300.0000
         0         0         0    1.0000


q2 =

   -4.7124    5.5204   -3.9401   -0.0000    1.5802
1.5708
```

The output result above had shown that to go to the target position (pick) at coordinate [600,0,-320] from [440,0,120], the changes of joint angle as stated in *qi* are as follows:

$\theta_1 = 0°$ $\theta_2 = 21.4°$ $\theta_3 = 10.6°$
$\theta_4 = 0°$ $\theta_5 = 32°$ $\theta_6 = 0°$

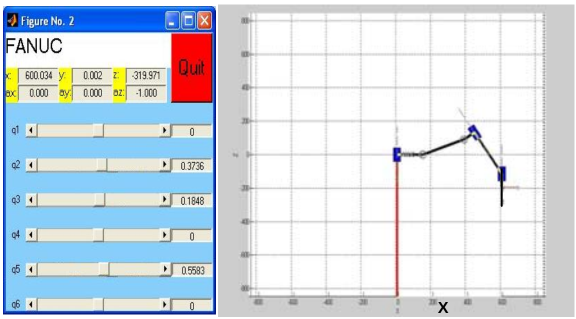
It is also shown that to go to the target position (place) at coordinate [0,620,-300] from [600,0,-320], the changes of joint angle as stated in *q2* are as follows:

$\theta_1 = 270°$ $\theta_2 = 316°$ $\theta_3 = 225°$
$\theta_4 = 0°$ $\theta_5 = 90.5°$ $\theta_6 = 90°$

3 Simulation

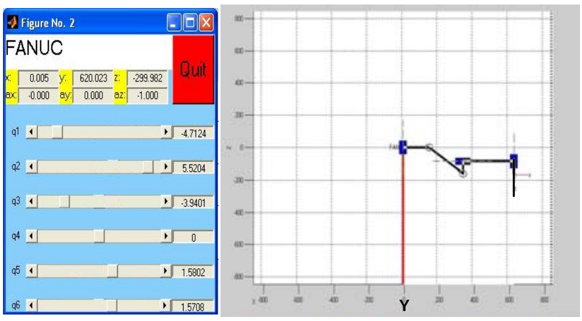
The data from the changing angle on the joint will be simulated using Robotic Tools in Matlab. This is to ensure every joint will produce accurate movement and the required end point.

Figure 1 describes the output of the simulation for the pick position of simulated Fanuc robot. It shown that the end point is located at coordinate [600,0,-320]. While figure 2 shows the output of the simulation for the place position and the end point is located at coordinate [0,620,-300].



x: 600.034 y: 0.002 z: -319.971
ax: 0.000 ay: 0.000 az: -1.000
q1 0
q2 0.3736
q3 0.1848
q4 0
q5 0.5583
q6 0

Figure 1: Simulated movement for picking an object.



x: 0.005 y: 620.023 z: -299.982
ax: -0.000 ay: 0.000 az: -1.000
q1 -4.7124
q2 5.5204
q3 -3.9401
q4 0
q5 1.5802
q6 1.5708

Figure 2: Simulated movement for placing an object.

From the data shown, the default position for the pick and place task is in figure 1 and figure 2. This default position is later described in text as Reference Movement. There are two types of movement that are modified so that it could achieve the same end point but with different joint angles. It is later described as Fixed Movement and Optimised Movement.

4 Energy Measurement

While the robot pick an object at position of [600, 0, -320] and place it at position of [0, 620, -300], the energy of overall task were measured using the Fluke 434 Three Phase Power Quality Analyser meter as shown in figure 3. This analyser meter has 4 BNC-inputs for current clamps and 5 banana-inputs for voltages.

Figure 3: Fluke 434 3-phase power quality analyser.

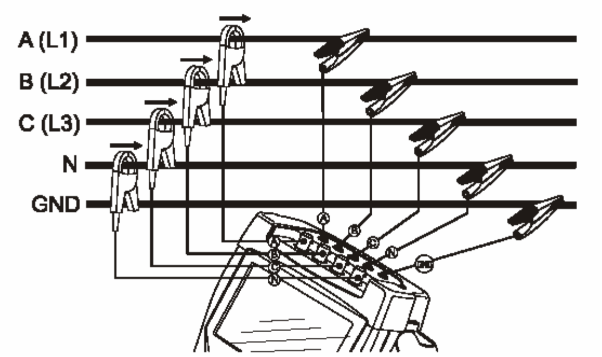


Figure 4: Connection of analyser to 3-phase distribution system.

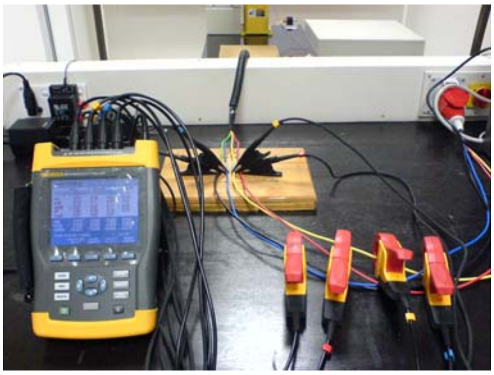


Figure 5: Real connection of analyser to 3-phase supply.

The robot movements for pick and place application involving the use of many motor of its joint and including the energy of its controller that attached together with the robot. So, this measurement was done for total of energy usage by the robot and its controller. The energy was measured at the input cable of the 3-phase power supply. Figure 4 explains how the connections of Analyser are made to the 3-phase distribution system and figure 5 describes the real connection of Analyser to the supply of the robot.

5 Experimental Result

Based on the data and information on simulation results done previously, it is time to test the movement of the manipulator in real application and measure their energy usage. The experiments were divided into three categories of movement which is Reference Movement, Fixed Movement and Optimised Movement. Those were experimented using real movement of Fanuc robot and run simultaneously together with Roboguide software. Each movement encompasses of seven steps from P[1] to P[7].

The steps are as follows:

- P[1] – Home position
- P[2] – Pick object
- P[3] – Back to home position
- P[4] – Rotate 90° left
- P[5] – Place object
- P[6] – Home position of 90° left
- P[7] – Back to home position

Table 1: End point coordinates (mm).

Step	X	Y	Z
P[1]	440	0	120
P[2]	600	0	-320
P[3]	440	0	120
P[4]	0	440	120
P[5]	0	620	-300
P[6]	0	440	120
P[7]	440	0	120

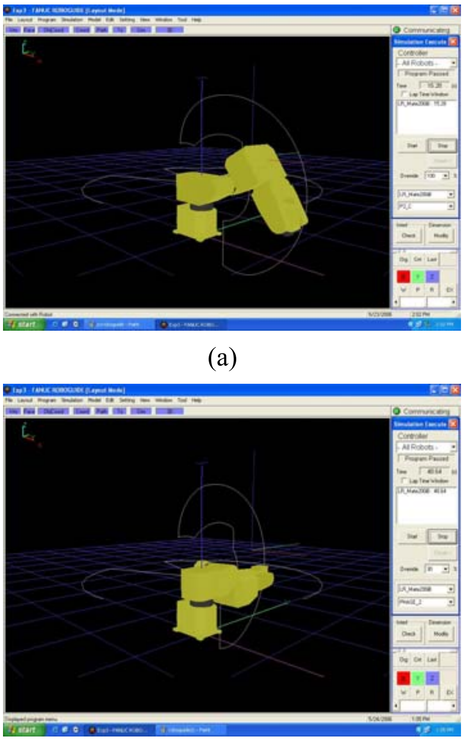
Table 1 shows the position of end point in coordinate format of [X,Y,Z] for each step. Those steps were considered as one cycle of task. While completing the task, the total energy usage by the system were measured and recorded. This measurement depends on process that involves the control of two parameters known as position of joint angle (ɵ) and speed of the motor (ω) of the 3 main axes of the robot. For each category of movement, the experiments are run with

- ω1=50% of motor's speed within one hour
- ω2=100% of motor's speed within 15 minutes.

The experiment of picking and placing the object will be repeated until it reaches the time limit. Within that time frame, the task cycle will be counted and recorded to show the effectiveness of those movements.

5.1 Reference Movement

The first category (Reference Movement) was done following the seven steps above. Figure 6 describes the Reference Movement while picking (P[2]) and placing (P[5]) an object by Roboguide software. This Reference Movement exactly follows the movement of default inverse kinematics result.



(a)

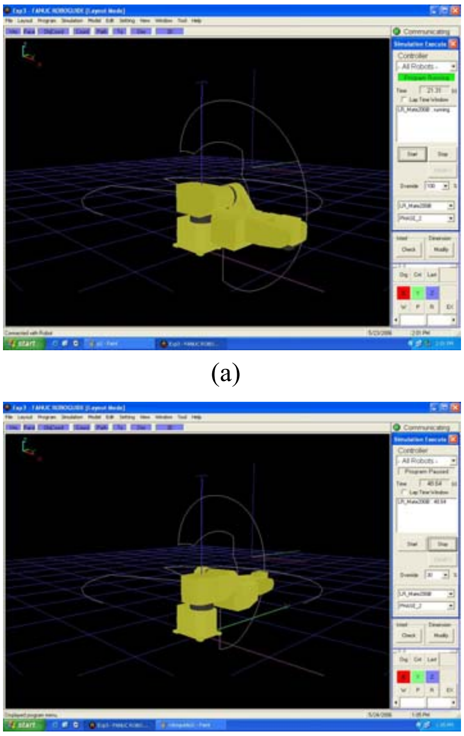
(b)

Figure 6: Reference Movements for (a) picking object, (b) placing object.

5.2 Fixed Movement

This type of movement was fixed to minimise the number of motor usage for the joint angle of 3 main axes to reach the target. So, it involves only the movements of motor for Joint 2.

It was done following the seven steps as stated earlier. Figure 7 describes the reference movement while picking (P[2]) and placing (P[5]) an object. There was a different kind of method on how the robot does the picking task that can be found from figure 6a and figure 7a. However, the same method is used for the robot placing the object as shown in figures 6b and 7b. Whatever it is, the robot can still find the same end point target for picking and placing the object.



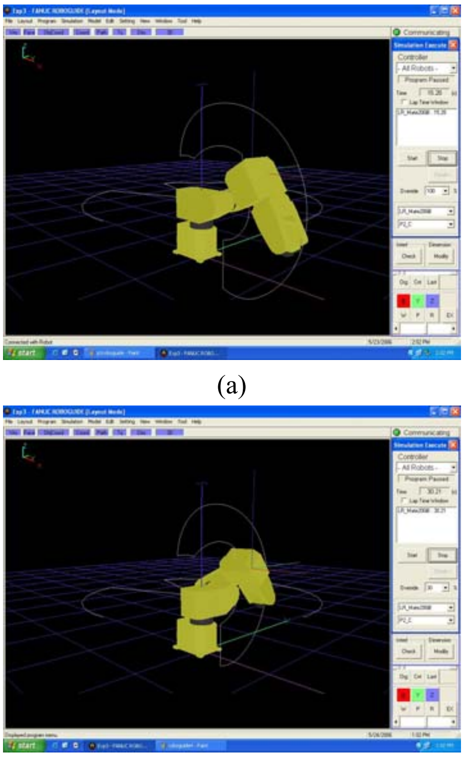
(a)

(b)

Figure 7: Fixed Movement for (a) picking object, (b) placing object.

5.3 Optimised Movement

Figure 8 shows the Optimised Movement for picking and placing an object.



(a)

(b)

Figure 8: Optimised Movement for (a) picking object, (b) placing object.

Table 2: Energy measurement for each movement category.

Movement	**Time**	**Motor speed**	**Cycle**	**Energy measured (kWh)**	**Energy usage (kWh)**	**1 Cycle energy usage (kWh)**	**1 Cycle time**	**Optimisation percentage**
Idle	1 hr	-	-	0.147	-	-	-	-
Reference	1 hr	50%	135	0.192	0.045	0.000333	26.67s	0.00%
Fixed	1 hr	50%	109	0.193	0.046	0.000422	33.03s	-26.61%
Optimised	1 hr	50%	170	0.190	0.043	0.000253	21.17s	24.12%
Idle	15min	-	-	0.037	-	-	-	-
Reference	15min	100%	65	0.057	0.020	0.000312	13.85s	6.54%
Fixed	15min	100%	53	0.055	0.018	0.000344	16.98s	-3.30%
Optimised	15min	100%	81	0.055	0.018	0.000225	11.11s	32.41%

To see the difference from the movement that had been done earlier, this movement were designed to minimise the changing of angle for each joint.

5.4 Performance Criteria

Those experiments were run whilst the number of cycle for completing an hour task were recorded with different speed of the motor. That means all type of the movement will be executed twice with different value of motor's speed. For this experiment, we had chosen the 50% and 100% of motor's speed.

The data on table 2 shows the Energy measurement and time motion taken for completing the cycle and repeating it within specific hour with different speed of the joint's motor.

It is seen that both Optimised Movement of 50% and 100% of motor speed shows the less cycle time taken for completing the task which are 21.17s and 11.11s respectively. It also shown that within one hour of repetitive task, it can produce a number of 170 task cycle which more than the other two movements.

The Optimised Movement can run single task only in 11.11s and a number of 81 cycles within fifteen minutes which produce 32.41% of energy saving depend on the Reference Movement (50% of motor's speed). That's more efficient than Reference Movement and Fixed Movement. It can be stated here that the Optimised Movement can produce better energy optimisation for the entire category.

6 Conclusion

This paper describes that to find the specific angle of FANUC robotic arm with given target coordinate, the inverse kinematics solution needs to be applied. The simulation technique and calculation method much helped to achieve the objective of this project. From the experiments that had been executed, it shows that the movement with less task's cycle time and with the fastest of joint motor's speed is more efficient in their overall dynamic performance. The lesser the motor's movement on a joint, especially on the three main axes, the lesser the time is taken to complete the task and save energy as well as increase the manipulator's performance. This performance can be really optimised by measuring the smallest energy usage for each of the category of movement.

7 Acknowledgements

The authors wish to acknowledge the financial assistance provided by Kolej Universiti Teknikal Kebangsaan Malaysia for the overall support of this research.

8 References

[1] J.J. Craig, *Introduction to Robotics* (3rd edition), Pearson Prentice Hall, USA (1989).

[2] Z. Shiller and S. Sundar, "Design of robotic manipulators for optimal dynamic performance", *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, pp 344-349 (1991).

[3] M.C. Ngunyen and V. Graefe, "Object manipulation by learning stereo vision-based robots", *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Maui, Hawaii, pp 146-151 (2001).

[4] M.C. Ngunyen and V. Graefe, "Self-learning vision-guided robots for searching and grasping objects", *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, pp 1633-1638 (2000).

[5] A. Bowling and O. Khatib, "Design of macro / mini manipulators for optimal dynamic performance", *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp 449-454 (1997).

[6] M. Herman, M. Azmi, S. Marizan and C.S. Horng, "Vision guided manipulator for optimal dynamic performance", *Proceedings of the IEEE Student Conference on Research and Development*, Selangor, Malaysia, pp 147-151 (2006).

[7] M. Herman, M. Azmi and S. Marizan, “Motion optimization of vision guided manipulator for optimal dynamic performance”, *Proceedings of the IEE International Conference on Man-Machine Systems*, Langkawi, Malaysia, pp 43-47 (2006).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Development of a Vision-Based Pick-and-Place Robot

Roneel V. Sharan and Godfrey C. Onwubolu
School of Engineering and Physics,
University of the South Pacific, Suva, Fiji Islands
sharan_r@usp.ac.fj

Abstract

Pick-and-place robots have their applications limited in a system where it is fed with all the information regarding its target. In other words, the robot is given a complete specification of each motion for a manipulation. This work considers the approach taken in automating the manipulation process of an in-house designed pick-and-place robot. It reports the studies of flexible manufacturing concepts using a combination of vision and motion. The vision system differentiates the work-pieces, placed on the work-plane of the robot, with respect to its shape and color and its position and orientation is also determined for manipulation. The pick-and-place robot used for manipulation has been designed as an educational robot in that it realizes pick-and-place operations with limited pay-load. A master control unit (MCU) manages the processing of the vision system and motion planning aspects for manipulation. In addition, a slave unit performs the actual manipulation based on instructions from the MCU.

Keywords: robotics, vision, image, colour, shape, automation

1 Introduction

Robotic manipulators are usually used for repetitive tasks whereby their operations are handled by simple position control strategies. To expand the applications of pick-and-place robots, it is vital to automate its manipulation process [1, 2]. Arguably, vision is thought to be the most valuable sense of automating a robotic system. Integration of a vision system in the workspace of the robot provides improved flexibility for adapting to various task requirements. It makes a robot manipulator more versatile by allowing it to deal with variations in work-piece position and orientation [3].

A vision system analyses and produces descriptions of what is imaged. In general, it takes in raw data and generates description or understanding as an output. The technique in robot vision is a combination of sensing and perception [4, 5]. Sensing is the process of acquiring the image into the computer memory and vision perception refers to the observation, collection, processing, and understanding of information from spatial measurements.

This work considers the approach taken to study flexible manufacturing concepts using a combination of vision and motion. It is carried out as part of development of a smart flexible manufacturing system (SFMS). The SFMS is to include an automatic guided vehicle (AGV) which will be transferring work-pieces, to be either milled or drilled, to the drilling and milling workstations. A pick-and-place robot is employed to pick and place work-pieces to/from the AGV from/to the workstations.

It involves manipulation of work-pieces with shapes ranging from rectangle, circle, and triangle and with possible colours of red, green, blue, yellow, and black using a five degree-of-freedom pick-and-place robot. Information obtained from an overhead mounted CCD camera is processed to differentiate the work-pieces with respect to its shape and colour. The extracted position and orientation information of the work-piece is then utilized to plan the motion of the robot such that the work-piece can be manipulated using its two-fingered gripper.

The work-pieces to be manipulated are placed at arbitrary positions and orientations on the two-dimensional work-plane of the robot covered by the field of view of the camera. On a user selection, using a graphical user interface (GUI), of the shape and colour of the work-piece to be manipulated, the robot, which has been appropriately programmed, then determines the angular rotation of its arms to pick the work-piece from the work-plane and place it at a predefined position. Figure 1 shows the architecture of the overall system.

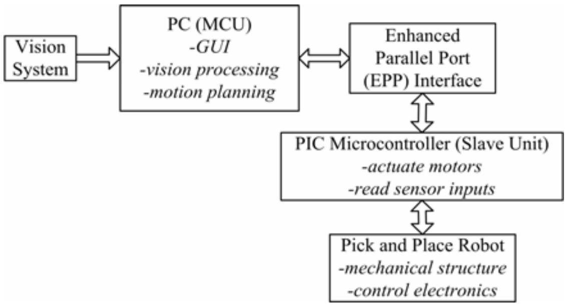


Figure 1: Architecture of the vision-based pick-and-place robot.

The vision system is equipped with an overhead CCD camera and a structured lighting system. It is linked to the MCU which handles the vision processing and motion planning of the robot. A GUI integrates the vision system with the pick-and-place robot and also provides a user interface to the integrated system.

The enhanced parallel port (EPP), through an interface card, is used as the communication medium between the MCU and the slave unit. A PIC microcontroller acts as a slave unit that carries out required motions of the robot based on commands from the MCU.

Moreover, the pick-and-place robot has been built in-house and has been named *RABBIT* (*r*obotic *a*rm for *b*roadly *b*ased *i*nter-station *t*ransfer). It is basically a five degree-of-freedom arm excluding the gripping/releasing movements. The joints of the arm are actuated using stepper motors and DC motors. Sensors in the form of a force sensor and a limit switch handle the gripping and releasing of the work-pieces respectively.

From here on, the paper is divided into the following sections: 2. vision system; 3. pick-and-place robot; 4. experimentation and results and 5. conclusion.

2 Vision System

2.1 Image Acquisition

The vision system includes a Sony digital video camera (model DCR HC42E) connected via the USB port to a PC running under the Windows XP platform. It is based under a structured lighting system such that specular reflections from the work-plane and the work-piece surface and shadows around the work-pieces could be minimized.

Image acquisition is performed in MATLAB using the image acquisition toolbox [6]. The rectangular view as seen by the camera is streamed into the MCU and a single snapshot image is taken for further processing. The captured image is represented using 320 x 240 pixels and is in the RGB colour format. That is, a single pixel is represented using the primary red, green, and blue colour values with 8-bit of information.

2.2 Image Processing

Feature extraction for shape recognition requires that the captured image be represented in binary format. That is, with pixel values of either 0 or 1 indicating either a black or a white pixel respectively. Conversion from the RGB colour format to the binary format follows spatial filtering, grayscale conversion, histogram equalisation, and finally binary conversion.

Spatial filtering performs neighbourhood operations on image pixels where the value of a filtered pixel depends on the value of its neighbouring pixels. This is achieved through correlation where the centre of the correlation filter, determined through trials with different size and coefficients, is superimposed on the pixel to be filtered. The coefficient of the correlation filter is then multiplied with the pixel value it is imposed on and the summation of the individual products estimates the value of the filtered pixel.

The filtered RGB image is then converted to a grayscale image, also known as an intensity image. It is in this case represented using 8-bit information where 0 represents black and 255 represents white. A grayscale image often has most of its intensity values concentrated within a particular range. Histogram equalization is performed to enhance the contrast of the grayscale image by transforming the intensity values of the grayscale image so that the histogram of the output image has the intensity values evenly distributed.

This enhanced image is finally converted to a binary image using a threshold where values greater than and equal to the threshold are converted to a pixel value of 1 (white) and values below the threshold are converted to a pixel value of 0 (black). A sample captured image with its corresponding binary image is shown in figure 2a and figure 2b respectively.

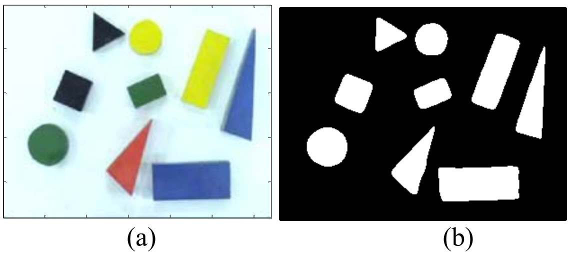

(a) (b)

Figure 2: (a) RGB format colour image and (b) its corresponding binary image.

The binary image is then segmented into different regions representing the number of objects, termed as image segmentation. This is done by determining the number of white pixels that are connected to each other which in turn denotes a connected component or object. The image processing steps are implemented using the image processing toolbox in MATLAB [7].

2.3 Feature Extraction

2.3.1 Shape Recognition

Based on the range on shapes to be differentiated, the feature of corner detection is utilized for shape recognition. This is a very realistic approach to the problem of shape recognition because it is usually the number of sides or corners that humans use for differentiating shapes. A corner, as defined by Jain [8], is a location on the boundary of an object where the curvature becomes unbounded and is defined as

$$|k(t)|^2 = \left(\frac{d^2 y}{dt^2}\right)^2 + \left(\frac{d^2 x}{dt^2}\right)^2 \qquad (1)$$

where *t* represents the distance along the boundary of the segmented regions and a corner is declared whenever |*k*(*t*)| assumes a large value.

Equation (1) can be modified for a digital binary image. If

$$\Delta x_z = x_{z+s} - x_z \tag{2}$$

$$\Delta y_z = y_{z+s} - y_z \tag{3}$$

where *s* is the sample length of the curvature in pixels, and with the approximation that

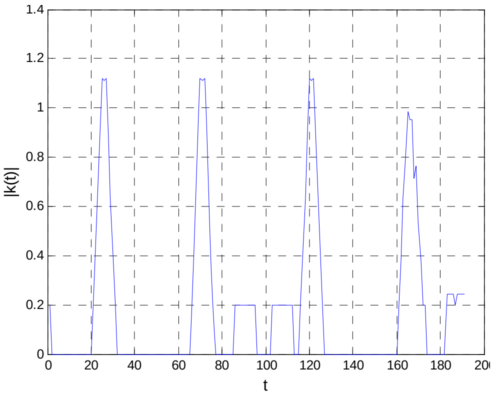
$$\Delta t_z = \sqrt{\Delta x_z{}^2 + \Delta y_z{}^2} \tag{4}$$

then the curvature of a digital binary image with boundary pixels *z* = 1, 2, 3, ..., *Z*, for a region, is given as

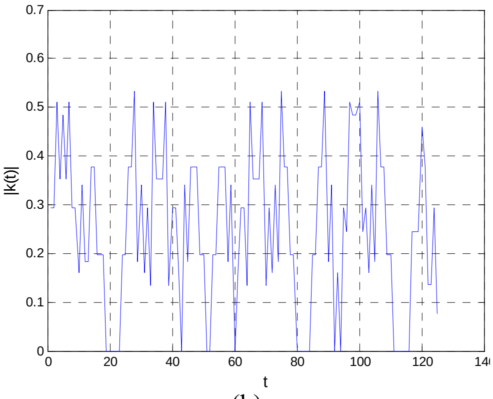
$$\left|k(t)\right|^2 = \left(\frac{\Delta y_{z+1}}{\Delta t_{z+1}} - \frac{\Delta y_z}{\Delta t_z}\right)^2 + \left(\frac{\Delta x_{z+1}}{\Delta t_{z+1}} - \frac{\Delta x_z}{\Delta t_z}\right)^2 . \tag{5}$$

A sample length of five pixels was chosen, through trials with different sample lengths, for best results. The curvature for a sample rectangle, circle, and triangle sampled at five pixels is shown in figures 3a, 3b and 3c respectively.

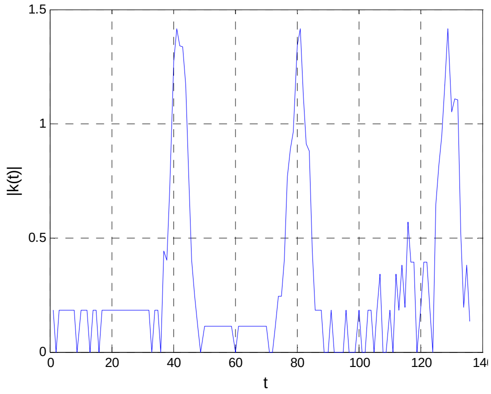
The four and three outstanding peaks in figures 3a and 3c correspond to the corners of a rectangle and a triangle respectively. For a circle, figure 3b, the curvature is nearly constant implying the non-presence of corners.



(a)



(b)

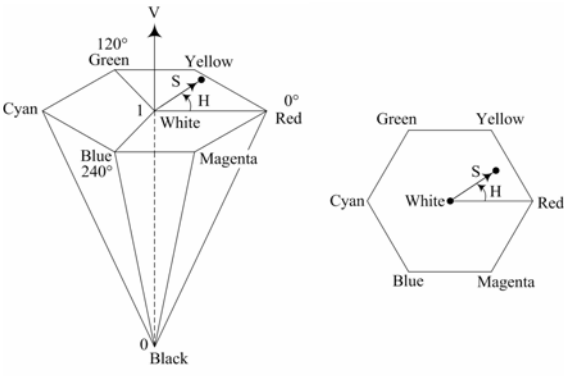


(c)

Figure 3: Curvature for sample of (a) rectangle, (b) circle and (c) triangle.

2.3.2 Colour Recognition

Various colour spaces are used for representing colour images for image processing. Colour spaces are where colours are specified by points in three-dimensional spaces. The colour spaces HSV (hue, saturation, and value) and HSI (hue, saturation, and intensity) are a more natural way to how humans perceive colour and is more often used for colour image processing than any other colour space. Due to its compatibility with the image processing toolbox, the HSV colour space, shown in figure 4, has been preferred.



(a) (b)

Figure 4: (a) The HSV colour model; (b) hue and saturation in the HSV colour model.

The HSV colour model has colours defined inside a hexcone as shown in figure 4a with the positioning of the hue and saturation in the model, for an arbitrary point, shown in figure 4b. *Hue* describes the colour type given as an angle from 0° to 360°. Typically 0° is red, 60° is yellow, 120° is green, 180° is cyan, 240° is blue, and 300° is magenta. *Saturation*, the purity of a colour or the amount of white added to the colour, has values from 0 to 1 where *S* = 1 specifies a pure colour, that is, no white. *Value* is referred to as the brightness of a colour and ranges from 0 to 1, where 0 is black. The transformation from the RGB colour space to the HSV colour space is given by Rogers [9].

2.4 Feature Classification

Feature classification aims to categorize an object within an image to a particular feature class from a number of possible feature classes. A feature class is defined as a group of objects that share common feature or properties. The classification system is operated in two modes: training (learning) and classification (testing). In the training mode, a database of images is built that contains known work-pieces. The knowledge gained from the extracted features forms the database for known work-pieces with which the classifier is trained to distinguish the different feature classes in the feature space.

2.4.1 Training

Since the number of shapes and colours to be differentiated is three and five respectively, there are fifteen different possible feature classes. For shape recognition, the a priori knowledge is utilized that the number of corners for a rectangle, triangle, and circle are always four, three, and zero respectively. However, for colour recognition, the H, S, and V colour values are obtained through samples of forty-five work-pieces to represent a feature class.

The mean RGB and HSV colour values are given in table 1. Also, the ideal RGB values for the five colours are given.

Table 1: The ideal RGB colour values and the mean RGB and HSV colour values.

Colour	Ideal RGB colour values	Mean RGB colour values	Mean HSV colour values
Red	[255, 0, 0]	[173.8, 64.0, 47.6]	[0.0271, 0.7260, 0.6816]
Green	[0, 255, 0]	[44.2, 88.1, 57.9]	[0.3850, 0.4984, 0.3454]
Blue	[0, 0, 255]	[47.7, 75.2, 142.9]	[0.6184, 0.6661, 0.5603]
Yellow	[255, 255, 0]	[176.5, 192.8, 19.7]	[0.1824, 0.8967, 0.7561]
Black	[0, 0, 0]	[45.5, 49.3, 46.3]	[0.5354, 0.2304, 0.2245]

2.4.2 Testing

In the testing phase, an input feature vector is assigned to one of the fifteen feature classes. Due to the distinct nature of the feature classes, the simplest approach of minimum distance classification is utilized for feature classification. A minimum distance classifier assigns an unknown feature vector based on the minimum distance between the unknown feature vector and each of the feature classes. The distance here refers to the Euclidean distance given as

$$d_j(\mathbf{x}) = \left\| \mathbf{x} - \mathbf{m}_j \right\| \qquad j = 1, 2, \ldots, W \tag{6}$$

where $\|\mathbf{a}\| = (\mathbf{a}^T\mathbf{a})^{1/2}$ is the Euclidean norm, **x** denotes the unknown feature vector and $\mathbf{m}_j$ denotes the *j*th feature class [10]. Hence, a feature class **x** is assigned to class w_i if $d_i(\mathbf{x})$ is the smallest distance.

3 Pick-and-Place Robot

3.1 Mechanical Design

The pick-and-place robot has been designed and built in-house. It is designed for a payload of 500 *g* and is constructed of aluminium due to its light weight. The joints of the robot include the base, shoulder, elbow, wrist, and gripper (rotates about the wrist). The gripper is used as an end-of-arm-tool (EOAT) which grips and releases work-pieces. The operation of the gripper is based on the principle of lead screw. One finger of the gripper is fixed while the second finger moves along the lead screw as it is rotated.

The calculated values of the torque for all the joints are given in table 2.

Table 2: Torque considerations for each movement.

Joint	Length/radius of joint link (*mm*)	Torque (*Nm*)
Base	$R_B = 75$	7.79
Shoulder	$L_S = 181$	2.79
Elbow	$L_E = 185$	1.87
Wrist	$L_W = 150$	0.66
Gripper (rotate)	$R_G = 25$	0.66
Grip/release	-	0.013

Stepper motors, due to their good position integrity, are utilized for rotating the base, shoulder, elbow, and the wrist. All the stepper motors have been mounted on the base so that the torque on the joints is reduced. Transformation of the torque from the motors to the joints is performed using timing belts and timing pulleys.

However, the DC motors to rotate the gripper about the wrist and to open/close the gripper have been mounted on the gripper itself. The gripper is also equipped with a force sensor and a limit switch that govern the gripping and releasing of a work-piece respectively.

The robot operates on a work-plane with dimensions of 348.14 *mm* and 261.11 *mm*. This is determined through the maximum reach of the robot and the ratio of the *x* and *y* pixels (320:240 = 4:3) of the captured image.

3.2 Electronics Design

An EPP interface card [11] is utilized for communication between the MCU and the slave unit for this system. The EPP interface card offers two 8-bit ports of input and output. A PIC 16F877 microcontroller is used as a slave unit which operates based on instructions from the MCU. These instructions are in the form of the joint to be actuated, its direction of rotation, and the angle it has to be rotated.

The four-phase, 1.8°, hybrid stepper motors are driven using unipolar drive boards, commercially available driver boards for stepper motors. In addition, the light duty DC motors are controlled using the quadruple half-H driver (SN754410) IC.

Furthermore, the force sensor and the limit switch are mounted on the gripper. The output of the force sensor is resistance whereby applying a force to the active sensing area causes the resistance at the terminals of the force sensor to decrease. Using voltage division, this output resistance is then converted to a voltage between 0 V and 5 V which, based on calibration, depicts a force between the work-piece and gripping fingers.

3.3 Motion Planning

Motion planning of the robot is done based on the predefined home (initial) and place positions while the pick position is determined after analysis of the captured image. The centroid of the work-piece in pixels (x_p, y_p) is first transformed into the dimensions of the work-plane (x_w, y_w) and then to the C-space of the pick-and-place robot (x_r, y_r) as shown in figure 5. The z component is always constant as the gripper points vertically downwards when picking or placing work-pieces.

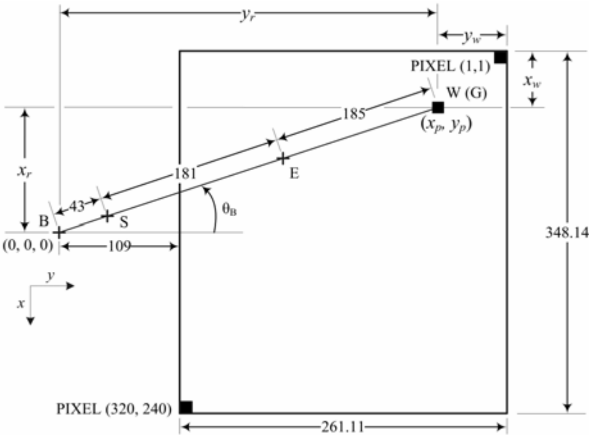


Figure 5: Representation of the pick position.

The angular rotation of the base is then defined as

$$\theta_B = \tan^{-1}\left(\frac{x_r}{y_r}\right). \tag{7}$$

Inverse kinematics is utilized to determine the angular positions of the shoulder, elbow, and wrist at the pick position as shown in figure 6.

The angle that the elbow link makes with the shoulder link, θ_E, is then given as

$$\theta_E = 180° - \beta \tag{8}$$

where

$$\beta = \cos^{-1}\left(\frac{H^2 - \left(L_S^2 + L_E^2\right)}{-2L_S L_E}\right) \tag{9}$$

and with reference to figure 5,

$$H = \sqrt{x_r^2 + y_r^2} - 43. \tag{10}$$

where 43 is the distance between the base joint and the shoulder joint.

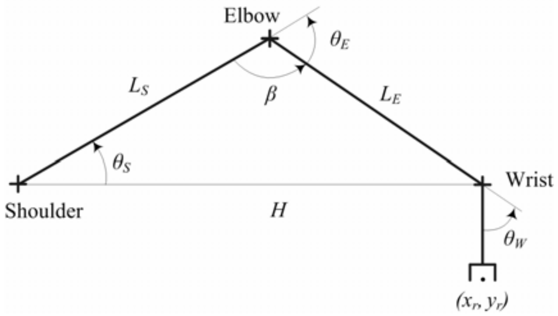


Figure 6: Angular position of the shoulder, elbow and wrist at the pick position.

Using the sine formula, the angle that the shoulder link makes with the xy-plane is given as

$$\frac{\sin\theta_S}{L_E} = \frac{\sin\beta}{H}$$
$$\theta_S = \sin^{-1}\left(\frac{L_E \sin\beta}{H}\right). \tag{11}$$

The angle that the wrist link makes with the elbow link is now defined as

$$\theta_W = 90° - \left(\theta_E - \theta_S\right). \tag{12}$$

The flowchart for motion planning is illustrated in figure 7 where n and N denote the number of work-pieces manipulated and the number of work-pieces in the specified class.

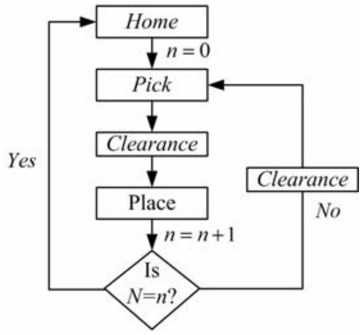
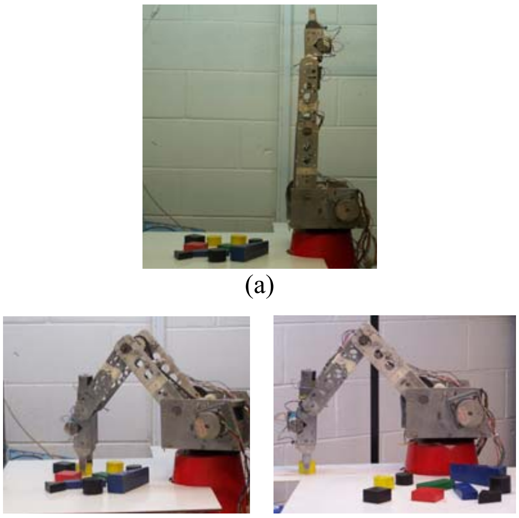


Figure 7: Motion planning of the robot.

From the home position, the robot picks the first work-piece, and moves through a clearance position to the place position. A clearance position is utilized since the pick and the place heights are same and it is not feasible to directly rotate the base, with the arm, to/from the pick and place positions. The pick-to-place and place-to-pick motions will continue until all the work-pieces of the selected class are manipulated, and then the robot returns to the home position. Figures 8a, 8b and 8c show the position of RABBIT at the home, pick and place positions respectively.



(a)

(b) (c)

Figure 8: The robot at (a) home, (b) pick and (c) place positions.

The robot operates in a calibrated environment. That is, the camera has been focused to the field of view of the rectangular work-plane. In addition, the joints, actuated using the stepper and DC motors, work based on calibration where a certain number of pulses imply a certain degree of movement of the arm link. However, calibration is not used for the gripper motor as it is controlled by the sensors mounted on it.

4 Experimentation and Results

The vision system gives 100% accuracy for both shape and colour recognition for the range of work-pieces considered for this work. For the pick-and-place robot, the accuracy of the joints is given in table 3. The elbow joint has two operation regions since it had different calibration relationship in its overall operation region.

Table 3: Results for the accuracy of the joints.

Movement / rotation		Error (°)	Repeatability (±3σ)
Base		$e_B = 1.18$	±0.27
Shoulder		$e_S = 0.59$	±0.30
Elbow	Region 1	$e_{E1} = 2.45$	±0.33
	Region 2	$e_{E2} = 1.58$	±0.27
Wrist		$e_W = 1.29$	±0.42

5 Conclusion

The automation of the manipulation process for a pick-and-place robot in its work-cell has been conveyed in this paper. Specially, the integration of vision in the robot work-cell and its implication in improving the flexibility of manipulation has been discussed. This flexibility can help eliminate the use of expensive and application specific manipulation in a variety of industrial applications. The work carried out has been very effective in studying flexible manipulation approaches using a combination of vision and motion.

6 Acknowledgements

This work was approved and funded by the Faculty of Science and Technology (FST) Research Committee and the University Research Committee (URC) as part of a Master of Science in engineering course at the University of the South Pacific (USP).

7 References

[1] T.L. Perez, J.L. Jones, E. Mazer and P.A. O'Donnell, "Task-level planning of pick-and-place robot motions", *IEEE Computer*, 22(3), pp 21-29 (1989).

[2] M.W. Spong, "Motion control of robot manipulators", in W. Levine (ed.), *Handbook of Control*, CRC Press, pp 1339-1350 (1996).

[3] A. Nagchaudhuri, S. Kuruganty and A. Shakur, "Mechatronics education using an industrial SCARA robot", *Proceedings 7th. Mechatronics Forum International Conference and Mechatronics Education Workshop (Mechatronics 2000)*, Atlanta, GA (2000).

[4] G.W. Awcock and R. Thomas, *Applied Image Processing*, McGraw Hill, New York (1995).

[5] P.G. Ranky, "Advanced machine vision systems and application examples", *Sensor Review*, 23(3), pp 242-245 (2003).

[6] Mathworks, *Image Acquisition Toolbox for Use with MATLAB User's Guide*, The Mathworks Inc. (2005).

[7] Mathworks, *Image Processing Toolbox for Use with MATLAB User's Guide*, The Mathworks Inc., (2004).

[8] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, New Jersey (1989).

[9] D.F. Rogers, *Procedural Elements of Computer Graphics* (2nd edition), McGraw Hill, New York (1997).

[10] R.C. Gonzalez, R.E. Woods and S.L. Eddins, *Digital Image Processing using MATLAB*, Pearson Prentice Hall, New Jersey (2004).

[11] G.C. Onwubolu, S. Aborhey, R. Singh, H. Reddy, M. Prasad, S. Kumar and S. Singh, "Development of a PC-based computer numerically controlled drilling machine", *Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture*, 216(B1), pp 1-7 (2002).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Reconstructed Surface Smoothing Using Tsallis Entropy

Hong Zhou, Yonghuai Liu
Department of Computer Science,
University of Wales, Aberystwyth, UK
hhz03@aber.ac.uk, yyl@aber.ac.uk

Abstract

3D modelling finds a wide range of applications in industry robotics. However due to the presence of surface scanning noise and image registration and fusion error, the finally reconstructed surfaces are often distorted and thus present obstacles to their applications. In this paper, we employ the entropy maximization (EntMax) principle in conjunction with the Tsallis entropy from statistical mechanics to optimize the mesh node locations and normals without altering the mesh node connectivity. A comparative study based on real images shows that the proposed algorithm is easy to implement and significantly smoothes the rough surfaces, while their geometric details are desirably kept.

Keywords: 3D modelling, reconstruction error, entropy maximization, Tsallis entropy, surface smoothing.

1 Introduction

3D modelling finds a wide range of applications in areas such as simultaneous localization. The quality of reconstructed surface models is of vital importance for reliable path planning and obstacle avoidance in robot navigation However, the finally reconstructed surfaces are usually rough in the smooth area of real surfaces due mainly to a rapid change of orientations and vertex locations of reconstructed surfaces caused by noise introduced in the process of surface scanning and image registration and integration. Therefore how to effectively filter noise from the reconstructed surface models, while preserving their desired details, is thus an active area of research.

1.1 Previous Work

Current smoothing methods can be classified into two main categories, namely Laplacian smoothing and optimization-based smoothing. Among these methods, Laplacian smoothing is the most commonly used [1, 9, 7]. Laplacian smoothing in its simplest form consists of recursively estimating an optimal position of each node as the average of the nodes connected to it. This technique generally works quite well for flat area in meshes. However, it can result in distorted or even inverted elements near concavities in the model.

Optimization-based smoothing [3, 6] techniques measure the quality of the surrounding elements of a node and attempt to optimize by computing the local gradient of the element quality with respect to the node location. The node is moved in the direction of the increasing gradient until an optimum value is reached. This optimum value is often defined as maximizing the minimum of mesh quality metrics such as the minimum interior angle of triangles and the internal angles at the internal points.

Based on difference of the smoothing elements, the above three main smoothing approaches can also be classified into two categories. One is to adjust vertex positions so that the overall surface becomes smoother [9, 7, 6]. The other is first to smooth the vertex normals and then to optimize vertex locations based on the smoothed normals [1].

1.2 Tsallis Entropy

While the traditional Shannon entropy: $E_S = -\sum_i w_i \ln w_i$ has been widely used in communications, hardly can it always succeed in describing the probability distribution w_i of the various microstates in statistical mechanics and thermodynamic systems. Thus the Shannon entropy was extended by C. Tsallis in 1988 [10] through introducing a parameter q into the entropy definition: $E_T^q = (1 - \sum_i w_i^q)/(q-1)$ where q is any positive real number except $q = 1$. Actually, $\lim_{q \to 1} E_T^q = E_S$. The generalization is considered to be one of the most viable and applicable candidates for formulating a theory of non-extensive thermodynamics.

One of the most attractive characteristics associated with the Tsallis entropy [10]: $E = (1 - \sum_i w_i^q)/(q-1)$ is the fact that it provides an easy access through the entropy maximization principle [5] to a rich set of probability distribution functions. Moreover, since the parameter q is introduced, the entropy can be adjusted more freely and q can be predicted in some specific applications, the system can be more accurately analysis and understood [11].

1.3 Our Work

While the most existing surface smoothing algorithms [1, 9, 7] are developed to mainly deal with scanning noise, in this paper, we develop a novel algorithm that can deal with both scanning noise and registration and fusion errors. While scanning noise can be modelled as Gaussian white noise, registration and fusion errors depend on specific registration and fusion algorithms and image data. Thus, while the former is relatively easy to handle, the latter is challenging.

Assume that 3D object computer models are represented as triangular meshes, the aim of surface smoothing in this paper is to optimize the meshes so that the edges of triangles have similar lengths and neighbouring vertices have similar normals. To this end, we construct objective functions minimizing the variation of triangle edge lengths and the normal variation of neighbouring vertices. However, such strategy is too crude and often leads the surface details to be smoothed out, since it equally treats all the edges of triangles and vertex normals in the mesh without considering the details they convey. Thus, a real number w in the unit interval has to be used to represent the extent to which the edges of triangles are of equal length and the normals of neighbouring vertices are similar. Consequently, weighted squared triangle edge length variation and neighbouring vertex normal variation need to be optimized for the optimal positions of mesh nodes without altering the original node connectivity and the optimal normals of vertices.

In order to estimate the parameter w, we employ the entropy maximization (EntMax) principle [5] in conjunction with the Tsallis entropy from statistical mechanics [10]. Since the EntMax principle leads to a least biased estimate possible on the given information, good results are expected.

The rest of this paper is organized as follows: reconstruction noise is described in section 2, the mathematical foundation is built and two smoothing methods are developed in section 3, while the experimental results are presented in section 4. Finally, some conclusions are drawn in section 5.

2 Noise Characterization

Most existing smoothing methods are developed to mainly deal with scanning noise [9, 7, 13, 1]. However, when a 3D model is reconstructed from several data sets such as range images, the surface reconstruction noise has also been unavoidably introduced due to the fact that accurate reconstruction often requires that the image registration error be an order of magnitude less than the measurement error, the actual automatic registration algorithm at best can only produce an average registration error of 1/3 or 1/2 scanning resolution. Inaccurate image registration and improper data fusion in the detected overlapping area between neighbouring views thus shift points from their actual positions and create artefacts in the reconstructed surfaces. As a matter of fact, reconstruction noise imposes more severe impact on the quality of reconstructed 3D object models since such noise usually distorts the shape of object as demonstrated by the figure 1.

As shown in the figure 1, by comparing the more accurate reconstructed surface on the left hand side with the one on the right hand side, it can be observed that while the former generates a limited number of artefacts on the head and tail of bird, the latter did not accurately reconstruct the bird model. While the lower mandible and breast are smooth, the upper mandible, head, wings, tail, and legs are not. The surface becomes uneven and more artefacts appear in areas with high curvature. Therefore the reconstruction noise needs special treatment for smoothing algorithms.

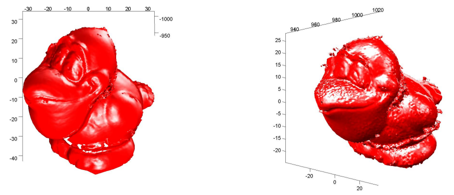


Figure 1: Left: More accurately reconstructed surface [12]. Right: Reconstructed model using the volumetric method [2].

3 Tsallis Entropy Based Smoothing Algorithm Development

In this section we develop new methods to deal with both scanning noise and reconstruction error. The new methods have ability not only to optimize location of vertices but also to filter noise in vertex normals. To this end, we employ the Tsallis entropy [10] in conjunction with the entropy maximization principle [5].

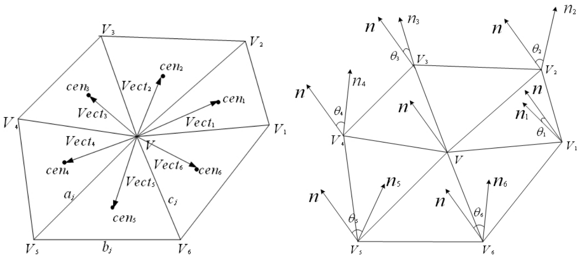


Figure 2: The principle of vertex location smoothing(left) and normal smoothing(right).

3.1 Vertex Location Optimization

Firstly, vertex location optimization is considered. If a surface is smooth and optimized, then each triangle should be as similar to equilateral one as possible and their edge lengths should be as equal to each other as possible. Then the edge length variation can be used to characterise the dissimilarity between the existing triangles and optimal ones:

$$\|\Delta L_{ij}\| = \|Max(a_j, b_j, c_j) - Min(a_j, b_j, c_j)\|^2 \qquad (1)$$

where a_j, b_j, c_j are the edge lengths of the triangle tri_j.

However, this strategy is too crude and it did not take into account the details of object surface represented as varied sizes, shapes, and orientations of triangles. Thus, a positive real number W_{ij} in the unit interval is introduced to characterize the extent to which the edges of triangles are of equal length. For this purpose, the following objective function is then constructed: $J_{EV} = \sum_{i=1}^{M}\sum_{j=1}^{N} W_{ij}\|\Delta L_{ij}\|$ where M is the number of vertices in the mesh and N is the number of incident triangles of the vertex V_i. In order to estimate the parameter W_{ij}, we employ the entropy maximization principle [5] and thus construct a new objective function, consisting of the cost J_{EV} minus the Tsallis entropy of these unknowns, as follows:

$$J = \sum_{i=1}^{M}\sum_{j=1}^{N} W_{ij}\|\Delta L_{ij}\| - \frac{1}{\beta}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{W_{ij} - W_{ij}^q}{q-1} \qquad (2)$$

where parameter β is the Lagrange multiplier controlling the relative contribution of the Tsallis entropy to the constructed objective function.

In order to estimate the unknown W_{ij}, we get the first order derivative of this objective function and set the result equal to zero:

$$\frac{\partial J}{\partial W_{ij}} = \|\Delta L_{ij}\| + \frac{1}{\beta(q-1)}(qW_{ij}^{q-1} - 1) = 0$$

Thus $W_{ij} = (\frac{1-\beta(q-1)\|\Delta L_{ij}\|}{q})^{\frac{1}{q-1}}$. In order to make sure $1 - \beta(q-1)\|\Delta L_{ij}\| \geq 0$, $0 < q < 1$ is required. In this case, vertices in a triangular mesh are simulated as a superextensive system.

Then the optimal position of vertex V_i is estimated as:

$$V_{inew} = V_i + \frac{1}{\sum_{j=1}^{N} W_{ij}}\sum_{j=1}^{N} W_{ij} * \vec{Vect}_{ij}, \qquad (3)$$

where $\vec{Vect}_{ij} = V_i - cen_{ij}$ and cen_{i1},cen_{i2} ,cen_{i3} $\cdots$, cen_{iN} are the centroids of incident triangles Tri_{ij} of vertex V_i. Equation 3 shows that the more similar a triangle is to a equilateral one, the smaller the parameter $\|\Delta L_{ij}\|$ is, the larger the weight W_{ij}. In this case, $\vec{Vect}_{ij}$ of non-equilateral triangle will not make a great contribution to the update of vertex location. Such property is useful to prevent details of surface from being smoothed out. Both parameters β and q control the relative contribution of different triangles. The principle for vertex location optimization is shown on the left hand side of figure 2. All vertices are optimized iteratively without changing neighbourhood information until the surface is desirably smoothed.

3.2 Vertex Normal Optimization

The accuracy of surface orientation estimation is very important for the final surface rendering quality. In this section, we follow the same methodology as vertex location optimization described in the last section for vertex normal optimization. The normal of each vertex in the mesh is firstly calculated by averaging the normals of all the triangles weighted by their areas [4] that share the vertex.

If a surface is smooth, then the normal of each vertex should be consistent with those of its neighbours. For each vertex V_i and its neighbouring vertices V_{i1},V_{i2} ,V_{i3} $\cdots$, V_{iN}, let the including angles in degrees between normal vectors N_i at V_i and N_{ij} at V_{ij} be $\|\Delta\theta_{ij}\|$. So for smooth surface, the weighted normal variation J_{OV} = $\sum_{i=1}^{M}\sum_{j=1}^{N}$ W_{ij} $\|\Delta\theta_{ij}\|$ should be minimum where M is the number of vertices in the mesh, N is the number of neighbouring vertices of a vertex and the parameter W_{ij} is in the unit interval indicating the extent to which the normals of neighbouring vertices are desired to be similar.

In order to estimate the parameter W_{ij}, we again employ the EntMax principle and construct the following new objective function that consist of the weighted normal variation minus the Tsallis entropy of these unknowns:

$$J = \sum_{i=1}^{M}\sum_{j=1}^{N} W_{ij}\|\Delta\theta_{ij}\| - \frac{1}{\beta}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{W_{ij} - W_{ij}^q}{q-1} \qquad (4)$$

where parameter β is again the Lagrange multiplier controlling relative contribution of the Tsallis entropy to the constructed objective function.

Then we again get the first order derivative of this objective function and set the result equal to zero:

$$\frac{\partial J}{\partial W_{ij}} = \|\Delta\theta_{ij}\| + \frac{1}{\beta(q-1)}(qW_{ij}^{q-1} - 1) = 0$$

Thus $W_{ij} = (\frac{1-\beta(q-1)\|\Delta\theta_{ij}\|}{q})^{\frac{1}{q-1}}$. Finally, the new normal N_i at vertex V_i is updated as a weighted sum of N_{ij}: $N_{inew} = N_i + \frac{1}{\sum_{j=1}^{N} W_{ij}}\sum_{j=1}^{N} W_{ij} * N_{ij}$ subject to normalization: $N_{inew} = \frac{N_{inew}}{\|N_{inew}\|}$. In order to make sure $1 - \beta(q-1)\|\Delta\theta_{ij}\| \geq 0$, $0 < q < 1$ is required. In this case, vertex normal in a triangular mesh are simulated as a superextensive system. This update shows that the more similar the normals of neighbouring vertices is to that of vertex of interest, the larger the weight W_{ij}. In this case, vertex normals that deviate considerably from that of interest will not make a great contribution to the update of vertex normal. Such a property is again useful to prevent details of surface from being smoothed out. Both parameters β and q control the relative contribution of different neighbouring vertex normals.

The principle of normal smoothing is illustrated in the right hand side of figure 2.

After normals of all vertices are optimized, the new vertex V_i is updated according to: $V_{inew} = V_i + \frac{1}{\sum_{j=1}^{N} W_{area_{ij}}} \sum_{j=1}^{N} W_{area_{ij}} (\vec{Vect_{ij}} \cdot N_{inew}) N_{inew}$ where $W_{area_{ij}}$ is the area of incident triangles Tri_{ij} of vertex V_i [1].

4 Experimental Results

To detail the advantage and disadvantage of the novel algorithm for smoothing the reconstructed 3D object computer models, the other three classical smoothing methods [13, 9, 7] are also implemented. To compare different algorithms and different noise sources, the following parameters are defined: (1) The distortion metric: the area of a triangle divided by the sums of the squares of the lengths of its edges and then normalized by a factor $2\sqrt{3}$. The value of distortion metric is in the unit interval $[0,1]$. The higher the distortion metric value, the higher the smoothing quality of the surface; (2) The distribution of interior angles of triangles. The angle distribution shows the global optimal degree of triangles. The closer the interior angles of triangles are to $60°$, the more similar they are to equilateral ones; (3) The computational time; and finally (4) The number of iterations. The experimental results are presented in figures 3 through 5 and table 1.

4.1 Vertex and Normal Smoothing

To compare the impact of surface orientation on 3D rendering, a single lobster surface rendered from its range image with noisy vertices and normals is used. The experimental results are presented in figure 3, showing that even the vertices are still corrupted by Gaussian white noise, after the new method has smoothed the noisy normals, the rendered surface is much smoother. Next, the noisy vertices are updated using the method [1] that first optimizes vertex normals and then locations. It is clear that the surface with smoothed normals and noisy vertices is very similar to that with smoothed normal and smoothed vertices. Not only do both of them look smoother but also are the details of surface kept. The bottom right figure is smoothed by firstly optimizing vertex location and then recalculating the surface orientation. The resulting surface is smooth but some details such as the abdomen of the lobster becomes blurred. This shows that details of a surface can be determined more easily and effectively by using surface normals rather than by using vertex positions, thus the surface normals impose a greater impact on the model's perceived quality.

4.2 Different Smoothing Algorithms

In this section we test them in handling the reconstruction noise. The mesh based reconstruction algorithm [8] and the volumetric reconstruction algorithm [2] were implemented to produce the noisy reconstructed teletubby and dinosaur surfaces with registered range images. The noisy reconstructed teletubby and dinosaur surfaces are shown in the left upper corners of figures 4 and 5 respectively. The telebubby surface was reconstructed from 20 range images and contains 33478 faces and 17243 vertices. The dinosaur surface was reconstructed from 8 range images and contains 88444 faces and 43965 vertices. The mesh based integration method [8] reconstructs the surface with the original meshes from different range images. So if the registration error is large, as is the case for the registration of the teletubby images, the connected meshes will depart away from each other, the final reconstructed surface will not be smooth, as illustrated by the chest and arms of teletubby. For the volumetric based method [2], since the voxel size is difficult to choose and data sampling is non-uniform, a lot of artefacts appear on the reconstructed surface, as illustrated by the profile of dinosaur.

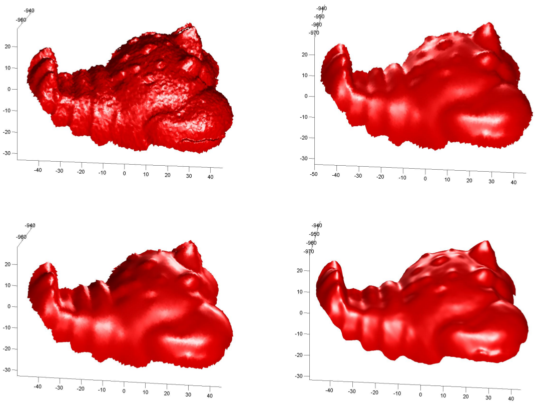


Figure 3: Top left: Noisy Lobster due to Gaussian white noise with a standard deviation of $\delta = 0.2$. Top right: Smoothing only vertex normals. Bottom left: Smoothing first vertex normals and then locations. Bottom right: Smoothing first vertex locations and then recalculating their normals.

The experimental results for smoothing the reconstructed teletubby and dinosaur surfaces are presented in figures 4 and 5 and table 1. figures 4 and 5 and table 1 show the new method perform best. While the distortion metric values of two surfaces are increased from 0.81 and 0.75 to 0.92 and 0.90 with 5 and 4 iterations. Meanwhile although the original Laplacian method [13] also produced the high distortion metric value 0.91 and 0.90 after 4 iterations, the original Laplacian method blurred the details of the chest of the teletubby and the profile of the dinosaur, as shown in the middle right of figure 4 and 5. However, after 7 iterations, the signal processing method [9] did not produce smooth teletubby and dinosaur surfaces with the distortion metric values increased only to 0.86, 0.80. The reason is that since the signal processing smoothing method [9] employed a low-pass filter and a high-pass filter alternately, some high frequency noise still remains after many iterations. Moreover, after 5 iterations, the mean curvature flow method [8] also did not produce smooth surfaces. The distortion metric values increase only from 0.81,0.75 to 0.82,0.77. Meanwhile, since the angles of triangles on reconstructed surface change rapidly, some artefacts are created using the mean curvature flow method [8]. It is illustrated by the waist of teletubby in the bottom of figure 4.

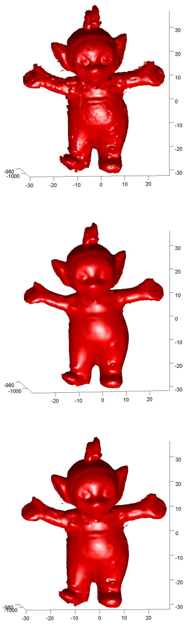


Figure 4: From top left to bottom: First: Reconstructed teletubby surface with 20 range images [8]. Second: Signal processing smoothing method [9]. Third: the new method. Fourth: the original Laplacian method [13]. Fifth: Mean curvature flow smoothing method[7].

It can also be seen from figure 6 that the angle distribution of noisy surfaces marked with blue line do not distribute smoothly and 40° and 85° have higher frequencies. However after smoothing using the original Laplacian method [13] and the new method the interior angles of all triangles have been driven closer to 60° and their distributions are smoother. This means that all triangles approach equilateral ones. While the signal processing method [9] also produces an optimal distribution of interior angles of triangles with a lower peak, the mean curvature flow method [7] hardly changes the angle distribution.

On the other hand, although the values of parameters of interest for the original Laplacian method and the new method are nearly the same in table 1 and figures 4 through 6. However, more details about the eyes of teletubby and the profile of dinosaur are kept by the new method but not by the original Laplacian method. This is because the Tsallis entropy leads to an effective optimization of vertex locations and normals. However, the Laplacian method is sensitive to the parameter β and thus can hardly distinguish and consequently smooth out details. The mean curvature flow method [7] is the most computationally expensive, since it tries to minimize the area of triangular meshes and averages the curvature of surface [8], while finally reconstructed teletubby and dinosaur surfaces include a lot of ill shaped triangles with small areas and the curvature does not change rapidly.

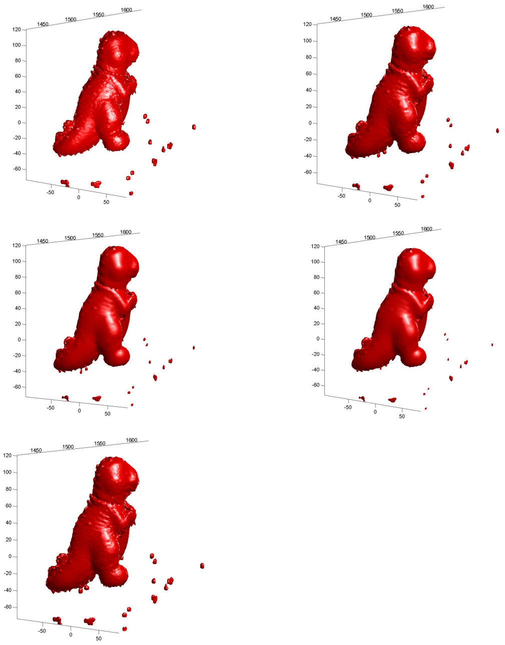


Figure 5: From top left to bottom: First: Reconstructed dinosaur surface with 8 range images [2]. Second: Signal processing smoothing method [9]. Third: The new method. Fourth: the original Laplacian method [13]. Fifth: Mean curvature flow smoothing method [7].

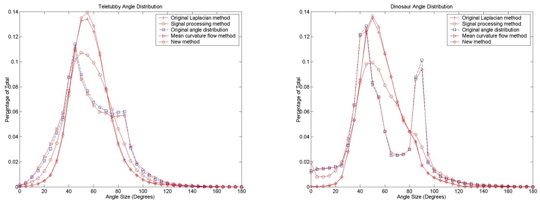


Figure 6: Angle distribution after using different smoothing methods. Left: Teletubby surface reconstructed using the mesh based reconstruction method [8] from 20 range images. Right: Dinosaur surface reconstructed using the volumetric method [2] from 8 range images.

5 Conclusion and Further Work

In this paper, we proposed a novel method for smoothing reconstructed surfaces corrupted by both scanning noise and reconstruction error due to inaccurate image registration and integration. To this end, we employed the entropy maximization (EntMax) principle

Object	Face	Vertex	Resolution(mm)	NOI	DistMetric	Iteration	Time(s)
Teletubby	33478	17243	0.68	20	0.81		
					0.86	7	179
					0.92	5	102
					0.91	4	106
					0.82	5	800
Dinosaur	88444	43965	1.40	8	0.75		
					0.80	10	1000
					0.90	4	689
					0.90	4	661
					0.77	5	1106

Table 1: Scanning noise and reconstruction error smoothing results using different algorithms and images. DistMetric: Distortion metric. NOI: Number of images. The 2nd, 7th rows: The signal processing method [9]. The 3rd, 8th rows: The new algorithm. The 4th, 9th rows: The original Laplacian method [13].The 5th, 10th rows: The mean curvature flow method [7].

in conjunction with the Tsallis entropy from statistical mechanics [10] to optimize the vertex locations and normals on reconstructed surface.

The proposed algorithm has an advantage of easy implementation. On the other hand, it has a linear computational complexity with regard to the number of nodes on surface. While the new method is as efficient as the original Laplacian method in dealing with high resolution surface or the flat area in surface, it outperforms the original Laplacian method in the sense of retaining details of surface when the surface resolution is low or surface contains a rapid change of depth. Comparing the novel method with the signal processing method [9] and the mean curvature flow method [7], the former is more computationally efficient and effective in producing high quality meshes with a better balance between details and smoothness. Experimental results based on real images have shown that the novel smoothing method yields on the whole the best smoothing results in the sense of distance metric, interior angle distribution, computational time, and the number of iterations for smoothing. This is because that the EntMax principle in conjunction with the Tsallis entropy are expressive in characterizing the probability distribution of triangle edge length variation and normal variation of neighbouring vertices. In the future, we are planning to apply various measures such as surface curvatures for the optimization of vertex locations and normals in our entropy optimization method.

6 References

[1] C.Y. Chen and K.Y. Cheng, "A sharpness dependent filter for mesh smoothing", *Computer Aided Geometric Design*, 22(5), pp 376-391 (2005).

[2] C. Dorai and G. Wang, "Registration and integration of multiple object views for 3D model construction", *IEEE Trans. PAMI*, 20(1), pp 83-89 (1998).

[3] L. Freitag and P. Knupp, "Tetrahedral mesh improvement via optimization of the element condition number", *Internat. J. Numer. Mehtods Engrg*, 53, pp 1377-1391 (2002).

[4] A. Hilton, A.J. Stoddart,J. Illingworth and T. Windeatt, " Implicit surface-based geometric fusion", *Journal of Computer Vision and Image Understanding*, 69(3), pp 273-291 (1998).

[5] E.T. Jaynes, "Information theory and statistical mechanics", *The Physical Review*, 106, pp 620-630 (1957).

[6] A.A. Mezentsev, "A generalized graph-theoretic mesh optimization model", *Proceedings 13th International Meshing Roundtable(IMR2004)*, Williamsburg, pp 255-264 (2004).

[7] K. Polthier, "Polyhedral surfaces of constant mean curvature", *Habilitationsschrift Technische University Berlin*, pp 1-212 (2002).

[8] Y. Sun and C. Dumont, "Mesh-based integration of range and color images", *Proc. SPIE*, Orlando, pp 110-117 (2000).

[9] G. Taubin, "Geometric signal processing on polygonal meshes", *Proc. Eurographics: State of the Art Report*, Manchester, . (2000).

[10] C. Tsallis, "Possible generalization of boltzmann-gibbs statistics", *J. Statistical Physics*, 52, pp 479-487 (1998).

[11] C. Tsallis, J.C. Anjos and E.P. Borges, "Fluxes of cosmic rays: a delicately balanced stationary state", *Physics Letters A* , 310, pp 372-376 (2003).

[12] H. Zhou and Y. Liu, "Flag guided integration of multiple registered range images", *Proceedings International Conference on Pattern Recognition*, Hong Kong,(2006).

[13] L.R. Herrman, "Laplacian-isoparametric grid generation scheme", *J. Engineering Mechanics*, 102, pp 749-756(1976).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Performance Analyses of Embedded Real-time Operating Systems using High-precision Counters

Kemal Köker, Richard Membarth, Reinhard German
Department of Computer Science, Computer Networks and Communication Systems
University of Erlangen-Nuremberg, Germany
koeker@informatik.uni-erlangen.de, richard.membarth@informatik.stud.uni-erlangen.de,
german@informatik.uni-erlangen.de

Abstract

To evaluate the performance of embedded real-time operating systems, we have built a scenario of soccer playing robots according to the F180 small-size league of the Robocup with a common embedded industrial PC/104 system. Thereafter we patched a common Linux kernel with the real-time application interface RTAI and installed it on a compact flash card for using it on the robots'. We connected IR distance sensors to the on-board embedded system and performed a response time analysis of the operating system, and implemented an interrupt service routine for the board's parallel port to generate a system response for externally caused hardware interrupts, e.g. from sensors. For a faster data collection we triggered interrupts by edges using a function signal generator. To monitor the embedded systems' response time we built a monitoring system by using a high-precision histogram scaler and counter. The data for the response time has been monitored in various system loads and been analysed statistically. Our system allows an easy and low-cost way for performance analyses of embedded real-time operating systems.

Keywords: embedded Linux systems, real-time operating system, performance analysis

1 Introduction

Nowadays the predictions about a systems response time of man-machine or machine-machine architectures are often based on simulation models. A lot of tools in this area require data for the input modelling. Often these data represent the response times for an input, at which the monitoring is done, e.g. by hardware. Analysing the performance of operating systems requires monitoring the response time for interrupts which are triggered by an input (e.g. pushing a button or getting new values of sensors). Mostly this response time depends on additionally running tasks in the system and the scheduling algorithm and varies with respect to the system load. Therefore increasing the system load increases e.g. the response time for a hardware interrupt.

Real-time operating systems guarantee a maximum time for the systems response for interrupt and represent a quite interesting possibility to be used in soccer robots. Smart scheduling algorithms and other techniques are used to keep the response time as low as possible. Mostly these techniques differ in case of a one-shot and periodic or multi-mode interrupt. In case of a one-shot interrupt, the operating system has to response as quickly as possible without any precautions for further interrupts. In case of a periodic mode, precautions for the scheduling are made to keep the response time lower. Considering the overhead for a one-shot mode, there should be a significant difference to the periodic-mode when the system has to response to higher interrupt frequencies. Therefore the mean of the response time in one-shot mode should lead to a higher value than the mean of the periodic mode.

We selected a setting in our labour assembly to confirm the upper conclusion also for embedded real-time operating systems. To keep it as easy as possible, we patched a Linux kernel with the real-time application interface RTAI [11] and installed a complete embedded Linux operating system on a compact flash card to use it in a PC/104 system [1]. The PC/104 system is a common hardware in industrial projects. We used the PC/104 module from Arbor [2] with an AMD Ultra Low Power Geode GX1-300-Mhz fan less CPU and an onboard compact flash socket. This embedded board is common for standard PC-like hardware and its interfaces like serial, parallel, LAN, USB ports.

The rest of the paper is organised as follows. Section 2 gives a brief overview of our embedded Linux system and the components used in it. Section 3 gives an overview of the real time application interface for Linux and explains the architecture. Section 4 describes very briefly the preparing for the performance analysis. Section 5 explains the measuring using an oscilloscope. Section 6 describes the measuring using the high precision counter. Section 7 gives the results of the performance analysis of the embedded Linux system used.

2 Embedded Linux System

The operating embedded Linux kernel is version 2.6.9 [8]. To keep the system small and truly (!) embedded, we used BusyBox [9]. BusyBox combines tiny versions of common Linux/UNIX utilities into a single small executable (including the Web server). For this reason, it's called a multi-call binary combining many common UNIX tools. The utilities in BusyBox generally have fewer options than their full-featured GNU originals; the options that are included provide the expected functionality and behave very much like their GNU counterparts.

For the systems C library we use uClibc [10] which is an optimised C library for developing embedded systems. It is much smaller than the common GNU C library, but nearly all applications supported by glibc work without problems with uClibc. One more benefit is that uClibc even supports shared libraries and threading. All robots allow remote logins to the running system via SSH connection. With the target to keep the system small, we used the SSH-2 client/server from Dropbear [14]. For further testing we included the Linux-based streaming system Palantir [15]. Palantir is designed to transmit live data (e.g. video, audio) over an existing TCP/IP network.

The system is stored on a compact flash card with the boot loader grub. Due to the limited write-cycles of a compact flash card, the system is running in ramdisk mode by loading a compressed initrd at boot time. Table 1 gives a brief overview of the storage size for our fully functionally embedded Linux system.

Table 1: Component size of embedded Linux.

Comp.	Name	Uncompr.	Compressed
Grub loader	grub	-	200 KB
Kernel 2.6.9	bzImage	-	1,100 KB
Root FS	initrd	4,400 KB	2,100 KB
		Total	3,400 KB

3 Real-Time Application Interface on PC/104

The well-known real-time application interface RTAI is an extension and modification of a common Linux kernel. For usage, the Linux kernel has to be patched with RTAI, which enables the operating system to response in a fast and predictable way. Fast means that it has low latency, thus i.e. it responds to external, asynchronous events in a short time. Predictable means in that case that it is able to determine task's completion time with certainty. The current architecture of an RTAI system consists of the Adeos Nanokernel layer [13], the hardware abstraction layer HAL and further modules for the requested functionality.

The Adeos Nanokernel layer is inserted as a division between the lower hardware and the operating system and has the task to transmit interrupts from the lower layer to the upper layer. Using Adeos enables running of different operating systems at the same time, in which each of the operating systems registers its interrupt handler to handle the interrupt. Each operating system is assigned to a domain, and each individual domain gets a certain priority. Depending on the priority level, Adeos manages the transmission of the interrupt to the corresponding operating system, where the higher priorities are provided first and the lower priorities get the interrupt signal at last. According to the architecture, Adeos is working at least with two domains and therefore there are at least two operating systems needed to make RTAI run.

The HAL again consists of an ARTI layer (Adeos-based real-time interface) and the real-time scheduler. The HAL enables an easy access of the OS to the interrupts via ARTI. The two OS domains mentioned above represent the real-time and a non-real-time domain. In this case, the real-time part is the RTAI domain with a higher priority, and the non-real-time part is the Linux kernel with a lower priority. Therefore, each interrupt for which both OS domains are registered as handler, is first send to the RTAI domain because of the higher priority. Figure 1 shows the schematic architecture of an RTAI system with the Linux domain as the non-real-time part.

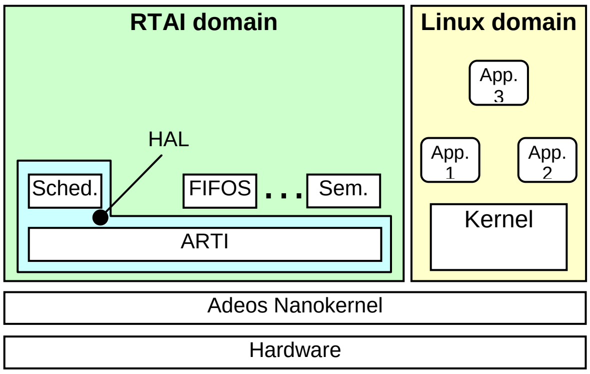


Figure 1: Schematic architecture of RTAI.

4 Preparing for Performance Analyses

To evaluate the performance of a system's response time means e.g. to measure latencies for an interrupt, in which an interrupt may be generated either via software or hardware. In case of hardware, an interrupt e.g. on the parallel port is usually recognised by the system when a TTL level is changed (e.g. from 0 to +5 V) on the corresponding pin. If everything is set up correctly, the CPU is signalled to activate the interrupt handler for the corresponding port. Now the CPU stops the currently running code and jumps to the code of the interrupt handler which has to be executed und deletes the flag for the interrupt signal. After finishing the execution of the interrupt service routine, the CPU executes the code before the interrupt was caused.

To conduct this recording we implemented an interrupt service routine ISR for the PC/104's parallel port, in which the implementation was done using RTAIs API [12]. Our ISR consists of three functions, called xinit_module, handler and xcleanup_module. The function xinit_module is called up automatically at module loading. It initialises the ISR and registers an interrupt handler for the interrupt 7 (parallel port). In case of an interrupt, a specific RTAI function is executed and a timer is started for the one-shot or periodic mode. The timer mode is depending on the mode the RTAI module was loaded at system booting. The handler function is the main function of the interrupt handler and generates just a short high signal and thereafter signals RTAI the end of service and readiness for the next interrupt. The end of the service is done by setting a low level on pin 2-9. The xcleanup_module is called up automatically at the time of unloading the module and cleans up the interrupt for the parallel port.

5 Monitoring Latency using an Oscilloscope

To verify the correct function of our ISR we performed some measurements by using a frequency generator and a digital sampling oscilloscope. To monitor the response, first the ISR has to be loaded.

We used the frequency generator GF266 from ELC [3] to generate rectangle signals at TTL level. The signal is connected to pin 10 of the parallel port and causes an interrupt. Each time a high level is on the pin, the ISR is called up and the corresponding response is served at pin 2-9. To monitor the response time for the generated signals, we connected the 4-channel, digital oscilloscope WaveSurfer 424 from LeCroy [4] to the frequency generator and to pin 4 of the parallel port. A high level generated by the frequency generator is connected parallel to channel one of the oscilloscope and on pin 10 of the parallel port (blue arrows). The response pin 4 is connected to channel 2 of the oscilloscope (pink arrows). Figure 2 displays the measurement infrastructure using the digital oscilloscope.

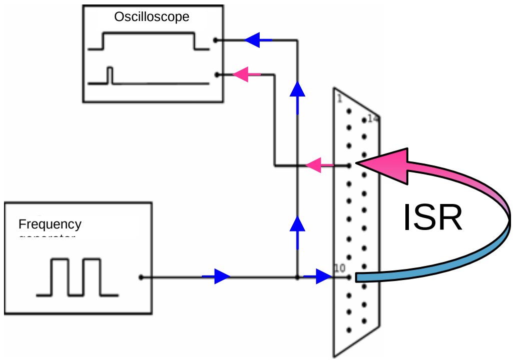


Figure 2: Monitoring via oscilloscope.

The first run of the monitoring for the latency was done at 10-kHz rectangle signals of the frequency generator. The embedded RTAI Linux system was idle, so the response time for an interrupt was not influenced by any other factor than the internal task of the running operating system. Figure 3 displays a screenshot of the oscilloscope with a latency of 4.3 µs for the response time at 10 kHz. The recording is truncated only for one signal from the frequency generator.

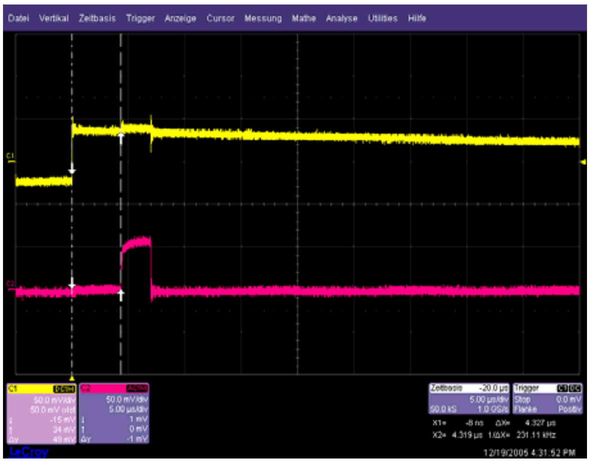


Figure 3: Screenshot of the oscilloscope - latency for response 4.3 µs.

Even though the system was idle, we monitored fluctuations for the response time. To measure and visualise this fluctuation, we used the cumulative sampling function of the oscilloscope. In this case, the sampling is recorded continuously and additionally overdrawn for the previous signal course. The number of observations of the values are now represented by coloured pixels, in which a red colours symbolises a higher and the violet colours for a lower appearance of the recorded value (figure 4). Accounting the oscilloscopes recordings, we observed response times in the range of about 4–17 µs when the system is idle.

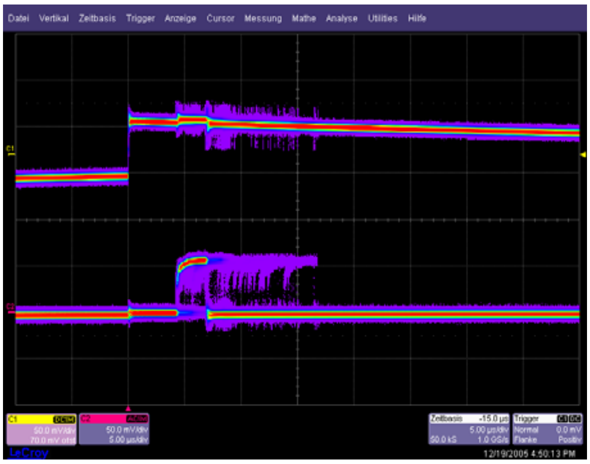


Figure 4: Cumulative recording for the response time – system: idle.

We increased the frequency of the rectangle signal up to 100 kHz to make a simple check for the performance of the embedded system. The monitored data was almost equal to the first run and the increased interrupt frequency did not affect the response time.

To evaluate the system performance, the system load usually should be increased by executing parallel programmes. We started a second run and executed just the ping flood to the embedded Linux system via a 100-MBit LAN connection and an interrupt frequency of 10 kHz. In this configuration, the signal course changed significantly using the cumulative recording of the system response (figure 5). We monitored a response time in the range of 4 to 35 µs and peeks up to 40 µs. Even though the interrupt frequency and the system load increased, the response was "still in time". Still in time means that we could relate each interrupt signal to its corresponding system response and vice versa. This bijectivity between interrupt and response is important for the evaluation and its statistical analyses, e.g. accounting for histograms.

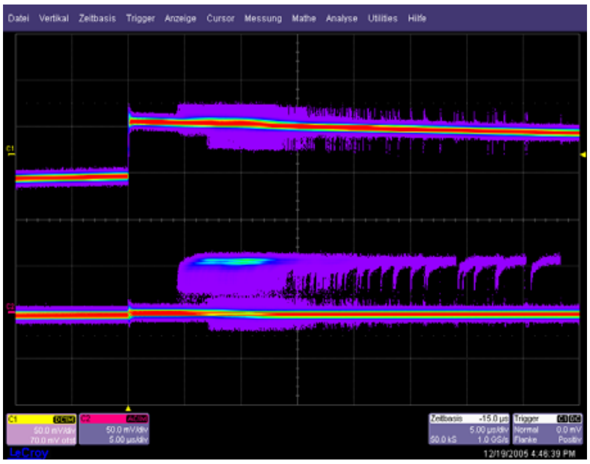


Figure 5: Signal course for the response at 10 kHz interrupt – system: load.

We started a third run and raised the interrupt frequency slowly up to 100 kHz. The system was loaded again by an executing ping flood via LAN from an external computer. In this configuration (frequency 100 kHz) it was not possible to relate the interrupt signals to the corresponding responses. We monitored the beginning of the lagged responses at about 83.5 kHz and also some mavericks before.

All mentioned variations of the test run are not useful for statistical analyses and performance evaluation. However, our lab construction allows a first overview for the system performance of our embedded Linux system and externally caused interrupts. Therefore our lab construction allows a screenshot of the monitoring time. Detailed analyses of the performance require to record time-related data for our lab construction for a long time period. The recorded data should contain the difference of the interrupt and the corresponding response time for each generated interrupt signal.

6 Measurement Infrastructure

We introduced a high-precision frequency counter SIS3820 [7] in the range of 50 ns to gather time-related data. The frequency counter has a start and a stop port to initiate and terminate the measuring. According to the monitoring via an oscilloscope, we connected the start port of the frequency counter with the output signal of the frequency generator and to pin 10 of the parallel port. The stop port of the counter is connected to one of the response pins of the parallel port (here it is pin 2). The counter uses an internal 50-MHz clock and therefore has a resolution of 20 ns. To measure the time between the start and stop signal, the counter is counting the ticks between the start and stop signal. Figure 6 displays the schematic for the measurement.

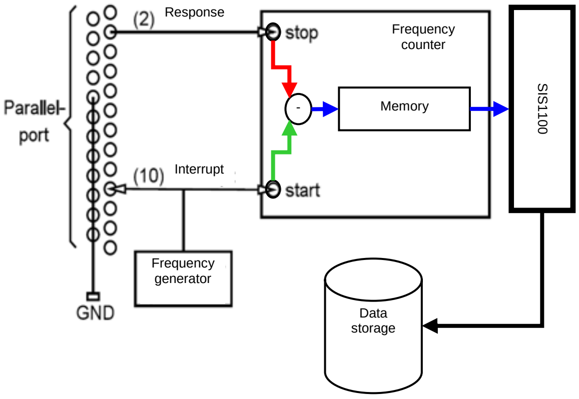


Figure 6: Schematic of the measurement via frequency counter.

To perform statistical analyses we need to collect the pair of interrupt signal and time difference for that interrupt and its response. Reliable statements need a recording for a long period of time or huge amounts of interrupt signals coming from the frequency generator. The frequency counter allows to store up to more then 16 million values in a 64-MByte memory in FIFO manner. Therefore, we need a fast transmission of the collected data to an external storage like a second personal computer. The data transmission from the counter to the computer is periodically done via the SIS1100 module [6] over a fibre-distributed data interface.

7 Statistical Analyses of Example Measurements

We used our lab construction to run data collection for 3.6 million interrupts in four different test runs. The test duration lasts 6 minutes at an interrupt frequency of 10 kHz. The tests only differ for the real-time mode one-shot or periodic, and whether the system is idle or loaded. Usually, the system load is generated by different types like kernel compiling, ping flood, etc. To be able to compare the results, we again generated the system load by executing a ping flood via LAN from an external computer and recorded the response times for the interrupts.

To evaluate the system performance based on statistical data, we took a closer look to the quartiles of the response times. The difference of the first and third quartile is defined as the distance of the quartile QD. This distance should cover 50% of all values and is used as the mass of the distribution. This mass is usually robust against runaways and is usable for statistical evaluations. Table 2 gives a brief overview of the collected data of the response time using the frequency counter.

Table 2: Quartiles for the response time.

RTAI	Quartile 1	Quartile 2	Quartile 3	QD
One-shot idle	4.20	4.30	4.40	0.2
Periodic idle	4.50	4.50	4.60	0.1
One-shot load	8.20	10.20	11.90	3.7
Periodic load	5.80	6.70	8.20	2.4

Comparing the response times for the quartile distance delivers no significant difference between both modes in the case of idle system. We can see a very small difference of 0.1 µs where the one-shot mode is faster. Setting the system under load, the quartiles for the response time also increase. When comparing with the system in idle phase, we notice a significant difference for the quartile distance and a faster response time for the periodic mode. This difference may be due to the scheduler in the periodic mode, in which the scheduler is periodically executed. This means that the scheduler is prepared for further interrupts and the system can response faster as in one-shot mode.

The results and the analyses of the quartiles are unique but do not contain detailed information for robust statements on the system performance.

Table 3: Jitter and standard deviation for latency.

RTAI	Min	Max	Mean	Jitter	SD
One-shot idle	0.6	45.7	**4.484**	0.6318733	0.7949046
Periodic idle	0.7	43.7	4.742	0.7626195	0.8732809
One-shot load	3.7	51.6	10.24	11.53628	**3.396510**
Periodic load	1.4	38.3	7.586	7.754853	2.784754

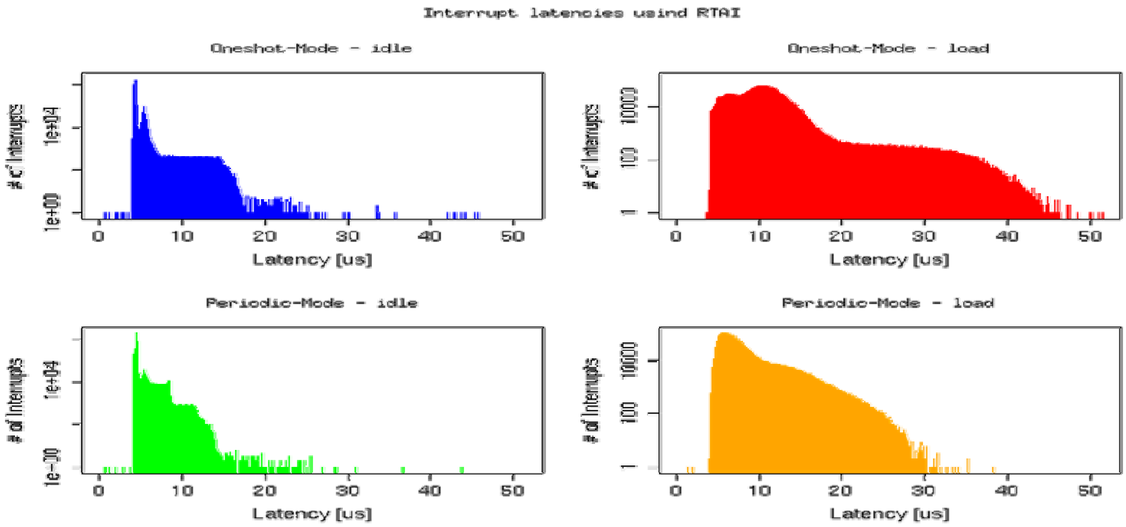


Figure 7: Histograms for the one-shot and periodic mode.

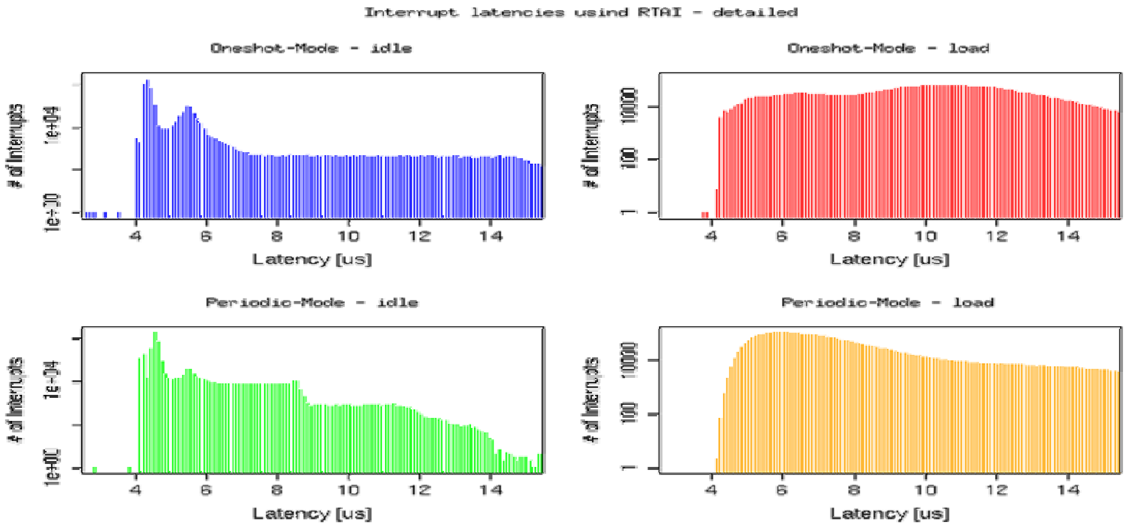


Figure 8: More detailed histograms for the one-shot and periodic mode - latency time 0-16 µs.

Therefore we considered values like minimum, maximum, mean, jitter and standard deviation for the latency, listed in table 3.

The mean in the one-shot idle phase is 4.484 µs and nearly equals the mean of the periodic idle phase with 4.742 µs, and the jitter is very low between 0.631 µs and 0.762 µs. Increasing the system load almost doubles the latency in the periodic mode and more than doubles it in the one-shot mode. At the same time, the jitter increases by up to 11.536 µs for the one-shot mode and 7.754 µs for the periodic mode. Increasing the system load also increases the standard deviation SD for both modes. It is remarkable that the standard deviation for the system load in the one-shot mode amounts to more than 50% of the mean value in the idle phase. The corresponding histograms for the values in table 3 are shown in figure 7 and confirm our theory that the periodic scheduling mode is the better choice, independent of whether the systems status is idle or load. The y-axis of the figures is scaled logarithmically when the system is idle, and scaled linear in the load phase and is counting the number of interrupts. The x-axis is representing the latency for the system response for an interrupt. As already mentioned above, the difference between the one-shot and periodic mode in the idle phase of the system is marginal and the result is shown in figure 8 with more detailed histograms for the range of 0-16 µs of latency time. On the other hand, when the system load increases, the histogram of the distribution for the latency in the periodic mode is slimmer in comparison to the one-shot mode.

8 Conclusion

We patched a Linux kernel with a real-time extension to evaluate the performance of an embedded real-time operating system on an embedded industrial PC/104 board.

Thereafter we monitored the system performance for hardware interrupts using a frequency generator connected to the parallel port of the PC/104 board. To enable a response for the interrupt, we implemented a service routine for the parallel port using the real-time API. We observed the interrupt signal and the response at the same time on the oscilloscope and found out that our embedded system is fast enough to respond in time for interrupts of up to 100 kHz when it is idle. Loading the system increased the response time and disabled it for interrupts higher 83.5 kHz.

We introduced a high-precision counter to gather time-related data for the latency. The analysis of the latency in different running modes confirmed our theory that the periodic mode in a real-time system gets a better performance nearly every time. Our measurement infrastructure can be easily adapted to evaluate other embedded real-time systems.

9 References

[1] PC/104 Embedded Consortium, PC/104 Spezifikation v2.5, “Home Page”, http://www.pc104.org, visited on 12/06/2004.

[2] Arbor Inc., PC/104 module Em104–n513/VL, “Home Page”, http://www.arbor.com.tw, visited on 12/12/2005.

[3] ELC, ELC Frequency Generator GF 2006, “Home Page”, http://elc.annecy.free.fr, visited on 15/12/2005.

[4] LeCroy Corporation, LeCroy WaveSurfer 424, “Home Page” http://www.lecroy.com, visited on 23/11/2005.

[5] Phillips Scientific, NIM Logic Unit 755, “Home Page”, http://www.phillipsscientific.com, visited on 9/12/2005.

[6] Struck Innovative Systeme GMBH, SIS1100/3100–PCI/VME link/interface, “Home Page”, http://www.struck.de, visited on 11/12/2005.

[7] Struck Innovative Systeme GMBH. SIS3820 multi purpose scaler, “Home Page”, http://www.struck.de/sis3820.htm, visited on 11/12/2005.

[8] Linux Kernel Archives, “Home Page”, http://www.kernel.org/, visited on 11/11/2005.

[9] Erik Andersen, “BusyBox: The Swiss Army Knife of Embedded Linux”, http://busybox.net/, visited on 15/7/2005.

[10] Erik Andersen, “uClibc: A C library for embedded Linux”, http://www.uclibc.org/, visited on 15/7/2005.

[11] RTAI Project, “RTAI: Real-Time Application Interface”, http://www.rtai.org, visited on 4/10/2005.

[12] RTAI Project, “RTAI API”, http://www.rtai.org/documentation/vesuvio/html/api/, visited on 4/10/2005.

[13] Adaptive Domain Environment for Operating Systems, “Home Page”, http://home.gna.org/adeos/, visited on 4/10/2005.

[14] Dropbear Small SSH2 Client, “Home Page”, http://matt.ucc.asn.au/dropbear/dropbear.html, visited on 15/7/2005.

[15] Palantir: Multichannel Interactive Streaming Solution, “Home Page”, http://www.fastpath.it/products/palantir, visited on 15/7/2005.

[16] Hannes Mayer, Programmbeispiele für RTAI, Adeos, Linux, “Home Page”, http://www.captain.at/rtai-adeos-linux.php, visited on 05/10/2005.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

A Comparative Study of Local Minima Escape in Potential Field Method

Suyeon Hong, Hadi Moradi, Seung Lee, Sukhan Lee
School of Information and Communication Engineering
Sungkyunkwan University, Suwon, Korea
kittytion@gmail.com, moradi@ece.skku.ac.kr, lsh@ece.skku.ac.kr

Abstract

Potential field method has been extensively used in motion planning. However, the major drawback of this method is the possibility of trapping in local minima. In this paper, we present a comparative study of different methods used to escape local minima, composed of well-known methods and several new methods proposed by the authors. The switching conditions, i.e., switching from local minima escape mode back to normal potential field method, have been compared and new switching conditions are proposed. The experimental results for a set of local minima escape methods and several switching conditions are provided and compared.

Keywords: motion planning, local minima, random walk, wall following, magnetic field

1 Introduction

The potential field algorithm is very simple and fast, which makes it suitable for real-time applications. However, it has a significant disadvantage, the 'local minima' problem. In potential field, trapping in a local minimum happens whenever the attractive and repulsive forces have the same magnitude but opposite directions relative to each other (figure 1). In this case, these two forces compensate each other, and then zero force will be induced to the robot. Consequently, the robot will not move any further and the goal cannot be reached [2].

There have been many studies to get over the local minima problem. These studies can be divided into two main classes: Local minima avoidance (LMA) and local minima escape (LME). In LMA, the potential field function is modified such that it never produces local minimum value anywhere in the environment except at the goal position. Navigation function [2], harmonic function [2], superquadric [18], and magnetic field [16] methods are examples of such approaches. Navigation function does not have local minima, but it needs to be transformed to the sphere coordinate and requires a prior knowledge of obstacles [3][8]. Consequently, it is not practical for real-time motion planning in dynamic or partially-known environments. Similarly, the harmonic function approach avoids having local minima in the path by applying Laplace's equation during path planning. However, it also needs prior information about obstacles. Thus, it also makes the approach unsuitable for dynamic and real-time applications [4]. Magnetic field method [10] is another local minima-free approach, in which the repulsive force is replaced by magnetic force to make the robot move along obstacles' boundaries instead of pushing it away from the obstacles. But, this algorithm cannot guarantee optimal path, and may result very long paths.

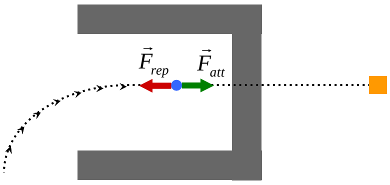


Figure 1: Local minima.

In contrast to LMA, the Local Minima Escape (LME) methods allow the robot to fall into a local minimum, and uses different algorithms to help the robot to escape the local minimum region. Methods such as random walk, multi-potential fields, straight line select, the best first, virtual obstacle, and wall following are examples of such approaches. Random walk algorithm [14] uses random motion to generate a path to escape the local minima. Although it is probabilistically complete, i.e. creates a path to escape local minima, it may take a long time to generate a path. And Multi-Potential Method [5] generates and keeps several potential maps with different resolutions at the same time. If the robot meets local minima in a potential map, then it converts to another map and continues potential guided path planning. The drawback of this method is that it may make a circular motion and come back to the original position after several moves. Another LME method is called the 'straight line select' (SLS) [6]; when a robot meets local minima, SLS finds a direction randomly and draws a straight line along the direction with a fixed length. If the ending point of the line has lower potential value than the local minima position, then the direction is selected for moving. By combining with the random walk method, its success rate can be increased. But, many directions are unsuccessfully explored which makes this approach not suitable for real-time motion planning. Considering the fact that local minima is basically a potential well, the best first method [2] fills up the potential well. This approach changes the potential well into saddle-shaped potential distribution, so the local minima region can be overcome. Unfortunately, this method also needs a-prior knowledge of the environments. Moreover, it is not appropriate for real-time motion planning since it takes very long time to fill up the entire local minima region [8]. Virtual obstacle approach [7] is similar to the best first method, except it is suitable for real-time application. It assumes a concave shape obstacle around the robot when it faces local minima. Thus a virtual polygon is generated inside the concavity and a virtual obstacle is placed in the polygon. Consequently, the robot would be guided out of the local minima region. It is important to mention that this approach uses heuristic polygon selection, which may perform inefficiently, if a bad heuristic is selected.

In this paper, we review many algorithms to overcome the local minima and propose the equi-repulsive and hybrid methods. Furthermore, we study the switching criteria, which stop LME and goes back to normal potential field method. In section 2, several local minima overcoming strategies are covered in more detail. In section 3, switching conditions are proposed and surveyed. Finally, section 4 reports some simulation results followed by the conclusion.

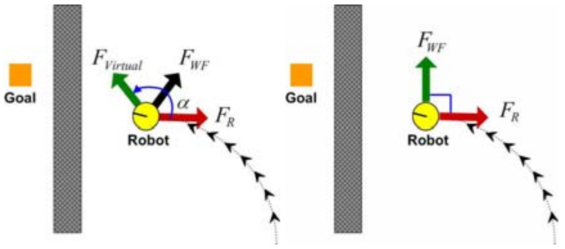
2 Local Minima Overcoming

2.1 Wall Following

Wall following has been the central method to navigate in an environment where the map is not given. By following the wall, either the left or the right side wall, one can go through the unknown environment until he/she can find a clue toward the goal. Escaping a maze or a cave using wall following are examples of such approaches. Currently, however, this kind of motion planning is less attractive, because wall-following motion is suitable only for very specific applications like a floor cleaning robot. But, it can be still one of the good solutions about the local minima problem. Here, several different methods based on wall following will be explained.

2.1.1 Fixed Distance Method

The simplest concept to implement wall following is maintaining a fixed distance to a specified wall while robot moves. The original algorithm [8] relies on deciding the wall to be followed, and then goes ahead with maintaining a constant distance to the wall. Keeping constant distance to a wall may be implemented using sensors or calculating distance based on the environment model. Although distance calculation in this method is very simple, fast, and providing real-time performance using sensors such as sonars, it is not suitable for advance path generation.



(a) angle method (b) normal to RF

Figure 2: Angle based wall following method.

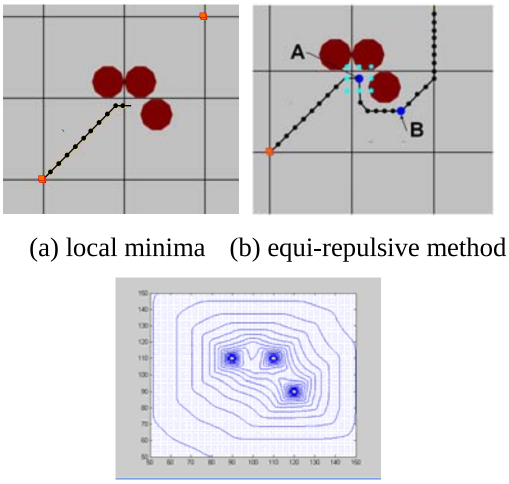
2.1.2 Angle Method

In complex environments, it is not feasible to determine the walls, and calculate the distance to them by keeping fixed distance to the wall. However, it is possible to determine the direction parallel to the wall using the repulsive force. This is based on the fact that parallel direction to the wall is orthogonal to the repulsive force generated by the wall. Borenstein and Koren [9] proposed the angle based method, in which the fixed angle such that $90^\circ < \alpha < 180^\circ$ is added to the repulsive force vector, θ_r. The resulting force, which is called a virtual attractive force, $F_{virtual}$ temporarily replaces the attractive force. (figure 2a). The combination of this virtual force and repulsive force generates the wall following force, i.e. F_{WF}, which moves the robot away from the local minima area. To follow a left-side wall, the algorithm adds an angle, α to θ_r. Similarly, it subtracts α from θ_r if robot wants to follow the right wall. Borenstein and Koren used, $\alpha = 145^\circ$ while Brock and Khatib [12] used $\alpha = 120^\circ$ based on their experimental results. This method is especially suitable when not very accurate sensors are used, since it is less sensitive to sensor misreading [9].

On the other hand, Hong et al. [15] proposed a method, which uses normal vector instead of virtual attractive force. This approach will generate a motion parallel to the surface of the obstacles, which would be the perfect wall following method. Moreover, this approach is very fast because of its simple calculation stage (figure 2b). In this approach, when the robot meets local minima, it starts moving along the normal direction of its repulsive force with a fixed step length.

2.2 Equi-Repulsive Method

When a human hill climber faces a tall hill, he/she will walk around the hill at the same level, i.e. the same potential, until it circumnavigate to the other side of the hill. Hong et al. [15] proposed the equi-repulsive method that uses the same approach by moving along a repulsive potential contour, i.e. a closed curve with the same repulsive potential value (figure 3c), to circumnavigate local minima. By following the equi-repulsive potential contour with avoiding past, it is guaranteed to go around obstacles.



(a) local minima (b) equi-repulsive method

(c) repulsive potential contour

Figure 3: Equi-repulsive method.

In this approach, if robot meets local minima (figure 3a), it checks 8 neighbouring sample points (figure 3b). Among these sampling points, the point with the closest repulsive potential value to the current point is selected as the next point. If the next point expects to make some collision with obstacles, it will be rejected and select second candidate point. Moreover, if the point is already in the history stack, it will also be refused. Then, the planner will select the next candidate point to go. The normal potential field approach will be resumed after the local minima region is passed. The criterion to restart the general potential field would be discussed in section 3. The advantage of this method is the ease of implementation; since it uses the repulsive potential component of the potential field, which has already been implemented to use negative gradient method. Furthermore, it provides alternative candidates in case a selected direction is not feasible. In contrast, the other methods, e.g. an angle method, it is hard to find an alternative direction if a selected direction results into collision. Consequently, the approach may fail.

2.3 Random Walk

Random walk algorithm [14] originated from the Brownian motion of Einstein, which has been used for describing a variety of natural phenomena, such as movement of molecules in gases. In general, Random walk method in motion planning consists of a series of random directional movement with a small step length. It randomly moves until getting away from the local minima region, q. To do so, it randomly selects a direction, and moves along it with constant length, to the collision-free area, q'. From q', the negative gradient is calculated to find the point, q''. If $\mathrm{p}(q'') < \mathrm{p}(q)$, in which p(q) represents the potential value at point q, then the random walk has succeeded, and the local minima is assumed to be escaped. Otherwise, random walk continues until detecting the right q''. After a predefined number of failures in finding q'', movement steps are gradually increased to search larger area. Probabilistically, random walk method guarantees to completely extricate the robot from the local minima region, by using the random method incorporated with avoid past. However, its performance becomes very poor in environments such as narrow passages [5].

2.4 Hybrid Method

Hong et al. [15] proposed the hybrid method, in which different LME methods are combined to improve LME performance. The authors combined random walks algorithm with wall following method and showed complete local minima escape in reasonable time frame. In this approach, if wall following method fails more than predefined number of trials, random walk is applied instead. By doing so, the algorithm is probabilistically complete and its speed becomes faster than the pure random walks method.

2.5 Magnetic Force

As mentioned before, local minima arises when the two forces, attractive and repulsive forces have the same magnitude but opposite direction. Singh et al. [10] proposed the magnetic force to replace the repulsive force to overcome this problem. In this approach, the magnetic force guides the robot to go around obstacles instead of pushing the robot away from obstacles.

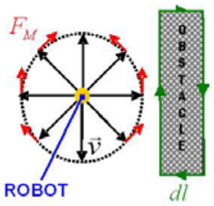


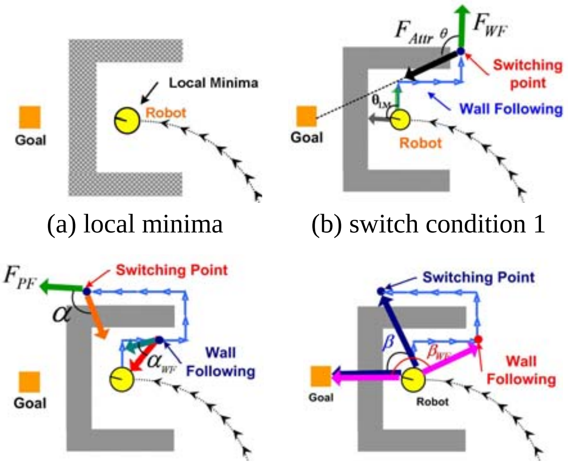
Figure 4: Magnetic force

To achieve such behaviour, the virtual current (dl) is defined at obstacles' boundaries (figure 4). The virtual current generates virtual magnetic field ($\vec{B}$), and the cross product of virtual magnetic field ($\vec{B}$) by current robot velocity ($\vec{v}$) generates the resulting magnetic force ($\vec{F}_M$) like equation (1).

$$\vec{B} = k_i \cdot \int \frac{dl \times \hat{v}}{r^2}$$

$$\vec{F}_M = \vec{v} \times \vec{B} \tag{1}$$

A drawback of this method is the assumption that the robot is never gets inside the obstacle, which is true in reality. However, if we use the elastic strip band [17] or elastic strip [13] methods, then parts of the path is in collision area from initial step, and the magnetic field method cannot be used. The other drawback of this method is the need to know obstacles' shape in order to determine the virtual current.



(a) local minima (b) switch condition 1

(c) switch condition 2 (d) switch condition 3

Figure 5: Switching conditions.

3 Switching Criteria

When the local minimum is passed, the motion planner should have criteria to switch from LME to normal potential field approach. Without these switching criteria, the robot will get into loop and move around the obstacles forever. There are three main classes of switching criteria: elapsed time, distance to the goal, and angle threshold. The elapsed time is the simplest criterion, in which a timer is set up and the original potential field is reactivated after a specific amount of time [8]. However, it easily gets back to the same local minima or gets into loops. In the second approach, when the distance between current robot's position and the goal position starts decreasing, the planner switches back to the original potential field method. And the third class is based on the angle of the robot with respect to the local minimum, goal, etc. In the following subsections, different angle criteria are discussed and surveyed.

3.1 The 1st Angle Criterion

Here, we can imagine two different force vectors. One is the wall following force, i.e. the combination of repulsive force and virtual attractive force as discussed in section 2.1.2. The other one is attractive force toward the goal position (figure 5b). The intuition behind this criterion is that the wall following force at the local minimum point, would move the robot away from the goal point [9]. If θ, the angle between above two forces becomes smaller than 90°, then heuristically, the robot is assumed to be enough away from the local minima to change its mode, and restarts the original potential field method.

3.2 The 2nd Angle Criterion

In the second criterion [15], there are also two vectors: the vector from robot's current position to the local minimum, and the force vector generated by the normal potential method induced on current robot position (figure 5c). Intuitively, if the angle between these two vectors, i.e. α, becomes larger than a threshold angle, more than 90° e.g. 120°, then empirically the robot can be assumed that the robot will not go back to the local minima. Consequently, the normal potential field approach can be resumed.

3.3 The 3rd Angle Criterion

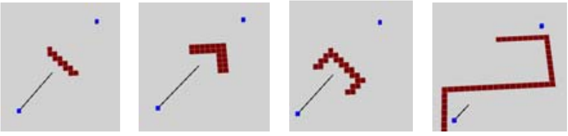
The above two switching criteria work pretty well in general cases but fail at the dead-ended type examples. The third criterion is proposed by Hong et al. [15], which performs better in such cases. Intuitively, if the angle between robot's current position and goal position with respect to the local minimum position, β, is less than 90°, then there is high probability that the local minimum is already passed (figure 5d).

3.4 The 4th Angle Criterion (Hybrid)

The hybrid criterion [15] suggests combining different criteria, e.g. the 2nd and the 3rd switching criteria, to get better result. In other words, one criterion is used, and if the robot gets into a local minimum again after trying a number of times, then the other criterion becomes active. In this way, the shortcomings of one criterion are covered by the other criterion.

4 Experimental Results

Above strategies to overcome the local minima and the switching conditions have been tested in our 2D simulator. Here, the size of workspace is 200*200 units and a circular shaped robot with 3 unit radius is considered.



(a) (b) (c) (d)

Figure 6: Typical local minima situation: (a) bench, (b) corner, (c) dead end, (d) internal wall.

Figure 6 shows four different kinds of local minima examples [8], bench, corner, dead-end, and internal wall, respectively. It contains most of typical local minima situations. In each picture, the lower left blue cube shows the starting point, the upper right blue cube represents the goal position, and the obstacles are shown in red.

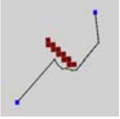
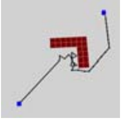
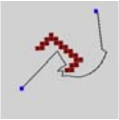
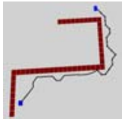
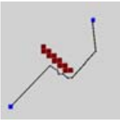
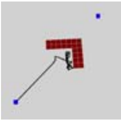
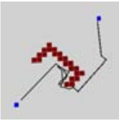
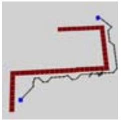
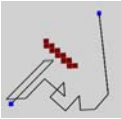
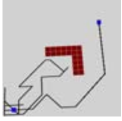
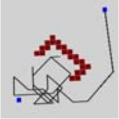
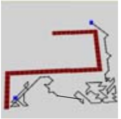
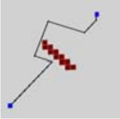
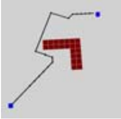
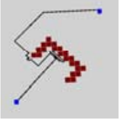
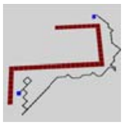
4.1 Overcoming Local Minima

The fixed angle method (ANG), the normal to repulsive force (NOR), the random walk (RAN), and the magnetic force method (MAG) have been tested, and the results are compared. In all these cases, the

2nd switching criterion, suggested in section 3.2, is used except dead-end case, in which the hybrid criterion, discussed in section 3.4, is used.

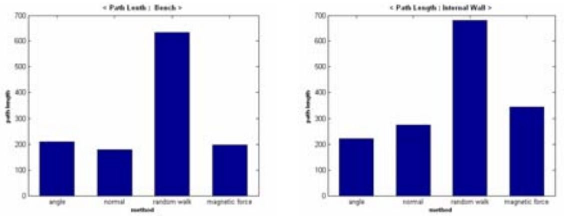
The results are reported in table 1. Each test result is represented in three rows. The first row shows the resulting path, and the second row shows whether the strategy succeeded or failed (S/F). Finally, total path length is written in the third row (PL). In the case of random walk, instead of total path length, average path length and their standard deviation (Std) is presented.

Table 1: Overcoming local minima. PL: path length, ANG: fixed angle method, NOR: normal to repulsive force, RAN: random walk, MAG: magnetic field method.

	Type	Bench	Corner	Dead end	Internal wall
A N G	Result				
	S/F	S	S	S	S
	PL	210	292.4	245.9	221.4
N O R	Result				
	S/F	S	F	S	S
	PL	178.137	–	284.35	276.1
R A N	Result				
	S/F	S	S	S	S
	PL (Std)	633.7. (15.5335)	770.4 (37.7508)	699 (42.65)	681.335 (27.35)
M A G	Result				
	S/F	S	S	S	S
	PL	198.35	193.3	325.5	343.29

Generally, ANG works well while NOR fails at the corner example. Probabilistically, RAN shows complete success rate. However, its path length is much longer than any other methods. Finally, magnetic force method also shows good success results and short path length. However, to implement the magnetic field method, we should know the obstacle's shape in advance to generate virtual current component [16].

Figures 7a and 7b are the path length for bench and internal wall, respectively. At the bench, the resulting path of magnetic force shows slightly shorter than angle method. On the other hand, at the internal wall, angle method is more efficient than the magnetic force method.



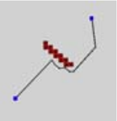
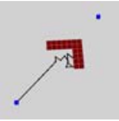
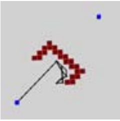
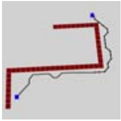
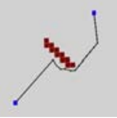
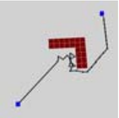
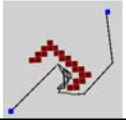
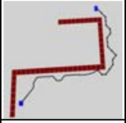
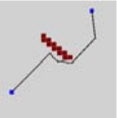
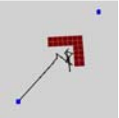
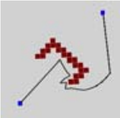
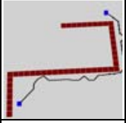
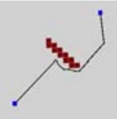
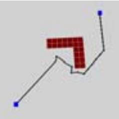
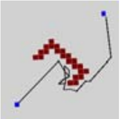
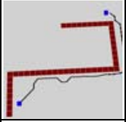
(a) bench (b) internal wall

Figure 7: Path length of each method.

4.2 Switching Conditions

Several switching conditions, which are suggested in section 3, are also tested for fixed angle method in section 2.1.2. Table 2 shows the resulting path and success/failure for each switching criterion. The rows C1 to C4 are the results of using the 1st to 4th angle criteria respectively. The 1st angle criterion fails at dead-end and corner. The 3rd angle criterion also fails at corner. However, by using hybrid method, which combines the 2nd and the 3rd angle criteria, the robot would be able to escape the local minima in all cases.

Table 2: Switching condition (C1: switching criterion 1).

C1				
	S	**F**	**F**	**S**
C2				
	S	**S**	**S**	**S**
C3				
	S	**F**	**S**	**S**
C4				
	S	**S**	**S**	**S**

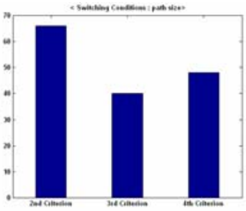


Figure 8: Path length with respect to switching conditions at dead end.

Figure 8 shows the path length at 2nd, 3rd, and 4th angle criteria for the dead-end, respectively. The 1st angle criterion is not compared since it fails at dead-end (table 2). As shown in table 2, each criterion has different path length in case of success. The 2nd angle criterion has difficulty escaping dead-end, while the 3rd angle criterion can easily escape the dead-end case. In general, however, 2nd angle criterion shows higher success rate in other cases. Finally, the 4th angle criterion, i.e. the hybrid case, works well at every case, especially for dead-end, in which it shows high success rate as well as short path length.

5 Conclusion

In this paper, we presented a comparative study of different methods used to escape local minima, composed of well-known methods and few new methods proposed by the authors. Many approaches are tested and their experimental results are presented. The switching conditions, i.e. switching from local minima escape mode back to normal potential field method, have been also compared and new switching conditions such as the 2nd, 3rd and 4th angle criteria are proposed. The experimental results show hybrid criteria, the 4th switching condition, make high success rate as well as short path length.

6 Acknowledgement

This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

7 Reference

[1] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA’85),* 2, pp 500-505 (1985).

[2] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers (1991).

[3] E. Rimon and D.E. Koditscheck, “Exact robot navigation using artificial potential functions”, *IEEE Transactions on Robotics and Automation,* 8(5), pp 501-518 (1992).

[4] C.I. Connolly, J.B. Burns and R. Weiss, “Path planning using Laplace’s equation”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA ’90)*, 3, pp. 2102-2106 (1990).

[5] H. Chang, “A new technique to handle local minimum for imperfect potential field based motion planning”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA ’96)*, Minnesota (1996).

[6] S. Caselli, M. Reggiani and R. Rocchi, “Heuristic methods for randomized path planning in potential fields”, *Proceedings of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation* (2001).

[7] C. Liu, M. Ang, K. Hariharan, S. Lim, “Virtual obstacle concept for local-minimum-recovery in potential-field based navigation”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA 2000)* (2000).

[8] X.P. Yun and K.C. Tan, “A wall-following method for escaping local minima in potential field based motion planning”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA ’97)*, pp 421-426 (1997).

[9] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots”, *IEEE Transactions on Systems, Man and Cybernetics,* pp 1179-1187 (1989).

[10] L. Singh, J. Wen and H. Stephanou, “Real-time robot motion control with circulatory fields”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA ’96)* (1996).

[11] A.M. Hussein and A. Elnagar, “Motion planning using Maxwell’s equations”, *Proceedings IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL* (2002).

[12] R.A. Brooks and J.H. Connell, “Asynchronous distributed control system for a mobile robot”, *Proc. SPIE,* 727, pp 77-84 (1986).

[13] O. Brock, O. Khatib and S. Viji, “Task-consistent obstacle avoidance and motion behaviour for mobile manipulation”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA 2002)* (2002).

[14] J. Barraquand and J.C. Latombe, “A Monte Carlo algorithm for path planning with many degrees of freedom”, *Proceedings IEEE International Conference on Robotics and Automation (ICRA’90),* 3, pp 1712-1717 (1990).

[15] H. Moradi, S. Hong, S. Lee and D. Lee, “TR-MP-20050913-0”, Intelligent Systems Research Center, http://isrc.skku.ac.kr/english/sub02/team04.html, visited on 18/10/2006.

[16] L. Singh, J. Wen and H. Stephanou, “Motion planning and dynamic control of a linked manipulator using modified magnetic fields”, *Proceedings of the 1997 IEEE International Conference on Control Applications* (1997).

[17] S. Lee, H. Moradi, G. Kardaras and S. Kim, “Geometric collision detection and potential field based time delay planning for dual-arm systems”, *Proceedings International Conference on Robotics and Automation (ICRA ’97)* (1997).

[18] R. Volpe and P. Khosla, “Manipulator control with superquadric artificial potential functions: theory and experiments”, *IEEE Transactions on Systems, Man, and Cybernetics* (1990).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Mapping and Localisation with Sparse Range Data

Jochen Schmidt, Chee K. Wong, and Wai K. Yeap
Institute for Information Technology Research
Auckland University of Technology, Auckland, New Zealand
jochen.schmidt@aut.ac.nz, chee.wong@aut.ac.nz, wai.yeap@aut.ac.nz

Abstract

We present an approach for indoor mapping and localisation with a mobile robot using sparse range data, without the need for solving the SLAM problem. The paper consists of two main parts. First, a split and merge based method for dividing a given metric map into distinct regions is presented, thus creating a topological map in a metric framework. Spatial information extracted from this map is then used for self-localisation. The robot computes local confidence maps for two simple localisation strategies based on distance and relative orientation of regions. The local confidence maps are then fused using an approach adapted from computer vision to produce overall confidence maps. Experiments on data acquired by mobile robots equipped with sonar sensors are presented.

Keywords: mobile robots, mapping, localisation

1 Introduction

Mapping and self-localisation play an important role when using mobile robots for the exploration of an unknown environment. Particularly for indoor applications, where a 2-D map is usually sufficient, geometric maps obtained from time-of-flight devices, such as laser or sonar, are widely used. In this paper, we present an algorithm for mapping and localisation which was developed originally for testing ideas about cognitive mapping [1]. In the latter, the concern is less on accuracy and speed but more on developing an appropriate process for testing ideas about cognitive mapping (as opposed to robot mapping). In particular, in cognitive mapping, the initial map computed is fragmented and highly inaccurate. Nonetheless, we believe our algorithm developed could be useful for roboticists who are interested only in the mapping and localisation problem. Consequently, in this paper we present our approach in that light and show further results. From a robot mapping perspective, what is most interesting about this work is that the robot can localise itself even with a map that is highly inaccurate in metric terms. In the first part of the paper a method for dividing a given metric map into distinct regions, e. g., corridors or rooms is presented, thus creating a metric-topological map. The region splitting is based on a metric map, which we obtain from sonar sensor readings that are processed into a map consisting of geometric boundaries. This is a more compact representation of the environment compared to grid maps, while details such as size and shape of objects are maintained. As we use only two sonar sensors, the available range data is very sparse, therefore making the map highly inaccurate in certain areas. We will show that this sparse and inaccurate data can nevertheless be used for self-localisation. Our work is inspired by [2], where a cognitive map is regarded as a network of local spaces, each space described by its shape and its exits to other local spaces. Related approaches can be found, e. g., in [3], which is a hybrid approach that combines topological and metric maps and identifies gateways and path fragments. In [4], topological maps are constructed from grid maps using Voronoi diagrams; the grid maps are split into regions and critical lines (gateways) are detected. In contrast to these methods, our approach is based on a region split and merge algorithm [5, 6, 7]. Many algorithms have been developed to solve the simultaneous mapping and localisation problem (SLAM) for mobile robots. For some examples of recent work in this area see [8] and the references therein. The approach followed in this paper (based on [2]), is different, as we do not solve the SLAM problem, but simulate a cognitive mapping process instead. The latter refers to the process in which humans and animals learn about their environment [9]. Our mobile robot explores the environment, moving from A to B, and computes the topological map mentioned earlier. The question we ask is: how does the robot use the map to find its way home from B to A? Recall that the map computed is inaccurate due to sensor errors and sparse sampling. Therefore, the map computed in the reverse journey will be different from the original one. We implemented two localisation strategies using both distance and orientation information extracted from the metric-topological representation. We show how local confidence maps can be computed using both strategies, and show how they can be fused into a global map based on the *Democratic Integration* approach [10, 11], which originated in image processing.

The paper is structured as follows: Section 2 describes the process used for generating a metric-topological map based on sparse range data, while section 3 shows how to use the information extracted from this map can be used for localisation. Experimental results are presented in section 4.

2 Mapping

The mapping process described in this section is used in two ways: First, the robot explores its environment and collects data. When this is finished, all acquired data are processed; the result of this initial mapping stage will be called the *original map* further on. This is the map the robot will use for returning home. On its way home, the robot basically performs the same data gathering and processing steps as described here, except that the data is processed each time the robot stops, in contrast to building the original map, where the data is processed only at the end of the data gathering stage. An overview over the map-processing algorithm will be given in the following. A more detailed evaluation of the algorithm including the influence of the parameters involved is given in [1].

2.1 Data Acquisition and Pre-Processing

For data acquisition we use a mobile robot equipped with eight sonar sensors and an odometer; as the emphasis in this paper is on processing sparse range data, only the two side sensors are actually used. However, it is important to note that the algorithms presented here are not restricted to sparse data or that type of sensors. In fact, the performance will be even better when more and densely sampled range data (e. g., from laser) are available. The robot acquires sonar readings while moving on a straight line (as far as the drift allows) until it runs into an obstacle. At this point an obstacle avoidance algorithm is used, after which the robot can wander straight on again. A single one of these straight movements will be called *robot path* throughout this paper. We store the sonar readings separately for each robot path; this is not mandatory for the presented methods, but it simplifies processing later on. Based on the raw sonar sensor readings we build a simplified geometric map containing the robot movement path as well as linear surfaces approximated from the sonar data. In a first step, the recorded sonar data is low-pass filtered and converted to surfaces, being a piecewise linear approximation of the sonar distances. The surfaces are simplified further by grouping them, thus removing small gaps. The pre-requisite for the algorithm presented in the following is a geometric map that contains the robot movement path as well as surfaces in terms of line approximations of the original range sensor data. The goal is to split the map into distinct regions, e. g., corridors and rooms. Splitting is done along the robot movement path, using an objective function that computes the quality of a region, based on criteria such as the average room width (corridors are long and narrow compared to rooms) and overall direction (e. g., a corridor is separated from another one by a sharp bend in the wall). Additionally, a regularisation term is used in order to avoid the formation of very small regions, which may originate from missing (gaps) or unreliable sensor data.

2.2 Split and Merge

The basis of the map-processing algorithm is the well-known split and merge method [7, 6, 5]. In pattern recognition this algorithm is traditionally used for finding piecewise linear approximations of a set of contour points. Other applications include segmentation of image regions given a homogeneity criterion, e. g., with respect to colour or texture [5]. The pre-requisite for applying split and merge is an ordered set of (contour) points, which is to be approximated. For this purpose a parametric family of functions $\mathscr{F}$ (e. g., lines) has to be chosen, as well as a metric for computing the residual error ε of the approximation (e. g., mean square error), or, when used for regions, a homogeneity or quality criterion. The result of the algorithm is a piecewise approximation of the original points, where every single residual error is below a given threshold θ_l. The single steps of the algorithm are as follows [5]:

1. **Start** with an initial set of points $\mathscr{S}^0$, which consists of n_0 parts, $\mathscr{S}^0 = \{\mathscr{S}^0_0, \ldots, \mathscr{S}^0_{n_0-1}\}$. Each part $\mathscr{S}^0_i$ is approximated by a function from $\mathscr{F}$. Compute the initial residual error ε^0_i for each part of $\mathscr{S}^0$.
2. **Split** each part $\mathscr{S}^k_i$ where $\varepsilon^k_i > \theta_\mathrm{l}$ into two parts $\mathscr{S}^{k+1}_j$ and $\mathscr{S}^{k+1}_{j+1}$, compute the approximation and residuals ε^{k+1}_j, ε^{k+1}_{j+1}. Repeat until $\varepsilon^k_i \leq \theta_\mathrm{l}$ $\forall i = 0, \ldots, n_k - 1$.
3. **Merge** two adjacent parts $\mathscr{S}^k_i$, $\mathscr{S}^k_{i+1}$ into one new part $\mathscr{S}^{k+1}_j$ if $\varepsilon^{k+1}_j \leq \theta_\mathrm{l}$. Repeat until merging is not possible any more.
4. **Shift** the split point shared by two adjacent parts $\mathscr{S}^k_i$, $\mathscr{S}^k_{i+1}$ to left and right while leaving the overall number of parts fixed. Keep the split that reduces the overall error, repeat until no further changes occur.

2.3 Splitting the Map

Before a region split and merge algorithm on the geometric map can be applied, it is necessary to create an initial split of the map. The easiest way to do so is to treat the whole map as a single large region defined by the start and end points of the journey. More sophisticated initialisations can be used as well, e. g., based solely on the robot movement without taking into account range data [1]. After the initialisation step, the actual division of the map into distinct regions is performed based on a split and merge that uses a residual error function $g(\mathscr{S}_i, \mathscr{S}_j)$ which compares two regions $\mathscr{S}_i$ and $\mathscr{S}_j$ and computes the homogeneity of the two regions (low values of $g(\mathscr{S}_i, \mathscr{S}_j)$ means homogeneous, high values very inhomogeneous). This function is used during the split phase for deciding whether a region $\mathscr{S}^k_i$ will be split again at a given position into two new regions $\mathscr{S}^{k+1}_j$ and $\mathscr{S}^{k+1}_{j+1}$, and in the merge (or shift) phase to determine whether two adjacent regions can be merged (or the splitting point be shifted). When the homogeneity is above a given threshold θ_r, the region will be split again (or not merged/shifted). The quality of a region now incorporates both, robot path as well as the range data. The basic idea is to use the average width of a region in the map as a criterion for splitting, as a width change resembles a changing environment, e. g., a transition from a corridor to a big room. The homogeneity (residual) function used is:

$$g(\mathscr{S}_i,\mathscr{S}_j) = \frac{\max\{f_\mathrm{w}(\mathscr{S}_i), f_\mathrm{w}(\mathscr{S}_j)\}}{\min\{f_\mathrm{w}(\mathscr{S}_i), f_\mathrm{w}(\mathscr{S}_j)\}} + s_\mathrm{r} r(\mathscr{S}_i,\mathscr{S}_j) \quad (1)$$

where $f_\mathrm{w}(\mathscr{S}_i)$ is the average width of region $\mathscr{S}_i$, and $r(\mathscr{S}_i,\mathscr{S}_j)$ is a regularisation term that takes care of additional constraints during splitting. The average width is given by $f_\mathrm{w}(\mathscr{S}_i) = \frac{A_{\mathscr{S}_i}}{l_{\mathscr{S}_i}}$, where $A_{\mathscr{S}_i}$ is the area of region $\mathscr{S}_i$, and $l_{\mathscr{S}_i}$ is its length. In practice, the computation of both needs a bit of attention. Particularly the definition of the length of a region is not always obvious, but can be handled using the robot movement paths, which are part of each region. The length $l_{\mathscr{S}_i}$ is then defined by the length of the line connecting the start point of the first robot path of a region and the end point of the last path of the region. This is a simple way to approximate a region's length without much disturbance caused by zig-zag movement of the robot during mapping.

Regarding the area computation, the gaps contained in the map have to be taken into account, either by closing all gaps, or by using a fixed maximum distance for gaps. Both approaches have their advantages as well as drawbacks, e. g., closing a gap is good when it originated from missing sensor data, but may distort the splitting result when the gap is an actual part of the environment, thus enlarging a room. We decided to use a combined approach, i. e., small gaps are closed in a pre-processing step already, while large ones are treated as distant surfaces.

The regularisation term $r(\mathscr{S}_i,\mathscr{S}_j)$ ensures that the regions do not get too small. In contrast to a threshold, which is a clear decision, a regularisation term penalises small regions but still allows to create them if the overall quality is very good. We use a sigmoid function that can have values between -1 and 0, centred at n, which is the desired minimum size of a region:

$$r(\mathscr{S}_i,\mathscr{S}_j) = \frac{1}{1+\exp\left(-\frac{\min\{A_{\mathscr{S}_i},A_{\mathscr{S}_j}\}}{A_\mathrm{max}}+n\right)} - 1\,. \quad (2)$$

The exponent is basically the area of the smaller region in relation to the maximum area A_max of the smallest allowed region. Thus, the smallest ratio is 1, and it increases when the region gets larger. This term only has an influence on small regions, making them less likely to be split again, while it has virtually no influence when the region is large, as the sigmoid reaches 0 asymptotically.

The influence of the regularisation can be controlled using the factor s_r (see (1)), which is given by $s_\mathrm{r} = s\theta_\mathrm{r}$, where $0 \leq s \leq 1$ is set manually and defines the percentage of the threshold θ_r that is to be used as a weight. θ_r is the threshold mentioned earlier, which determines that a region should be split into two when the first region is θ_r times as large as the second one.

3 Localisation

Once the original map has been generated, we instruct the robot to return home based on the acquired map. In the following, we will describe the strategies that we use for localisation based on this information, and a data fusion algorithm that allows for an overall position estimate computed from the single localisation methods. On its way home, the robot basically performs the same data gathering and processing steps as described in section 2. Each time the robot stops on its return journey, which is normally because of an obstacle in its way, it performs a map processing. Therefore, at each of these intermediate stops a new high-level representation (in terms of regions) is available and can be used in combination with the original map for localisation. The result of this step is the index of the region the robot believes it is currently in, which is a rough estimate of its global position. As it is argued in [2], this estimate is sufficient for navigation, and an accurate map will not be necessary as long the robot can find the exits to adjacent regions.

Two different strategies for localising the robot based on the original map generated on its way to the current position are presented in section 3.2. Each method computes a local confidence map that contains a confidence value for each region of the original map. All local confidence maps are then fused into a single global one using the method described in section 3.1.

3.1 Fusion of Strategies

The fusion of all local confidence maps, which may have been generated by different robot localisation methods with varying reliability, is based on the idea of *Democratic Integration* introduced in [10, 11]. It was developed for the purpose of sensor data fusion in computer vision and computes confidence maps directly on images. The original method has been extended and embedded into a probabilistic framework in [12, 13], still within the area of machine vision. We extend the original approach in a way that we do not use images as an input, but rather generate local confidence maps using various — more or less reliable — techniques for robot localisation. A main advantage of this approach is that the extension to more than two strategies is straightforward, as is the replacement of a method by a more sophisticated one. Each local confidence map contains a confidence value between 0 and 1 for each region of the original map. As in [10, 11] these confidence values are not probabilities, and they do not sum up to one; the interval has been chosen for convenience, and different intervals can be used as desired.

The actual fusion is straightforward, as it is done by computing a weighted sum of all local confidence maps. The main advantage of using democratic integration becomes visible only after that stage, when the weights get adjusted dynamically over time, dependent on the reliabilities of the local map. Given M local confidence maps $\boldsymbol{c}_{\mathrm{loc}\,i}(t) \in \mathrm{I\!R}^N$ (N being the total number of regions in the original map) at time

t generated using different strategies, the global map $\boldsymbol{c}_{\text{glob}}(t)$ is computed as:

$$\boldsymbol{c}_{\text{glob}}(t) = \sum_{i=0}^{M-1} w_i(t)\boldsymbol{c}_{\text{loc}i}(t) \quad , \tag{3}$$

where $w_i(t)$ are weighting factors that add up to one. An estimate of the current position of the robot with respect to the original map can now be computed by determining the largest confidence value in $\boldsymbol{c}_{\text{glob}}(t)$. Its position b in $\boldsymbol{c}_{\text{glob}}(t)$ is the index of the region that the robot believes it is in. The confidence value c_{glob_b} at that index gives an impression about how reliable the position estimate is in absolute terms, while comparing it to the second best one (and maybe even third best one) shows the reliability relative to other regions.

In order to update the weighting factors, the local confidence maps have to be normalised first. The normalised map $\boldsymbol{c}'_{\text{loc}i}(t)$ is given by $\boldsymbol{c}'_{\text{loc}i}(t) = \frac{1}{N}\boldsymbol{c}_{\text{loc}i}(t)$. The idea when updating the weights is that local confidence maps that provide very reliable data get higher weights than those which are unreliable. Different ways for determining the quality of each local confidence map are presented in [11]. We use the normalised local confidence values at index b, which has been determined from the global confidence map as shown above, i. e., the quality $q_i(t)$ of each local map $\boldsymbol{c}_{\text{loc}i}(t)$ is given by $c'_{\text{loc}b}(t)$. Normalised qualities $q'_i(t)$ are computed by:

$$q'_i(t) = \frac{q_i(t)}{\sum_{j=0}^{M-1} q_j(t)} \quad . \tag{4}$$

The new weighting factors $w_i(t+1)$ can now be computed from the old ones:

$$w_i(t+1) = w_i(t) + \frac{1}{t+1}\left(q'_i(t) - w_i(t)\right) \quad . \tag{5}$$

This is a recursive formulation of the average over all qualities from time zero to t. Using this update equation and the normalisation of the qualities in (4) ensures that the sum of the weights equals one at all times [11].

3.2 Localisation Strategies

Two strategies for computing local confidence maps will be described in the following, one based on distance travelled, the other based on orientation information obtained from the maps. Note that these strategies are mainly used to illustrate how local confidence maps can be computed from information that is readily available. Depending on the sensors used, more sophisticated ones can be added to enhance the localisation accuracy. A main feature of using a fusion approach is that each strategy taken on its own may be quite simple and not very useful for localisation; it is the *combination* of different strategies which makes localisation possible.

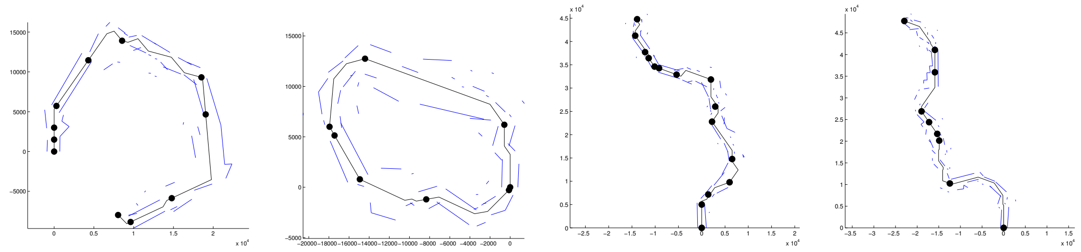
3.2.1 Distance

The first strategy is based on using the distance the robot travelled from its return point to the current position. Note that neither do we care about an exact measurement, nor do we use the actual distance travelled as provided by odometry. Using the odometry data directly would result in very different distances for each journey, as the robot normally moves in a zig-zag fashion. Instead we use distance information computed from the region splitting of the maps, i. e., region length, which is defined by the distance between the "entrance" and the "exit" (split points) the robot used when passing through a particular region. The basic strategy is to compare the distance d travelled when returning home, measured in region lengths taken from the intermediate map computed on the return journey, to the lengths taken from the original map computed during the mapping process.

The confidence for each region in the local confidence map $\boldsymbol{c}_{\text{Dist}}$ depends on the overall distance d travelled on the return journey; the closer a region is to this distance from the origin, the more likely it is the one the robot is in currently. As the distance travelled is an unreliable estimate, adjacent regions should be considered as well, the more the closer they are to the most likely one. We decided to use a Gaussian to model the confidences for each region, the horizontal axis being the distance travelled in mm. The Gaussian is centred at the current overall distance travelled d. Its standard deviation σ is dependent on the distance travelled, and was chosen as $\sigma = 0.05d$. Note that although a Gaussian is used here, we do not try to model a probability density. A Gaussian was rather chosen for a number of reasons making it most suitable for our purpose: It allows for a smooth transition between regions, and the width can be easily adjusted by altering the standard deviation. This is necessary as the overall distance travelled gets more and more unreliable (due to slippage and drift) the farther the robot travels. The confidence value for a region is determined by sampling the Gaussian at the position given by the accumulated distances from the origin (i. e., where the robot started the homeward journey) to the end of this region. After a value for each region is computed, the local confidence map $\boldsymbol{c}_{\text{Dist}}$ is normalised to the interval $[0;1]$.

3.2.2 Relative Orientation

The second method for computing estimates of the robot's position is based on using relative orientation information generated while dividing the map into regions. During its journey, the robot enters a region at one location and exits at a different one, usually including zig-zag movements in between. We define the direction of a region as the direction of the line connecting the entrance and exit points. Certainly this direction information varies every time the robot travels through the environment, but the overall shape between adjacent regions is relatively stable. Therefore, we propose to use angles between region directions as a simple measure of the current position of the robot. It has the advantage that angles between adjacent region directions are a local measure of direction changes, thus keeping the influence of odometry errors due to drift and slippage to a minimum.



(a) (b) (c) (d)

Figure 1: (a) original map for Experiment 1, (b) map generated during the homeward journey in Experiment 1, (c) original map for Experiment 2, (d) homeward journey in Experiment 2. The black dots indicate the points where the map is split into separate regions. The robot movement always starts at the origin. The origin of the homeward journey is (approximately) the same position as the end point coordinate of the respective original map; in particular this means, that the map of the homeward journey for Experiment 2 (d) is upside-down compared to the original map (c).

Firstly, all angles $\alpha_1, \ldots, \alpha_{N-1}$ between adjacent regions in the original map are computed. In the remapping process while going home, new regions are computed in the new map based on data gathered while the robot travels. Using the direction information contained in this map, the angle β between the current region and the previous one can be computed. “Comparing” this angle to all angles of the original map gives a clue (or many) for the current location of the robot, resulting in a local confidence map c_{Dir}:

$$c_{\text{Dir}i} = \frac{1}{2}(\cos|\alpha_i - \beta| + 1), \quad i = 1, \ldots, N-1 \quad . \quad (6)$$

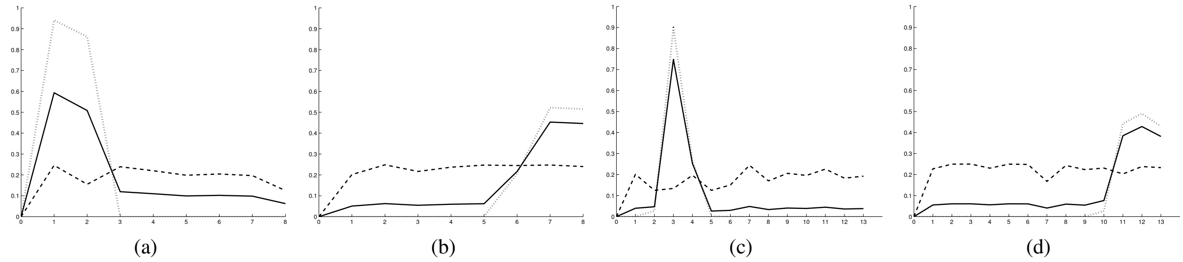
This results in high values for similar angles and low values for dissimilar ones.

4 Experimental Results

The main features of the office environment where we conducted the experiments are corridors, which open into bigger areas at certain locations, doors that are located on the left and right of the corridors, and obstacles like waste paper baskets that can be found on the floor in various positions. The acquisition of the original maps and the experiments for using the map in order to find the way back to the starting position was done on different days, so the environment was different for each experiment (e. g., doors open/closed). Furthermore, we used two different robots, one for generating the original map, the other one for using the map for localisation, to demonstrate the robustness of the methods. Both robots are from Activmedia (a Pioneer 2 and a Pioneer 3), equipped with eight sonar sensors and an odometer. Only the two side sensors were used in order to obtain sparse range data. Results for two experiments are shown in the following.

Figure 1 shows four maps, including the locations of the splitting points marked by dots. These are located on a set of connected lines that resembles the path the robot took while mapping the environment. To the left and right of that path, the (simplified) surfaces representing the environment can be seen. For splitting purposes, gaps were treated as distant surfaces, having a distance of 6000 mm from the position of the robot. Units are given in millimetres, and the robot started the mapping process at the origin. All maps were processed using the same parameter values, namely $\theta_r = 2.0$ and $s = 0.1$; the desired minimum size of a region was 1500 mm. We found that the overall robustness to changes in the parameters is quite high, i. e., the choice of the actual values is usually noncritical; for an evaluation see [1]. It can be observed that the splits are located at the desired positions, i. e., where the environment changes, either from corridor to big room or at sharp bends in the corridor. Note that gaps imply a rapid change as well, because they are treated like distant surfaces, which sometimes leads to splits at positions that may be undesired, but do not pose a problem when using the map for localisation. The maps shown in figures 1(a) and 1(b) were generated from the mapping and going home processes respectively for Experiment 1; the figures 1(c) and 1(d) are the maps generated from the mapping and going home processes respectively for Experiment 2. Comparing the maps generated during mapping and going home highlights the difficulty in using these maps directly for localisation. Each time, the robot goes through the same environment, it will generate different representations due to sensory inaccuracies.

Figure 2 shows two confidence maps for each experiment computed at different locations during the return home journey. The light dotted lines represent the region estimate using the region length information (distance method) and the dark dashed lines depicts the region estimate using the angles between regions (relative orientation method). The solid line is the overall region estimate for the corresponding region. The confidence maps in figure 2 illustrate different situations during localisation. A narrow peak for the overall confidence signifies the robot being very confident of being in a particular region. A wider confidence curve shows that the robot is at the transition from one region to another, as more than one region has a high confidence value, and the robot is unsure which of the regions it is in. Comparisons of the estimated position to the actual position have shown that the localisation is



(a) (b) (c) (d)

Figure 2: Confidence maps computed at different locations during the return home journey. (a),(b) Experiment 1. (c), (d) Experiment 2. The plots show: distance (light dotted), relative orientation (dark dashed), overall confidence (solid). Horizontal axis: region index; vertical axis: confidence (0 to 1).

usually correct, with possible deviations of ±1 in areas where the regions are extremely small.

5 Conclusion

We have presented methods for mapping and localisation using sparse range data without the need for solving the SLAM problem. An initial metric map obtained from sonar sensor readings is divided into distinct regions, thus creating a topological map on top of the metric one. A split and merge approach has been used for this purpose, based on an objective function that computes the quality of a region, including a regularisation term that avoids the formation of very small regions. Based on spatial information derived from these maps, we showed how simple localisation strategies can be used to compute local confidence maps. These are fused into a single global one, which provides an estimate in the form of confidence values, one for each region, that reflects the confidence of the robot being in a particular region. The main advantages of the fusion approach are that it can easily be extended by more sophisticated methods or additional sensors, and that the influence of unreliable strategies on the global confidence map automatically decreases over time. Experimental results show that the robot was highly successful in using our proposed method in localising itself in the environment. Even though this approach does not provide the robot's exact pose, we believe the current output is sufficient for navigation purposes.

6 References

[1] J. Schmidt, C. K. Wong and W. K. Yeap, "A split & merge approach to metric-topological mapbuilding", *Proceedings Int. Conf. on Pattern Recognition (ICPR)*, Hong Kong, 3, pp 1069–1072 (2006).

[2] W. K. Yeap and M. E. Jefferies, "Computing a representation of the local environment", *Artificial Intelligence*, 107(2), pp 265–301 (1999).

[3] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy", *Proceedings Int. Conf. on Robotics and Automation*, New Orleans, LA, pp 4845–4851 (2004).

[4] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation", *Artificial Intelligence*, 99(1), pp 21–71 (1998).

[5] H. Niemann, *Pattern Analysis and Understanding* (2nd edition), Springer, Berlin (1990).

[6] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves", *IEEE Trans. on Computers*, C-23, pp 860 – 870 (1974).

[7] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York (1973).

[8] C. Estrada, J. Neira and J. D. Tardos, "Hierarchical SLAM: real-time accurate mapping of large environments", *IEEE Trans. on Robotics*, 21(4), pp 588–596 (2005).

[9] R. M. Downs and D. Stea, *Image and Environment: Cognitive Mapping and Spatial Behaviour*, Aldine Publishing, Chicago (1973).

[10] J. Triesch, *Vision-Based Robotic Gesture Recognition*, Shaker Verlag, Aachen (1999).

[11] J. Triesch and Ch. von der Malsburg, "Democratic integration: self-organized integration of adaptive cues", *Neural Computation*, 13(9), pp 2049–2074 (2001).

[12] J. Denzler, M. Zobel and J. Triesch, "Probabilistic integration of cues from multiple cameras", in R. Würtz (ed.), *Dynamic Perception*, Aka, Berlin, pp 309–314 (2002).

[13] O. Kähler, J. Denzler and J. Triesch, "Hierarchical sensor data fusion by probabilistic cue integration for robust 3-D object tracking", *Proceedings 6th IEEE Southwest Symp. on Image Analysis and Interpretation*, Nevada, pp 216–220 (2004).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Design of a Simulation Programme for DNA Computing Processes

Mohd Saufee Muhammad and Osamu Ono
Institute of Applied DNA Computing,
Meiji University, Tokyo, Japan
msaufee@isc.meiji.ac.jp, ono@isc.meiji.ac.jp

Abstract

Parallel overlap assembly (POA) and polymerase chain reaction (PCR) are the main processes in DNA computing before the gel electrophoresis image is captured for computing output. The design of the DNA oligonucleotides to represent a problem is quite complicated and is prone to errors. In this paper, we present a simulation programme capable of simulating the POA and PCR processes in order to assist in the design of DNA computing before the actual in vitro experiment is being carried out. This software programme capability is unlimited where problem of any size and complexity can be simulated, thus saving cost due to possible errors during the design process. With this software programme, information regarding the DNA sequence during the computing process as well as the computing output can be extracted at the same time.

Keywords: parallel overlap assembly, polymerase chain reaction, DNA computing, simulation programme, in vitro experiment.

1 Introduction

Interest in using molecules of Deoxyribonucleic Acid or DNA as a medium for computation has increased tremendously ever since Adleman [1] demonstrated it to solve a directed Hamiltonian Path Problem (HPP). Many research results of similar combinatorial problems which are mainly in the realm of computer science and mathematics have been presented [2, 3, 4, 5] since then. The application in solving engineering related problems however, has not been very well established. A proposed method on how DNA computing approach can be used to solve an engineering scheduling problem in the case of an elevator travel path optimisation for a typical building of *N* floors with *M* elevators has been presented [6] based on the researches mentioned.

One of the main problems during the design process is the synthesising of DNA oligonucleotides to represent both the input and output of the problem. Although an in vitro experimental work has been carried out successfully for the case of an elevator travel path optimisation problem, a mechanism for implementing the DNA computing approach for a much larger and complex problem is needed. The main aim of this research is therefore to develop a simulation programme that is capable of simulating the DNA computing process that can verify the expected result before the actual in vitro experiment is carried out. Since the complexity and costs of the DNA oligonucleotides increases for larger and complex problem, this software programme will provide a helpful guide for the DNA computing implementation as to eliminate errors during the design process. Information regarding the DNA oligonucleotides sequences for both the input and output could also be extracted from the simulation programme.

Before the discussion on the simulation programme is presented, an overview of the elevator scheduling problem and its DNA computing solution design is first explained.

2 Elevator Scheduling Problem

Consider a typical building consisting of 2 elevators and 6 floors as illustrated in table 1.

Table 1: Typical elevator positioning of a building.

Floor No	Elevator *A*	Elevator *B*	Hall Call
6		(3 , 2)	
5			
4			↑
3			↓
2			
1	(3 , 5)		

These elevator travel paths can be represented using a weighted graph by representing each elevator position at floors 1, 2, 3, 4, 5, and 6 with nodes V_1, V_2, V_3, V_4, V_5, and V_6 respectively. The weight between each node representing the elevator's travel time between each floor can be formulated as

$$\omega_{|j-i|} = (\,|j-i|\,)\,T_C + T_S \qquad (1)$$

where

i = elevator's present floor position
j = elevator's destination floor position
$|j-i|$ = total number of floors of elevator's movement
T_C = elevator's travelling time between two consecutive floors
T_S = elevator's stopping time at a floor

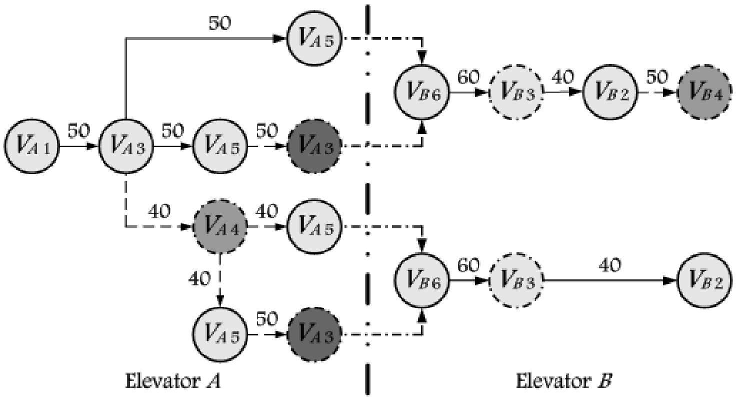


Figure 1: Weighted graph showing all travel path combinations.

Table 2: Total graph output of all travel path combinations.

Hall Calls	Elevator travel path combinations	Total graph output
$A = –$ $B = 3 , 4$	$V_{A1} \rightarrow V_{A3} \rightarrow V_{A5} \rightarrow$ $V_{B6} \rightarrow V_{B3} \rightarrow V_{B2} \rightarrow V_{B4}$	$G(A, B)_1 = 100 + 150$ 250
$A = 3$ $B = 4$	$V_{A1} \rightarrow V_{A3} \rightarrow V_{A5} \rightarrow V_{A3} \rightarrow$ $V_{B6} \rightarrow V_{B3} \rightarrow V_{B2} \rightarrow V_{B4}$	$G(A, B)_2 = 150 + 150$ 300
$A = 4$ $B = 3$	$V_{A1} \rightarrow V_{A3} \rightarrow V_{A4} \rightarrow V_{A5} \rightarrow$ $V_{B6} \rightarrow V_{B3} \rightarrow V_{B2}$	$G(A, B)_3 = 130 + 100$ 230
$A = 3 , 4$ $B = –$	$V_{A1} \rightarrow V_{A3} \rightarrow V_{A4} \rightarrow V_{A5} \rightarrow V_{A3} \rightarrow$ $V_{B6} \rightarrow V_{B3} \rightarrow V_{B2}$	$G(A, B)_4 = 180 + 100$ 280

The output of the graph, given by sum of the graph weights thus represents the total travelling time of the elevator, i.e.

$$G(E_x) = \sum \omega_{|j-i|} \qquad (2)$$

The total travelling time of both the elevators can now be calculated by summing up each of the elevator's travelling time, i.e.

$$G(E_A, E_B) = G(E_A) + G(E_B) \qquad (3)$$

The minimum total travelling time of both elevators thus gives the optimal elevator travel path, i.e.

$$\text{Optimal Travel Path} = G(E_A, E_B)_{min} \qquad (4)$$

Since the building is 6 floors high, the maximum number of floors that the elevator can travel is (6 – 1) = 5 floors. Now, if we assume that $T_C = 5$ s, $T_S = 15$ s, and representing 5 s of time with 10 units we have from (1)

$$\omega_1 = 40 \quad , \quad \omega_2 = 50 \quad , \quad \omega_3 = 60$$

$$\omega_4 = 70 \quad , \quad \omega_5 = 80$$

A weighted graph representing all possible travel path combinations of elevators *A* and *B* with either elevator answering one or both of the hall calls can now be constructed as shown in figure 1. Note that all possible end paths of elevator *A* are joined with the start paths of elevator *B*. This is done in order that the total output of the graph *G* (*A*, *B*) representing the travel path combinations of the elevators can be calculated.

Since there are two hall calls with two available elevators, it is clearly seen that there are $2^2 = 4$ possible travel path combinations for both elevators as tabulated in table 2. The required solution for the elevator scheduling problem is thus the optimal path weight $G(A, B)_2 = 230 = 115$ s.

3 DNA Computing Solution

In order to solve the elevator scheduling problem using DNA computing approach, the weighted graph of figure 1 is first transformed to represent the start, intermediate and end nodes, and also to differentiate the nodes of different travel path combinations as depicted in figure 2. The nodes are then assigned with a specific DNA sequence [7]. All the possible travel path combinations of the elevator are then synthesised so that the DNA oligonucleotides sequence length will directly represent the weight between the nodes as tabulated in table 3. Parallel overlap assembly (POA) [8] is then employed for initial pool generation to generate all the possible travel path combinations, and polymerase chain reaction (PCR) [9, 10] for the amplification of the required optimal path. Finally, gel electrophoresis [11] is performed to separate all the possible travel path combinations according to its length, and the image is captured where the DNA duplex representing the shortest path could be visualised representing the required optimal path solution of the problem as illustrated in figure 3.

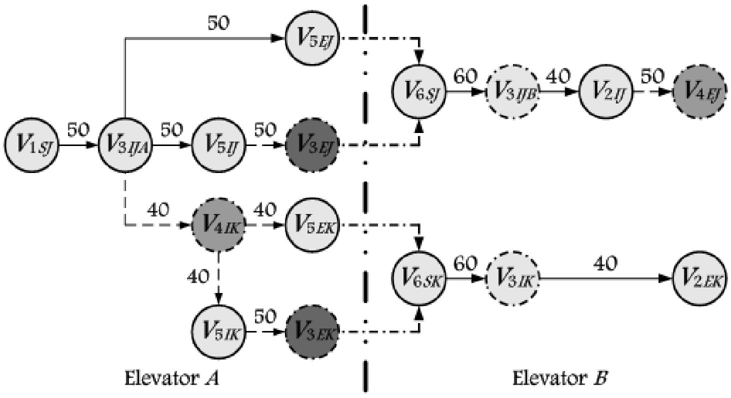


Figure 2: Weighted graph for DNA computing approach showing different node locations and travel paths.

Table 3: Synthesised DNA oligonucleotides for all elevator travel path combinations.

Travel Path	DNA sequences (5’ – 3’)
$V_{1SJ} \rightarrow V_{3IJA}$	CGGCGGTCCACTAAATACTA*aggtcgtttaaggaagtacg*CACTCTTTGTGAACGCCTTC
$V_{3IJA} \rightarrow V_{4IK}$	CACTCTTTGTGAACGCCTTC*acgtcgtgtaacgaagtcct*GTGGGTTAGAGGTAGTCCGG
$V_{3IJA} \rightarrow V_{5IJ}$	CACTCTTTGTGAACGCCTTC*ccgtcggttaagcaagtaatgtactatgct*TGAACCGGCCCTTTATATCT
$V_{3IJA} \rightarrow V_{5EJ}$	CACTCTTTGTGAACGCCTTC*gcgtcgcttaccgaagcacg*CTATAAGGCCAAAGCAGTCG
$V_{4IK} \rightarrow V_{5IK}$	GTGGGTTAGAGGTAGTCCGG*cgctcgttgaagccagtacc*CCGCTGATCCTTGCTAAGTA
$V_{4IK} \rightarrow V_{5EK}$	GTGGGTTAGAGGTAGTCCGG*gcgtcttta*ATGCCTGGCTAAAGTGAGAC
$V_{5SJ} \rightarrow V_{3EJ}$	TGAACCGGCCCTTTATATCT*acgtgttttacccaagtcag*TCATTCGAGTTATTCCTGGG
$V_{5IK} \rightarrow V_{3EK}$	CCGCTGATCCTTGCTAAGTA*gcggcgtgtcacgaactacg*AAATGACCTTTTTAACGGCA
$V_{3EJ} \rightarrow V_{6SJ}$	TCATTCGAGTTATTCCTGGGGGACCTGCATCATACCAGTT
$V_{5EJ} \rightarrow V_{6SJ}$	CTATAAGGCCAAAGCAGTCGGGACCTGCATCATACCAGTT
$V_{3EK} \rightarrow V_{6SK}$	AAATGACCTTTTTAACGGCATGCACGCAAAACTATTTCAT
$V_{5EK} \rightarrow V_{6SK}$	ATGCCTGGCTAAAGTGAGACTGCACGCAAAACTATTTCAT
$V_{6SJ} \rightarrow V_{3IJB}$	GGACCTGCATCATACCAGTT*acgtggtttaaggaagtacggtactatgct*CACTCTTTGTGAACGCCTTC
$V_{6SK} \rightarrow V_{3IK}$	TGCACGCAAAACTATTTCAT*ccgtgggttaaagaagtcctgtactctcct*TCTGCACTGTTAATGAGCCA
$V_{2IJ} \rightarrow V_{4EJ}$	AAAGCCCGTCGGTTAAGTTA*ggtcttttaatcaactaatg*GGAATCCATTGATCGCTTTA
$V_{3IJB} \rightarrow V_{2IJ}$	CACTCTTTGTGAACGCCTTC*acgtcgctgcaagaactacg*AAAGCCCGTCGGTTAAGTTA
$V_{3IK} \rightarrow V_{2EK}$	TCTGCACTGTTAATGAGCCA*acgtcttgtc*CTACGGATAGGTGTCTGGGA

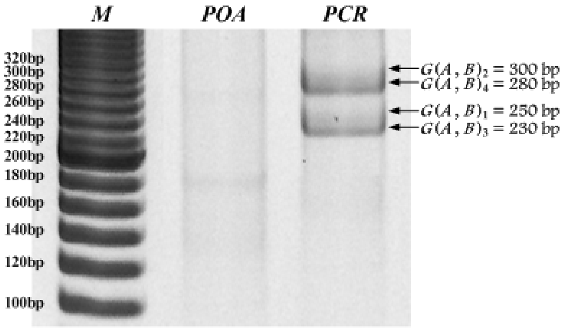


Figure 3: Gel electrophoresis image showing computing output.

The PCR gel image shows 4 bands indicating all the 4 possible travel paths, i.e. $G(A, B)_3$ =230bp, $G(A, B)_1$ = 250bp, $G(A, B)_4$ = 280bp and $G(A, B)_2$ = 300bp. This confirms the expected result that the optimal elevator travel path is given by $G(A, B)_3$ = 230bp = 115 s.

4 Simulation Software Design

The DNA computing implementation involves oligonucleotides sequences design to represent both the problem inputs and outputs. As the problem grows larger, the complexity of the oligonucleotides design gets complicated. In order to assist in the design, we developed a simple simulation programme that is able to simulate the expected DNA process during the computational stage.

The Microsoft Visual Basic platform software programme developed is able to simulate the POA and PCR physical processes of the computation. The flowchart of the designed software programme is shown in figure 4. It is a simple user friendly programme that allows the user to choose between the two processes as shown in figure 5.

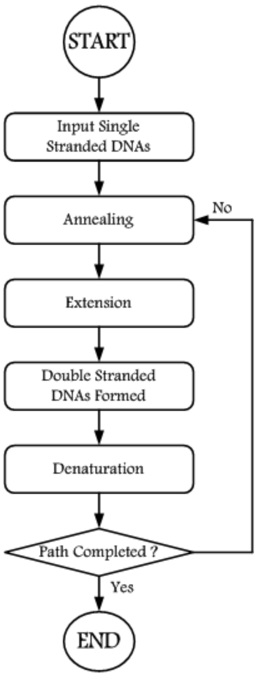


Figure 4: Simulation software flowchart.

Once the process is chosen, a new menu appears that guides the user to enter the input data, start the simulation process and save the output results. Note that for convenience, the weighted graph of figure 2 is relabelled as shown in figure 6. The simulation result is stored in the MS Access database and can be manipulated using the MS Excel. The simulation results for the POA and PCR processes of the problem discussed is shown in figures 7 and 8 respectively.

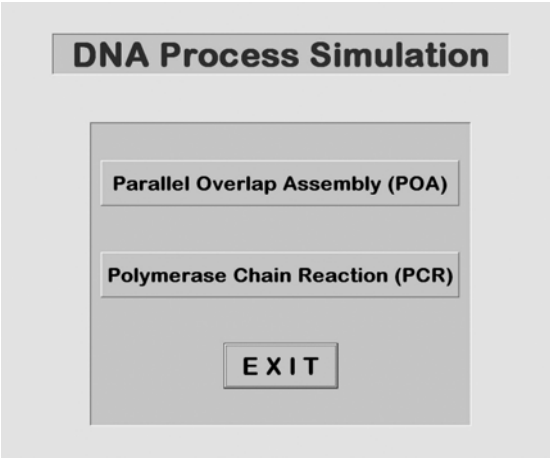

DNA Process Simulation
Parallel Overlap Assembly (POA)
Polymerase Chain Reaction (PCR)
EXIT

Figure 5: Simulation software process menu.

Here, it can be seen that after 4 cycles of POA, all the possible travel path combinations of the elevator are generated. The sequence and length of DNA oligonucleotides representing the travel time is also shown in the simulation output. This verifies the theoretical as well as the in vitro experimental results of the problem. Finally, for the PCR simulation process, as expected, after 2 cycles of PCR process, $2^2 = 4$ sequences are replicated that will represent the DNA computational output of the problem.

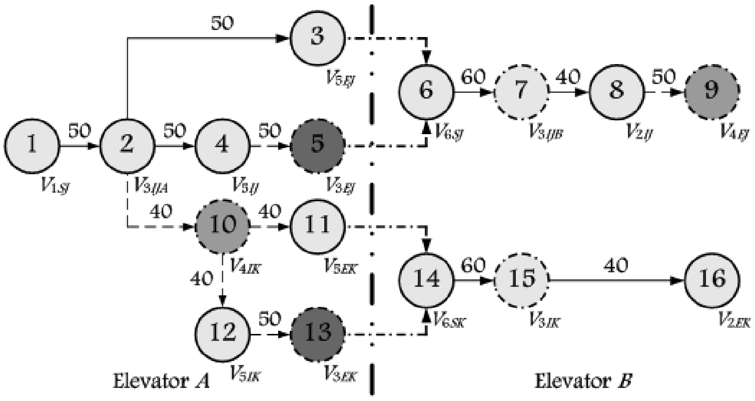
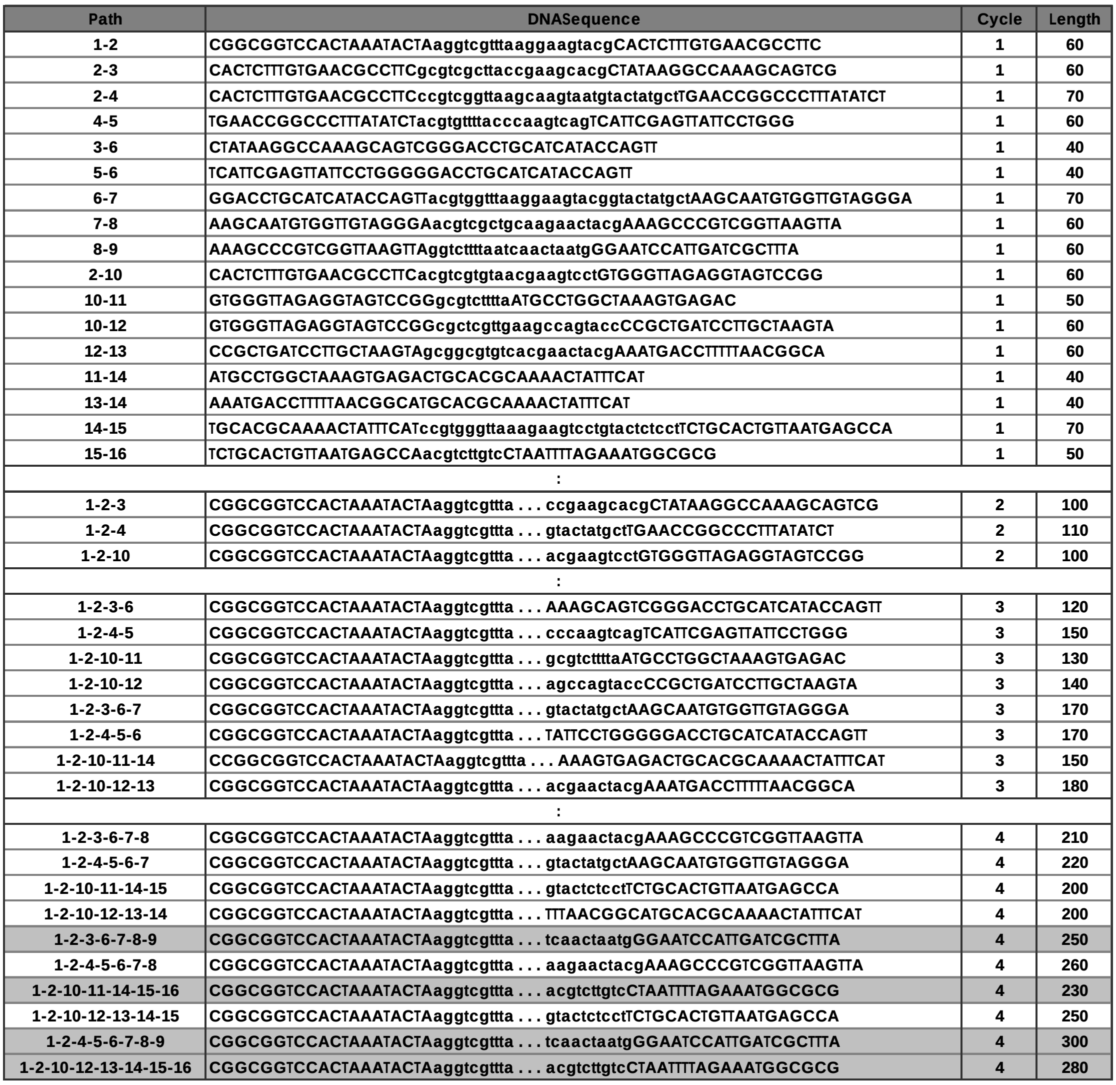
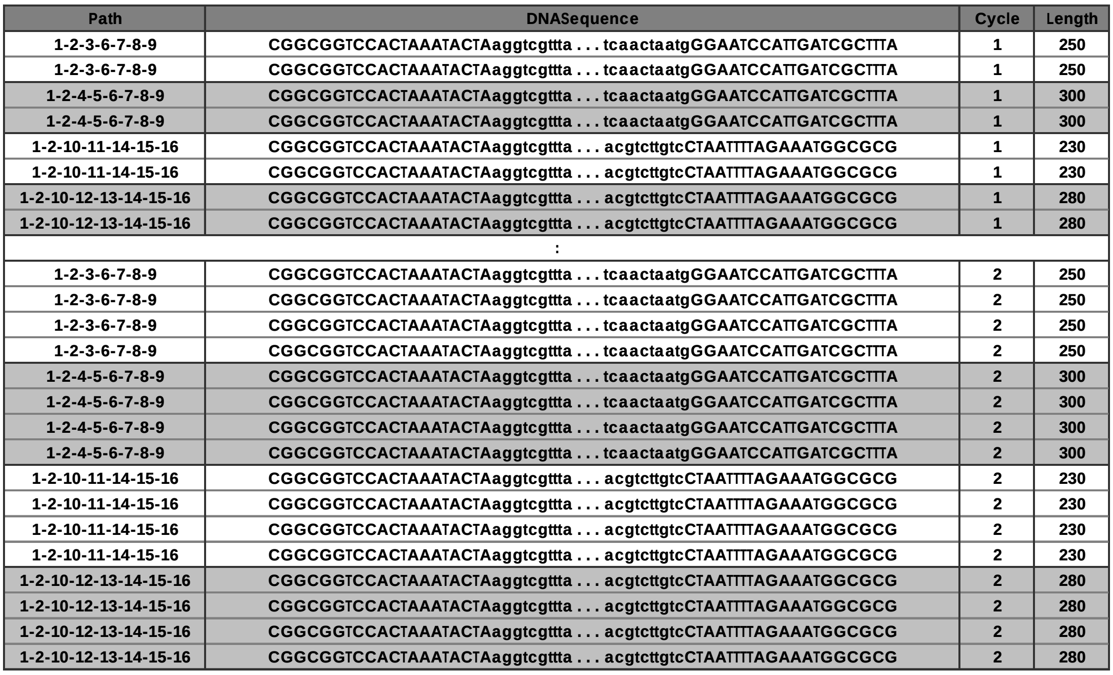


Figure 6: Relabelled weighted graph for simulation software.



Path	DNASequence	Cycle	Length
1-2	CGGCGGTCCACTAAATACTAaggtcgtttaaggaagtacgCACTCTTTGTGAACGCCTTC	1	60
2-3	CACTCTTTGTGAACGCCTTCgcgtcgcttaccgaagcacgCTATAAGGCCAAAGCAGTCG	1	60
2-4	CACTCTTTGTGAACGCCTTCccgtcggttaagcaagtaatgtactatgctTGAACCGGCCCTTTATATCT	1	70
4-5	TGAACCGGCCCTTTATATCTacgtgttttacccaagtcagTCATTCGAGTTATTCCTGGG	1	60
3-6	CTATAAGGCCAAAGCAGTCGGGACCTGCATCATACCAGTT	1	40
5-6	TCATTCGAGTTATTCCTGGGGGACCTGCATCATACCAGTT	1	40
6-7	GGACCTGCATCATACCAGTTacgtggtttaaggaagtacggtactatgctAAGCAATGTGGTTGTAGGGA	1	70
7-8	AAGCAATGTGGTTGTAGGGAacgtcgctgcaagaactacgAAAGCCCGTCGGTTAAGTTA	1	60
8-9	AAAGCCCGTCGGTTAAGTTAggtcttttaatcaactaatgGGAATCCATTGATCGCTTTA	1	60
2-10	CACTCTTTGTGAACGCCTTCacgtcgtgtaacgaagtcctGTGGGTTAGAGGTAGTCCGG	1	60
10-11	GTGGGTTAGAGGTAGTCCGGgcgtcttttaATGCCTGGCTAAAGTGAGAC	1	50
10-12	GTGGGTTAGAGGTAGTCCGGcgctcgttgaagccagtaccCCGCTGATCCTTGCTAAGTA	1	60
12-13	CCGCTGATCCTTGCTAAGTAgcggcgtgtcacgaactacgAAATGACCTTTTTAACGGCA	1	60
11-14	ATGCCTGGCTAAAGTGAGACTGCACGCAAAACTATTTCAT	1	40
13-14	AAATGACCTTTTTAACGGCATGCACGCAAAACTATTTCAT	1	40
14-15	TGCACGCAAAACTATTTCATccgtgggttaaagaagtcctgtactctcctTCTGCACTGTTAATGAGCCA	1	70
15-16	TCTGCACTGTTAATGAGCCAacgtcttgtcCTAATTTTAGAAATGGCGCG	1	50
:			
1-2-3	CGGCGGTCCACTAAATACTAaggtcgttta . . . ccgaagcacgCTATAAGGCCAAAGCAGTCG	2	100
1-2-4	CGGCGGTCCACTAAATACTAaggtcgttta . . . gtactatgctTGAACCGGCCCTTTATATCT	2	110
1-2-10	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgaagtcctGTGGGTTAGAGGTAGTCCGG	2	100
:			
1-2-3-6	CGGCGGTCCACTAAATACTAaggtcgttta . . . AAAGCAGTCGGGACCTGCATCATACCAGTT	3	120
1-2-4-5	CGGCGGTCCACTAAATACTAaggtcgttta . . . cccaagtcagTCATTCGAGTTATTCCTGGG	3	150
1-2-10-11	CGGCGGTCCACTAAATACTAaggtcgttta . . . gcgtcttttaATGCCTGGCTAAAGTGAGAC	3	130
1-2-10-12	CGGCGGTCCACTAAATACTAaggtcgttta . . . agccagtaccCCGCTGATCCTTGCTAAGTA	3	140
1-2-3-6-7	CGGCGGTCCACTAAATACTAaggtcgttta . . . gtactatgctAAGCAATGTGGTTGTAGGGA	3	170
1-2-4-5-6	CGGCGGTCCACTAAATACTAaggtcgttta . . . TATTCCTGGGGGACCTGCATCATACCAGTT	3	170
1-2-10-11-14	CCGGCGGTCCACTAAATACTAaggtcgttta . . . AAAGTGAGACTGCACGCAAAACTATTTCAT	3	150
1-2-10-12-13	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgaactacgAAATGACCTTTTTAACGGCA	3	180
:			
1-2-3-6-7-8	CGGCGGTCCACTAAATACTAaggtcgttta . . . aagaactacgAAAGCCCGTCGGTTAAGTTA	4	210
1-2-4-5-6-7	CGGCGGTCCACTAAATACTAaggtcgttta . . . gtactatgctAAGCAATGTGGTTGTAGGGA	4	220
1-2-10-11-14-15	CGGCGGTCCACTAAATACTAaggtcgttta . . . gtactctcctTCTGCACTGTTAATGAGCCA	4	200
1-2-10-12-13-14	CGGCGGTCCACTAAATACTAaggtcgttta . . . TTTAACGGCATGCACGCAAAACTATTTCAT	4	200
1-2-3-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	4	250
1-2-4-5-6-7-8	CGGCGGTCCACTAAATACTAaggtcgttta . . . aagaactacgAAAGCCCGTCGGTTAAGTTA	4	260
1-2-10-11-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	4	230
1-2-10-12-13-14-15	CGGCGGTCCACTAAATACTAaggtcgttta . . . gtactctcctTCTGCACTGTTAATGAGCCA	4	250
1-2-4-5-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	4	300
1-2-10-12-13-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	4	280

Figure 7: POA process simulation showing all the possible travel path combinations.



Path	DNASequence	Cycle	Length
1-2-3-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	1	250
1-2-3-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	1	250
1-2-4-5-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	1	300
1-2-4-5-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	1	300
1-2-10-11-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	1	230
1-2-10-11-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	1	230
1-2-10-12-13-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	1	280
1-2-10-12-13-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	1	280
:			
1-2-3-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	250
1-2-3-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	250
1-2-3-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	250
1-2-3-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	250
1-2-4-5-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	300
1-2-4-5-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	300
1-2-4-5-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	300
1-2-4-5-6-7-8-9	CGGCGGTCCACTAAATACTAaggtcgttta . . . tcaactaatgGGAATCCATTGATCGCTTTA	2	300
1-2-10-11-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	230
1-2-10-11-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	230
1-2-10-11-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	230
1-2-10-11-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	230
1-2-10-12-13-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	280
1-2-10-12-13-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	280
1-2-10-12-13-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	280
1-2-10-12-13-14-15-16	CGGCGGTCCACTAAATACTAaggtcgttta . . . acgtcttgtcCTAATTTTAGAAATGGCGCG	2	280

Figure 8: PCR process simulation of travel path combinations for 2 cycles.

5 Conclusions and Discussions

In this paper, we first discussed the proposed DNA computing approach to solve an elevator scheduling optimisation problem. The in vitro experimental result is then presented that verifies the expected computation output. In order to assist in designing and synthesising the DNA oligonucleotides for a larger and complex problem, a simulation programme capable to simulate the POA and PCR physical processes is developed. With this successful DNA computing design, in vitro experimental implementation and simulation software, the applicability and feasibility of the DNA computing approach could therefore be extended into many more complex problems of this type of nature.

6 Acknowledgements

The first author is very grateful to Universiti Malaysia Sarawak (UNIMAS), Malaysia for granting a study leave at Meiji University, Japan.

7 References

[1] L.M. Adleman, “Molecular computation of solutions to combinatorial problems”, *Science*, 266, pp 1021-1024 (1994).

[2] A. Narayanan and S. Zorbalas, “DNA algorithms for computing shortest paths”, *Proceedings of Genetic Programming*, Wisconsin, pp 718-723 (1998).

[3] Y. Yamamoto, A. Kameda, N. Matsuura, T. Shiba, Y. Kawazoe and A. Ahochi, “Local search by concentration-controlled DNA computing”, *International Journal of Computational Intelligence and Applications*, 2, pp 447-455 (2002).

[4] J.Y. Lee, S.Y. Shin, S.J. Augh, T.H. Park and B.T. Zhang, “Temperature gradient-based DNA computing for graph problems with weighted edges”, *Lecture Notes in Computer Science*, 2568, pp 73-84 (2003).

[5] Z. Ibrahim, Y. Tsuboi, O. Ono and M. Khalid, “Direct-proportional length-based DNA computing for shortest path problem”, *International Journal of Computer Science and Applications*, 1(1), pp 46-60 (2004).

[6] M.S. Muhammad, Z. Ibrahim, O. Ono and M. Khalid, “Direct-proportional length-based DNA computing implementation for elevator scheduling problem”, *Proceedings of the IEEE International Region 10 Conference (TENCON2005),* Melbourne, pp 711-715 (2005).

[7] F. Udo, S. Sam, B. Wolfgang and R. Hilmar, “DNA sequence generator: a program for the construction of DNA sequences”, *Proceedings 7th International Workshop on DNA Based Computers,* Florida, pp 23-32 (2001).

[8] P.D. Kaplan, Q. Ouyang, D.S. Thaler and A. Libchaber, “Parallel overlap assembly for the construction of computational DNA libraries”, *Journal of Theoretical Biology*, 188(3), pp 333-341 (1997).

[9] J.Y. Lee, H.W. Lim, S.I. Yoo, B.T. Zhang and T.H. Park, “Efficient initial pool generation for weighted graph problems using parallel overlap assembly”, *Proceedings 10th International Meeting on DNA Computing*, Milan, pp 357-364 (2004).

[10] J.P. Fitch, *Engineering Introduction to Biotechnology,* SPIE Press, Washington (2001).

[11] G. Paun, G. Rozenberg and A. Salomaa, “DNA computing: new computing paradigms”, *Lecture Notes in Computer Science*, 1644, pp 106-118 (1998).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

RoboSim: Multimode 3D Robotic Simulator For Study of Co-operative Robot Behaviour

G. Seet[1], S.K. Sim[1], W.C. Pang[1], Z.X. Wu[2], T. Asokan[3]
[1]Robotics Research Centre, School of Mechanical & Aerospace Engineering,
Nanyang Technological University, Singapore
[2]Agency for Science, Technology and Research, Singapore
[3]Department of Mechanical Engineering, Indian Institute of Technology, Chennai, India
mglseet@ntu.edu.sg, msksim@ntu.edu.sg, weeching@ntu.edu.sg,
wuzx@ihpc.a-star.edu.sg, asok@iitm.ac.in

Abstract

The design and development of an interactive 3D robotic simulator, as a platform for the study of co-operative robot behaviour, is described in this paper. The simulator provides a visually realistic 3D world with simulated robot agents and geographical terrain, for the purpose of evaluating the performance of robots in various scenarios. The simulator was developed in C++ using open source packages, resulting in a modular and cost effective package. The modelling and simulation of robotic vehicles, terrains, sensors and behaviours are presented. Artificial Intelligent agent behaviours in co-operative robotic scenarios were developed and tested using the simulator and the results presented.

Keywords: robotics, simulator, co-operative behaviour, artificial agents, multimode

1 Introduction

Simulations [1],[2],[3],[4] are frequently used to test the performance of robots in different operating scenarios before deploying them to real tasks. One of the main reasons for this is the cost involved in the deployment, as well as the difficulties in executing hundreds of trials using real physical robots.

‘RoboSim’ is a 3D simulator developed at the Nanyang Technological University, Robotics Research Centre, for the study of co-operative behaviour of mobile robots. Different operating modes like tele-operation, semi-autonomous and autonomous operation are built into the simulator. This gives the simulator a multi-mode simulation capability.

The three-dimensional terrain, environment and entities provide the operator a high level of realism in scenario simulations and in the development of deployment strategy. Inclusion of geo-specific imagery and digital elevation maps (DEM), for terrain modelling, helps to model and simulate any specific operating terrain as required.

In RoboSim, different types of robots, e.g. ATRV-JR and Pioneer, are simulated. Different sensors, equipment payloads and other accessories are modelled as plug-ins for a modular design and facilitate integration of additional components. The simulator structure, components and some of the major developmental contributions are highlighted in the following sections. The virtual environment is also simulated to change its sky, terrain and weather conditions according to time or humans’ desire to demonstrate the effects environment has on different sensors and robots during the simulation.

RoboSim has been created on the Linux operating system together with some open source graphic libraries [5],[6]. The simulator unifies the software development, control operation and behaviour of real and virtual robots. The Network Collaborative Virtual Environment is designed as one of its components such that RoboSim would be network distributed – simulation and rendering can be done on different machines connected through a network.

2 Related Works

Over the last three decades, significant advancements in computer graphics, processor capability, networking and Artificial Intelligence (AI) have created exciting new options in simulation technology that provides high realism environments for military training and robotics research. For this paper, the focus is on simulators that attempt to model the real world in a sufficient level of detail, preferably network distributed, and provide a platform for developing and testing algorithms for multiple mobile robots.

Research on networked simulators has been ongoing and one of the earliest network simulators is the SIMNET [7] project funded by DARPA to investigate the feasibility of creating a real-time distributed simulator for combat simulation. Distributed Interactive Simulation (DIS [8]), a follow-on from the SIMNET system, defined object-event architecture and 27 protocol data units (PDU), which allowed any type of player or machine complying with its protocol to participate in the simulation.

In the academic community, the Swedish Institute of Computer Science developed a Distributed Interactive Virtual Environment (DIVE) [9], an internet-based multi-user VR system, to support the development of virtual environments, user interfaces and applications based on shared 3D synthetic environments by simulating a large shared memory over a network. BrickNet [10] and VLNet [11] were developed in the late 1990s with emphasis on multiple servers, object sharing and user representation.

In the military and robotics research community, there are several software simulators that have modelled the real world in a high level of detail. Topscene [12] is one simulator that includes real-time geo-specific visualisation systems, with aerial and satellite images and a high-resolution three-dimensional database of regions. Some simulators are derived from 3D engines. For instance, the Urban Search and Rescue (USAR) robot simulator [13], derived from the Unreal^{TM} engine, is visually compelling and offers a truly 3D environment.

Among all the above-mentioned research works, it can be noticed that the focus is placed on the virtual environment simulation. There has been little study on how realistic these simulations are when compared to the real world. The dynamics in such simulations are only approximate at best. As such, these simulators lack sufficient accuracy to enable development of algorithms to transfer from simulation to reality with any confidence. The lack of consistency and interface between the simulated world and the actual one are the major drawbacks of the various previous works, especially when automated systems like mobile robots are involved.

Other simulators, such as Gazebo [3] and Webots [4], may provide a dynamically accurate motion, a variety of actuator primitives and support a wide range of sensor suites, but most of the time these simulators are designed to simulate a small population and compromise graphical realism, especially for large outdoor environments with trees, buildings and roads. Some of these softwares are tied to a particular hardware in a specific application area. RoboSim is a piece of work that attempts to provide a generic robotic simulator application, supporting networked multi-users, multi-modes, with realistic terrene environment, sensors and robots modelling. It also provides the ability to test scenarios and port algorithms between the real and virtual environments.

3 System Structure

3.1 System Components

The three major components of the simulator are:

- Networked collaborative virtual environment (NCVE)
- Entity simulator
- Sensor module

NCVE functions as a virtual environment simulator with network features. Simulation and rendering can be done on a single, or on different, machines connected through a network.

The entity simulator is responsible for simulating the behaviour/performance of entities like robot vehicles and payloads.

The sensor module is the carrier of codes for the various sensors, which simulate the physical characteristics of each type of sensor and its behaviour at various levels of detail and conditions.

3.2 System Configuration

The overall system configuration is identified as shown in figure 1.

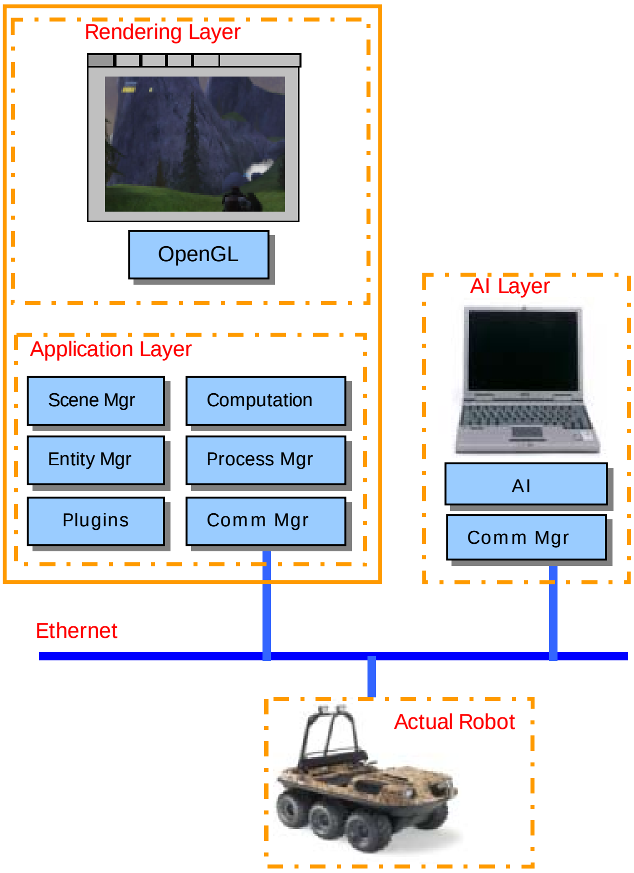


Figure 1: System configuration.

There are three distinct layers at the simulator level:

- Rendering layer
- Application layer
- AI layer

The rendering layer is responsible for rendering graphical the models of simulated environment using hardware accelerated OpenGL script.

The application layer consists of the scene manager, entity manager, the process manager and network communication manager. The scene manager organises the entire object in the simulation to provide realistic scene and environments at the lowest possible processing cost. It incorporates advanced 3D rendering techniques like view culling and the level of detail (LOD). Each entity in the simulator has a graphical model, a collision model and a kinematics model. The entity is simulated as an independent computational unit.

The AI layer consists of different AI behaviours and is designed to be a separate module and run independent of the simulator. This option was chosen due to its complexity and high computational requirement. Inputs to the AI layer, like data for some sensors, are however collected from the entity running in the simulator. Control commands, from the AI layer, are sent back to the entity. Most of the components are developed as plug-ins to the simulator. This structure provides a clearer interface and ease of software maintenance.

3.2.1 Communication Manager and Network Layout

The communication manger implements a TCP/IP network interface. We have used two communication protocols in our network: UDP/IP and TCP/IP. The simplicity of UDP/IP and the reliability of TCP/IP are conveniently exploited to maximise performance. Critical information like tasks assignment is sent through TCP whilst information like robot status, payload status is through the UDP protocol.

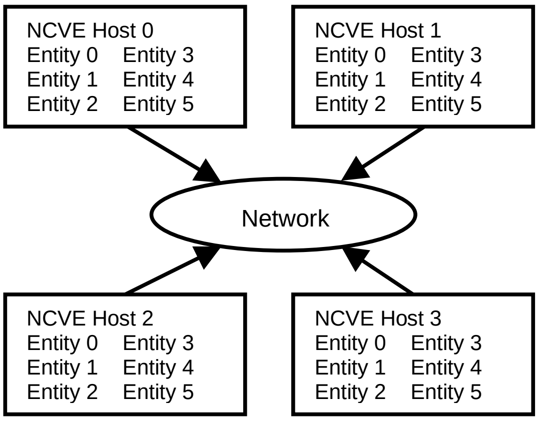


Figure 2: Robotic simulator network layout.

Figure 2 shows the network layout of the robotic simulator running on various hosts connected by a TCP/IP network.

All the hosts differ only as being either a local or a remote system, when viewed by the simulated entity. Computational load is distributed to individual systems. The communication manager only handles system level communication tasks, like discovering running hosts, establishing connection to remote hosts and creating the entity to simulate. Once the entity is created it has its own means of communication capability.

3.2.2 Process Manager

Computers were typically equipped with one CPU. The majority of games and virtual environments were designed to use a single process or thread. Dual-core and multiple chip computers are becoming more widely available in the consumer market. RoboSim is developed as a multi-thread application following this trend. A parallel process, however, cannot read data that is being written by an upstream process concurrently. In order to achieve full multi-process with multi-thread safety, a large overhead of data buffering, protection and synchronisation needs to be added to the system processes. General multi-threaded read and write access is computational-inefficient and complicates its design. In our Robotic Simulator we employed a synchronous write, asynchronous read (SWAR) approach (figure 3).

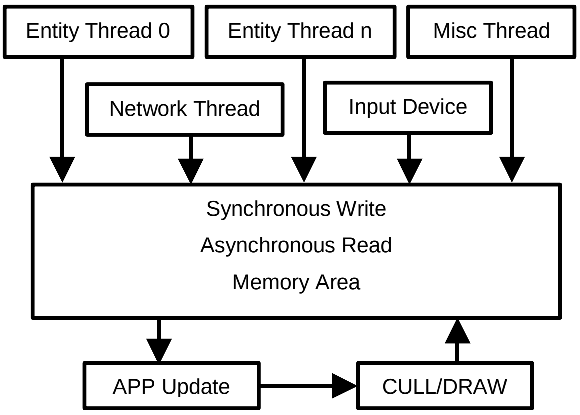


Figure 3: Robotic simulator process model.

Shared data can only be written by a single thread and read by multiple threads. At the top of figure 3 are shown various threads, such as entity threads and network threads. These are run independently.

3.2.3 Device Manager

The device manager is for managing the peripheral devices like the joystick and steering wheel attached to the simulator. It would read data from the hardware device and pre-process the data. Most of the controlling devices have a USB interface which unifies the control data.

3.2.4 Event Manager

The event Manager implements an event queue. Event types such as input events, command events and network events are generated during the simulation process and are stored into the queue. Each module or plug-in that implements the event interface can register with the event queue and is notified when certain events happen. The event queue is generally used for system level control.

3.2.5 Entity Simulation

On each host running RoboSim, there are two types of entity: local and remote. Local entities are those that run on the local system and broadcast their status to other systems. A remote entity serves as a duplicated copy of those entities running on other remote systems. Computations are only done for local entities while remote entities are updated only upon receiving update information.

3.2.6 Sensor Simulation

The sensor simulator simulates the physical characteristics of each type of sensor, including laser scanner, differential GPS, collision detectors, radio communication, electronic compass, inclinometer and specialised payload. Environment conditions usually have an effect some sensor characteristics. This requires the sensors to be incorporated with the global simulation condition. Most of the sensors of interest are ranging devices.

3.3 Graphical User Interface

The graphical user interface (GUI) acts as a user-friendly interface between the human operator and the simulator. The hallmark of GUI programming lies in its graphical control features such as toolbar, buttons and icons. There are many cross platform GUI packages available (motif, gtk, qt, wxwidget, etc.).

The GUI is based on OpenGL, a platform similar to that for graphical rendering. This maximises the portability of the software and provides a unified programming interface. A variety of input devices such as mouse, keyboard and joysticks are provided to allow different functionalities for the operator to interact efficiently with the simulator and control its activities.

3.3.1 GUI Design Structure

The basic GUI design concept of the simulator is to develop a separate and independent class for the GUI package.

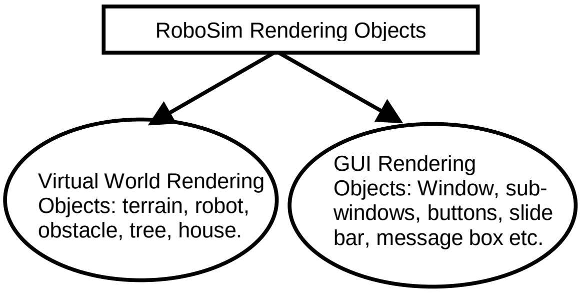

RoboSim Rendering Objects
Virtual World Rendering Objects: terrain, robot, obstacle, tree, house.
GUI Rendering Objects: Window, sub-windows, buttons, slide bar, message box etc.

Figure 4: Robotic simulator rendering object.

As shown in figure 4, the GUI works as a separate rendering object. This allows the concurrent development of the virtual world and GUI, and testing, to be performed separately. Figure 5 shows a detailed view of the OpenGL GUI and figure 6 shows an overview of the simulator running multiple robots, together with its GUI. The GUI includes the main menu, map view window, camera view and robot information windows.

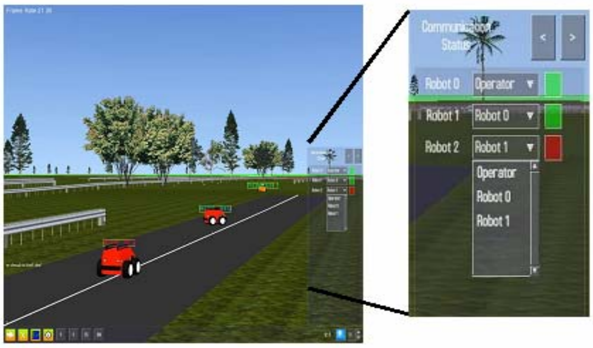


Figure 5: Robotic simulator OpenGL GUI.

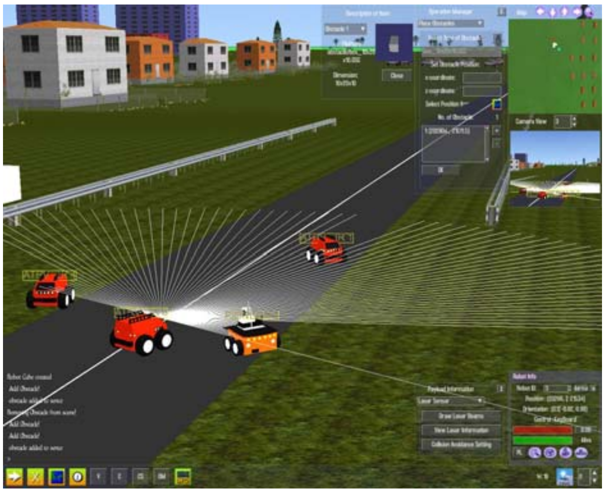


Figure 6: Robotic simulator overview.

3.4 Real World Modelling

Rendering the virtual environment only forms part of the simulation. We require a collision detection system to prevent the moving object penetrating through other objects and a dynamic system to simulate the motion of these moving objects. Recent advances in the graphical processing unit allow us to render objects at a satisfactory level of details and speed. In the current development of the virtual environment simulation, the bottleneck is in the CPU, which performs the detailed computation on collision and dynamics. Hence different level of detail models are used for the graphical system and the collision system. Real-time dynamic level of detail terrain rendering algorithms provide a better visual effect. The Seumas McNally's Tread Marks algorithm [14] is used in rendering the terrain. It is fast and memory efficient, requiring 3 bytes/pixel plus 20 bytes per drawn triangle. Tested with a terrain grid of 2049*2049 on a Pentium IV 2.8GHz machine, it achieved a rendering speed generally higher than 60GHz.

3.4.1 Collision Detection

Fast and accurate collision detection is critical for real time responsiveness. It is central to the modelling of ranging sensors used in the simulation. The collision system must be able to detect the occurrence of collision and also determine the point of contact.

Collision detection is one of the most complicated problems in computer graphics. To provide fast detection, bounding volumes are applied as a first check before performing further detailed calculations for determining the point of contact. Due to the relative ease of determining the overlapping of bounding boxes, this enhances the efficiency of the collision detection process. In the simulator, bounding boxes for all the physical models are constructed to facilitate the process.

3.5 AI Planner

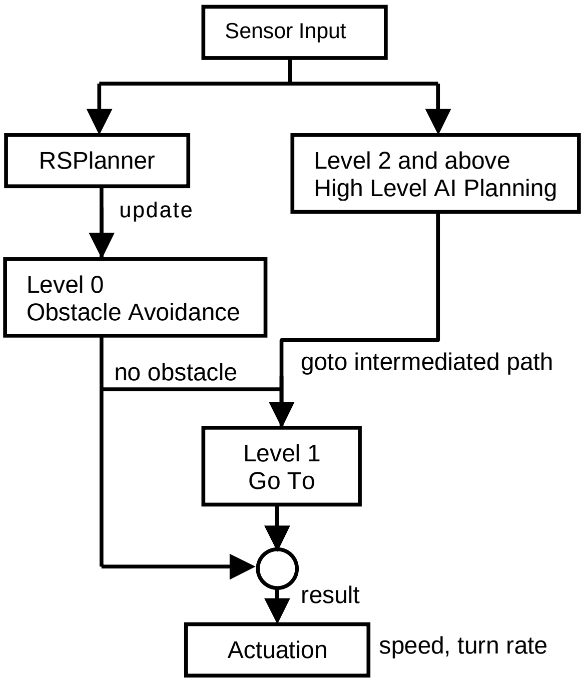


Figure 7: Robotic simulator AI planning.

The AI planner in the RoboSim provides the basic navigation capability to the simulated entities and serves as a foundation for higher level AI processing. Basic AI behaviours includes simple obstacle avoidance, path planning, ‘goto’ behaviour, leader following and road following.

Figure 7 shows the hierarchy of the AI planner. In the planning process, lower level of planning implies higher priority. The simulator also provides interfaces for the option of adding a plug-in for an external AI control algorithm.

4 Testing of AI Algorithms

One of the objectives of developing the simulator is to test different external AI algorithms for co-operative robotic tasks. Different operating scenarios requiring specialised algorithms can be tested. Simple algorithms like vehicle following and obstacle avoidance were tested to analyse the simulator’s performance.

A third party AI algorithm for buried object detection, using multiple robots, was tested to study the client-server features as well as the robot co-operation behaviour. The results showed a good co-ordination between the robots and achieved comprehensive coverage of the search area.

5 Conclusion

The design and development of a multimode 3D robotic simulator for a PC host was described. The multi-mode capability allows the use of a virtual environment and robots to test and verify advance AI algorithms.

The simulator is further being developed as a platform to integrate both virtual and real robots, unifying the development of intelligent mobile robotics. The RoboSim simulator combines the benefits of the multi-mode configuration described and the use of simulators for fast effective evaluation of the behaviour based code described, previously reported in [15],[16].

6 Acknowledgements

The authors wish to acknowledge the support provided by the Defence Science and Technology Agency and the Agency for Science, Technology and Research, Singapore, for this work.

7 References

[1] J.S. Dahmann, “High level architecture for simulation”, *1st International Workshop on Distributed Interactive Simulation and Real-Time Application (DIS-RT 97)*, p9 (1997).

[2] H.W. Holbrook, S.K. Singhal and D.R. Cheriton, “Log-based receiver-reliable multicast for distributed interactive simulation”, *Computer Communications Review*, 25(4), pp 328-341 (1995).

[3] N. Koenig and A. Howard, *Gazebo - 3D Multiple Robot Simulator with Dynamics*, http://playerstage.sourceforge.net/gazebo/ gazebo.html, visited on 1/9/2006.

[4] O. Michel, “Webots^{TM}: professional mobile robot simulation”, *International Journal of Advanced Robotics Systems*, 1(1), pp 39-42 (2004).

[5] D. Burns, *A New Processing Model for Multithreaded, Multidisplay Scene Graphs*, http://www.donburns.net/OSG/Articles/OSGMP/ , visited on 28/3/2006.

[6] P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, N. Faust and G. Turner, “Real-time,

continuous level of detail rendering of height fields", *Proceedings SIGGRAPH 96*, pp 109-118 (1996).

[7] R.S. Johnston, "The SIMNET visual system", *Proceedings of the 9th ITEC Conference*, Washington, D.C., pp 264-273 (1987).

[8] L.B. McDonald and C. Bouwens, *Rationale Document for Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation*, Institute for Simulation and Training Publication IST-PD-90-1-Revised, University of Central Florida, Orlando, FL. (1991).

[9] E. Frcon and M. Stenius, "DIVE: a scaleable network architecture for distributed virtual environments", *Distributed Systems Engineering Journal* (Special Issue on Distributed Virtual Environments), 5(3), pp 91-100 (1998).

[10] G. Singh, L. Serra, W. Png and H. Ng, "BrickNet: a software toolkit for network-based virtual worlds", *Presence*, MIT Press, 3(1), pp 19-34 (1994).

[11] I.S. Pandzic, T.K. Capin, N.M. Thalmann and D. Thalmann, "VLNET: a networked multimedia 3D environment with virtual humans", *Proceedings Multi-Media Modelling (MMM 95)*, World Scientific, Singapore, pp 21-32 (1995).

[12] Lockheed Martin Corp, *TOPSCENE, Rehearsal & Simulation Systems*, http://www.topscene.com, visited on 28/3/2006.

[13] M. Lewis, K. Sycara and I. Nourbakhsh, "Developing a testbed for studying human-robot interaction in urban search and rescue", *Proceedings 10th International Conference on Human Computer Interaction (HCII 03)*, Crete, Greece, pp 270-274 (2003).

[14] B. Turner, *Real-time Dynamic Level of Detail Terrain Rendering with ROAM*, Gamasutra, http://www.gamasutra.com/features/20000403/turner_pfv.htm, visited on 29/3/2006.

[15] X. Wang, G.G.L. Seet, M.W.S. Lau, E. Low and K.C. Tan, "Exploiting force feedback in pilot training and control of an underwater robotics vehicle: an implementation in LabVIEW", *OCEANS 2000*, MTS/IEEE, pp 2037-42 (2000).

[16] J. Loh, J. Heng and G. Seet, "Behaviour-based search using small autonomous mobile robot vehicles", *2nd International Conference on Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, pp 294-301 (1998).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Efficient Robotic Search Strategies for Finding Disaster Victims

Raymond A. Jarvis and Mohamed S. Marzouqi
Australian Research Council Centre for Perceptive and Intelligent Machines
in Complex Environments:
Intelligent Robotics Research Centre,
Monash University, Clayton, Australia
ray.jarvis@eng.monash.edu.au

Abstract

The early discovery of disaster victims after a catastrophe could well be critical to their lives. This paper considers a robot search strategy which minimises the average time to find victims using a combination of convex visibility region segmentation, probabilistic spread functions which operate in obstacle strewn environments and travelling salesman visit strategies. Whilst the ideas presented here lack aspects of reality, they make a good start in understanding these kinds of problems.

Keywords: robotic search, disaster victims, travelling salesman, visibility regions.

1 Introduction

In a disaster search and rescue situation over wide areas, perhaps initially known but possibly now devastated, it is clearly of value to find victims as quickly as possible, since their lives and limbs may depend on a timely rescue. But not everyone can be found first, so how best should a search for victims be carried out? The assumption here is that only the finding (sighting) of victims is to be dealt with. Rescue after location is outside the scope of this paper.

A reasonable optimality criterion to adopt is that of minimal average time for sighting, since this will take both numbers of victims and search times into consideration. Ultimately, diligence requires that the search extends to everywhere possible. However, the order in which different areas are searched will critically alter the optimality value calculated.

It is to be admitted that many realistic details are not included in this study. These include the existing dangers to searchers, the practicality of entering various areas, visibility limitations in smoke, dust and fire or night conditions, whether victims can be communicated to by cell phone and so on. Thus this paper is just a small beginning in thinking about search (followed by rescue) strategies. Simplistic assumptions can only be justified as a means of making a start which should ultimately lead to better solutions able to cope with the anticipated realities of genuine disasters, including chaos.

The next section outlines the framework of the problem dealt with here. Next, some basic tools for providing solutions are described (5 sections). Dealing with incremental remappings of the initially known environment subjected to devastation is next touched on. Results to date follow. Future developments are then suggested. A conclusions section closes the paper.

2 Visibility, Convex Region Segmentation and Portals

Visibility is taken to mean the existence of a line of sight between the searcher and a victim. Only obstacles are considered as impediments to visibility, not smoke, dust, fire or other possibilities, such as size on the image plane. There exist algorithms for determining this kind of visibility in polyhedral obstacled space [1] and others that use simple ray tracings in rectangularly tessellated space [2]. Only two-dimensional spaces are considered here. This means that obstacles are projected on the flat ground plane. Three-dimensional extensions are not difficult to consider, but, nevertheless, will not be considered here. A convex region in the plane is one which collects together (and finds the perimeter) spaces (or cells) within which every point is visible to every other point. Interfaces between convex regions allow us to define ‘portals’, which are single points, arbitrarily chosen, one for each such interface. These ‘portals’ can be used as nodes in a search path matrix where each path is of minimal length in a collision-free sense. However, for simplicity, in this paper the centre of area of each convex area is considered the local target. Full use of the ‘portal’ notion can be considered an extension.

Thus, it is adopted here that, once a searcher enters any part of a convex region (via a portal on the way to the local centre of area), all parts of that region become visible and thus all victims in that region are considered found. If the initially known search environment were not altered, then visibility regions could be calculated ‘a priori’ with respect to the search. We will start with this unlikely situation and later consider what could be done about incrementally discovered changes. An alternative approach would be to consider that an aerial view has provided the knowledge of the environment after the disaster.

Again, for simplicity, we will assume, initially, there is only one searcher and deal with multiple, cooperative searchers later.

3 Visibility-Based Map Segmentation

The map segmentation algorithm in [3] deals with stationary, obstacle-strewn environments. It divides a tessellated free space into convex regions (i.e. the visibility line connecting any two cells within a region does not penetrate an obstacle). Examples are shown in figures 2, 4, and 6. Obstacles are in black. The different convex regions are labelled by different colours.

Briefly, the first step in the process of map segmentation is to find its ‘visibility map’. Each cell’s value is the number of free space cells visible to it (i.e. its exposure). In the second step, the visibility map is used to find the cell with the lowest visibility value. This cell is given the next region number (ID) and treated as a seed for that region. Iteratively, the neighbour cells (that have not yet been added to any region) are added to the current region (by labelling them with the same ID), but if and only if the neighbour cell can observe all cells that have already joined the region. Once no more neighbour cells can be added, the visibility map is used again to find the next seed for the next region. The whole process is repeated until no more seed cells are found.

In the related figures, it should be noted that the free space cells adjacent to the obstacles (i.e. the white borders surrounding the obstacles) do not belong to any region. The reason for skipping those cells in the segmentation process is the nature of the occupancy grid representations where the obstacles might be in any shape and size (in comparison to polygonal representations). This usually leads to the creation of a large number of unnecessary convex regions, even as small in size as one cell (especially near the walls). More details can be found in [3].

4 Distance Transform Based Gaussian Distributions

One final component of the framework remains. This concerns the disposition of victims in terms of their spread amongst the convex regions of the environment. At least two factors relate to this aspect. The first is the expectation that knowledge of the type of activities likely to have been taken place in the various parts of the environment (e.g. street market, domestic activities, office operations, recreation concentrations, etc.) will allow some estimation of starting points of locations of victims and in what concentration. However, one would expect a spreading out after the disaster, or perhaps during a disaster, due to panic and logic, but limited by obstacles, time and injuries so that some kind of spread function could be applied to the initial position/concentration estimates. In such situations, Gaussian distributions are favoured with standard deviations reflecting the spread extent. However the spread is impeded by obstacles which cannot be penetrated. It is ludicrous to imagine concentrations along the edges of obstacles to reflect truncation of Gaussian functions. What we have discovered is a non-linear distance transform (DT) [4] based value which, when substituted into the Gaussian exponent, nicely models spreads favouring directions of accessibility in a realistic manner. The DT propagates cell step distances away from specified goals throughout all of reachable free space, flowing around the obstacles. The steepest descent path from any reachable free space cell to the goal is optimal. The method extends to higher dimensional spaces and can accommodate space/time considerations. Details are to be found in equations (1),(2) representing the modified Gaussian distribution function which calculates the probability at each free-space cell (x, y) given its already calculated transformed distance d_{xy}. Normalising the calculated probabilities is a necessary step to adjust their values to accurately sum to 1.0 (dividing each cell’s value by the sum of all free space cell values).

$$G_{DT}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{d_{xy}{}^2}{2\sigma^2}} \tag{1}$$

$$G(x,y) = \frac{G_{DT}(x,y)}{\sum_{x,y=0,0}^{x,y=m,n} G_{DT}(x,y)} \tag{2}$$

5 Optimality Considerations

The search strategy aims to provide the order in which to visit all convex regions (via portals) so that the average time to finding (sighting in convex region) victims is minimal. This is a type of visiting salesman problem, except that a return to the start is not required and there are considerations other than minimal tour length to take into account.

Each convex region can be allocated a utility function which reflects the expected value of the number of victims to be sighted there. This is done by aggregating the probability spread functions which have non-negligible values over the convex region. Of course these DT based Gaussian spread values are weighted by concentration estimates at their sources. Thus a convex area at the intersection of many escape routes may accumulate a large utility value and a cul-de-sac not so large a value.

For every proposed convex region visiting order, the time to victim sightings for each region (path length so far) needs to be divided by the utility value to get a time average for a sighting in this region. The sum of each total path length divided by the utility value as each convex region is visited is itself summed over the whole circuit and then divided by the number of nodes to obtain the average sought. It is this value which should be minimised over all possible orderings.

6 Genetic Algorithm Approach

If the number of convex regions to be searched is large, the number of possible visit orderings is very large and an exhaustive search to find that ordering which minimises the average time to finding a victim is ill-advised. The simple travelling salesman problem only considers tour length. Here we must consider the utility values as well and the overall average is an average of averages. There are a number of ingenious ways of dealing with the travelling salesman problem using genetic algorithms, which can yield good but not necessarily optimal results. Ordering problems are usually dealt with in a special way. However, even a GA algorithm can be cumbersome, so we have developed a different method as a preliminary attack on this problem. Only the best result so far found is kept as the search continues.

7 Alternative TSP Search

To obtain very good search order, we have developed a new solution for the travelling salesman problem which gives a very close to optimal tour. The processing time is suitable for our application as it took ~2 seconds for 100 nodes. The processing time increases exponentially (1000 nodes may take a few days).

The new method can be described as follows. Firstly, a list is created where the first stored node is the robot's start location (the region's centre of area where it is standing is ignored as a node). Each next node to be added to the list should be the closest (in terms of time to victims sighting) to the previously added node. This initial sort of node is not necessary for optimality but it would accelerate the processing time for the next steps.

Secondly, each possible continuous sequence of nodes (2 or more, not including the robot node) is flipped in its position. The flipping instance that gives the best tour found so far is stored. The flipping process is repeated until no better tour is found.

Thirdly, each possible continuous sequence of nodes is shifted to all positions that follow its location in the list. Similar to the flipping process, the shifted instance that gives the best tour is stored. The shifting process is repeated until no better tour is found.

The second and third step are repeated one after the other until no better tour can be obtained. The method has been tested using instances taken from the World TSP on the web. Three instances have been tested: 29, 89 and 194 nodes. The method found the optimal tour for the first two instances. A tour within 4% to optimal has been found for the third instance.

8 Devastation-Changed Environments

If the disaster is an earthquake or bomb attack, some of the structural elements of what would have been regarded as obstacles in the 'a priori' map will have changed. These changes (unless they can be monitored as a whole from the air so that the changed environment now becomes the 'given' environment for planning the search at a strategic level) will be discovered incrementally as the search proceeds, region by region.

Unless some vital specific piece of new knowledge suggests a change of strategy, the search can continue at the global level as planned, but each original convex region is explored before moving on. A simple exploration procedure, using distance transforms, is to treat all parts (obstacles and free space) in each original convex region as goals and approach the closest such cell, filling in the map with detail as the searcher views new parts not originally visible. This 'greedy' strategy works quite well and is easily implemented [5].

Thus, the only essential change of search plan is to discover the new details of each region as the search continues. It may also be the case that new 'portals' between regions will need to be discovered and used when old ones are closed.

9 Results

In figures 1 to 6, three cases are presented where each case is described by two maps. The first map is the victims' distribution probability using DT based Gaussian distribution function. The robot start location is represented by 'R'. The victims' locations are represented by 'people'. Obstacles are in the dotted areas. A cell's brightness represents the sum of the Gaussian values at that cell for each victim location (a victim location represent a fixed number of people).

The second map is the visibility regions map. The different colours represent the generated convex hull regions. The white cells surrounding the (black) obstacles are free spaces that do not belong to any region (to reduce the number of generated regions). Victims are represented by red circles here. Each region has two values in its centre: The lower value is the percentage of the region's utility value to the total utility value of all regions; the upper value is the generated order of visit index.

So far, the search path that would start from the robot location would visit each region's centre in the provided order. A more efficient path (the shown red line) is generated here by taking into account two facts:

a. the robot doesn't need to reach a region's centre once it has reached its border (since all parts of that region become visible at that point);

b. the next region to visit is skipped if it has already been passed through earlier while visiting other regions.

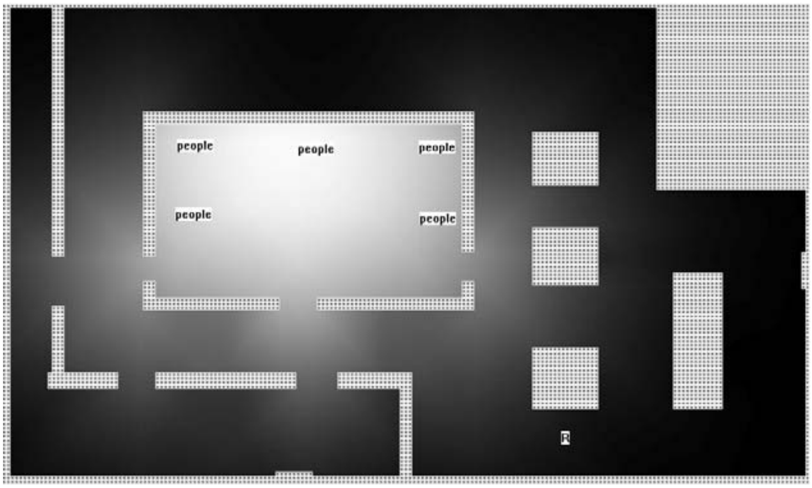


Figure 1: A generated Gaussian map for victims whose locations are represented by 'people'. Obstacles are the dotted areas. The calculated probabilities of the free-space cells are represented by their brightness. A cell's brightness is proportional to its value. σ is set to 40. The map resolution is 200x120.

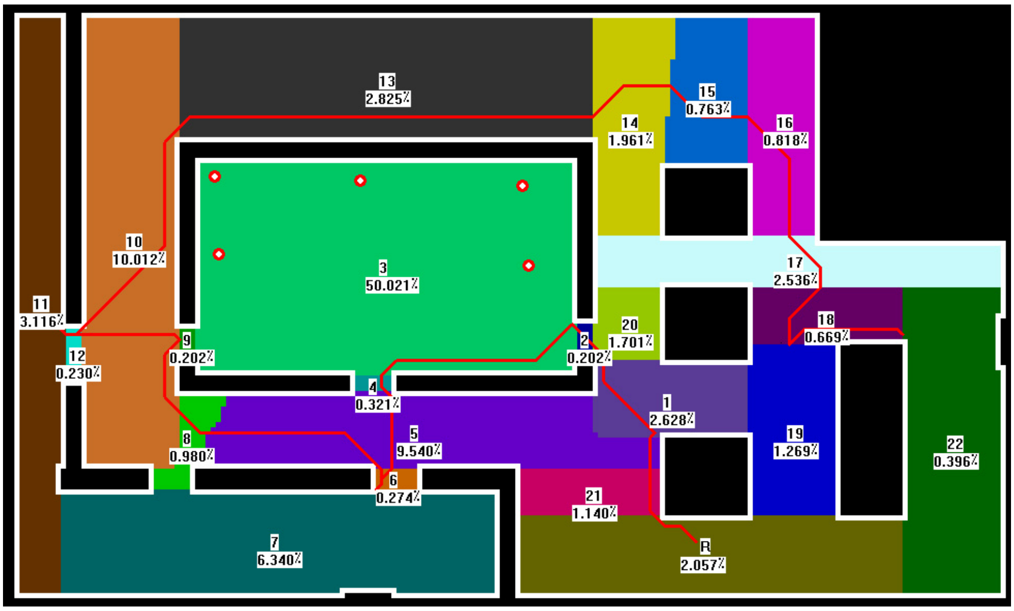


Figure 2: The visibility regions represented in different colours. At each centre, the lower value is the region's utility percentage while the upper value is the best order of visit. The generated path here (red line) is more optimal than a path that would visit each region's centre since it considers ignoring a region once its order is visited.

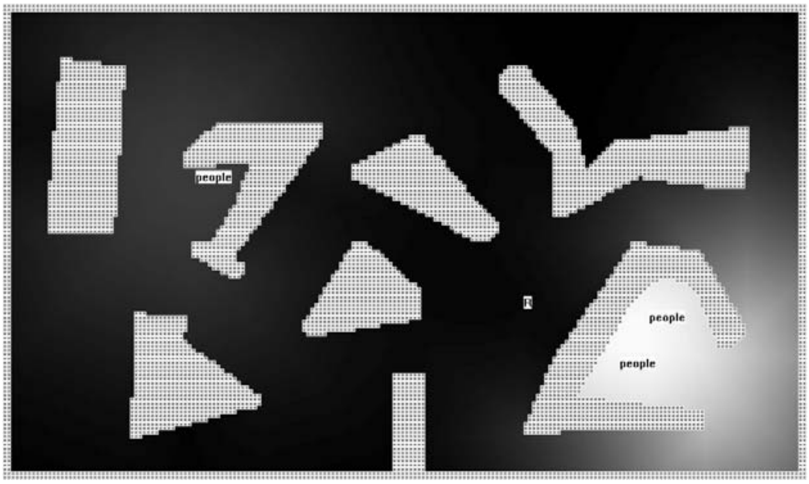


Figure 3

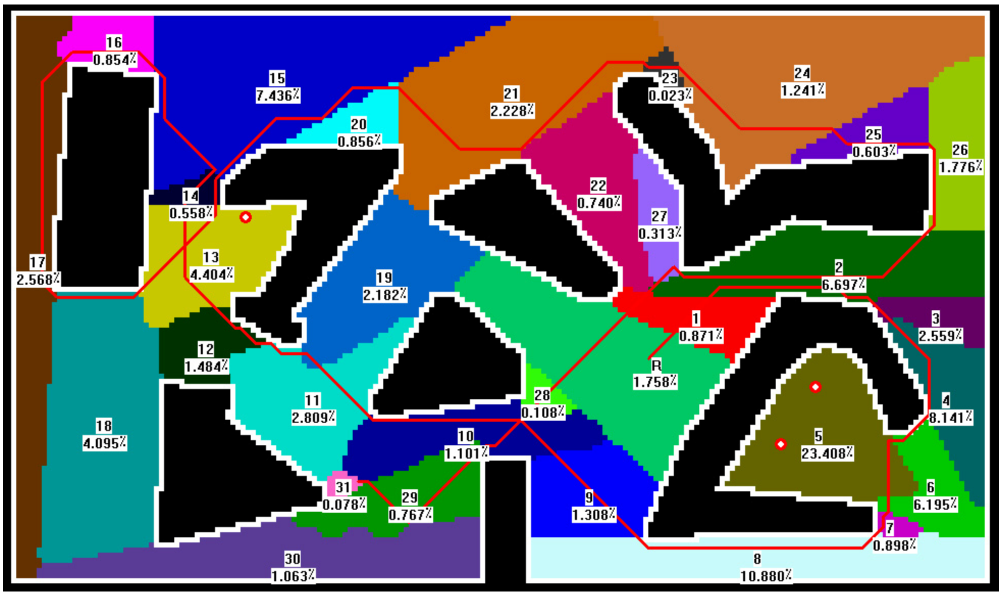


Figure 4

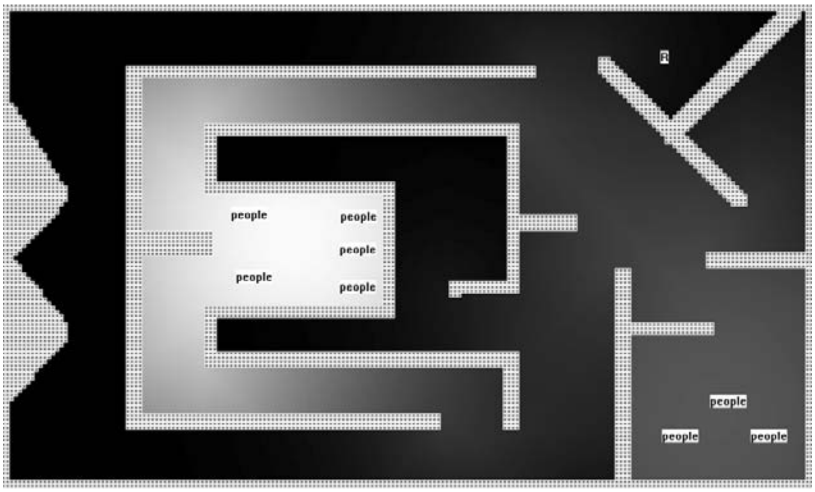


Figure 5

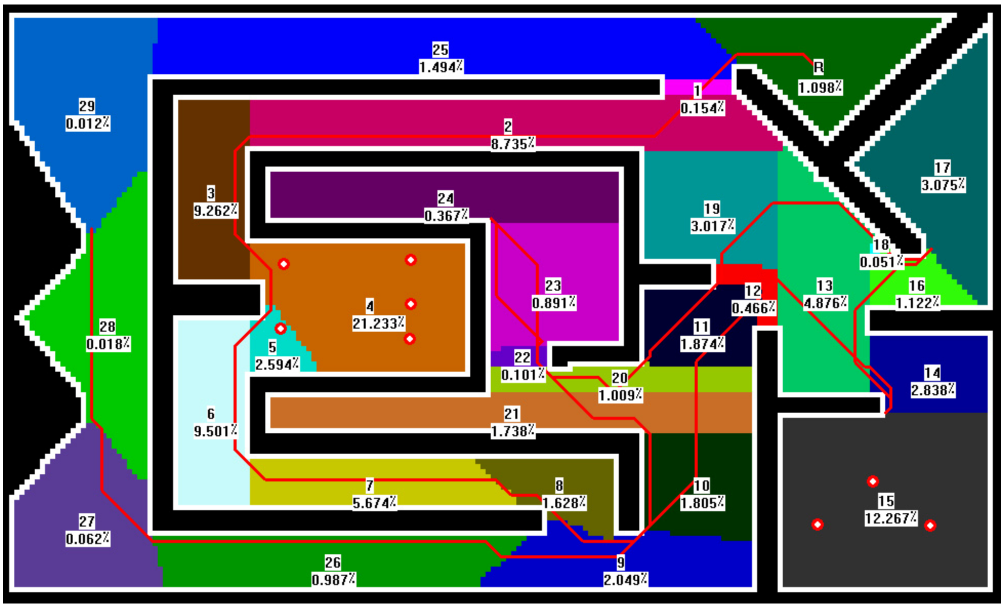


Figure 6

10 Discussion and Future Work

The spread of victims following a disaster clearly does not depend simply on spreading an estimated concentration of people in a structured way using a single σ value. What was happening before the disaster, the physical condition of victims immediately after the disaster and whether the disaster was progressive or global are all considerations worth including. However, many of these types of variations can be fairly easily included by considering the time of day, events which may be starting, finishing or in progress, the type of disaster (e.g. fire, earthquake, nuclear attack, flood) and the extent of the damage to existing structures. What has been presented here, despite the simplifying assumptions, is a good basis for further refining developments to determine an efficient search strategy. Future research could provide a more sophisticated modelling process that could better provide estimates of the numbers of people at various nominal points and the appropriate σ values to apply. The distance transform based Gaussian remains a very useful tool to accommodate obstacles in spread functions. Better TSP solutions might also be attempted. The deployment of multiple robotic agents to work cooperatively in the rescue operation is also of interest.

11 Conclusion

This paper has presented a means by which victims might be found after a disaster by a single robotic agent such that the average time to sight a victim is minimised. Many simplifying assumptions have been made but the notion of using a distance transform based Gaussian spread function, novel TSP based strategy of search order and the construction of 'visibility regions' is a sound basis for further development.

12 References

[1] J.H. Munson, *Efficient Calculations on Points and Lines in the Euclidean Plane*, SRI AI Group Technical Note No. 61, (1971).

[2] R.A. Jarvis, "Collision-free trajectory planning using distance transforms", *Mechanical Engineering Transactions*, ME10(3), pp 187-191 (1995).

[3] M. Marzouqi and J. Jarvis, "Efficient robotic pursuit of a moving target in a known environment using a novel convex region segmentation", *Proceedings 2nd International Conference on Autonomous Robots and Agents*, Palmerston North, New Zealand, pp 13-18 (2004).

[4] M. Marzouqi and R. Jarvis, "The dark path algorithm for covert and overt path planning with uncertainties involved," to be published in *Proceedings IEEE TENCON*, Melbourne (2005).

[5] R.A. Jarvis, "Distance transform based collision-free path planning for robot navigation in known, unknown and time-varying environments", in Y.F. Zheng (ed.), *Recent Trends in Mobile Robots*, World Scientific Publishing, pp 3-31 (1994).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Mixture Strategy of Mowing Path Plan for Small Mowing Robots

Tomonori Sato, Keiji Suzuki,
Future University Hakodate, Japan
g2105011@fun.ac.jp, suzukj@fun.ac.jp

Abstract

In this paper, the robustness and the efficiency of the mowing strategy is proposed for the small mowing robots and the charge station. The mowing robots will be expected to work at open areas such as parks. To work in the open area, the robots must keep the safety of the people first. To keep the safety, small and low power type robot is selected. The characteristic causes stuck and unstable moving. To overcome the weakness, the cooperative multiple robots strategy is introduced. The robots communicate with others other using Ad-hoc network and SOAP communication. Based on the communication, the mowing area is managed as the divided grid areas. In the grid areas, the mowing path designed by the proposed mixture moving strategy. The approach strategies well works on the mowing task in the real robots experiment.

Keywords: multi robot, path planning, mowing robot, genetic algorithm, GPS localisation, SOAP communication

1 Introduction

The developed small mowing robots, called Robomowers, will be expected to do mowing work at open areas such as parks. To work in the open area, the primary concern would be the safety of people. For keeping the safety, small and low power type robot is selected. The characteristics cause stuck and unstable moving. To overcome the weakness, the cooperative multiple robots strategy is introduced. The Robomowers are GPS based small mowing robots. It has many sensors: Differential GPS, electric compass, ultrasonic wave sensors, tactile bumper switches, voltage sensor and etc. Robomowers communicate with each other and the charge station by ad-hoc network and SOAP communication.

The mowing work of the small robots has 4 problems. The robots get stuck in rough terrain and are hard to keep running directly to their destination because their motors are low powered. The wireless communication is losable because robots always move. The robots need to return to the charge station to charge up, if task place is a large range. Our approach to these problems is to use many mowing robots, the robots share mowing task cooperatively based on message exchanging. The message consists of each robot coordinate, target coordinate and all working area's state. The robots should communicate with other robots to convey the mowed grids so repeated mowing of the same grids will not occur.

The mowing strategy is proposed as the mixture strategy that consists of the path trace strategy and the nearest target strategy. Therefore, to cover the weakness of the running ability, the path trace strategy is applied first. Then, the nearest target strategy is applied to the rest of the working unit areas.

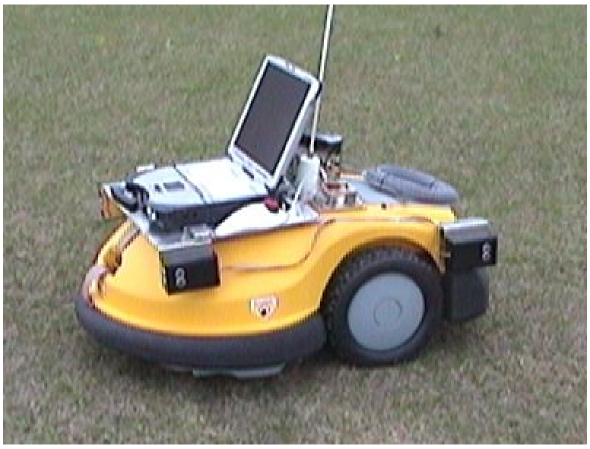


Figure 1: Mowing robot “Robomower”.

The target is determined by self tracing and communication with other robots. The most nearest untraced unit working area is selected as the target.

The previous research [1] showed the mowing system using mixture strategy had resistance to the communication instability. In this paper, the GA making path function was added to the mowing system for the work efficiency improvement.

2 Mowing Robot

The Robomower is GPS based small mowing robot. The body and chassis are based on the Robomower RL500 that made by Friendly Robotics Inc. It was customised by researching [2]. Figure 1 shows the Robomower body. The size is 490x730x620(mm), weight is 40.1(kg).

2.1 Equipment

Figure 2 shows the mowing system overview and robot equipment. It has many sensors: ultrasonic wave sensors, tactile bumper switches, voltage sensor, acceleration sensor, angular speed sensor and beacon receiver. The control board drives for motors and collects sensor data. The control board communicates with the notebook computer equipped on the front body using serial communication interface. The notebook computer controls the robot autonomously and communicates with other robots using wireless LAN. The Robomower has 5 motors: 2 traction motors and 3 mowing blade motors. The positioning data are received from Differential GPS with NMEA style. In the NMEA sentences GGA, GLL, RMC sentences are used. The latitude and the longitude are converted to Plane Rectangular Coordinate System 2001.

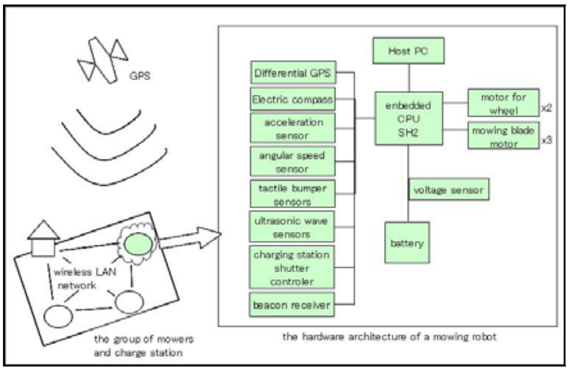
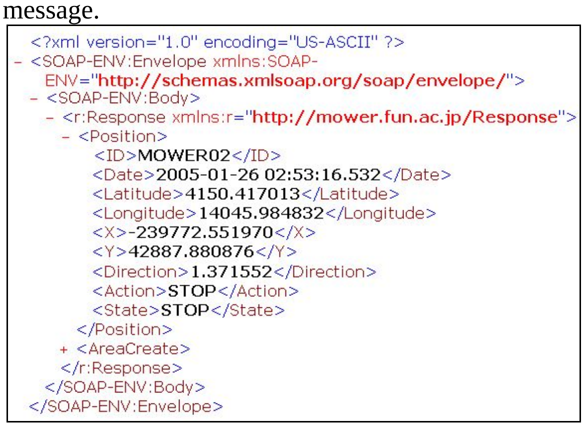


Figure 2: Mowing system and robot hardware.

2.2 Communication Environment

The Robomowers communication uses wireless network. As the communication protocol, HTTP and SOAP are used. The SOAP message handles the XML data. Figure 3 shows example the SOAP message.



```
  <?xml version="1.0" encoding="US-ASCII" ?>
- <SOAP-ENV:Envelope xmlns:SOAP-
    ENV="http://schemas.xmlsoap.org/soap/envelope/">
  - <SOAP-ENV:Body>
    - <r:Response xmlns:r="http://mower.fun.ac.jp/Response">
      - <Position>
          <ID>MOWER02</ID>
          <Date>2005-01-26 02:53:16.532</Date>
          <Latitude>4150.417013</Latitude>
          <Longitude>14045.984832</Longitude>
          <X>-239772.551970</X>
          <Y>42887.880876</Y>
          <Direction>1.371552</Direction>
          <Action>STOP</Action>
          <State>STOP</State>
        </Position>
      + <AreaCreate>
      </r:Response>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Figure 3: SOAP message example.

OpenSOAP [4] which is open source implementation for SOAP communication is used. The OpenSOAP can make the SOAP message easily. In addition, the SOAP message can analyse things such as Domain Object Model. The SOAP can pass the fire wall. Thus changing the message contents and adding the other robot to the Robomowers team can be done easily. But the communication is losable because this communication is wireless and the robots are always moving. The robots should communicate with other robots to convey the mowed grids so repeated mowing of the same grids will not occur.

3 Charging Station

The Charging station was developed by the previous research [3]. It consists of the garage, the tow charging machines and the charging control computer. The charging control computer operates the garage's shutter and the charging machines. When the garage's shutter is opened, the charging machine sends out the beacon. The beacon navigates the Robomower. After the Robomower is stocked the charging machine, the control computer starts the charging.

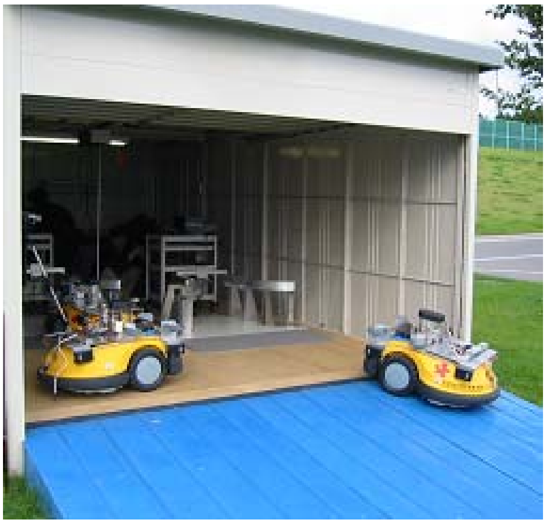


Figure 4: Mowing robots and charge station.

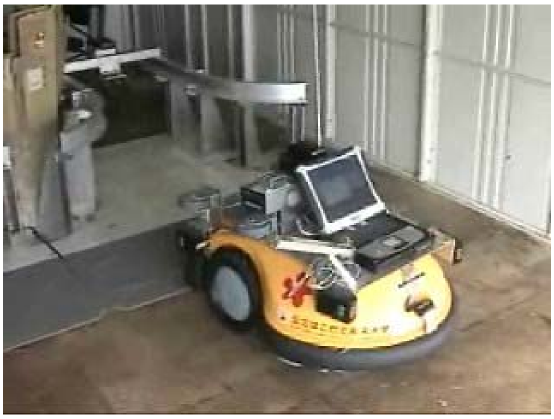


Figure 5: Returning to the charge machine.

Figure 4 shows the charging station and the Robomowers. Figure 5 shows the Robomower returning to the Charge Station. The mowing robots always consume the electrical power at a proportional rate to the motor power and the basic usage. Thus, when the mowing robots electrical power is low, their must return to the charging station. The charge station's capacity is 2 robots.

4 Strategy

4.1 Basic Running Strategy

The target position is driven by the path plan. The robot position and the attitude follow the Differential GPS and the electric compass. The positioning data are received from Differential GPS using NMEA format. In the NMEA sentences, GGA, GLL, RMC sentences are used. The latitude and the longitude are converted to the plane rectangular coordinate system 2001.

Figure 6 shows the basic control scheme of the Robomower. Figure 7 shows target decision rule. The Robomower decides running motors speed from the target position and the robot state, using PID control. The stopping condition of the robot consists two. That is, the Robomower hits an obstacle. The motors for the moving and the mowing become overload. Under the condition, when the Robomower hit something, the robot stops and goes back about 1 metre. Then it turns right or left alternately. When it gets stuck, its motor powers increase.

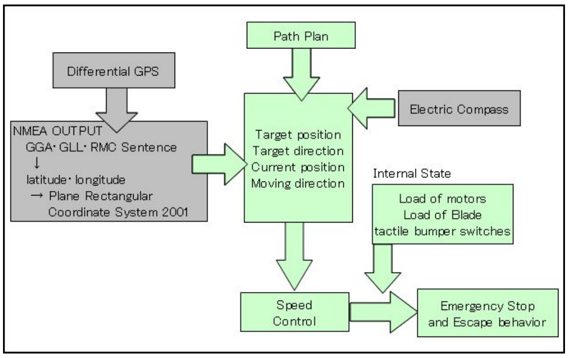


Figure 6: Basic control scheme of the Robomower.

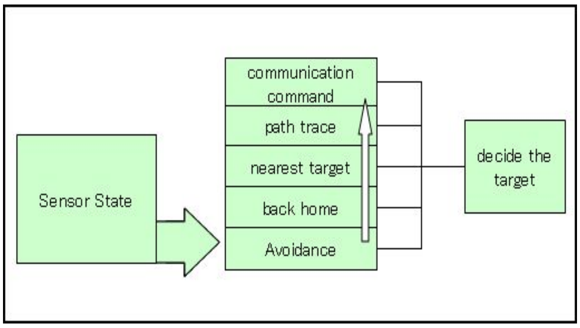


Figure 7: Decision scheme of the Robomower.

4.2 Charging and Return Home Strategy

When the electric power is low, the Robomower sets target place to the charging station. As the Robomower approach the charging station, the Robomower send shutter open command to the charging station. Then the Robomower waits for the beacon from the charging machine. The beacon navigates the Robomower. Finally the Robomower is connected to the charging machine, the station starts charging.

4.3 Mowing Strategy

The mowing work of the small robots has 4 problems. The robots get stuck by rough terrain and are hard to keep running directly to their destination because their motors are low powered. The wireless communication is losable because the robots always moving. The robots need to return to charge station to charge up, if the task place is a large range. Our approach to these problems is to use many mowing robots. The robots share mowing task cooperatively based on the message exchanging. The message consists of each robot coordinate, the target coordinate and all working area's state. The robots should communicate with other robots to convey the mowed grids so repeated mowing of the same grids will not occur.

The mowing strategy is selected as the mixture strategy that consists of the path trace strategy and the nearest target strategy. Namely, to cover the weakness of the running ability, the path trace strategy is applied first. Then, the nearest target strategy is applied to the rest of the working unit areas. The target is determined by self tracing and the communication with other robots. The most nearest untraced unit working area is selected as the target.

4.4 Pathmaking Strategy

There are many size and shapes of lawns mowing areas. It is difficult and bothersome to make path plan for these situations. It will need the automatic pathmaking function. The function uses the genetic algorithm. The genetic algorithm is used in order to make a better path plan.

The genetic algorithm, two point crossover, the bit inversion mutation and the elite strategy selection are used. The evaluation function consists of the covering rate against the mowing area, the total path length and the total turn angles. The best path plan may be that the covering rate is high, the total path length and the total turn angle is low. The equation (1) is the evaluation function. The individual gene is bit strength. The bit strength has path line segments and the segment quantity. The upper 5 bits are declared as segment quantity. The following bits are declared path line segments. One path line segment consists of the angle (5 bits) and the length (5 bits). Use the bit strength to make the path plan. The path starting point is the place where the mowing robot is now. The path plan defines the new point that subject to the path line segment and the last point. The declaration points repeat until all the segments are read. The Robomower must take short time for pathmaking. Therefore, it has pathmaking terms. The segment quantity is fixed 8, the elite individual quantity is 1 and the individual quantity decrease less than 10 and goes to the next generation. This continues until 300 generations.

$$V = \text{coveringRate} + 1/\text{length} + 10/\text{angle} \qquad (1)$$

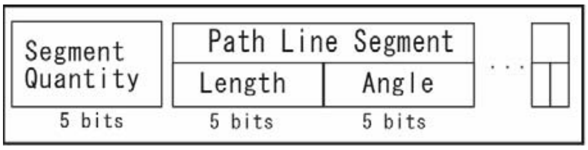


Figure 8: Gene structure for pathmaking.

5 Experiments

The efficiency of robots using the automatic pathmaking approach that uses the GA compared to that of non-use of the automatic pathmaking approach ware conducted.

The real robots experimented on 10 square-metre turf area by two robots using the mixtures strategy. The charge station is in a place 12 metres away from the centre of the working area. The starting positions of the robots are in front of charging station. When the experiment started, the robots did not move until the pathmaking function determined the path, but the experiment without this function, the robots moved immediately. Figure 9 shows the robots working on the area.

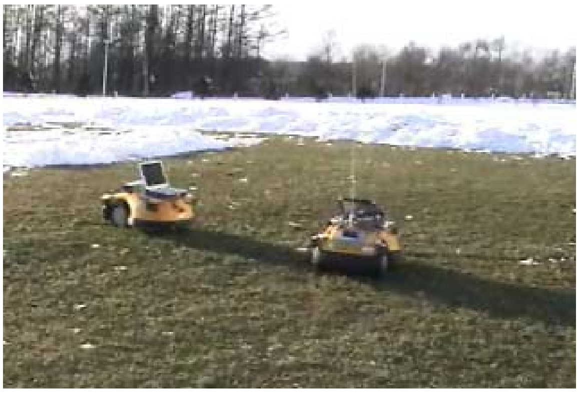


Figure 9: Working mowing robots.

Table 1 shows experimental result and compare 2 strategies. The Robomowers with the pathmaking function finished the path trace strategy at 774 seconds. The performance factor is 37.5%.

Table 1: Experimental results.

	with pathmaking function	without pathmaking function
Path trace time	774 sec	x
Performance factor (until 774sec)	37.5%	35.0%
Work finality time	2710 sec	2806 sec

Figure 10 shows generated a path plan by the function. Figure 11 shows the Robomowers using the pathmaking function orbit (774 seconds). Using the Robomowers without the pathmaking function, the performance factor is 35% utilising the same time. Figure 12 shows the Robomowers without the pathmaking function orbit utilising the same time.

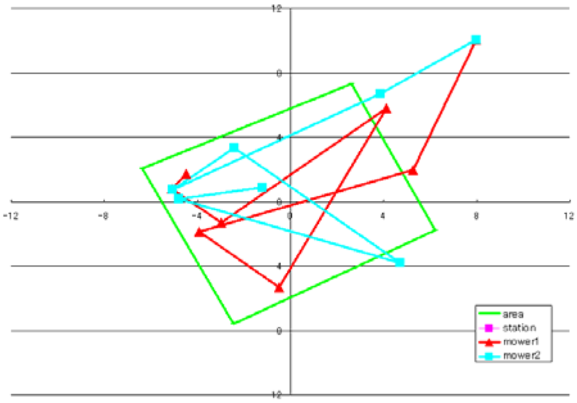


Figure 10: Generated path plan.

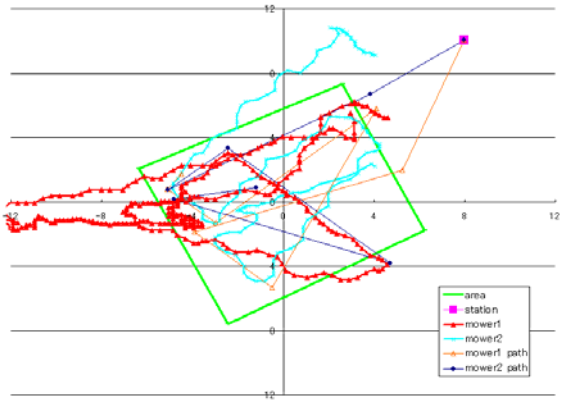


Figure 11: Robomowers with pathmaking orbit utilising 774 sec.

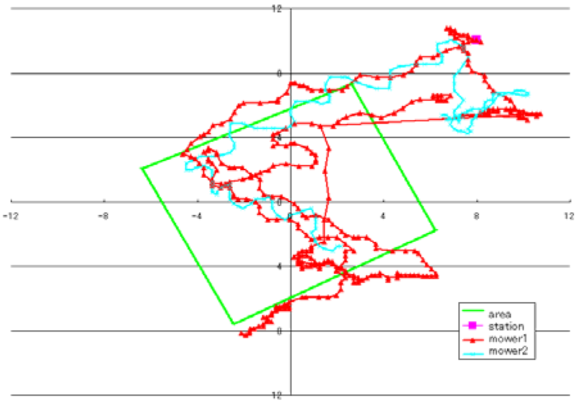


Figure 12: Robomowers without pathmaking orbit utilising 774 sec.

The Robomowers with the pathmaking function finished all work at 2710 seconds, the ones without the pathmaking function one finished all work at 2806 seconds. Figures 13 and 14 show the finished orbit of each Robomower's strategy.

From these result, the pathmaking function contributes slightly to working efficiency. It can be confirmed that implementing strategy, the robots show the robustness and the efficiency under the unstable communication circumstance.

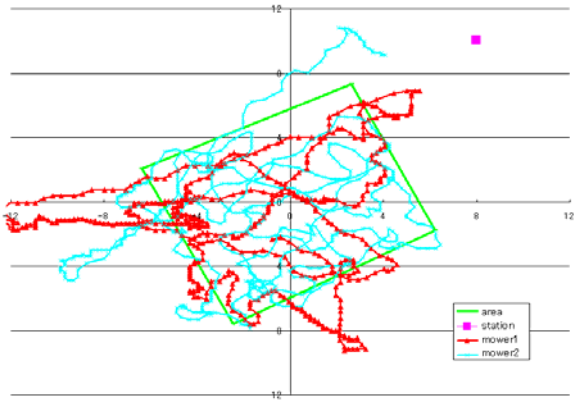


Figure 13: Robomowers with pathmaking orbit finished work.

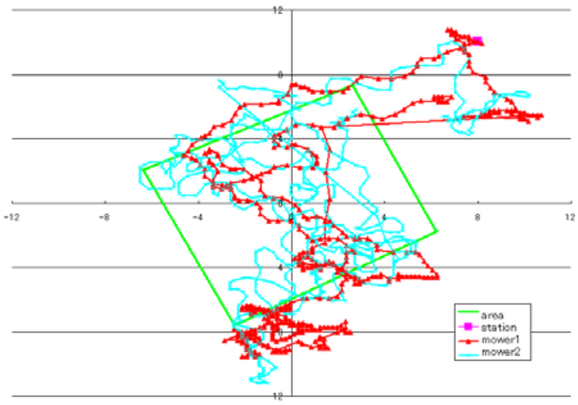


Figure 14: Robomowers without pathmaking orbit finished work.

6 Conclusion

GPS based small mowing robots work that using SOAP message communication, the automatic pathmaking function, the path trace strategy and the nearest target strategy becomes the robustness and the efficiency system. It can be confirmed that implementing strategy, the robots robustness and efficiency will work even in unstable communication circumstance.

7 References

[1] T. Sato and K. Suzuki, “Control of the small mowing robots using SOAP”, *Proceedings ISR2005*, Tokyo (2005)

[2] K. Suzuki, S. Mikami, J. Akita and E. Osawa, “Development of cooperative small mowing robots”, *Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Kobe, Japan, 3, pp 1498-1502 (2003)

[3] K. Suzuki, S. Mikami and E. Osawa, “Development of cooperative small mowing robots with charging station”, *Proceedings 4th International Conference on Advanced Mechatronics (ICAM04)*, Asahikawa, Japan, pp 617-622 (2004)

[4] SOAP Specifications, “Home Page”, http://www.w3.org/TR/soap/, visited on 8/3/2005.

[5] OpenSOAP Project, “Home Page”, http://opensoap.jp/, visited on 8/3/2005.

[6] Future University Hakodate, *OpenSOAP for Web Services: Basic Technical Training*, Future University Hakodate, Hakodate (in Japanese) (2005).

[7] L. Davis, Y. Kakazu, S. Mikami, M. MasaAki, K. Takashi, T. Norihiko, K. Suzuki, *Genetic Algorithm Handbook*, Morikita Publications, Tokyo (in Japanese) (1994).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

A New Control Method for Robot Gripper Using Vision Based Tactile Sensor

Goro Obinata and Norinao Watanabe
Graduate School of Engineering,
Nagoya University, Nagoya, Japan
obinata@mech.nagoya-u.ac.jp

Abstract

A new type tactile sensor has been proposed recently. One of the good features is the function to estimate friction coefficient. Based on the function, we can achieve handling the object without slipping out. In this paper, we propose a control method with an actuator for adjusting the grip force. The experimental result shows that the proposed control method is effective for preventing from the slippage while handling the object.

Keywords: tactile sensor, robot gripper, dextrous handling, vision-based sensing, control system design

1 Introduction

Humans have the ability to sense the weight and friction coefficient of the grasped object with their distributed tactile receptors. The ability makes it possible for humans to prevent from the slippage of manipulated object or collapsing the object. Such dexterous handlings are achieved by feeding back the signals from the receptors to muscle control system through neural networks. Therefore, it may be a key point for establishing dexterous handlings of robots when we try to mimic skilled human functions.

For tactile sensing of robots, several methods and sensors have been proposed by using electrical resistance, capacitance, electromagnetic component, piezoelectric component, ultrasonic component, optical component, and strain gauge [1], [2]. There exist many problems of these sensors to be solved for establishing practical ones. For an example, the sensor which consists of elastic body and strain gauges requires a lot of gauges and the wiring. Moreover, the signal processing is not simple to obtain the values of the contact forces and the friction coefficients [3]. On the other hand, optical sensors have been introduced because wiring is not required in the contact part to the object [4], [5], [6]. The introduction of optical sensor makes the size small and the wiring simple. However, the sensing of friction coefficient is not considered in those papers.

It is required for establishing dexterous handlings of robots to provide a purpose-built sensor for the measurement of friction coefficients between robot hand and the target surfaces. So as to avoid multiple usage of tactile sensors, we have proposed a new design of tactile sensors for multiple measuring of contact information including friction coefficient [7].

In this paper, we propose a control method for robot gripper using the tactile sensor. The purpose of control is to achieve dexterous handling. The main feature of the control system is to hold the object without slipping out against several disturbances. First, the principle of tactile sensing is described. Then, a new method for controlling the grip force is proposed. The experimental result is given to illustrate the effectiveness of proposed method.

2 Vision Based Tactile Sensor

2.1 Structure

We have proposed a vision-based sensor for multiple measuring of contact information including friction coefficient. The system consists of a CCD camera, LED lights, acrylic plate and elastic body, as illustrated in figure 1. The elastic body, which is made of transparent silicon rubber and has grid pattern or dotted pattern on the spherical surface as shown in figure 2, is to contact the object. The CCD camera is to take pictures of the spherical surface from the flat surface side of the elastic body. The experimental setup is shown in figure 3. We can apply not only arbitrary normal and tangential forces but also moments around normal axis of contact with the sliding mechanisms.

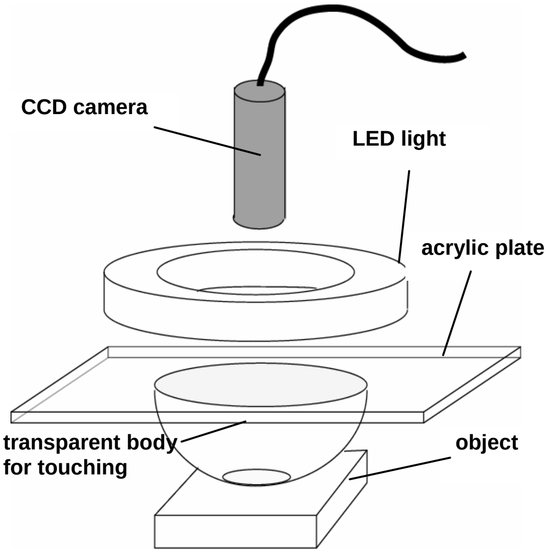


Figure 1: Structure of vision-based tactile sensor.

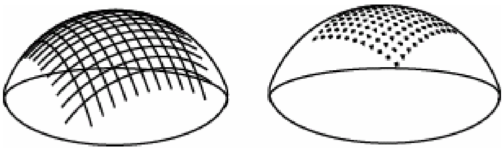


Figure 2: Examples of shape and pattern on the surface of the elastic body.

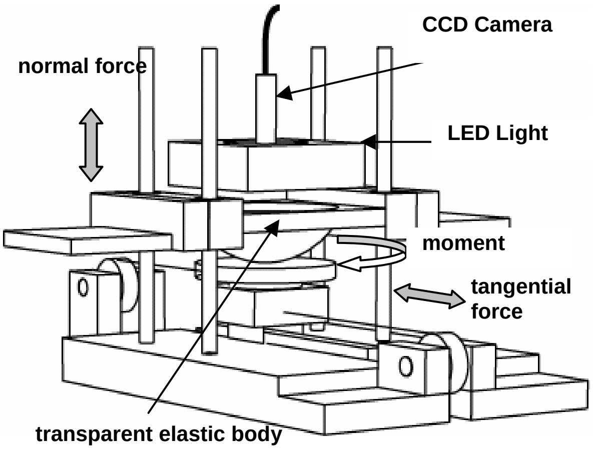


Figure 3: Experimental setup for the sensor.

2.2 Measurement of Contact Force and Moment

2.2.1 Normal Force

The relation between contact radius and normal contact force has been analysed in cases of hemispherical surface [8]. The result provides the following relation:

$$a = cN^{\gamma} \qquad (1)$$

where a is radius of contact area, c is a constant depending upon the material, γ is a constant depending upon the shape, N is normal force. This means that the normal force can be obtained from the contact area once the values of c and γ are determined. The picture of figure 4 is an example when only a normal force is applied. We can estimate the contact area from the difference of brightness of each pixel; that is, we can obtain the estimation of the normal force. The dotted circle in the picture shows the estimated contact area with a certain threshold of the brightness. The experimental results are summarised as shown in figure 5. The obtained values agree with the relation (1). The solid line in figure 5 shows the curve with c = 4.35 and γ = 0.17. We can estimate normal contact forces based on this relation using the proposed sensor.

2.2.2 Tangential Force

The picture in Figure 6 shows an example when normal and tangential forces are simultaneously applied. Central four dots on the surface of the elastic body were coloured with red, and are used as the origin and axes of the coordinate while identifying the displacements of all dots from the pre-contact positions. When the four dots are included in the contact area, the displacements of the dots allows us to determine the direction of applied tangential force.

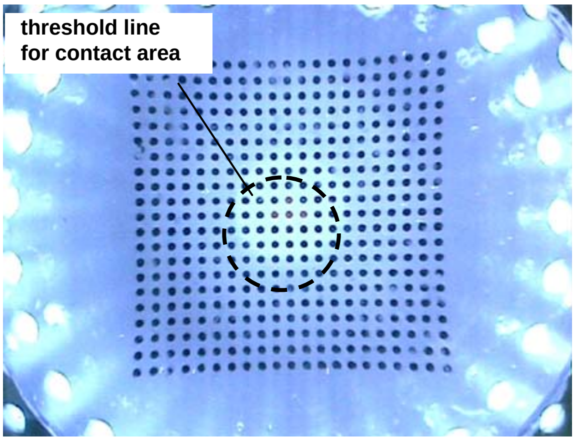


Figure 4: Example of the picture (only a normal force is applied).

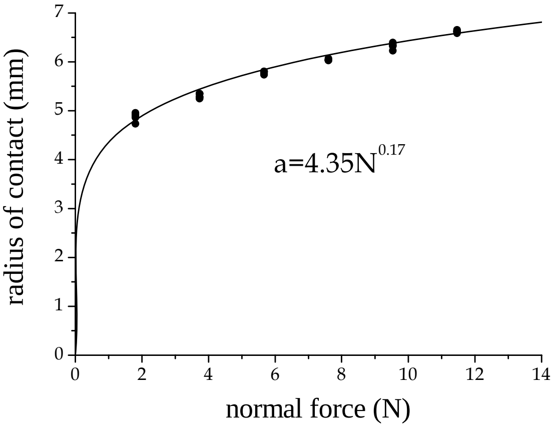


Figure 5: Relation of radius of contact area to the applied normal force.

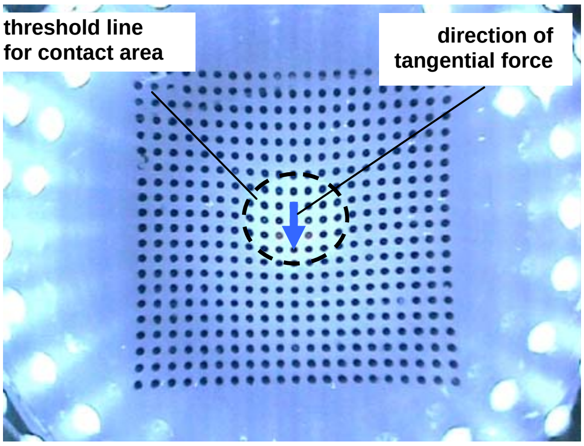


Figure 6: Example of the picture (both normal and tangential forces are simultaneously applied).

We recognised that the displacements depend on the applied normal forces. On the other hand, the contact radius is independent on the tangential forces; thus, we estimate at first the normal force from the contact radius, and then estimate the tangential force with the estimated normal force. We found out the method for eliminating the dependency of tangential force on the normal force by cut and try.

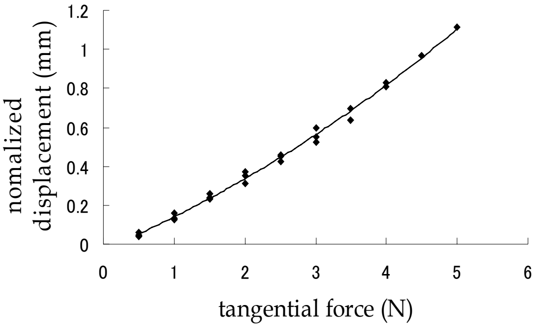


Figure 7: Relation between normalised displacement and tangential force.

Multiplying the ratio of contact radius to the measured displacements in tangential direction yields the normalised displacements. Then the relation between the normalised displacements and the tangential forces becomes one to one correspondence which is nearly expressed by one curve. The result of the normalisation is summarised in figure 7. Based on the relation, we can estimate the magnitudes of the applied tangential forces.

2.2.3 Moment

We can identify all dots on the surface of sensor in the coordinate frame defined by the coloured dots. This fact allows us to obtain the vectors corresponding to the all dots which start from the positions of pre-contact phase and end at the positions of post-contact phase. We can estimate the centre of rotation from the vectors when a moment is applied. Then, we can calculate the rotation angle from the position of the centre and the vectors by using appropriate technique, for an example, least square method.

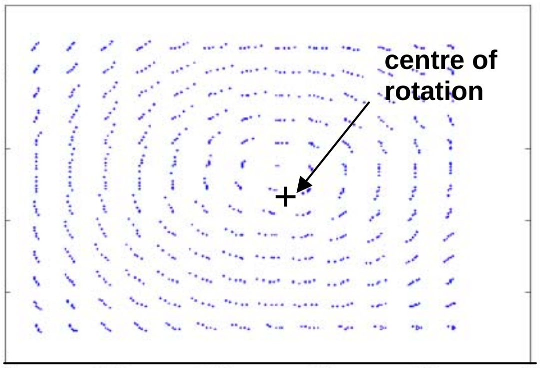


Figure 8: Vectors for dots and centre of rotarion.

Figure 8 shows how to estimate the centre of rotation, for the example. We recognised that the estimated angles depend on the applied normal forces. We used the same method as the case of tangential force to eliminate the dependency; that is, the obtained angles were normalised by multiplying the ratio of contact radius. However, the result has a relatively large deviation and is summarised in figure 9.

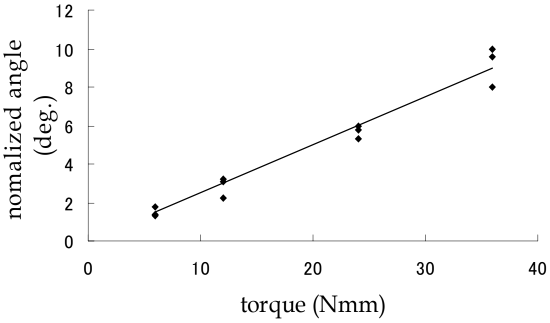


Figure 9: Relation between normalised rotation angle and applied moment.

3 Estimation of Friction Coefficient

So as to prevent from the slippage of the manipulated object, we need to obtain the conditions of contact surface between the gripper and the object. The coefficient of static friction is important for handling the object without slipping. When the contact occurs between curved surface and flat surface, the pressure between the two surfaces distributes in the contact area. If the pressure of contact surface takes a lower value than the constant which is determined by both the surface conditions and the materials, the relative motion in tangential direction is possible in the area. This leads that the pressure distribution between the gripper and the object divides the contact area into two parts in general. In one part of contact surface, the relative motion in tangential direction is possible.

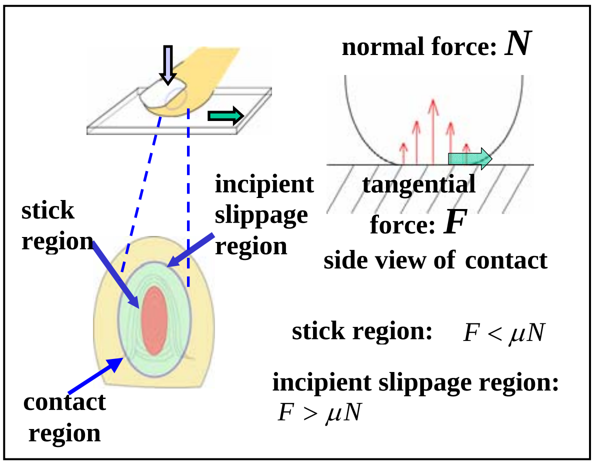


Figure 10: Schematic view of finger contact and definition of incipient slippage region.

We call the part as incipient slippage region. In the other part, the relative motion is impossible. This always occurs when human contact the object with fingertips. Several receptors in cutaneous sensory system of human distinguish these two areas. This means that human can sense the degree of slippage without occurring macroscopically slippage. The schematic view of finger contact and the definition of the two regions are illustrated in figure 10. If we distinguish the two parts from the picture of CCD camera with our sensor, we can estimate the degree of slippage from the ratio of area of the incipient region to the total contact area. The ratio is defined by

$$\rho = \frac{S_s}{S_c} = \frac{r_s^2}{r_c^2} \tag{2}$$

where S_S is stick area, S_C is contact area, r_s is radius of stick area, and r_c is radius of contact area. We call this ratio as stick ratio, and it relates to the friction coefficient. In the case of spherical and flat surfaces, the incipient slippage occurs in peripheral part of contact. So as to confirm the phenomenon experimentally with spherical and flat surfaces, we add a displacement sensor and a force sensor to the experimental setup of figure 3. The force sensor measures directly the applied tangential force. The displacement sensor measures the movement of object. First, we identified the positions of all dots when only a normal force was applied. Next, we applied small additional force in tangential direction and increased the magnitude gradually. The dots in stick region moved in the direction, and the displacements were almost equivalent to that of the object. Note that macroscopic slippage did not occur at this moment while the surface in stick region moved with the object. On the other hand, the dots in incipient slippage region moved a shorter distance in tangential direction because slippage occurred in the region. The relation between the initial distances of dots and the displacements for three cases of different tangential forces is shown in figure 11. It should be noted that the radius of stick region decreased as the applied tangential force increased. When the radius reaches to zero, macroscopic slippage will occur. This leads to the possibility of estimating the degree of slippage from the displacements of central and peripheral dots. We propose a method for estimating stick ratio only from measurements of the sensor. The method uses the relative displacements of peripheral dots to the central dot. The radius of stick region can be determined by comparing the relative displacements with the threshold. In order to show the effectiveness of the proposed method, we carried out another experiments under different friction coefficients. We controlled the friction coefficient by using talcum powder on the contact surface and obtained the relation of the friction coefficients to the estimated stick ratios. The values of friction coefficient were determined with the ratio of the tangential force to the normal force at occurrence of the macroscopic slippage. We express the result with five lines in figure 12. Each line corresponds to each friction coefficient. The lower stick ratio with the same displacement of central dot means the lower friction coefficient. Although the proposed method cannot estimate smaller stick ratio under 0.1, we can use the estimated value to prevent from slippage because we can keep it over a certain value by feedback control of the contact force.

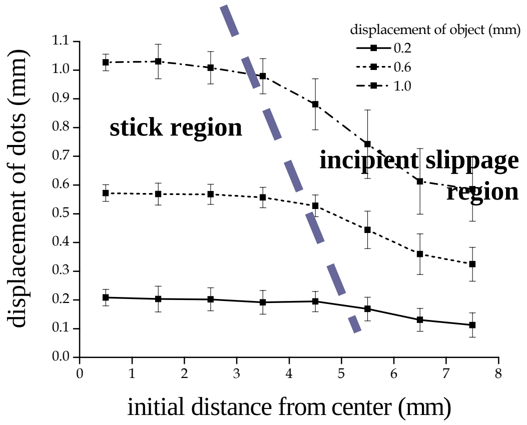


Figure 11: Identifying incipient slippage region.

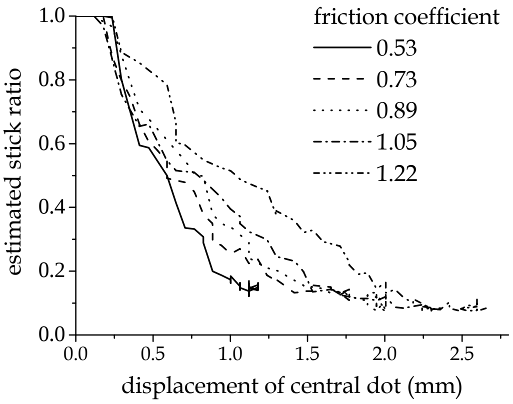


Figure 12: Estimated stick ratio in different friction coefficients.

4 Control System for Robot Grippers Using Tactile Sensor

In this section, we describe the control system design for robot grippers to prevent from the slippage of the manipulated object. The feedback signal from the proposed tactile sensor is used to control the grip strength for stable handling of the object. The control system consists of the tactile sensor with image processing software, a voice coil motor, and a simple proportional controller with gain K, which is shown as the block diagram in figure 13. The reference S_S^* is the set point for the stick area S_S. The controller amplifies the deviation $S_S^* - S_S$ by K, and transmits the calculated value to the voice coil motor. The voice coil motor generates the grip force under the control.

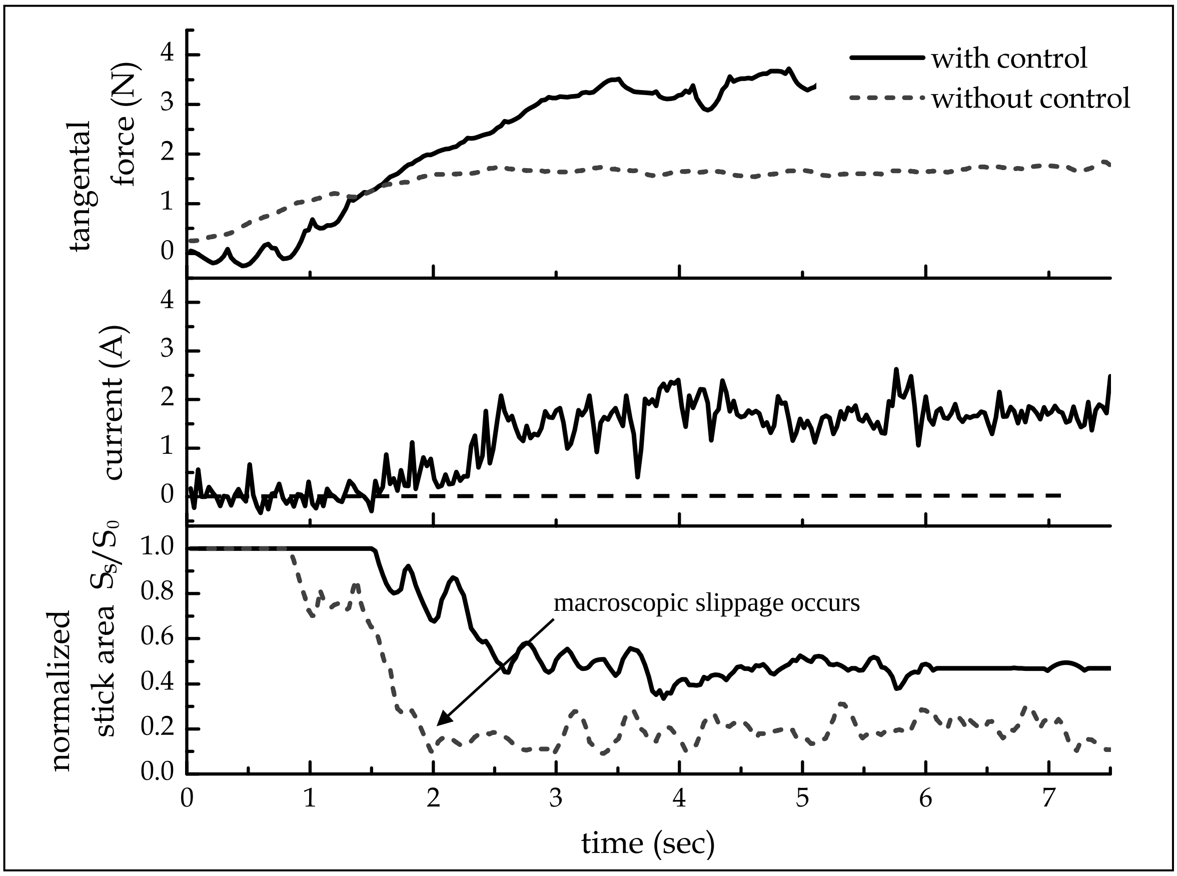


Figure 14: Result of stick area control: comparison between with and without control.

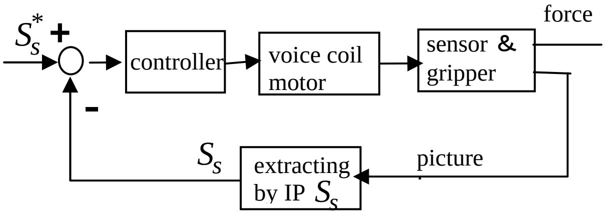


Figure 13: Block diagram of grip force control system.

The generated force is in proportion to the current in the voice coil motor. This feedback mechanism keeps the stick area around the set point. The experimental results of this control system are given by figure 14. The tangential forces were applied manually; so, the curves in figure 13 are not typical functions of time. The manipulated variables of control system correspond to curves in “current” of the second row which are in proportion to the generated normal forces. The current of one ampere is corresponding to the force of 3.5 N. The estimated stick ratios were normalised by the initial contact area and shows that a macroscopic slippage occurred while the control did not work. Moreover, it is shown in the figure that the control resulted in keeping the values of estimated stick ratio around the set point 0.5. This proves prevention from slippage by the control. It should be noted that the control system works only using signals from the tactile sensor.

5 Conclusion

We have explained on a new tactile sensor for measuring multi-dimensional force and moment of contact. The structure of sensor is simple because the method is vision-based. The method to obtain the values of force and moment has been confirmed by several experiments. It is shown that the accurate estimation of contact force and moment is possible with the proposed sensor although there is a trade-off between the resolution and the computational time. There exist small interpositions between the tangential force, the moment and the normal force while measuring. Clear understanding the interposition between applied forces and moments will be required in further research. We defined the stick ratio as an index for indicating the degree of slippage. We have also proposed a new method to estimate the stick ratio for preventing from slippage of manipulated object. The exact relation of the defined stick ratio or the estimated stick ratio to the friction coefficient is an important problem to be solved. We demonstrated the control system for keeping the estimated stick ratio around a set point. The purpose-built integrated circuit for the image processing of this sensor may be required to achieve high speed control against disturbances in high frequency band. It is shown that the proposed sensor has the potential for dexterous handling like human.

6 Reference

[1] H. Shinoda, "Contact sensing: a state of the art", *Trans. of Japanese Robotic Society*, 20(4), pp 385-388 (in Japanese) (2002).

[2] M.H. Lee and H.R. Nicholls, "Tactile sensing for mechatronics: a state of the art survey", *Mechatronics*, 9, pp 1-31 (1999).

[3] T. Maeno, K. Kobayashi, T. Kawai and Y. Hirano, "Grip force control by detecting the internal strain distribution inside the elastic finger having curved surface", *Trans. of Japanese Society of Mechanical Engineers*, 64(620), pp 1258-1265 (in Japanese) (1998).

[4] M. Ohka, Y. Mitsuya, Y. Matsunaga and S. Takeuchi, "Sensing characteristics of an optical three-axis tactile sensor under combined loading", *Robotica*, 22, pp 213-221 (2004).

[5] N.J. Ferrier and R.W. Brockett, "Reconstructing the shape of a deformable membrane from image data", *International Journal of Robotics Research*, 19(9), pp 795-816 (2000).

[6] K. Kamiyama, H. Kajimoto, M. Inami, N. Kawakami and S. Tachi, "Development of a vision-based tactile sensor", *Trans. of Society of Electrical Engineers of Japan*, Series E, 123(1), pp 16-22 (in Japanese) (2003).

[7] G. Obinata, T. Kurashima and N. Moriyama, "Vision-based tactile sensor using transparent elastic fingertip for dexterous handling", *Proceedings 36th International Symposium on Robotics*, Tokyo, (2005).

[8] N. Xydas and I. Kao, "Contact mechanics and friction limit surfaces for soft fingers in robotics with experimental results", *International Journal of Robotics Research*, 18(8), pp 941-950 (1999).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Robotics in Over-Constrained Environments

Peter Kowski, Tim Short and Ernest Appleton
School of Engineering
Durham University, Durham, United Kingdom
peter.kowski@durham.ac.uk

Abstract

This paper has focussed on the specific design features that enable robots to deal with instances where the operating environment changes significantly with respect to one of the constraints. The term ‘constrained’ is used extensively in the robotic context. There are generally two established classifications of robot operating environment, namely a *fully-constrained* environment as seen in a purpose built facility; or the contrasting *ill-constrained* environment such as that found in nature. A newly defined case presented by this research has been termed an *over-constrained* environment. This case is shown to be an automation environment outside of the readily accepted classifications. These ‘over-constraints’ have been researched in order to determine specific robot design characteristics which are most suitable for overcoming the imposed constraints. This has resulted in a unique form of *constraint-aided design* with the potential to benefit the developers of such automation as it will allow rapid product development in a market where quick time to market is critical, due to environmental and financial pressures. In addition the process should yield superior designs because the finished product becomes increasingly task-compliant as it is optimised specifically to match the constraints of the application area. Autonomous mechanical equipment operating in industrial pipelines provides a case study example for robotics in such over-constrained environments.

Keywords: constraint, compliance, robot environment, pipeline robot, pig, ill-constrained, fully-constrained

1 Introduction

This paper has focussed on the specific design features that enable robots to deal with instances where the operating environment changes significantly with respect to one of the constraints. “The state of being restricted or confined within prescribed bounds” [1] is a dictionary definition of the term ‘constrained’ which is particularly pertinent to robotic operation. Every mechanical body can be regarded as having its motion restricted by an imposed stricture which is formed by a unique set of constraints. Every robot must therefore be designed to operate within these boundaries, which relate to either geometric or environmental constraints.

Robots are machines, and as machines they are subject to all of the constraints which are common in the study of mechanics. Mechanical constraints are boundaries that include internal limits such as the reach of an arm or external factors such as the operating headroom or the volume of a cavity. A robot is built up from various combinations of the six classical joints recognised by traditional kinematics but notably the two most constraining of all: the prismatic joint and revolute joint. It is the limit of each such joint and their interaction that ultimately defines the workspace of a robot. The external mechanical constraints are those that are imposed by the operating situation such as those related to particular obstacles. These should not be confused with environmental constraints such as temperature or pressure which can also have an important impact on robot design.

Environmental constraints are defined by the ambient operating conditions. These include the ability for the robot to operate within certain temperature norms as well as over minimum-service duration. Global parameters are constraints that influence the design and performance of a machine and examples include quantifiable measures such as operating temperatures, pressures and radiation as well as less quantifiable measures such as wear-and-tear and operational risk.

This paper focusses on only the primary geometric constraint group, and how machines can be better designed to overcome such mechanical constraints, especially in certain environments where these are particularly significant. Such constraints are generally dealt with by either intelligence and corresponding mechanism or alternatively by giving the machine increased mechanical compliance. The latter method has been traditionally overlooked and this study aims to address distinct relationships between compliant design features and specific geometric constraints.

Robotic environments are generally categorised by the significance of the constraints present and their impact on mechanical equipment operating in that environment. There are also many common misconceptions. Robots operating in nature are sometimes wrongly referred to in various literature as being ‘highly-constrained’. They may be subject to several simultaneous environmental and mechanical constraints, though the constraints are actually largely unknown which properly makes the environment ‘ill-constrained'. There are obvious differences between such an environment and a well-ordered factory. Whereas one environment was never designed for robotic use, the other is specifically adapted for automation. The latter environment is referred to as being 'fully-constrained'. These can be regarded as the two classical orders of robotic scenario. This research has suggested that there are other overly-standardised environments which do not fit in to either of these structures and may now require a possible further classification, namely the over-constrained environment.

2 Classical Robotic Environments

This work concentrates on the interrelationships between environmental constraints and the design issues they create in the development of robots. Therefore it is useful at this point to introduce a constraints perspective on classical robotic environments.

Ill-constrained environments (also know as under-constrained) are those where the environment is of a random nature that cannot be easily adapted to the utilisation of robotics or other automation. Examples include situations in nature such as marine exploration, outer-space, internal medicine and even most agricultural environments such as forestry. In the man-made 'built' environments they would also include construction sites and many domestic environments such as the home. In these environments, the robot utilised has to have a high level of flexibility, even to achieve the simple tasks such as being mobile. The use of automation tends to be complex especially in its control, as the environment often has to be analysed or measured before the automated task sequence can be initiated. Characteristically robots deployed have multiple sensors along with several joints for extra-ordinary degrees of freedom. Often the costs of this level of sophistication offset the necessity of a fully automated task, so it is quite common to find human operators in these environments. If this is not possible, human controlled remote operated vehicles (ROVs) are often deployed before a fully automated robotic sequence is considered. Several significant advancements in the fields of robotic vision and recognition systems, as well as in walking robots, have been made as a result of an interest in ill-constrained environments.

There are important distinctions between the terms *constrained* and *defined* when describing robotic environments. A continuously changing environment such as a battlefield or an unknown 'random' environment such as a demolished building are both *ill-constrained* and *ill-defined*, where constraints are primarily geometrical. High magnitudes of intelligence and kinematic complexity are the order of the robotic solution to such an environment. On the other hand, sub-sea exploration environments are still ill-constrained but now increasingly *defined* in that the constraints are primarily environmental constraints such as pressure, temperature and risk.

A fully-constrained environment is one which has been specifically designed for the use of robotics and automation. The environment is typically highly standardised so that relatively simple automation can be utilised. It is quite common to see dedicated single task robotic manipulators with few degrees of freedom. Often, many such robots will work together to undertake a sequence of operations. The prime example of such an environment is the mass manufacturing industry but other examples include certain types of mass-agriculture. Characteristically, changes to the automation may be implemented with ease, including changes to the task or even raising the speed or efficiency of a process, as the level of sophistication increases. The pinnacle of mass manufacturing automation technology was in the 1980s with the introduction of flexible machining centres and experiments in manufacturing system design carried out by established automobile manufacturers such as VW, Fiat, and GM.

3 Over-Constrained Environments

A key contribution to be made with this paper is the definition and discussion of what has been termed "an over-constrained" environment. Initially, this is seen to differentiate itself from the fully-constrained classification, in that it has not been exclusively designed for robotic use. It may seem that every overly-standardised environment such as a flat factory floor or any regular cross sectional cavity such as the inside of a pipe should by definition be considered *fully-constrained*. The standardisation lends itself well to automation tasks just as in the specially designed case, however robotics are not the primary function of these environments. Additional imposed constraints make these environments particularly challenging to the robot designer. Consider the case of an industrial pipeline which is designed for efficient product flow. Common features such as bends, valves, branches as well as the product in the pipeline itself all hinder the ability to support any robotic tool trying to share the environment. In addition, the current conditions of the pipeline are often completely unknown, so there are additional constraints imposed by damage, corrosion or deposits left inside the pipeline.

In a straight, smooth pipeline there are ordinary geometric constraints which make this akin to a fully constrained environment. Such pipelines are accessible to conventional pipeline inspection machinery such as pipeline inspection gauges commonly known as *pigs*. Such a pipeline is *fully-defined* and in common pipeline terminology this implies that the pipe is *pigable* [2,3] When considering the kinematics of a pipe traversing object such as a pig moving in such a fully-constrained environment, it is noted that it only has two kinematical degrees of freedom: Firstly, it can translate along the length of the pipe and secondly it is able to roll on its axis due to the circular cross section of both the pipe and the pig. This is demonstrated by the two arrows shown in Figure 1.

If a bend is introduced into the pipe, (as shown on the right hand side of Figure 1) the device is forced to perform a pitch or yaw action, which under normal circumstances the device is not capable of accomplishing. The environment can now be regarded as having become *over-constrained*. This simple analogy helps to further establish a definition of the new state: "An over-constrained environment is one where the environment changes with respect to one of the already constrained degrees of freedom."

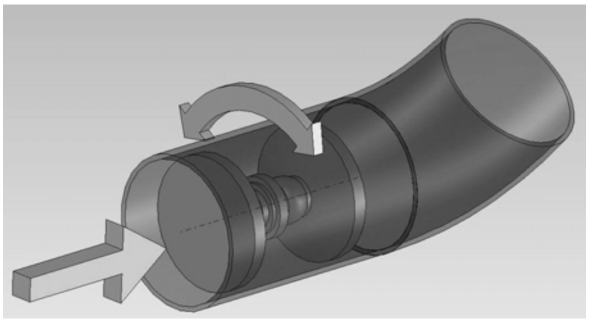


Figure 1: Ordinary kinematic degrees of freedom of a pipe traversing device.

How can a robot operate efficiently in these unique environments? In an ill-constrained environment there is an option of building sensors, intelligence and actuators into a machine in order to process and negate likely constraints. In the over-constrained context, there is relatively little that intelligence can accomplish in counteracting potentially critical geometric constraints. More significantly, the automation could be designed to be submissive to constraints so that they can restructure the automation automatically so that it can continue on its primary function of accomplishing the task.

Consider the example in nature of an earth worm that is able to negotiate highly-constrained, ill-defined environments using basic intelligence and sensors. It is highly flexible or compliant in terms of structure. A more rigid natural structure such as an ant requires higher magnitudes of intelligence and dexterity for such an environment, but it still largely depends on task-focus. Therefore, when considering a spectrum of different over-constrained and ill-defined environments it can be concluded that robots require effective combinations of compliance, flexibility, intelligence, dexterity and task focus.

4 Kinematic Features of Robots for Over-Constrained Environments

Having defined an over-constrained environment it is now possible to move on to suggest how robotic devices can be designed to accommodate this severity of constraint. An over-constrained environment imposes increased requirements on the kinematic functionality of the robot. For example, a device such as that shown in Figure 1 may utilise articulated joints to overcome the environment change. In this work, the normal pitch and yaw capabilities of joints [4] developed for fully-constrained environments were found to be unsuitable for the very sharp bends typical in over-constrained environments. A *hook-joint* (or conventional universal joint) pivots only about its centre, which was seen as a limit on the degree of flexure of such a joint operating in pipelines. The new requirement was to have a joint which was capable of moving its pivot point off centre depending on the exact nature of the imposed constraint. In this investigation a completely new joint has been devised and is shown in Figure 2. This special joint is capable of achieving up to 100 degrees of flex in any direction as it can pivot about the inside of the pipe bend if required to do so. It also exhibits unique features including non-rotating properties, which make it particularly suitable for the needs of over-constrained pipeline environments.

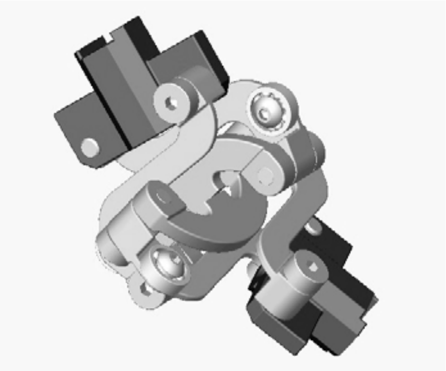


Figure 2: Model of Hood-Kowski joint.

This joint, named the Hood-Kowski joint, has the interesting characteristic of consisting of only seven parts (excluding the fittings) and only three types of component which consist of: the end connection pieces; identical 'S' linkages; and a symmetrical central structure. These components can be cheaply manufactured using common laser cutting technology. This minimum component strategy has been beneficial not only for reducing the component and assembly cost, but also for improving mechanical reliability by reducing the mechanical complexity.

The Hood-Kowski joint has been successfully implemented on a pipeline robot developed for a 2 inch minimum diameter pipeline system consisting of multiple 5 inch radius pipe bends. Such a pneumatically driven robot consists of three rigid components as shown in Figure 3, which have been articulated at two points as shown by use of Hood-Kowski joints. This has enabled the robot to function in this over-constrained environment.

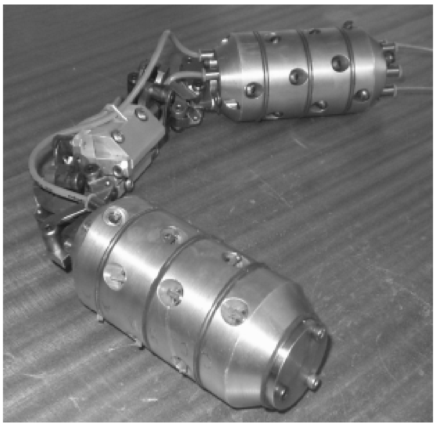


Figure 3: Pipeline robot using Hood-Kowski joint.

5 Compliance Features of Robots

A further significant contribution made from this work relates to the use of structural compliance to deal with over-constraint of the robotic environment. In most instances robots are designed to be rigid structures limited by a series of kinematic joints. A less usual way of articulating a robotic device is to use compliance.

It has been deduced that increasing the *compliance* of a robot operating in an over-constrained environment such as a pipeline can be used as an alternative to using an intelligent mechanism in order to overcome geometric constraints. Compliance can be regarded as the ability of a device to adapt to the features of an environment. In a geometric context this particularly relates to matching to the shape [5] of the environment in order to move with the use of minimum energy. Several studies have shown that shape compliance and ease of passage are closely related especially in anknown ill-constrained environments such as in buildings demolished in an earthquake. [6]

In order to understand such compliance, it is best to look at natures methods. Several studies have been undertaken in this field such as one by the mathematician and designer Pierce [7] who began a comprehensive study more than thirty years ago into how structure in nature could be utilised as a strategy for design. In nature structure is proven to be based on "*minimum inventory and maximum diversity*". The complex forms and structures such as those found in snowflakes are typical examples of how nature utilises constraints in a similar rubric of minimum energy to form diverse shapes. Similarly DNA is a relatively simple molecular structure that creates complex long chain shapes of itself consisting of diverse structures.

The same philosophies of low inventory and maximum diversity can be applied to the compliance requirement of pipeline robots. Geometric constraints are determined by the features of a pipeline, which include bends, diameter restrictions and obstructions due to valves, welds, damage or unwanted debris. An initial requirement is for the robot to take on a diversity of shapes and have the ability to quickly reconfigure itself into a new shape. Rather than achieving this through mechanism and intelligence, research has been undertaken on allowing a robot to be passively receptive of the constraints, and becoming compliant.

Compliance relates to two conditions: global compliance and local compliance. The first relates to the overall function of the system as well as the interaction of various system components. The latter condition relates to the performance and characteristics of individual components. Research has been undertaken on these two distinct levels. For example, component material characteristics such as flexibility refer to local compliance, whereas mechanical joints are usually ways of addressing the global compliance issue. Minimum inventory characteristics also relate to global compliance, as simpler structures consist of fewer moving components and can be rapidly reconfigured, which makes them more compliant.

In the case of pipeline robotics, most of the calculations of shape are concerned with the length of a component, and its relationship with the diameter (the aspect ratio). Due to functional requirements of the traction mechanism or tool, it is most desirable to incorporate both maximum length and maximum diameter. There are two methods of allowing optimal device length namely to introduce a joint or move to a non-rigid (flexible) design. As already indicated any reduction in diameter of the device, to allow the passing of bends will remove the yaw and pitch constraint in a straight length of pipe. In practical terms this means that a device may be susceptible to *jack-knifing* in straight portions of pipe. This can be inhibited without reducing bend passing capability by the design of appropriate suspension systems and by using appropriately selected joints (with appropriate stiffness) between rigid robot components.

The two processes of overcoming constraints in a pipeline provided a structure to this study: Firstly, the flexibility issue became the basis of a study into flexible elements used by robotics in over-constrained environments. In the pipeline context, various flexible elements are already commonly in use and are known as flexible probes and umbilicals. The second issue prompted research into the kinematic behaviour of robotic elements in over-constrained environments. In the pipeline context, an investigation into pipeline centralisation techniques, traction mechanisms and the special purpose joints already mentioned form the basis of investigations into compliant robot systems.

6 Constraint-Aided Design

The definition of over-constraint has led to a consideration of the use of kinematic and compliance-based designs of robotic devices. It is now possible to bring together these ideas into a wider consideration of "constraint-aided design".

The over-constrained environment can be regarded as a unique type of robotic scenario in that the design of robotic tools is primarily dominated by the constraints and not the function. Just as a robot utilises constraints beneficially to accomplish its task it is imperative to recognise the effective use of the constraints from the product-design stage onwards. The objective of the research was to identify any constraint-aided process of solving the environment constraint problems, with the aim of developing a knowledge-set that can be applied for the future design of robots in any over-constrained environment.

Many problems in computer science, Artificial Intelligence and business modelling are now being interpreted by way of constraint processing. Constraints exist in any logical decision making process. The technique of dealing with constraints in reaching a solution is a simple method of constraint-processing. If a set of constraints is defined intuitively on some base set of parameters, then a more structured constraint-processing algorithm could be utilised directly to efficiently aid the decision making process. In this way, the design of robots for over-constrained environments could be formulated as a *constraint satisfaction problem* (CSP). These are increasingly gaining impetus in the engineering design discipline using emerging techniques such as "constraint processing" as described by O'Sullivan [8].

Pipelines have been identified as a perfect context of an over-constrained environment and the design of compliant pipeline access equipment as well as more complex autonomous pipeline machinery is the vehicle of study. To develop a parameterised set of constraints for the pipeline robot design context it was necessary to investigate the nature of constraints in the most over-constrained pipeline environments. In-depth case studies of nuclear as well as oil-production industries were used to investigate and catalogue compliant designs for solving constraints in these environments.

The developers of pipeline robots are often required to provide a bespoke design as a result of some particular problem that is affecting a pipeline system. Due to environmental and financial pressures caused by the non-functionality of a commercial pipeline, a rapid product development is often demanded of the market. New designs can quickly be realised, if the constraints are immediately recognised and if the robotic system to be deployed is based on a common platform and incorporates known design features for overcoming particular constraints. Various studies have investigated parameterised constraint processing using CSPs [8] and constraint driven *design spaces* [9] which could represent different base functions of the robot.

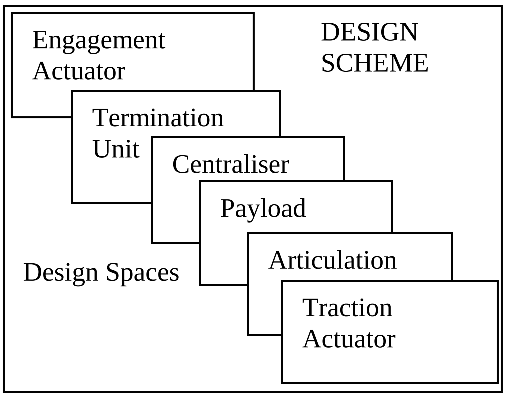


Figure 4: An example design scheme structure for a generic pipeline robot.

The development of a *designers' toolbox* has become a central feature of the constraint-aided design process. This contains the *design scheme* of the complete pipeline robot made up of design spaces of individual functional components as shown in Figure 4. Each design space may consist of numerous designs related to typical constraints found in industrial pipelines. Several design aspects were selected for populating such a structure, based upon their significance in overcoming an individual constraint. The top level scheme is based upon the characteristic that any pipeline traversing device consists of a number of base features which are all used to achieve the primary trajectory objective of passing down a pipe with minimum energy and without damaging the pipe-wall. These include traction, centralisation and articulation. As an example of one such feature: centralisation, it can be shown that any pipeline device will try and maintain an axial relationship with the centreline of the pipe by utilising some base constraints in much the same way that a robot in a fully-constrained environment may behave. Particularly essential to this toolkit are the limits and characteristics of how such a feature copes with the increased requirements beyond the simple fully-constrained environment.

7 Pipeline Robot Tool for Nuclear Environment: A Case Study

The understanding presented in this paper stemmed from a number of practical design challenges which came to light after technical collaborations with a number of companies. The following section describes a particular instance that is considered particularly illustrative of such concepts. The development of a robotic tool for a radioactive environment provided one of several case studies for implementing the design procedures suggested by this study. The requirement of this particular tool was to measure corrosion levels of a coil in a glass vitrification cell in a vitally important live nuclear facility, which was to be achieved by a careful visual and ultra-sound inspection procedure. Environmental constraints included radiation, high operating temperature and proven recovery in all failure modes. The geometric constraints were even more critical, as the pipeline contained large diameter changes, consecutive bends, unknown weld beads and pipe misalignments.

Initial lines of design development took place using the toolbox strategy. Specialist miniature crawler technology based on compliant bristled traction devices were necessary for the kinematic requirements of the application, but were found to be unsuitable in dealing with the critical reliability issues imposed by the CSP framework. Equally, an un-powered conventional probe type solution limited the distance of deployment as a result of friction and the capstan effect due to multiple consecutive bends.

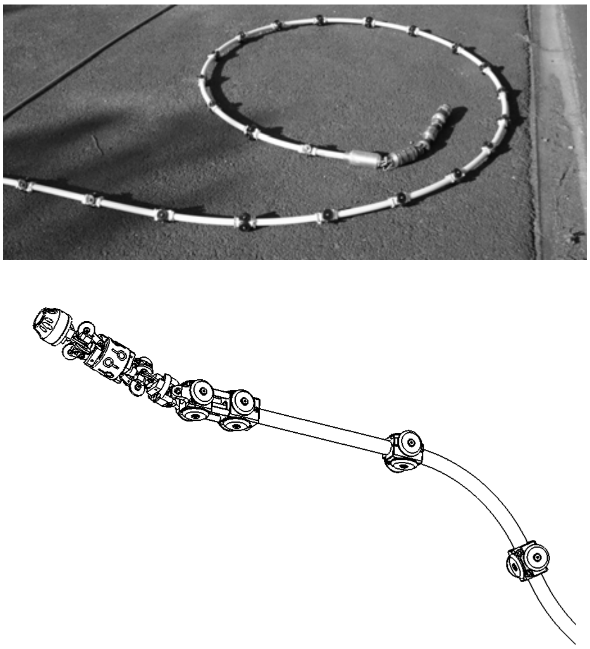


Figure 5: Nuclear facilities inspection probe.

A purely compliant solution was suggested in the form of a probe-type tool as shown in Figure 5. It may not seem that such a design came about as a result of the same functional design scheme used in powered pipeline robots however it can be proven that it consists of the same base features as all pipeline robots investigated by this study.

The basis of compliance and minimum energy dissipation was to find a balance between flexibility and stiffness of the base probe material, whilst maintaining high tensile strength. This also required low diameter and weight due to constraints imposed by deployment. Energy dissipation was found to be both in strain energy (in shaping the probe into the new shape imposed by the environment) and the frictional losses already mentioned. To achieve the desired trajectory objective over the length of inspection, low rolling friction wheels were added along the length of the probe. These were given increased compliance by making them large enough to overcome step changes in the pipeline schedule as well as weld beads.

As a result of this research, the robotic probe called the *D-Flex* has now been utilised to deploy ultra-sonic and visual sensors through over-constrained pipelines deep into a live nuclear environment.

8 Conclusion

A robot operating in an over-constrained environment such as a pipeline needs to have in-built capability to either passively or actively deal with the constraints that are placed on them. The nature of these constraints is such that they can bring about a significant environment change with respect to an already constrained degree of freedom. This research suggests that the constraints in these unique over-constrained environments should have an over-riding influence on the robot development process from the design stage onwards. Specific methods of how devices can accommodate the characteristics of an over-constrained situation have been examined, and suggestions on improving this process have been made. In addition, the robot design process has been optimised as a result of efficient methods of constraint recognition, parameterisation and processing by use of emerging techniques such as the CSP.

9 References

[1] The American Heritage Dictionary of the English Language (4th edition), “constraint”, http://www.thefreedictionary.com/constraint, visited on 30/05/06.

[2] J. Cordell and H. Vanzant, *An Introduction to Pipeline Pigging*, (5th edition), Pipes and Pipelines International, Beaconsfield (2003).

[3] J. Southgate. *Wax Removal Using Pipeline Pigs*, PhD Thesis, University of Durham (2004).

[4] N.W. Stutchbury, *Design Characteristics of a Pipe Crawling Robot*, PhD Thesis, University of Durham (1999).

[5] W. Li, *Positioning Technology for Stepwise Underground Robots*, PhD Thesis, University of Durham (1998).

[6] Z. Wang, *Design Study of an Earthquake Rescue Robot*,PhD Thesis, University of Durham (2004).

[7] P. Pierce, *Structure in Nature is a Strategy for Design*, MIT Press, Cambridge, MA (1978).

[8] B. O’Sullivan, *Constraint-Aided Conceptual Design*, Professional Engineering Publishing, London (2002).

[9] A. Csabai, J.G. Taiber and P.C. Xirouchakis, “Design support using constraint-driven design spaces”, in B. Bruederlin and D. Roller (eds), *Geometric Constraint Solving and Applications*, Springer, Heidelberg, pp 82-106 (1998).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Validation of a Catadioptric Stereo System

Donald G. Bailey, Jonathan Seal and Gourab Sen Gupta
Institute of Information Sciences and Technology,
Massey University, Palmerston North, New Zealand
d.g.bailey@massey.ac.nz

Abstract

A catadioptric imaging system was constructed using low precision mirrors. As a consequence, the captured image is subject to a number of local distortions. The perspective, lens and mirror distortions are removed using a non-parametric calibration approach. The effectiveness of the catadioptric system, and associated calibration procedure is validated by capturing an image of a known scene, calculating the disparity map and depth map, and comparing the measured ranges of features with their true depths. A simple dynamic programming based stereo matching algorithm was used to validate the setup. The locations of significant object edges within the image corresponded well with the actual object range, to within 1 pixel disparity.

Keywords: catadioptric stereo, calibration, mirrors, depth measurement, range map

1 Introduction

Stereo imaging is a commonly used method for depth perception, particularly in the area of mobile robotics. Stereo imaging uses multiple images of the same scene taken from different camera locations. The multiple (usually two) images are related in such a way as to provide disparity. Disparity is defined as the relative movement of an object between two or more views. In a stereo imaging system, the cameras are spatially separated, resulting in the disparity being a function of depth. The disparity is found by matching corresponding points in the input images.

In practise, the stereo matching problem is simplified by making use of epipolar geometry. This allows a 1-D search to be used for matching points rather than a full 2-D search. Consider one object point. That point, and the two points corresponding to the camera centres define a plane in 3-D space. All object points on that plane will be imaged to a single line in each of the captured images. Therefore, in finding a matching point, the search may be restricted to a corresponding line in the other image. The epipolar lines do not, in general, coincide with the image scanlines.

While image resolution is the ultimate limiting factor in terms of precision, lens distortion can introduce significant errors. The field of photogrammetry has developed techniques for characterising and compensating for any distortions present, enabling accurate measurements to be made. Camera calibration involves determining the relationships between the two cameras and the scene (enabling the epipolar lines to be determined) and correcting for distortions resulting from the imaging process. For stereo imaging, absolute positioning with respect to a world coordinate system is not as critical as determining the relative positions of points on the objects in both views [1].

1.1 Our Catadioptric Stereo System

A catadioptric system uses both lenses and mirrors as focusing elements. For stereo imaging the principle is that the lenses and mirrors are designed to act in a way that produces two images on the same sensor [2]. The basic arrangement of our system is illustrated in figure 1. The detailed design of the mirror system is described elsewhere [3]; however, a key factor to reduce the cost in the design is to use polished aluminium mirrors. Such mirrors are only approximately planar, and can introduce local distortions. An image of a regular rectangular grid is shown in figure 2.

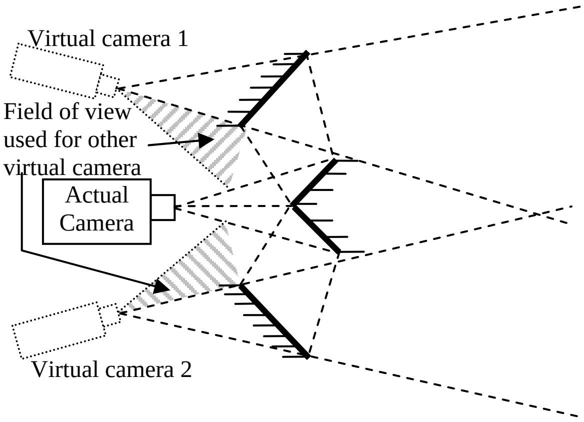


Figure 1: Our catadioptric stereo vision system.

2 Calibration

There are at least three sources of distortion apparent in figure 2. The first of these is perspective distortion. This results from the convergence of the centrelines of the two cameras. Such convergence is unavoidable in this particular configuration if there is to be any overlap between the field of view of the two virtual cameras. The perspective distortion means that the epipolar lines are no longer parallel, and this complicates the matching of corresponding points. Matching is further complicated by the position dependent change in scale factor between the left and right image of each view.

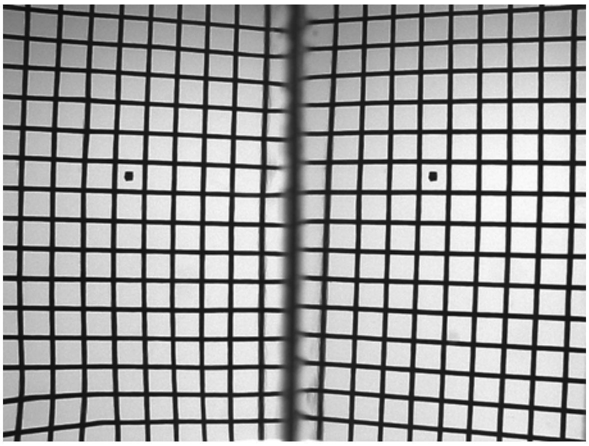


Figure 2: The image of a regular calibration grid seen through the vision system.

The second source of distortion is barrel distortion resulting from the use of a wide angle lens. It results from the lens having a slightly higher magnification in the centre of the image than at the periphery. In figure 2, this effect, although significant, is dominated by the perspective distortion. Barrel distortion is particularly significant for catadioptric stereo imaging because the two different views have the distortion in opposite senses. The right side of the left view and the left side of the right view are in the centre of the image and have relatively little lens distortion. Without calibration, barrel distortion will contribute a significant systematic error in the measured range.

The third source of distortion results from minor distortions in the polished aluminium mirrors. These result in apparently random deviations or twists in the lines. The distortions from the mirrors are different for the two views because each view uses a different pair of mirrors. Again the effect is a significant position-dependent systematic error in the measured range.

The perspective and barrel distortions can be modelled with parametric models with relatively few coefficients. However, the same is not true of the mirror distortions. Therefore successful calibration in this application requires a non-parametric approach.

2.1 Calibration Procedure [4]

Non-parametric calibration is effectively determining the relationship between the captured image coordinates, (x,y), and an undistorted coordinate system (u,v) for every pixel in the image.

Although the perspective and barrel distortions are global, the local nature of the mirror distortions requires calibration at a dense set of points. Unless each pixel is calibrated (as is done in [5]) it is necessary to assume that the mapping is locally smooth.

A rectangular grid is placed at the designed working distance of the system in such a way that it fills the field of view of both virtual cameras. The grid intersections provide a set of corresponding points where the u and v locations are known in both views. The lines themselves provide a set of equally spaced (in terms of the real world) samples of both the u and v axes of the desired undistorted image.

The centres of each of the gridlines are detected [6], and approximated by a smoothing spline. These gridline splines effectively characterise the distortions present in the image from all three sources. A distortion map is then produced for each of the two ‘views’ by fitting interpolation splines to each row and column approximately perpendicular to the gridlines. That is, the vertical gridlines give the horizontal distortion map and vice versa.

The mappings obtained in this way are forward maps (they specify where each distorted pixel should map to in the undistorted image). To apply them to an image to correct the distortion requires reverse maps (where each undistorted output pixel, indicates where it should come from in the input image). For simplicity, a two-pass mapping process is used [7] to perform the correction as a set of independent 1-D corrections. The corrected stereo pair for the calibration image is shown in figure 3.

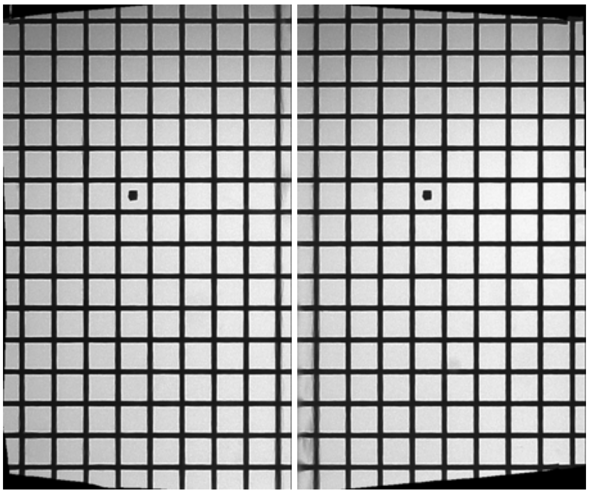


Figure 3: The calibration image after correcting for lens, mirror and perspective distortions.

3 Stereo Matching

The removal of perspective distortion results in the more conventional parallel camera geometry. If each horizontal gridline is co-planar with the two virtual cameras then after correction the epipolar lines will correspond to horizontal lines in each of the transformed views. This enables a simpler 1-D matching algorithm to be used. Mapping both views to the same coordinate system means that objects at the working distance will have no disparity, effectively setting the location of the calibration target as the reference range. Calibration is therefore relative to the camera, rather than in terms of absolute world coordinates. Points with positive disparity are closer to the camera than the working distance, and points with negative disparity are beyond the target range.

3.1 Obtaining the Stereo Pair

A single image captured by the catadioptric stereo camera will contain two views. The mirror system effectively enables the scene to be simultaneously viewed by from two slightly offset directions as shown in figure 4. When the distortion maps obtained from the calibration procedure are applied, the image is split into its two components (see figure 5). While most of the distortion has been removed, a small amount of residual distortion remains. This is because the calibration grid provides a relatively sparse sampling of the image and has not captured all of the local distortion.

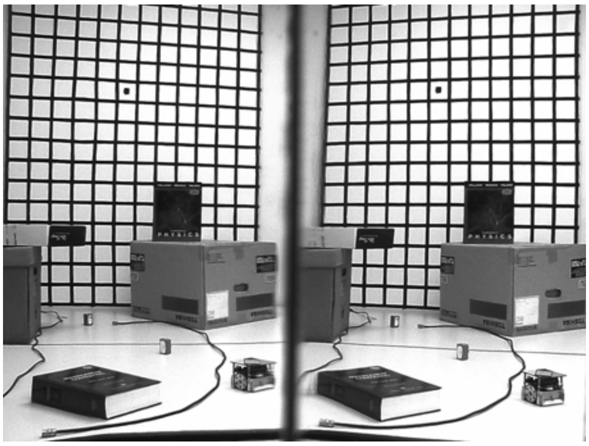


Figure 4: A scene captured using the catadioptric stereo camera.

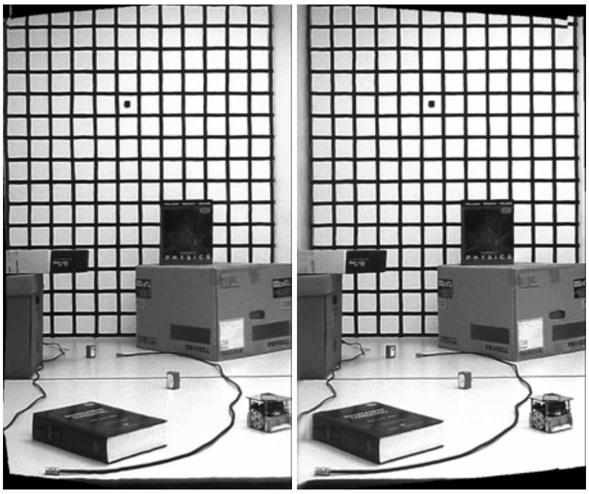


Figure 5: The image of figure 4 corrected for distortion and split into the left and right images.

To build a depth map of this scene, each point in the left image must be matched to the corresponding scene point in the right image. Depth can then be inferred from the offset or disparity between the corresponding points.

3.2 Image Normalisation

One problem with simple intensity matching (or local correlation) is that the intensity can vary significantly between the two images of the pair. While the intensity differences in figure 5 appear relatively minor, even a small local intensity gradient can result in a significant matching error, especially in regions containing very little detail.

One approach to overcoming this problem is to only perform the correspondence matching on edge pixels. The approach we took was to apply intensity normalisation to the image to correct for the intensity differences between the images. First the local average intensity is estimated by using Gaussian filter with a standard deviation of 8.5 pixels. Normalisation is achieved by dividing the input image by the local average. This effectively provides a form of local gain control that normalises the image intensity. The result is that edges, and any other texture details, are enhanced, while any subtle lighting gradients are completely suppressed (see figure 6).

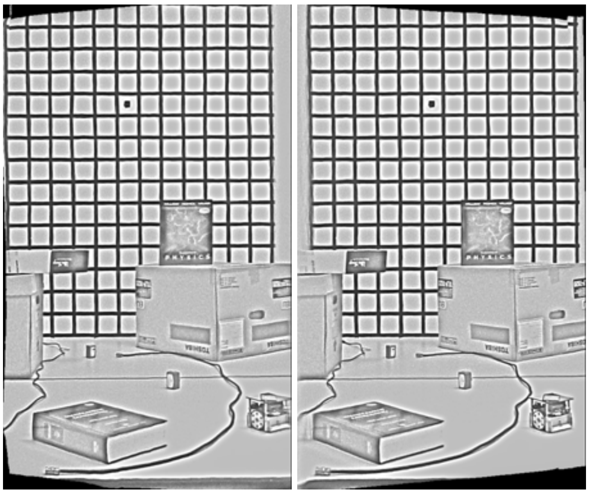


Figure 6: Intensity normalised stereo image pair.

3.3 Correspondence Matching

Since the calibration procedure makes the epipolar lines horizontal, any point in one image will appear on the corresponding row within the other image. Therefore each row within the stereo pair may be matched independently.

While there is a wide range of stereo matching algorithms available [8], for the purpose of validating the setup a simple correspondence matrix approach was used. A correspondence matrix considers the ‘compatibility’ between each pixel in the left image with each pixel in the right image. The absolute difference between pixel values is used as the compatibility measure. The correspondence matrix therefore shows all possible correspondences. Points in the matrix with low values represent similar pixel values between the two images, and therefore potential matches. High matrix values represent strong incompatibilities. An example correspondence matrix is shown in figure 7.

Correspondence matching is therefore equivalent to finding a set of compatible matches within the correspondence matrix. Any deviation of the matches from the central diagonal represents the disparity in the image.

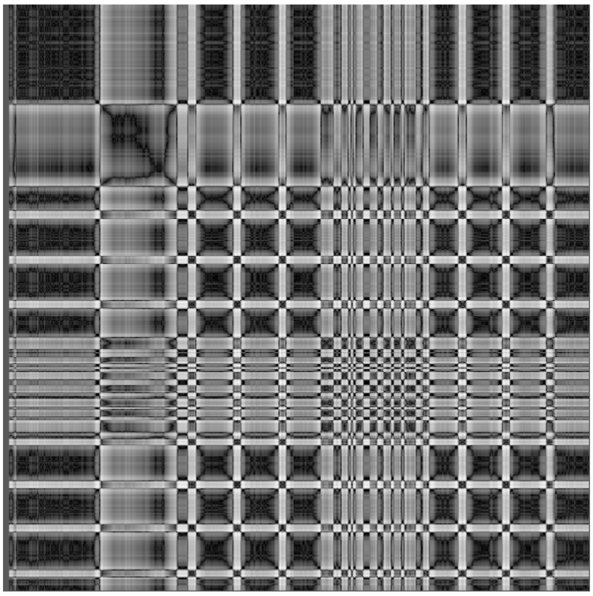


Figure 7: Correspondence matrix for line 512 of the stereo image pair. The left image is represented horizontally, and the right image vertically.

Generally we are interested in matching over a relatively narrow range of disparity. This corresponds to a diagonal band in the correspondence matrix. If the correspondence matrix is rotated, the disparity becomes more obvious. The rotated and constrained correspondence matrix is shown in figure 8. This is scaled so that the horizontal axis corresponds to the image position (from partway between the two views). The vertical axis directly measures the disparity.

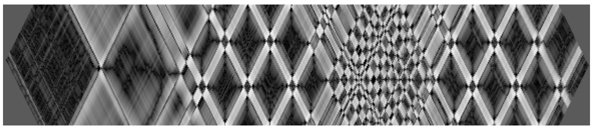


Figure 8: Rotated and constrained correspondence matrix. The disparity is represented vertically, and the image position is represented horizontally.

3.4 Determining Minimum Cost

The values in the correspondence matrix are considered as a penalty function, with dynamic programming techniques used to minimise the total cost to get from one side of the matrix to the other. The path directly gives the disparity as a function of position (figure 9).

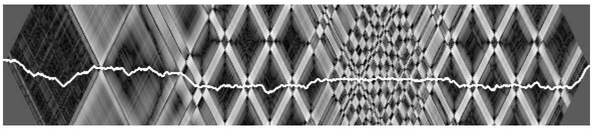


Figure 9: Minimum cost path through the correspondence matrix.

An assumption implicit in this dynamic programming approach is that the relative order of the pixels in the two images is the same [8]. In many circumstances, this will be the case, although with narrow foreground objects this assumption will be violated.

A second assumption is that every pixel in one image will have a correspondence in the other image. This does not take into account occlusions, both of background objects occluded by foreground objects, and of surfaces oriented such that they are visible in one image but not the other.

The minimum cost approach is effectively performing a correlation between the left and right images with a 1 pixel wide window. Generally better results may be obtained by using a wider correlation window. This may be simulated by smoothing the correspondence matrix of figure 9 with a horizontal box average. The effect of smoothing with a 9 pixel window is shown in figure 10.

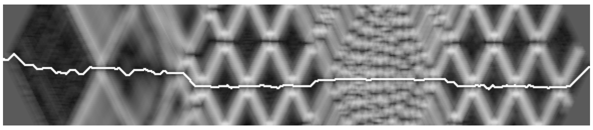


Figure 10: Effect of smoothing on the minimum cost path.

3.5 Disparity Maps

This process directly gives the disparity for each row in the image. The rotated correspondence matrix gives the disparity for a ‘central’ view, halfway between the left and right views. The raw disparity map for this stereo pair is shown in figure 11. The results are consistent from row to row where there are vertical edges or other features; however the disparity of featureless regions is somewhat random. The image may be improved by post-processing with a vertical median filter [9]. Using a median filter significantly reduces this noise by enforcing some consistency between nearby rows, without significantly affecting edges. The effect of filtering using a 13x1 median is also shown in figure 11.

The disparity may be referred to the two side views to obtain disparity maps for the individual views (figure 12). Significant edges are detected using a Sobel filter and overlaid on these images for reference.

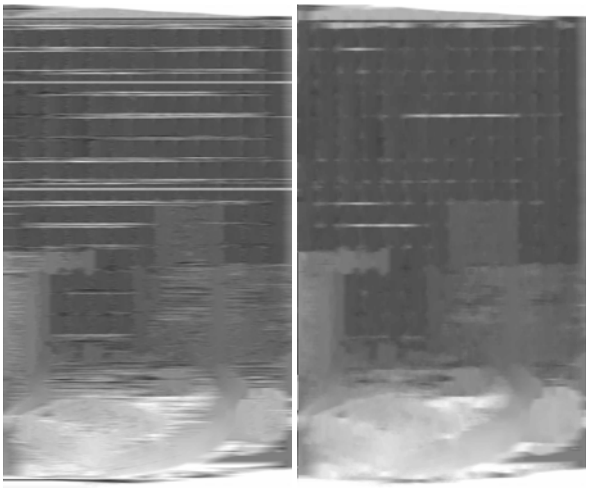


Figure 11: Raw disparity map (left) and filtered using a 13x1 vertical median filter (right). Mid grey corresponds to 0 disparity.

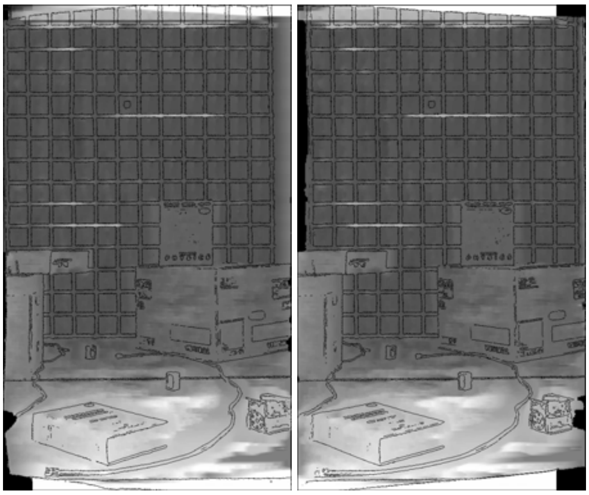


Figure 12: Filtered disparity maps for the left and right images with significant edges.

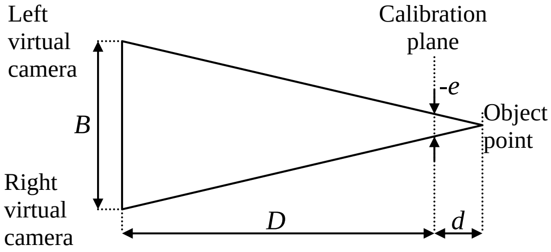


Figure 13: Imaging geometry after calibration.

3.6 Range Map

The disparity map can be converted to a depth or range map. The geometry of the vision system after calibration is shown in figure 13. *B* is the baseline between the virtual images, and is obtained from the arrangement of mirrors relative to the camera [3]. *D* is the working distance; the distance of the calibration plane from the virtual cameras. *e* is the disparity between the calibrated left and right images. After calibration, points at the working distance, *D*, have no disparity. Points closer than the working distance have positive disparity, and those beyond the working distance have negative disparity. *d* is the offset of an object point from the working distance.

From similar triangles:

$$\frac{-e}{d} = \frac{B}{D+d} \tag{1}$$

This may be solved for the displacement from the working plane:

$$d = \frac{-eD}{B+e} \tag{2}$$

Therefore the range is related to the disparity by:

$$D+d = \frac{DB}{B+e} \tag{3}$$

Figure 14 shows this relationship between range and disparity for the configuration used in these tests.

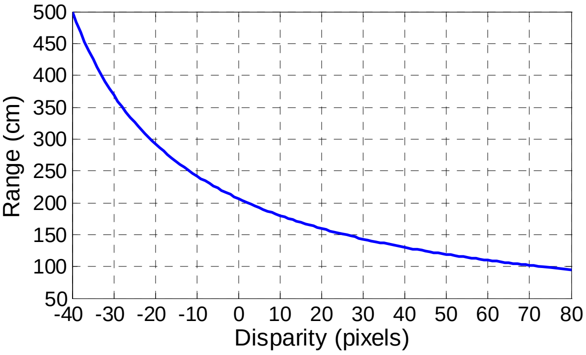


Figure 14: Relationship between range and disparity. The calibration range, *D*=206.5 cm, the baseline *B*=10.5 cm and the calibration scale is 6.5 pixels/cm.

4 Results

As a simple stereo matching algorithm is being used, the results are really only valid where there are significant edges or texture within the image. Textureless regions between the sharp edges have had very few constraints (for example a smoothness constraint could be applied [8]) and as a result, the depth map has less validity in these regions.

Two tests were performed to validate the results. Within the test image, the locations of 25 corners were measured, and their depths estimated from the stereo pair. The objects spanned the full range of the scene, between 1 and 3 metres from the camera. The corresponding disparity range was -23 to +60 pixels disparity. A disparity range of -60 to +60 pixels was used for the matching process. The errors associated with each of the measured ranges are shown in figure 15. The errors indicate that the measured disparity was up to 1 pixel out. Ideally, all of the errors should have been within 0.5 pixels disparity.

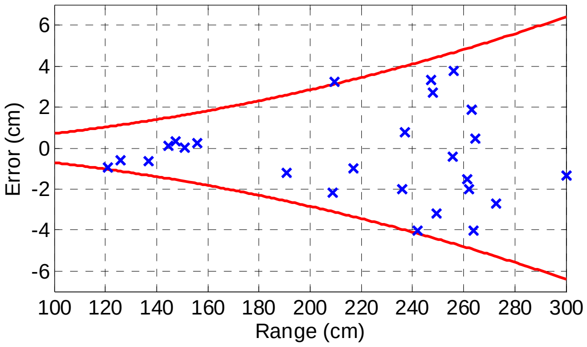


Figure 15: Errors from a set of 25 sample vertical edge points. The solid lines show the error associated with ±1 pixel disparity.

The second test was to measure the disparity of all of the vertical line pixels from the background grid (at a range of 3 m). This will give an indication of the extent of the distortion remaining after calibration. Only the line pixels are measured, because the spaces between the lines have very little texture, and the matching is more random at these locations. This data is shown in table 1, along with the mean and standard deviation range.

Table 1: Measurements made on the background grid. Columns 3 and 4 give a count of line pixels that have a particular disparity.

Disparity	Range (cm)	Raw image	After filtering
-24	319.0	214	199
-23	311.9	472	459
-22	305.1	3964	3627
-21	298.7	26149	27274
-20	292.4	6624	6835
-19	286.5	45	37
Mean range		**298.5 cm**	**298.4 cm**
Range standard deviation		**4.0 cm**	**3.9 cm**

In the raw image (before median filtering), there were several lines that matched incorrectly (see figure 11). This is because the regular grid pattern is ambiguous, and the gridlines were matched with an offset of 1 grid spacing. Such errors are unavoidable without enforcing consistency from one row to the next. These errors are removed by applying the median filter.

The results indicate that the median filter has very little effect on the range data where there is strong edge information. There is some spread of the disparity values, although the RMS error (or standard deviation) is significantly less than 1 pixel.

5 Conclusions

We have demonstrated that it is feasible to use a catadioptric system with low cost polished aluminium mirrors for stereo image capture. It is essential to calibrate for the lens and mirror distortions, and an appropriate calibration can significantly simplify the correspondence matching by mapping the epipolar lines to the image scanlines.

The effectiveness of such an arrangement for stereo imaging has been demonstrated by measuring the range of a set of feature points. Range errors indicated that the feature points had disparities that were up to 1 pixel in error. A larger sample of background pixels identified some points where the error in the disparity was 2 or 3 pixels. These indicate that the distortions have not been completely removed by the calibration process, and suggest that a finer calibration grid is required to more accurately model the distortions.

A more sophisticated stereo matching algorithm should give more accurate range mapping within the relatively featureless areas of the image.

6 References

[1] M. Qui and S.D. Ma, “The nonparametric approach for camera calibration”, *Proceedings 5th International Conference on Computer Vision (ICCV'95)*, Cambridge, MA, pp 224-229 (1995).

[2] J. Gluckman and S.K. Nayar, “Catadioptric stereo using planar mirrors”, *International Journal of Computer Vision*, 44, pp 65-79 (2001).

[3] J.R. Seal, D.G. Bailey and G. Sen Gupta, “Depth perception with a single camera”, *Proceedings International Conference on Sensing Technology*, Palmerston North, New Zealand, pp 96-101 (2005).

[4] D.G. Bailey, J.R. Seal and G. Sen Gupta, “Nonparametric calibration for catadioptric stereo”, *Proceedings Image and Vision Computing New Zealand (IVCNZ'05)*, Dunedin, New Zealand, pp 404-409 (2005).

[5] J. Wang, O. Hall-Holt, P. Konecny and A.E. Kaufman, “Per pixel camera calibration for 3D range scanning”, *Videometrics VIII*, San Jose, CA, 5665, pp 342-352 (2005).

[6] D.G. Bailey, “Sub-pixel estimation of local extrema”, *Proceedings Image and Vision Computing New Zealand 2003*, Palmerston North, New Zealand, pp 414-419 (2003).

[7] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, Los Alamitos, CA (1990).

[8] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, *International Journal of Computer Vision*, 47(1), pp 7-42 (2002).

[9] Q. Shi and G. Gimel’farb, “Postprocessing for dynamic programming stereo”, *Proceedings Image and Vision Computing New Zealand (IVCNZ'00)*, Hamilton, New Zealand, pp 210-215 (2000).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Integrated IPMC Sensor-Actuator Devices for Walking Robots

Gurvinder S. Virk, Dave R. Harvey, Ioannis Chochlidakis, Sanja Dogramadzi and Abbas Dehghani
School of Mechanical Engineering, Intelligent Systems Group
University of Leeds, Leeds, United Kingdom
g.s.virk@leeds.ac.uk, d.r.harvey@leeds.ac.uk, men2ic@leeds.ac.uk, s.dogramadzi@leeds.ac.uk

Abstract

Ionic polymer metal composites (IPMC) are dual purpose flexible materials capable of converting electrical energy to mechanical force for actuation, or converting mechanical movement to electrical signals for sensing force and/or displacement. The paper presents a methodology for designing integrated IPMC sensor-actuator devices (iSADs) to realise force/displacement feedback for IPMC actuators. Two IPMC strips are used, one to provide the actuation and the second for the sensing. In this way multiple IPMC strip devices can be designed for sophisticated iSAD elements that could have potential use in various CLAWAR robot applications. One such application, to construct walking robots based on iSAD components, is discussed.

Keywords: Ionic polymers, IPMC devices, iSAD, sensors, actuators, integrated sensor-actuator, beam theory, walking robots.

1 Introduction

Electroactive polymeric materials (EAPs) are a class of material that have been the subject of intensive research since 1992. Ionic polymer metal composites (IPMCs) are a class of EAPs that can be used as either a sensor or an actuator in small robot/mechatronic applications. IPMCs consist of an ion exchange membrane (e.g. Nafion®) with a metal layer deposited on its two sides that acts as electrical contacts. Their low density, low voltage requirements, mechanical flexibility and ease of processing offer several advantages over traditional electroactive materials [1]. These advantages of IPMCs have led to consideration of such materials in various applications [2]-[5]. As an actuator, they exhibit a bending motion in their transversal direction when a voltage is applied across the metal coatings. As a sensor, a charge is generated when the IPMC is deflected, providing an electrical signal depending on the deflection. The forces produced by IPMC actuators are extremely small (around 1-5mN) and hence their use is limited to micro-applications.

The dimensions of the IPMCs influence the performance of the sensor or actuator. For example, maximum actuator tip deflection is achieved when the IPMC is long, whilst maximum force is achieved when a short IPMC strip is used. The range of usable validated working dimensions of current IPMC strips is 10-60mm in length and 3-6mm in width and they are available in only a few thicknesses. For example, DuPont supply IPMCs in the following thicknesses: Nafion 112 (50µm), Nafion 1035 (89µm), Nafion 105 (127µm), Nafion 115 (140µm), Nafion 117 (178µm), Nafion 350 (254µm) and Nafion 417 (432µm). If other thicknesses are needed it is necessary to produce them by purchasing Nafion resin.

A further factor to consider is the hydration levels of the IPMCs. IPMC actuators work best when wet, whilst sensors operate better when dry. Hence for operation in air, the actuators need to be coated appropriately to prevent dehydration or else they have a very limited working life (approximately 1-3 minutes) due to drying out and require hydration for further operation.

At present IPMC actuators are mainly used in an open loop fashion with no feedback to regulate the deflection achieved. To overcome this, it has been proposed to integrate a second element onto an IPMC actuation strip to act as a sensor for the actuator strip. In this way an IPMC integrated sensor-actuator device (iSAD) can be realised. As the actuation forces are small, the deflection of the actuator in these cases will be affected not only by its own characteristics but also by those of the integrated IPMC sensor and the integration/attachment mechanism. A design methodology for integrating sensor-actuator devices has been produced to realise the required closed-loop actuator specifications and initial results have been presented in [6]. This paper extends these results and presents a potential application for using the iSAD devices in realising micro-walking robots.

An EC funded research project "Ionic polymer-metal composite as sensor and actuator: Application in motion control" (ISAMCO, EC contract FP6-505275-1, see www.mediainnovation.it/progetti/isamco/) aims to fabricate IPMC devices for sensing and actuation applications. To achieve this, dedicated Matlab software has been developed to assist in the design of individual IPMC actuator and IPMC sensor devices. This design software has been used in conjunction with beam theory [7] to realise a design methodology for producing iSAD components. The overall approach is as follows:

1. Design required IPMC actuator via ISAMCO software.
2. Design required IPMC sensor via ISAMCO software.
3. Determine portion of actuator's performance that is needed to deflect the sensor via beam theory.
4. If performance degradation is too severe and not acceptable, repeat steps 1-3 until satisfied.

Taking a simplistic view, it seems obvious that when a sensor strip needs to be added to an IPMC actuator, the sizing of the actuator should just be increased to accommodate the parasitic force required to deflect the sensor strip. However this may not be easily possible due to the small output forces that can be generated by IPMCs, so it is important to consider the design of iSADs in the integrated manner described above to determine a suitable algorithm.

The final algorithm developed can be used to size the sensor-actuator pair in a sensible manner and to reduce the interaction effects between the actuator and sensor to an acceptable level. Beam theory has been used in the design procedure in a straightforward manner because each IPMC strip can be viewed as a simple beam vibrating as a cantilever. The Euler-Bernoulli (E-B) equation for the deflection of such a beam can be used to determine the force required for an IPMC strip of length l to cause a specific deflection (although this can be used to calculate the deflection at any point along the beam, we assume this is measured at the end point for convenience). It is sensible that in determining good design rules the actuator IPMC has to be sized to provide sufficient force for "real actuation" and not simply to deflect the sensor IPMCs. Hence a sufficient safety margin must be introduced to size the actuator and sensor elements in a holistic iSAD design approach.

From an engineering perspective we can assume that the sensor IPMC should not "consume" more than 10% of the actuator's force during normal operation while providing a reasonable output signal.

2 Design of iSAD Devices

In this section we summarise the methodology for analysing and modelling of the integrated sensor-actuator devices. In order to do this we first consider the IPMC actuator model. This can be viewed as a simple single-input / single-output system as shown in figure 1. The input is the applied voltage $v(t)$ and the output can be either the force generated $g(t)$ or the deflection of the IPMC strip $d(t)$ (in some cases both force and deflection may be necessary but for convenience we assume only one output is needed here).

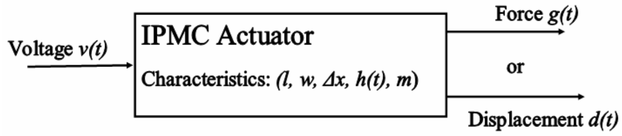


Figure 1: Block diagram of an IPMC actuator.

The work carried out in the EC ISAMCO project [8] has shown that the model for the IPMC actuator is given by $d_a(t)=g_1(w,l,h(t),m,\Delta x,v(t))$ where $g_1(\cdot)$ is a function of the variables inside the brackets. These variables depend on the characteristics of the IPMC strip used to make the actuator, namely, w is the width, l is the length, h is the hydration level, m is the type of material used and Δx is the thickness of the strip. Also the "a" subscript represents the fact that the IPMC is being used as an actuator. A similar equation holds for the force generated by the actuator, namely, $f(t)=g_2(w,l,h(t),m,\Delta x,v(t))$. The IPMC modelling details are presented in [10] where it can be seen that a high order non-linear time varying model is needed to adequately represent the behaviour. For convenience this model can be simplified to $d_a(t)=K_1(t)v(t)$ and even to $d_a(t)=K_a v(t)$ if the time varying aspects of the IPMC strip are ignored. In this way a simple constant of proportionality can be used to determine the output deflection of the IPMC actuator for a particular situation.

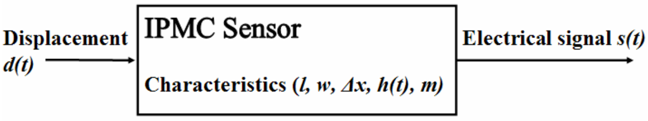


Figure 2: Block diagram of an IPMC sensor.

The IPMC sensor model has also been modelled within the ISAMCO project [8] and an appropriate block diagram is shown in figure 2. In a similar way to the case above, the single-input / single-output model to determine the electrical signal obtained from the IPMC sensor is given by $s(t)=g_3(w,l,h(t),m,\Delta x,d_s(t))$, where $s(t)$ represents the output signal and the "s" subscript represents the fact that the IPMC is being used as a sensor. This model can be simplified to $s(t)=K_2(t)d_s(t)$ and $s(t)=K_s d_s(t)$ when the time varying aspects are ignored. In this way a simple constant of proportionality can be used to determine the output signal from the IPMC sensor as it is actively deflected.

In order to determine a design methodology for integrated sensor-actuator devices, the two single-input / single-output models presented above need to be combined so that we have a two-input / two-output model. It is straightforward to see that the model for this situation is as shown in figure 3, which gives the multivariable model for the integrated sensor-actuator IPMC device as:

$$\begin{bmatrix} d_a(t) \\ s_s(t) \end{bmatrix} = \begin{bmatrix} K_a & C_{as} \\ C_{sa} & K_s \end{bmatrix} \begin{bmatrix} v_a(t) \\ d_s(t) \end{bmatrix}.$$

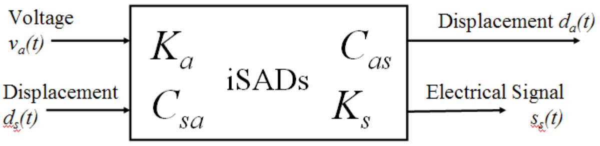


Figure 3: Block diagram of an IPMC iSAD device.

It is clear that when the cross coupling terms C_{as} and C_{sa} are non-zero, they will affect the performance of the integrated sensor-actuator device and so it is not possible to design iSADs to the required specification by designing the sensor and actuation elements separately.

As the sensor and actuator are physically connected at their tips and have equal lengths, the two end-point deflections are identical, that is, $d_s(t) = d_a(t)$. Hence we have:

$$\begin{bmatrix} d_s(t) \\ s_s(t) \end{bmatrix} = \begin{bmatrix} K_a & C_{as} \\ C_{sa} & K_s \end{bmatrix} \begin{bmatrix} v_a(t) \\ d_s(t) \end{bmatrix}$$

and the equation for d_s gives the following:

$$\begin{aligned} d_s(t) &= K_a v_a(t) + C_{as} d_s(t) \\ &= \frac{K_a v_a(t)}{1 - C_{as}} = d_a(t) \end{aligned}.$$

Therefore it can be seen that the actuator deflection changes from $d_a(t) = K_a v_a(t)$ when only an IPMC actuator is designed to $d_s(t) = d_a(t) = \frac{K_a v_a(t)}{1 - C_{as}}$ when an iSAD is built. It is clear that we need C_{as} to be as small as possible so that the actuator deflection is affected as little as possible due to the addition of the sensor IPMC.

It is also plausible to assume that $C_{sa} = 0$, because the electrical signal generated will be due entirely to the deflection of the overall sensor-actuator strip and the sensor IPMC will be insulated from the applied voltage. Hence the model for designing integrated IPMC sensor-actuator devices when the two strips have equal length and identical end points is given by: $\begin{bmatrix} d_s(t) \\ s_s(t) \end{bmatrix} = \begin{bmatrix} K_a & C_{as} \\ 0 & K_s \end{bmatrix} \begin{bmatrix} v_a(t) \\ d_s(t) \end{bmatrix}$ with C_{as} designed to be "as small as possible" so that the effect of the sensing IPMC is small on the iSAD element.

2.1 Beam Theory for IPMCs

The design aspects can be investigated by assuming that the motions of IPMC strips can be approximated by flat beams to which standard beam theory [7] can be applied. It is well known that the Euler-Bernoulli equation gives the deflection of a flat beam of length l as $d = \frac{Fl^3}{3EI}$, where F is the force, E is Young's modulus, and the moment of inertia, $I = \frac{w(\Delta x)^3}{12}$.

Rearranging this we get $F = \frac{Ew(\Delta x)^3}{4l^3} d$ where Δx is the thickness and w is the width of the beam.

It is straightforward to see that the force needed to deflect the sensing IPMC by some specified amount can be reduced if the width and thickness of the strip is reduced and/or its length increased. Another approach that can be adopted is to oversize the actuator as stated above so that it can perform as required with the extra load due to the integration of the IPMC sensor. However, this option may not be available.

The strip dimensions affect the bending force in the following way:

- beam width – linear variation
- beam thickness – cube law variation
- beam length – inverse cube law variation

From a design point of view, varying the beam thickness and length will have a more pronounced effect on the bending force, hence these should be used in the first approach when designing iSADs.

Varying strip length is easy because IPMCs are available in small sheets and the required strip sizes can be simply cut out. Varying strip thickness is not so straightforward, hence IPMC strips of various thicknesses are required so that thick ones can be used for actuators and thinner ones for sensors. At present only a limited range of thicknesses is available as stated above and thus the design of iSADs needs to focus on increasing strip length and reducing strip width to reduce the force to required levels.

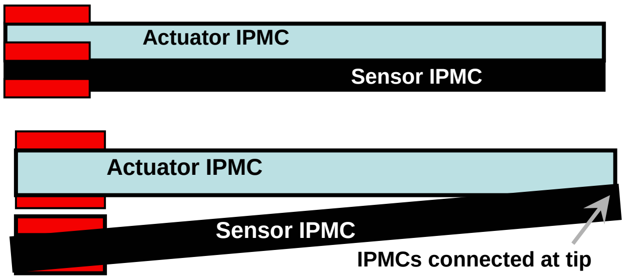


Figure 4: Configurations of Case 1 iSAD devices.

Varying the IPMC strip dimensions in this way leads to several configurations of iSAD devices, e.g. those with equal length and width strips but different thicknesses, or equal length and thickness strips but different widths, etc. We will consider two cases here.

2.2 Case 1: IPMCs with Equal l and Δx but Different w

A few configurations for this setup are shown in figure 4, where the lengths and thickness of the actuator and sensor IPMCs strips are the same but the widths are different (not shown). From above the force is given by $F = \frac{Ew(\Delta x)^3}{4l^3} d$. As l and Δx are fixed (equal to the dimensions of the IPMC actuator), the only parameter varied in this case in designing the IPMC sensor is reducing its width to reduce the force. As stated above this has a linear effect only, which may be insufficient to give the required design flexibility. From a generic viewpoint, increasing the length has a greater effect (cubic order) than reducing the width in reducing the force needed to move the sensor IPMC. This leads to the Case 2 type iSADs.

2.3 Case 2: IPMCs with Equal Δx and w but Different l

The discussion above for Case 1 iSADS has indicated that changing IPMC length has a more significant effect in varying the bending force (inverse cubic order); hence increasing strip length will reduce the force by a third order. However the configurations of iSADs with different lengths of actuator and sensor require specific design to suit the application so that the two IPMCs can move as required. Figure 5 shows two configurations.

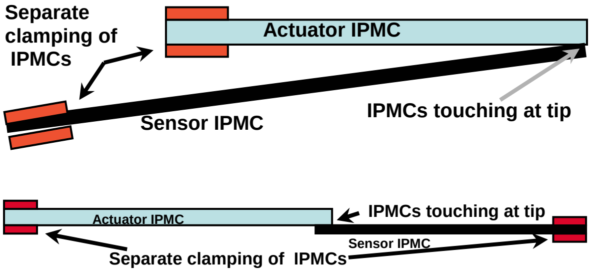


Figure 5: Configurations of Case 2 iSAD devices.

Separate clamping for the sensor and actuator elements is needed but this leads to a more profound capability of reducing the bending force needed by the sensing IPMC.

3 Experimental Validation

3.1 Test Rigs

The design procedure is experimentally validated by building and testing individual IPMC sensor and actuator strips as well as integrated sensor-actuator devices (iSADs). A number of test rigs have been developed for this purpose and they are briefly presented here.

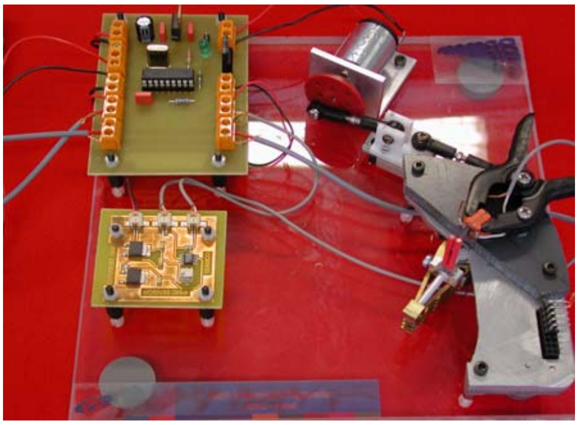


Figure 6: IPMC sensor test rig.

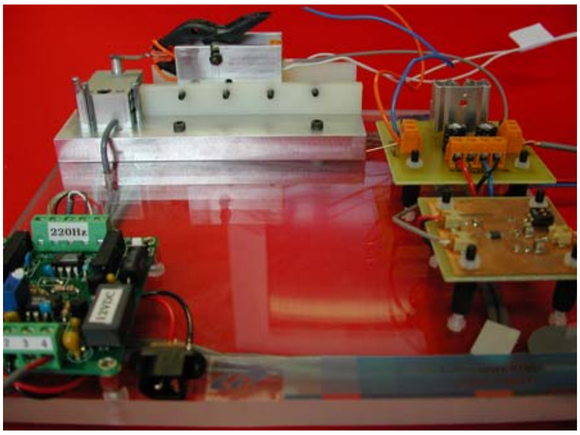


Figure 7: IPMC actuator test rig.

Figure 6 shows the sensor test rig, which allows an IPMC to be clamped at one end to provide electrical contacts, whilst the other end is attached to an oscillating arm. The oscillating arm is shown in the lower right corner of the photograph. The arm is attached to a slider-crank mechanism whose speed is controlled using the electronics in the top left of the photo. The signal conditioning circuitry of the sensor is shown in the lower left of the photo.

The actuator test rig is shown in figure 7. The IPMC can be seen in the top left of the photo. One end is clamped into the clamp to provide electrical contacts, whilst the free end is positioned on a force sensor. The IPMC is powered using the electronics on the right hand side of the figure. Applying a voltage to the IPMC causes it to bend, pushing down on the force sensor. The conditioning circuit for the force sensor is shown in the lower left of the photo.

These test rigs were used to monitor and verify the effect of integrating an IPMC sensor strip onto an actuator in a quantifiable way, based on the design approach and examples presented in the previous sections. It is conceivable that different methods of attaching the sensor strip will need to be investigated to minimise the impact of integration.

3.2 Validation of Results

The theoretical results produced by the iSADs design approach have been validated by building and testing actual IPMC strips. An example of an iSAD device design and experimental validation procedure is given here.

3.2.1 Individual IPMC Actuator Design

1. IPMC actuator force required = 3mN (say)
2. ISAMCO software gives the result that a Nafion 115 Li IPMC of length=15mm, width=5.2mm is required to form the actuator. The software predicts a deflection of 2mm and an actuation force of 3mN.

3.2.2 Individual IPMC Sensor Design

1. Use sensor deflection of 2mm and that a sensor output of 0.5μA (say) is required.
2. ISAMCO software gives the result that a Nafion 117 Na IPMC of length=15mm, width=3.2mm is required.

3.2.3 Assessment of Designed iSADs

1. Beam theory gives the force needed to deflect the IPMC sensor designed by 2mm as 0.9mN.
2. This parasitic force is 30% of the actuator's force, which is felt to be excessive (if we assume that 10% is what we are aiming for), and hence the design needs to be improved. For this we can go back to redesign both the actuator and the sensor, but for convenience here we focus on redesigning only the sensor so that it needs a lower bending force.
3. We repeat the IPMC sensor design by assuming a deflection of 2mm and that it outputs a lower signal of 0.2μA (say). In this case, the ISAMCO software gives the result that a Nafion 117 Na IPMC of length=15mm, width=1.2mm is required.
4. Beam theory in this case gives the force needed to deflect the improved IPMC sensor designed by 2mm as 0.34mN.
5. This sensor bending force for this second sensor is 11.3% which is felt to be low enough for the 3mN actuator and so the design has been completed.

The following iSAD has therefore been designed to give an actuation force of 3mN and a defection of 2mm:

- Actuator: Nafion 115 Li IPMC of length=15mm, width=5.2mm
- Sensor: Nafion 117 Na IPMC of length=15mm, width=1.2mm

The above data is used to cut and shape the two IPMCs for experimental validation of the iSAD designed. The experimental results use the rigs that have been developed for the IPMC research. These latest results are presented in the next section.

3.3 Experimental Validation

The full experimental validation of the design methodology is current under progress but the following initial results have been achieved:

Actuator tests: Two IPMC actuators have been produced. A force of 3.4mN has been measured in both actuators and the following deflections have been measured:

- Actuator 1: 2-7.5mm
- Actuator 2: 1-5mm

Sensor tests: Three sensors have been produced. For a deflection of 2mm these were found to produce output signals of 1μA, 0.75μA and 0.5μA.

The results are in broad agreement with what is predicted by the theoretical models. This gives us confidence to continue the research. However it is clear that there is a wide variation in the experimental results obtained; this is not a surprise as the methods of manufacturing IPMC strips are rather crude and prone to considerable human error. The IPMC elements are extremely small, delicate and fiddly to handle and the measurements have to be carried out manually. This introduces considerable errors in the quality of the strips produced. Hence the work needs to be repeated several times to reduce these human errors of measuring, cutting and handling the small IPMC strips in a reliable and repeatable manner. It is clear that specialised tools may also be needed.

4 Walking Robot Application

In this section we present the use of iSAD elements in realising miniaturised walking robots. Work is currently underway for this and only the initial results are included here.

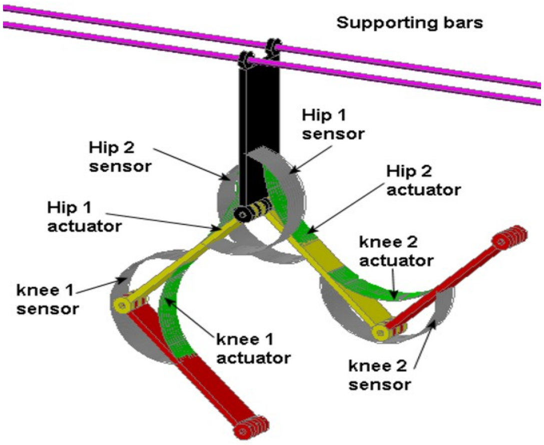


Figure 8: Concept drawing of IPMC walking robot.

The iSAD elements are to be used as active elements in realising a walking system and precise motion control of the strips is needed in order that the system is able to move itself in a walking manner. Precise trajectory following in this way allows us to perform different walking gaits. A concept diagram for the walking system is shown in figure 8 which has been designed to allow the IPMC actuator to move the walking system.

4.1 Modelling and Control of iSADs

In order to control the behaviour of systems comprising iSAD devices it is necessary to model IPMC actuators and sensors. A step response of the IPMC actuators used in our research is shown in figure 9; the response exhibits an interesting behaviour in that the IPMC strip deflects quite rapidly initially in direct response to the step input but is then seen to slowly relax to its original state, even though the step input signal is still present. In other words, the IPMC strip is unable to maintain its DC position. This behaviour indicates that such an IPMC cannot be used for maintaining a stationary position, but actuation for a walking system could be possible if the dynamics of the walking is kept within the “fast dynamics” of the IPMC.

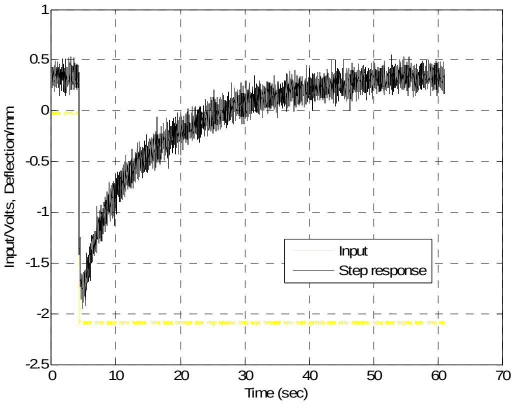


Figure 9: Step response of IPMC actuator.

Pseudo random binary signal excitation experiments are currently being carried out on IPMC actuators and sensor strips to enable good dynamic models of iSAD devices to be obtained. These models are then expected to be used for motion control of the walking links used in the robot depicted in figure 8.

5 Conclusions

The paper has presented a design methodology to produce IPMC integrated sensor-actuator devices so that feedback of the position of the IPMC actuator can be provided. The ISAMCO models and design software is used to develop a methodology to allow integrated sensor-actuator devices to be produced to the desired specifications. The theoretical results are being validated experimentally by using them to build and test the iSAD units. This work is continuing to design and validate the results on a miniaturised walking robot. The results obtained to date indicate that various configurations of iSAD devices can be designed. For good designs, it is necessary that sensor IPMC strips are longer and thinner than the actuator IPMC strips so that parasitic forces are reduced as much as possible.

6 Acknowledgements

The authors would like to acknowledge the work carried out by colleagues within the ISAMCO project and the valuable suggestions made.

7 References

[1] H. Tamagawa, K. Yagasaki and F. Nogata, “Mechanical characteristics of ionic polymer-metal composite in the process of self-bending”, *Journal of Applied Physics*, 92(12), pp 7614-7618 (2002).

[2] M. Shahinpoor and K.J. Kim, “Novel ionic polymer-metal composites equipped with physically loaded particulate electrodes as biomimetic sensors, actuators and artificial muscles”, *Sensors and Actuators A* 96(1), pp 125-132 (2002).

[3] Z. Chen, X. Tan and M. Shahinpoor, “Quasi-static positioning of ionic polymers-metal composite (IPMC) actuators”, *Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, CA, pp 60-65 (2005).

[4] M. Shahinpoor and K.J. Kim, “Ionic polymer-metal composites IV: industrial and medical applications”, *Smart Materials and Structures*, 14, pp 197-214 (2005).

[5] M. Shahinpoor and K.J. Kim, “Ionic polymer–metal composites III: modelling and simulation as biomimetic sensors, actuators, transducers, and artificial muscles”, *Smart Materials and Structures*, 13, pp 1362-1388 (2004).

[6] G.S. Virk, D.R. Harvey, I. Chochlidakis and S. Dogramadzi, “Design of integrated sensor-actuator ionic polymer-metal devices”, *Proceedings 9th International Conference on Climbing and Walking Robots (CLAWAR’06)*, Brussels, pp 706-711 (2006).

[7] S.S. Rao, *Mechanical Vibrations* (4th edition), Prentice Hall, Upper Saddle River, N.J. / London (2004).

[8] P. Arena, C. Bonomo, L. Fortuna, P. Giannone, S. Graziani and S. Strazzeri, *ISAMCO Deliverable 3.2: The Electromechanical Properties of IPMC Materials*, EC Contract #NMP2-CT-2003-505275, Department of Electrical, Electronic and Systems Engineering, University of Catania, Italy (2005).

Force-Displacement Modelling of Soft Tissue-Tool Interactions using Circular Indentors during Indentation Test

Talal Aljaafreh[1], Yahya Zweiri[2], Lakmal Seneviratne[1] and Kaspar Alothoefer[1]
[1]Kings College London, Mechanical Engineering Department,
London, UK
[2]Mutah University, Mechanical Engineering Department,
Karak, Jordan
talal.al-ja'afreh@kcl.ac.uk, yhzweiri@mutah.edu.jo, lakmal.seneviratne@kcl.ac.uk,
kaspar.althoefer@kcl.ac.uk

Abstract

Force-displacement characteristics prediction for soft tissue-tool interaction is important for its many applications in medical field such as accurate estimation of feedback forces, palpation simulation, disease diagnosis and robotic assistance. A new indentation model, motivated by a force-displacement soil-tool model, is presented in this paper. A six degree of freedom robot manipulator and a data acquisition system are used to collect data from a lamb liver sample to validate the indentation model. Measured data of force versus tool displacement for the soft tissue sample under indentation test conditions are presented and compared with the forces predicted by the model, showing very good agreement. The proposed model is a relatively simple algorithm with low computational cost. Most indentation models proposed to date deal with specific tools; the proposed method is more generic and applicable to a wider range of tool sizes.

Keywords: Soft tissue properties, Bekker model, indentation testing

1 Introduction and Related Work

Soft tissue indentation modelling is an important problem, dealing with predicting force-displacement characteristics. Soft tissue indentation models are required for the modelling and control of surgical tool interactions during minimally invasive surgery (MIS) and for the clinical palpation of patients to measure tissue firmness and stiffness. Such models can be also used to detect and diagnose tumours; every year about 8400 people are diagnosed in the UK with liver and kidney cancers [1]. Characterisation of soft tissue such as the heel pad would assist in the development of athletic and orthopaedic shoes. Residual limb tissue (soft tissue remaining after an amputation) and plantar soft tissue of patients with diabetes can be assessed for optimal fitting of prosthetic and orthotic devices. Determining tissue mechanics is also important for finite element modelling of biomechanical systems, such as the foot [2]. Skin harvesting to cover burns [3], grasping and manipulation of soft organs [4] also depend on soft tissue properties. Considerable work has been carried out to predict the nonlinear behaviour of soft tissue.

Many researchers have developed a number of useful empirical formulae to predict the force-displacement characteristics of soft tissue [5-7]. Okamura and Simon [5] used a second order polynomial function to fit experimental data for bovine liver. Ahmadian and Asadi [6] proposed two empirical equations to predict uniaxial force-displacement characteristics, for two categories of soft tissue. Han et al. [7] presented an exponential equation to describe deformations in human breasts, extending earlier work proposed by Fung [8]. Most of these researchers employed simple data fitting techniques to predict force variation for a specific tool size. Another group of researchers used complex mathematical models to model and simulate soft tissue behaviour [9-11]. For example, Tönük et al. [9] used nonlinear finite-element analysis to simulate force-displacement indentation behaviour of residual limb tissue. Guillaune et al. [10] proposed a new FE model for soft tissue deformation estimation based on non-linear elasticity and anisotropic behaviour. Yongmin et al. [11] presented a new methodology for predicting the deformations of soft objects, based on analogy between heat conduction and elastic deformation. However, most of the methods these researchers have used are complicated and have high computational cost. In general, it is noted that most of the research proposed above uses either FE modelling or empirically based techniques to predict variations of the indentation force during soft tissue deformation. It is noted that FE models are time consuming and unsuitable for real-time applications. Empirical models overcome this drawback, but are derived for specific cases, such as a given tool size, and have to be tuned to cope with different operating conditions. It is also noted that most force-displacement models proposed for soft tissue are based on non-linear elasticity and ignore viscoelastic properties. This is because in many applications, the load can be approximated by a quasi-static function. In medical applications such as rehabilitation, palpation and manipulation of organs, accurate estimation of the indentation force is required.

This paper develops a new methodology for estimating soft tissue indentation forces, motivated by a force-displacement soil-tool model proposed by Bekker [12]. Most indentation models proposed to date deal with specific tool sizes; the proposed method is more generic and applicable to a wider range of tools.

This paper is organised as follows: after the introduction, the Bekker model and the modified Bekker model are presented in section 2. Parameter estimation and experimental setup are presented in section 3. Results are analysed in section 4 and conclusions are drawn in section 5.

2 Bekker Model for Soil Sinkage and its Adaptation to the Force-Displacement Estimation of Soft Tissue

The basic principles for describing soil-vehicle force-displacement characteristics were first proposed by Bekker [15]:

$$P = (\frac{k_c}{d} + k_f)z^n \qquad (1)$$

Equation (1) gives the pressure-sinkage relationship of a stationary vehicle in homogenous soil, where z is the vehicle sinkage, P is the ground pressure generated by the vehicle, d is the contact patch diameter, Kc is the cohesive modulus, $K\phi$ is the internal friction modulus and n is the exponent of terrain deformation. Parameters Kc, $K\phi$ and n are constant physical properties of the soil, determined empirically.

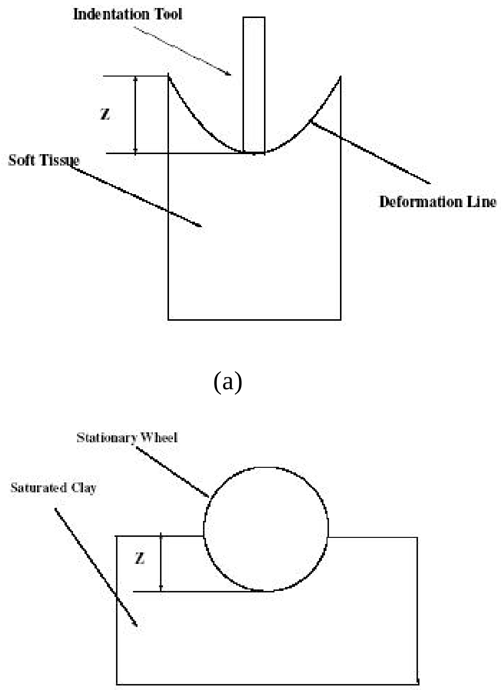
In this paper, an analogy based on the Bekker model is used to estimate the forces resulting from an indentation test carried out on soft tissue samples. The motivations for this approach are: (a) the physical analogy between soft tissue indentation and a stationary vehicle sinking in a homogenous terrain (i.e. vertical displacement due to force loading), figure 1; and (b) the Bekker model includes the vehicle contact patch, which can be equivalent to tool size, and hence the parameters Kc, $K\phi$ and n need not be tuned each time the tool size changes. Employing this analogy, the soil is replaced by a soft tissue sample and the vehicle wheel is replaced by the indentation tool, figure 1. It is noted that equation (1) is derived based on a static equilibrium analysis of soil deformation. It gives the pressure-displacement relationship between a general vehicle and soil, as a function of soil properties. The main feature of this paper is the extension of the functional form of equation (1) to soft tissue deformation modelling. Since $P = \frac{F}{A}$, equation (1) can be reformulated for the force estimation with a circular indentor as follows:

$$F = \frac{p}{4}d^2(\frac{k_c}{d} + k_f)z^n \qquad (2)$$

where A is the cross-sectional area of the circular indentor and F is the force. However, in order to describe force variation for different tool sizes, equation (2) is modified as follows to cope with different tools:

$$F = \frac{p}{4}d^{2p}(\frac{k_c}{d^p} + k_f)z^n \qquad (3)$$

where p is the new exponent coefficient, aiming to account for tool size (diameter) variations. The main differences between the proposed model and past research [5-7] are: (a) the proposed model explicitly includes tool diameter, and hence has the potential to deal with tool diameter variations and (b) the exponent, p, is not restricted to being an integer (as is the case with vehicle-soil interactions). To the best of the authors' knowledge, this is the first time these two concepts have been applied to soft tissue modelling.



(a)

(b)

Figure 1: (a) Soft tissue-tool interaction schematic diagram; (b) soil-wheel interaction schematic diagram.

3 Parameter Estimation and Experimental Setup

To use equation (3), it is necessary to estimate the values of the four constants, *Kc*, *K*ϕ, *p* and *n*. However, *K*ϕ, the internal friction coefficient, can be neglected since soft tissue resembles to some degree frictionless clay, with both having a relatively high percentage of fluid content [12]. Hence, equation (3) can be rewritten as follows:

$$F = \frac{p}{4} d^{p} k_{c} z^{n} \tag{4}$$

The parameter estimation based on experimental data, table 1, shows that the value of *K*ϕ is negligible when compared to the value of *Kc* which supports the above assumption.

Table 1: Parameter values based on modified Bekker model.

Tested sample	*kc*	*k*ϕ	*p*	n
Liver	0.0019	9.8×10^{-7}	1.21	1.81

It is noted that equation (4) has the functional form:

$$F = Cd^{p} z^{n} \tag{5}$$

where *C* is constant that depends on tissue type, Cd^{p} is tissue stiffness and *n* is the exponent of tissue deformation (for *n* = 1, the tissue shows linear elastic response and for *n* > 1, it shows non-linear elastic response). It is noted that equation (5) is a subset of the polynomial modelling of soft tissue deformations mentioned in [5]. The other three parameters *Kc*, *p* and *n* are determined using three experimental measurements with two tool sizes. First, *F* is measured at *z*1 = 1 unit (i.e. 1 cm) for tool diameters *d*1 and *d*2. This gives:

$$p = \frac{\log_{10}(F|_{z_1,d_1} / F|_{z_1,d_2})}{\log_{10}(d_1 / d_2)} \tag{6}$$

and

$$k_c = 10^{[\log_{10}(F|_{z_1,d_2}) - \log_{10}(\frac{p}{4}) - p\log_{10}(d_2)]} \tag{7}$$

To find *n*, a third measurement is taken for *d*2 at *z*2:

$$n = \frac{\log_{10}(F|_{z_2,d_2}) - \log_{10}(\frac{p}{4}) - p\log_{10}(d_2) - \log_{10}(k_c)}{\log_{10}(z_2)} \tag{8}$$

where *z*2 ≠ 1 unit. The experimental protocol (procedure) adopted in this paper is similar to that used by many researchers [5, 13]. Lamb liver was chosen as the test material. The experiments were carried out on the test sample at room temperature using a robotised test rig using a Mitsubishi RV-6SL robot manipulator. A personal computer with data acquisition software (LABVIEW) was used to control the manipulator. A six-axis ATI Mini40 force/torque sensor was used to acquire the forces between the tool and the test sample (figure 2).

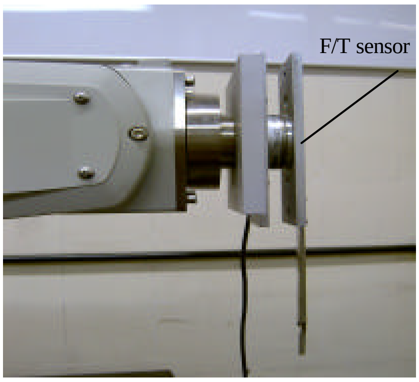


Figure 2: Mini40 force/torque sensor.

In the indentation procedure, each tool was first positioned 4 mm above the surface of the sample, moved down 20 mm, then stopped for 1 sec only, and finally the tool was retracted to its initial position. The tool speed is fixed at 5.54 mm/sec for all trials (at this speed, soft tissue shows quasi-static behaviour [13]). The liver sample weighed 385 grams (figure 3).

Figure 3: Lamb liver sample used in the experiments.

Due to the irregular thickness of the sample, and since the indentation force depends on thickness [7], the focus was on one sample area during the experiments, with thickness of 31 mm. It is noted that other boundary conditions such as shape and thickness variations are not investigated in this study.

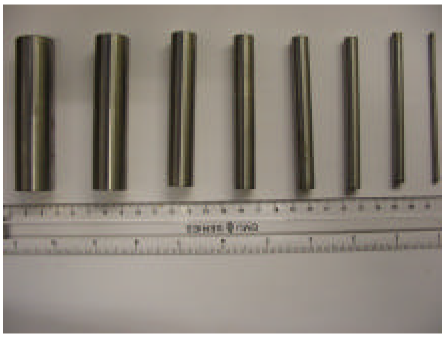


Figure 4: Different tools used in the experiments.

Eight flat round tools of 22, 20, 15, 12, 10, 8, 6 and 3 mm diameter were attached to the manipulator and used in the indentation tests (figure 4). The tool sizes chosen are similar to those mentioned by [2]. Three trials using each tool were carried out and then the average and standard deviations are presented.

4 Results and Analysis

In this section, predicted and measured force-displacement curves are given for eight tool sizes, using the liver sample. The predicted curves were obtained from the modified Bekker model, equation (4). The values of *Kc*, *K*ϕ, *p* and *n* were evaluated using tools d_1 = 3 mm and d_2 = 22 mm (table 1). Experimental results during the loading phase (i.e. compression) are presented and compared to results predicted by the model (figures 5 to 8).

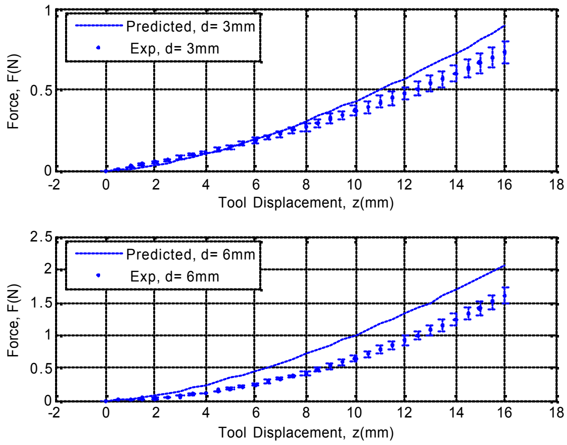


Figure 5: Comparison between measured data and predicted results using modified Bekker model for tool diameters *d* = 3 mm and *d* = 6 mm, *Kc* = 0.0019, *p* = 1.21 and *n* = 1.81

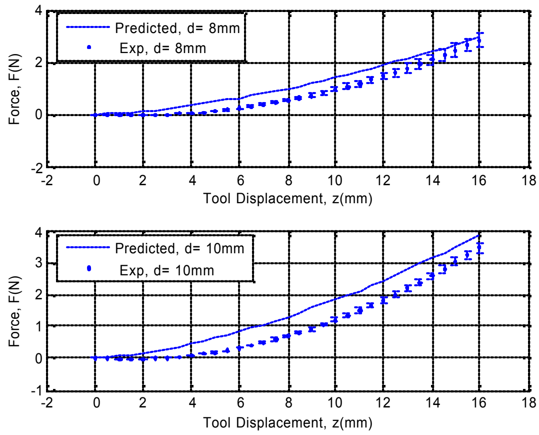


Figure 6: Comparison between measured data and predicted results using modified Bekker model for tool diameters *d* = 8 mm and *d* = 10 mm, *Kc* = 0.0019, *p* = 1.21 and *n* = 1.81

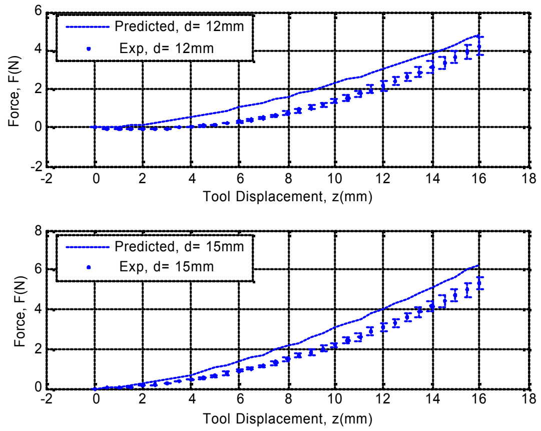


Figure 7: Comparison between measured data and predicted results using modified Bekker model for tool diameters *d* = 12 mm and *d* = 15 mm, *Kc* = 0.0019, *p* = 1.21 and *n* = 1.81

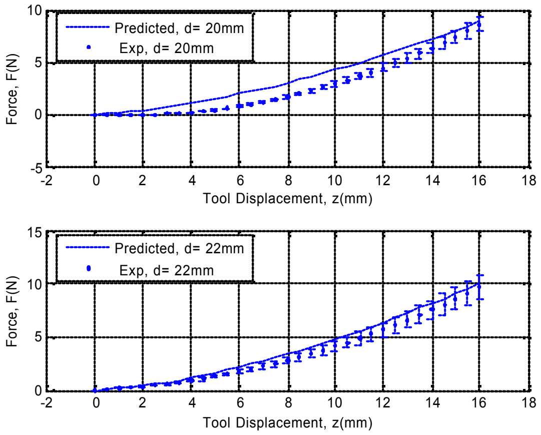


Figure 8: Comparison between measured data and predicted results using modified Bekker model for tool diameters *d* = 20 mm and *d* = 22mm, *Kc* = 0.0019, *p* = 1.21 and *n* = 1.81

To validate the proposed approach, the measured results are compared with the model results using the relative mean square error (RMSE) and cross correlation factor (CCF). All simulations were carried out using Matlab/Simulink and executed on a 3.4 GHz Pentium IV computer with 990 MB of RAM. The relative mean square error and cross correlation factor formulae are computed as follows [17]:

$$RMSE = 100 \times \frac{\sum_i (x_i - y_i)^2}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}} \qquad (9)$$

$$CC = 100 \times \frac{\sum_i (x_i \times y_i)^2}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}} \qquad (10)$$

where *xi* and *yi* are the measured and theoretical values of force respectively. Table 2 shows the RMSE and CCF for the eight tool sizes. The maximum RMSE value is 4.53%. CCF values indicate excellent agreement between predicted results and measured ones for all the tools tested.

Table 2: RMSE and CCF analysis using modified Bekker model.

d(mm)	RMSE (%)	CC (%)
22	0.07	99.99
20	1.82	99.65
15	2.33	99.49
12	3.58	99.30
10	3.62	99.51
8	2.31	99.64
6	4.53	99.85
3	1.53	99.53

The particular reason why the relative mean square error equation (9) is used is that, in addition to being used by other researchers [14], it gives the same potential (weight) to both the measured and predicted data; hence, it allows more accurate error estimation.

For further validation, the proposed model is compared to the models proposed by Okamura [5], Han [7] and Ahmadian [6] (equations (11), (12) and (13) respectively (Appendix A)). The proposed model has the significant advantage of tool size generalisation ability. Once the parameters Kc, p and n are found, the force for any tool size between the two tools used for tuning can be predicted. The parameters of the other models have to be tuned for each tool size. The Ahmadian model shows that its parameters at d_2 = 22 mm are g_1 = 47.45, g_2 = 6.82, m_1 = 28.80, m_2 = 16.10, n_1 = 7.65, n_2 = 8.48; but at d_1 = 3 mm these parameters change to g_1 = 63.57, g_2 = 0.54, m_1 = 40.18, m_2 = 15.66, n_1 = 10.01, n_2 = 9.10. Han's model shows that the parameters at d_2 = 22 mm are a = 1.935, b= 0.113; these change to a = 0.373, b = 0.068 at d_1 = 3 mm. Okamura's model also reveals the same trend as parameters change from k_1 = 0.100, k_2 = 0.032 at d = 22 mm to k_1 = 0.022, k_2 = 0.001 at d_1 = 3 mm. Table 3 shows a sample of parameter variation using the Okamura and Han models.

Table 3: Variation of Okamura and Han model parameters.

	Okamura model		Han model	
d(mm)	k_1	K_2	a	ß
22	0.100	0.032	1.935	0.113
20	0.001	0.033	0.773	0.159
15	0.032	0.022	0.983	0.124
12	0.001	0.020	0.483	0.155
10	0.003	0.015	0.452	0.144
8	0.001	0.010	0.173	0.183
6	0.024	0.005	0.392	0.104
3	0.022	0.001	0.373	0.068

The maximum errors using the Okamura, Han and Ahmadian models are 0.22%, 0.64% and 0.07% respectively. The errors are relatively low, since the parameters of these models are tuned for every tool. The proposed model shows comparable errors for the tools where the parameter estimation was conducted; the RMSE is 0.07% at d_2 = 22 mm and for d_1 = 3 mm, the RMSE is 1.53% (table 4).

Table 4: Comparison between various models in terms of RMSE (%). 1, 2, 3 are tuned at each tool size; MBM (modified Bekker model) is tuned at d = 22 mm and d = 3 mm.

d(mm)	MBM	Okamura Model (1)	Han Model (2)	Ahm. Model (3)
22	0.07	0.01	0.08	0.02
20	1.82	0.21	0.26	0.01
15	2.33	0.00	0.12	0.02
12	3.58	0.20	0.23	0.02
10	3.62	0.22	0.17	0.02
8	2.31	1.10	0.64	0.06
6	4.53	0.15	0.11	0.00
3	1.53	0.04	0.01	0.07

However, when these models are tuned at one tool and extended to predict the force variation for other tools, relatively high errors are found. Taking the Okamura model tuned at d = 10 mm, the RMSE is found to be 282% for d_1 = 3 mm and 101% for d_2 = 22 mm (table 5).

Table 5: RMSE (%) analysis using Okamura model tuned at d = 10 mm.

d(mm)	22	20	15	12	10	8	6	3
RMSE (%)	101	66	24	7	0.22	14	64	282

5 Conclusions

The paper presents a new model to predict the force-displacement characteristics between soft tissue and a circular indentor. It shows the validity of utilising the modified Bekker model to estimate forces occurring during the uniaxial indentation of soft tissue for various tool sizes. The study shows good agreement between measured and predicted results; the relative mean square error (RMSE) between predicted and measured data is low (less than 5%) and the cross correlation factor (CCF) shows very good matching between predicted and measured data. However, an error of up to 10% is accepted [14]. Furthermore, the main feature of the proposed model that it is more generic: once the modified Bekker model is tuned for two tools, other force-displacement curves for the tools between can be predicted, whereas other existing models require parameter tuning for each tool size.

6 Acknowledgments

The authors would like to thank A. Mendelssohn and S. Wootton for technical advice.

7 References

[1] Cancer Research UK Information Research Centre, http://info.cancerresearchuk.org/, visited 3/11/2005.

[2] J.W. Klenser, P.K. Commean, M.K. Hastings, D. Zou and M.J. Mueller, “Accuracy and reliability testing of a portable soft tissue indentor”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9, pp 232-240 (2001).

[3] G. Duchemin, P. Maillet, P. Poignet, E. Dombre and F. Pierrot, “A hybrid position/force control approach for indentation of deformation models of skin and underlying tissues”, *IEEE Transactions on Biomedical Engineering*, 52, pp 160-170 (2005).

[4] T. Gregory, C. Teeranoot, H. Tie, P.D. Jaydev and L. Alan, “Measuring grasping and cutting forces for reality-based haptic modeling”, *International Congress Series*, 1256, pp 794-800 (2003).

[5] A. Okamura and C. Simone, “Modeling of needle insertion forces for robot-assisted percutaneous therapy”, *Proceedings IEEE International Conference on Robotics and Automation*, Washington, D.C., pp 2085-2091 (2002).

[6] M.T. Ahmadian and A. Asadi Nikooyan, “Modeling and prediction of soft tissue directional stiffness using in vitro force-displacement data”, *International Journal of Science*, 15, pp 123-126 (2005).

[7] L. Han, J.A. Noble and M. Burcher, “A novel ultrasound indentation system for measuring biomechanics properties of in vivo soft tissue”, *Ultrasound in Medicine and Biology*, 29, pp 813-823 (2003).

[8] Y.C. Fung, *Biomechanics: Mechanical Properties of Living Tissues* (2nd edition), Springer-Verlag, Berlin (1993).

[9] E. Tönük and S. Barbara, “Nonlinear elastic material property estimation of lower extremity residual limb tissues”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11, pp 43-53 (2003).

[10] G. Picinbono, D. Herve and A. Nicholas, “Nonlinear anisotropic elasticity for real-time surgery simulation”, *Graphical Models*, 65, pp 305-321 (2003).

[11] Z. Yongmin, S. Bijan, A. Gursel and S. Julian, “A new methodology for deformable object simulation”, *Proceedings IEEE International Conference on Robotics and Automation*, Barcelona, pp 1914-1919 (2005).

[12] J.Y. Wong, *Theory of Ground Vehicles*, John Wiley and Sons, New York (2001).

[13] M. Heverly, P. Dupont and J. Triedman, “Trajectory optimization for dynamic needle insertion”, *Proceedings IEEE International Conference on Robotics and Automation*, Barcelona, pp 1658-1663 (2005).

[14] F. Mobasser and K. Hashtrudi-Zaad, “Hand force estimation using electromyography signals”, *Proceedings IEEE International Conference on Robotics and Automation*, Barcelona, pp 2642-2647 (2005).

Appendix A

The following models are used to predict forces resulting from tool-tissue interaction:

Okamura [5]:

$$F(z) = k_1 z + k_2 z^2 \qquad (11)$$

Han [7]:

$$F(z) = a\,(e^{bz} - 1) \qquad (12)$$

Ahmadian [6]:

$$F(z) = \sum_{i=1}^{2} g_i e^{-[\frac{m_i - z}{n_i}]^2} \qquad (13)$$

where *F* is the indentation force, *z* is the tool displacement and *k*2, *k*1, a, *ß*, *gi*, *mi* and *ni* are constants that are determined experimentally.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Smart Battery Charger for Soccer Robots: A Non-Contact Power Transfer Based Approach

S.D. Karunanayaka, V. Suri, S.C.Mukhopadhyay and G. Sen Gupta
Institute of Information Sciences and Technology,
Massey University, Palmerston North, New Zealand
s.c.mukhopadhyay@massey.ac.nz

Abstract

In recent years, robots and advancement in robotic automation technology have provided a platform that can be utilised to support a variety of services in commercial facilities, government and municipal offices. A variety of robots, including entertainment robots, are being developed for these purposes. In most cases, the robots rely on batteries for the supply of the driving power. With use in carrying out operations, the batteries get depleted and need to be charged again for continuation of smooth operation. This paper presents an overview of the theoretical and practical design considerations related to a smart charger for the batteries used in a soccer robot. It is based on a non-contact power transfer technology. Of particular interest are the theoretical and practical factors affecting the performance and hence the design of the charger.

Keywords: Soccer robot, smart charger, non-contact charging, flux, pickup, high permeability core

1 Introduction

Non-contact power transfer methods offer several advantages over power transfer by contact, such as avoidance of moveable cables and associated connectors, saving of time and effort in replacing discharged batteries and the ability to operate in hazardous environmental conditions. When using rechargeable batteries, the discharged batteries need to be removed from the robot and charged before reinstalling and using again. This creates a disruption to operation. Such general problems and inconveniences associated with the charging of batteries using traditional methods have led to the quest for a more convenient charging method.

The principle of operation of contactless power transfer relies on Faraday's law of electromagnetic induction. Research on different forms of contactless power transfer started very early and is very useful to those applications of power transfer where the use of conventional brush-and-conductor combinations, trailing cables or cable drums incur penalties [1]. The principle of contactless power transfer has been exploited in a service robot by Fujitsu [2], in which the robot has the ability to autonomously self-charge by means of high-density non-contact charging via electromagnetic induction. The non-contact power transfer is very similar to a transformer but the major differences are low magnetising inductance and high leakage inductance. Usually the devices which operate based on this principle are inefficient and have very poor voltage regulation [3]. In spite of these problems the usefulness of non-contact energy transfer in many environments is very high, especially where the relative motion between transmitter and receiver has clear disadvantages [4-9]. Design issues and different design strategies based on class C converters [10], sufficient exciting inductance and low leakage inductance at large air gaps [11] and a resonant converter with parallel capacitor of the secondary coil [12] are reported. The issue of noise radiation is also very important for non-contact power transfer. The geometry of the device can be designed so as to reduce noise [13]. Special design considerations have to be taken into account while using non-contact power transfer for supplying power to mobile loads such as electric vehicles [14 -16]. Contactless energy transmission has also been used to charge a battery used in an implanted medical device [17]. The intensity of the magnetic field necessary for charging a pacemaker battery has been computed. It has been estimated that to charge the battery used in a pacemaker, a 4 W generating station is required which can charge the battery within 3 hours.

In this paper we report the theoretical background and effects of some major design parameters on the performance of a smart battery charger, based on a non-contact power transfer method, used for charging the batteries used in a soccer robot. In most cases, the robot relies on batteries to supply driving power. With time the batteries get discharged and need to be charged for continuation of smooth operation.

2 Operating Principle of Non-Contact Charger

In its most elementary form, the charger consists of a power source (and associated circuitry), a track and conductor system that is used to carry high frequency alternating current and transfer power to a specially designed pickup system located on the device.

The three modules that comprise the smart charger include:

(i) Power supply and associated fast switching circuitry to generate a high frequency current to be carried by the conductor placed in the track.

(ii) Track and conductor to carry the current (and over which the pickup is positioned).

(iii) Pickup and associated circuitry involved in picking up induced voltage, rectification and charging.

A simple schematic representation of the complete set-up consisting of power circuit, track and conductor, pickup coil and rectifier for charging the battery is shown in figure 1. The primary side is based on an inverter which is operated at a high frequency of several tens of kilohertz. The operating frequency is selected based on several conditions such as the cost of the power circuit, amount of power transfer, radio interference to neighbouring systems, etc. The details of the different parts are provided later.

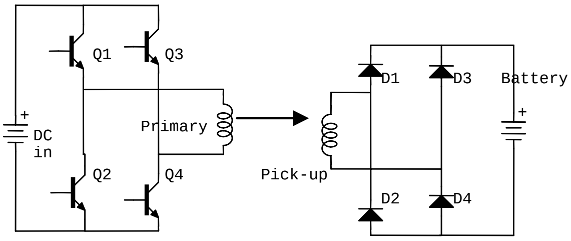


Figure 1: Schematic representation of non-contact charger.

Figure 2 represents the conductor carrying high frequency current and the core on which the pickup coil is wound.

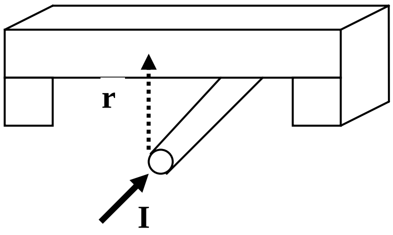


Figure 2: Representation of conductor and the core of the pickup coil.

The magnetic field intensity *H* created by the exciting current *I* at a distance *r* is given by:

$$H = \frac{I}{2\pi r} \tag{1}$$

The magnetic flux density *B* is given by:

$$B = \mu_0 H = \frac{\mu_0 I}{2\pi r} \tag{2}$$

Since the air gap is very large, the flux density inside the magnetic core will be decided by the magnetic boundary condition and will be very close to *B*.

The magnitude of the induced voltage *E* across the pickup coil is thus given by:

$$E = N\frac{d\phi}{dt} = NA\frac{dB}{dt} = 4NAfB_m = \frac{2NAf\mu_0 I_{peak}}{\pi r} \tag{3}$$

where *A* is the core area, *N* is the number of turns of the pickup coil, *f* is the frequency of operation and B_m is the peak value of flux density inside the core. It is assumed that the flux density changes from $+B_m$ to $-B_m$ in half of the switching time period *T*/2; *T* is the time period and *T* = 1/*f*; I_{peak} is the peak value of the exciting current.

For a typical situation with a pickup coil of 100 turns, core area of 5 mm x 5 mm, an operating frequency of 20 kHz and a peak current of 5 A, the induced voltage for the pickup system at a distance of 5 mm will be 0.628 V.

So in order to design an efficient battery charger, the influence of each and every parameter needs to be studied. Our study has been carried out by finite element analysis. Since the magnetic flux density is the main parameter which decides the amount of induced voltage, the effect of other parameters on magnetic flux density is studied.

3 Description of Soccer Robot and the Need for a Battery Charger

The smart battery charger discussed in this paper is for a soccer robot. The soccer robots and the playground on which they play soccer are shown in figure 3. The ground is approximately 2 m x 1.5 m wide for the 3-a-side competition. In other categories, each team may have 5 to 11 robots as players.

Figure 3: A robot soccer playground.

The position of the battery inside a robot is shown in figure 4. The battery compartment is very compact. When the batteries get discharged, they are taken out of the robot and connected to a charger. Usually the batteries are 9 V. With time there has been a great improvement in the quality of the batteries. For the same size the Ampere-Hour rating has increased. A few different types of batteries and a conventional charging unit to charge the batteries are shown in figure 5. The aim of this paper is to design and develop a charger which will make it possible to leave the batteries in the robot and charge it while the robot is on the soccer field.

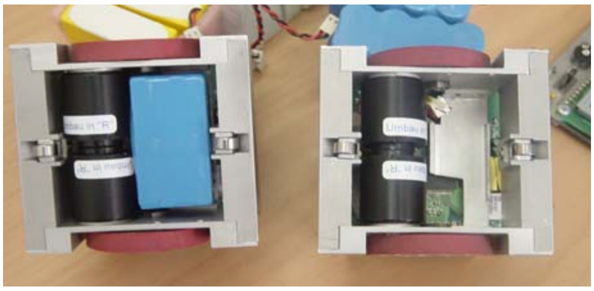


Figure 4: The battery inside a soccer robot.

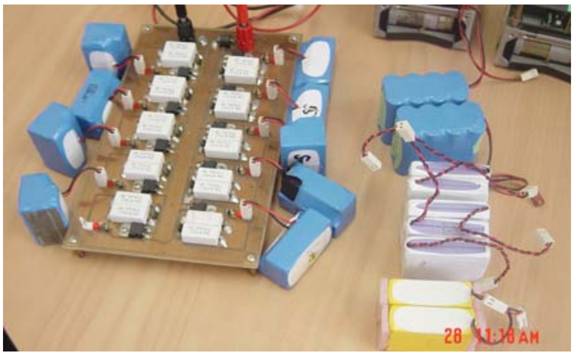


Figure 5: Different types of batteries and a conventional battery charger.

4 Finite Element Analysis of the Pickup Section

Due to the non-contact nature of its operation, there are many factors that affect the performance of the charger. These include both theoretical and practical considerations, including frequency of operation, high permeability of the pickup core, and the air gap between the track and the pickup coil. Figure 6 shows the 2-dimensional model of the pickup section and exciting coil of the charger.

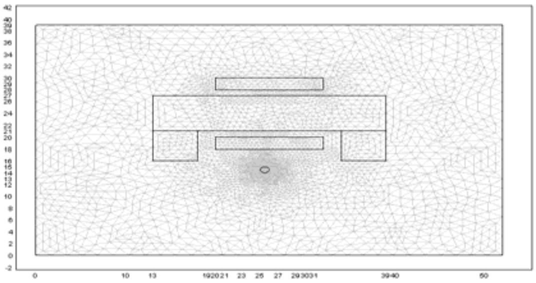


Figure 6: 2-D finite element model of the charger for field analysis.

By providing the necessary sub-domain settings and boundary conditions, the solution of the field distribution has been obtained. Figure 7 shows a typical field distribution of the model.

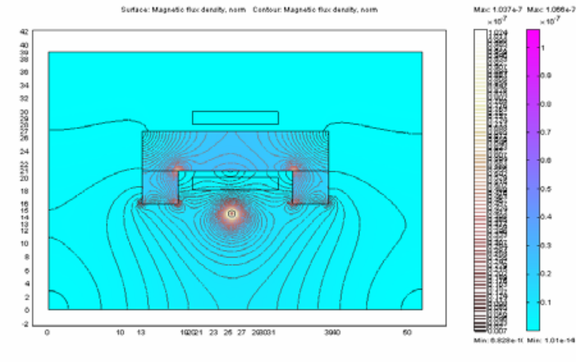

Surface: Magnetic flux density, norm Contour: Magnetic flux density, norm
Max: 1.037e-7
Max: 1.066e-7
×10⁻⁷
Min: 6.828e-11
Min: 1.01e-14

Figure 7: Solution of the field distribution.

4.1 Effect of Exciting Current

The exciting current is the source which creates the magnetic field. The magnetic flux density, as well the amount of induced voltage across the pickup coil, is directly proportional to the exciting current, as seen from equations (2) and (3). This has been verified by finite element analysis. By changing the input setting of the model, the field has been analysed for different currents. The variation of flux density inside the core as a function of current is shown in figure 8. It is seen that the variation of flux density is a linear function of current.

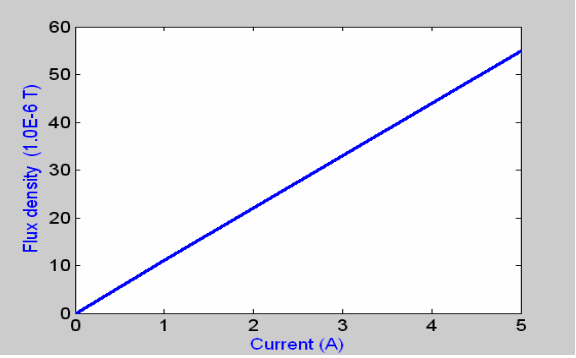


Figure 8: Variation of flux density with the magnitude of current

One way of increasing the induced voltage is by increasing the exciting current. This increases the size of the conductor and yet does not utilise the complete core geometry. Instead of using a thick conductor, it is better to design the track with multiple conductors, as shown in figure 9.

Based on this idea, the field distribution has been carried out for various configurations using 2 to 7 conductors. Figures 10, 11 and 12 show the solution of field distribution for the configurations using 3, 5 and 7 conductors respectively, carrying the same amount of current.

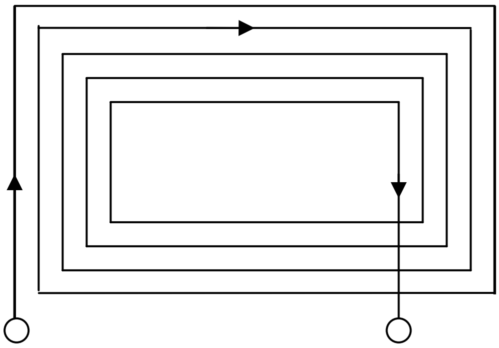


Figure 9: Multiple conductors for track.

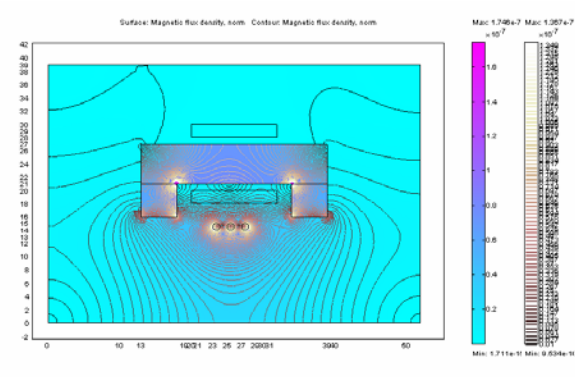


Figure 10: Field distribution with 3 conductors.

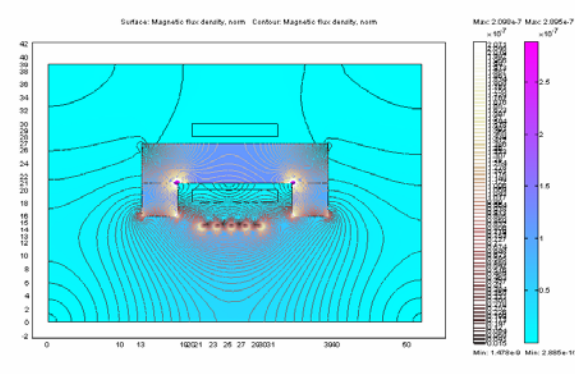


Figure 11: Field distribution with 5 conductors.

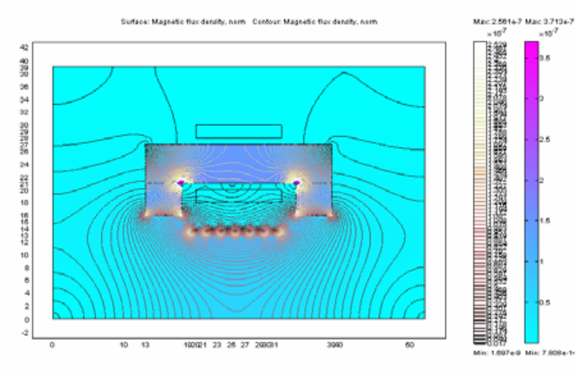


Figure 12: Field distribution with 7 conductors.

Figure 13 shows the variation of flux density as a function of the number of conductors while the current carried by the conductor remains same. It is seen that the flux density increases almost linearly with the number of conductors. This result is very similar to that shown in figure 8, but in this approach the size of the conductor need not be thick.

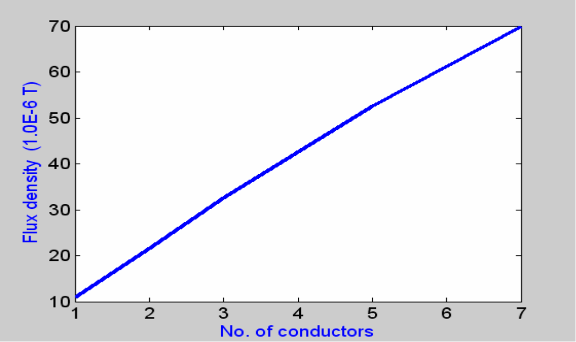


Figure 13: Variation of flux density with the number of conductors.

4.2 Effect of Permeability of Core

The core is a very important element of the charger as it not only provides support for winding the pickup coil, it also provides a low reluctance path for the magnetic flux. Apparently there is a case for using a high-permeability core, but for this particular case a high permeability core will not be of much benefit and will be very expensive. Figure 14 shows the variation of flux density as a function of the relative permeability of the core and it is seen that the increase of flux density with permeability is not at all significant.

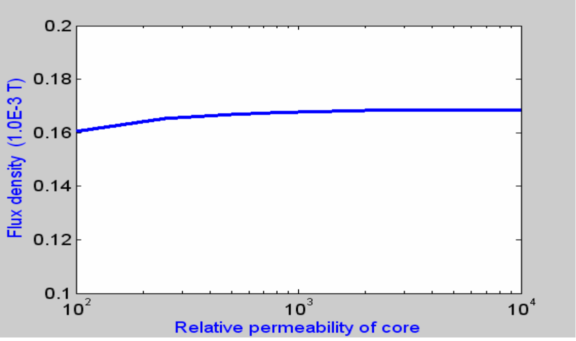


Figure 14: Variation of flux density with relative permeability of core.

4.3 Effect of Operating Frequency

The operating frequency is very important for designing a smart battery charger. Equation (3) shows that the induced voltage across the pickup coil is directly proportional to the frequency. An inverter with a very high frequency option will increase the induced voltage across the pickup coil but will make the system expensive. Figure 15 shows the variation of the magnetic flux density as a function of the operating frequency. It is seen that the magnetic flux density starts dropping when the frequency becomes more than 20 KHz. A suitable frequency, based on cost and other consideration, has to be chosen.

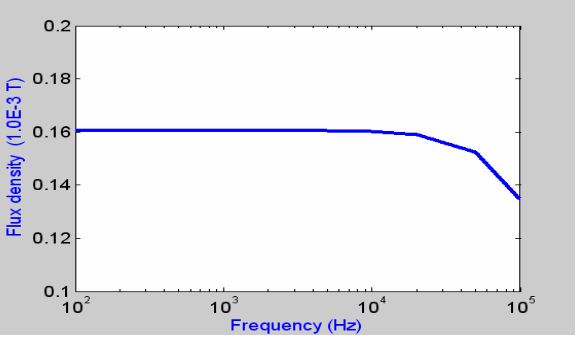


Figure 15: Variation of flux density with operating frequency.

4.4 Effect of Air Gap

The most important parameter for an efficient battery charger is the air gap. The air gap can decide the size of the charger. The magnetic flux density at any point is inversely proportional to the distance of the point from the conductor. From the performance point of view, the air gap should be as minimal as practicable. Sometimes it is not possible to have a very small air gap. For the soccer robot there should be a minimum clearance between the lower surface of the robot and the upper surface of the field. The minimum air gap also depends on the clearance of the upper layer of the soccer field from the tracks. A minimum value of 5 mm can be considered as a very good value for this design. Figures 16 and 17 show the solution of field distribution for air gaps of 0 and 5 mm respectively.

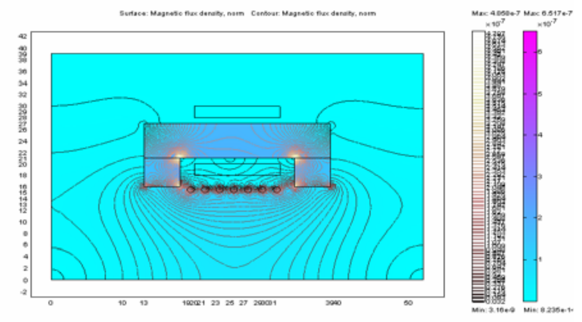

Surface: Magnetic flux density, norm Contour: Magnetic flux density, norm
Max: 4.658e-7 Max: 6.517e-7
Min: 3.16e-9 Min: 8.235e-1

Figure 16: Field distribution for air gap of 0 mm.

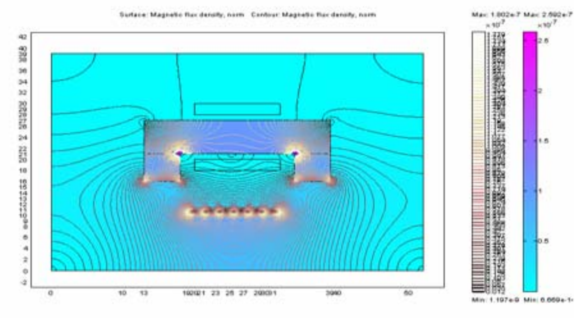

Surface: Magnetic flux density, norm Contour: Magnetic flux density, norm
Max: 1.802e-7 Max: 2.692e-7
Min: 1.197e-9 Min: 6.669e-1

Figure 17: Field distribution for air gap of 5 mm.

Figure 18 shows the variation of magnetic flux density as a function of the air gap distance. As expected, the magnitude of the flux density decreases with the increase of the air gap distance.

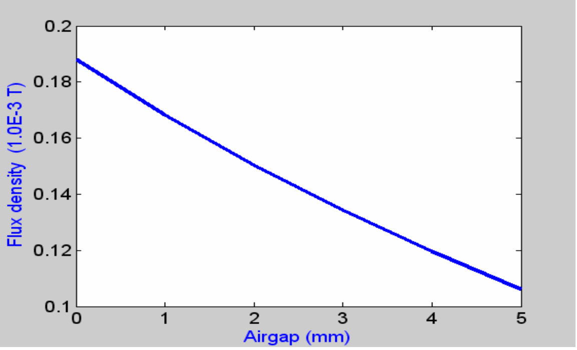


Figure 18: Variation of flux density with air gap.

5 Implementation and Future Plans

The full implementation of the smart charger will be realised for a set of autonomous robots playing a game of soccer and charging themselves without human intervention while the robots are idle. The soccer field will be provided with tracks parallel to the boundary walls and just under the upper surface. The robots can align themselves in such a way that the magnetic pickup is maximal while they are not playing. A typical pickup coil, as shown in figure 19, has been fabricated and experimented with. It provides satisfactory output voltage, but is quite large in size. Different options of reducing the size to accommodate it under the soccer robot are being investigated.

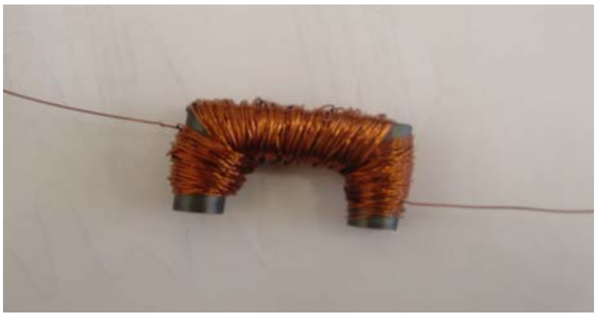


Figure 19: A typical pickup coil.

6 Conclusions

In this paper an overview has been provided on the design and performance of a smart charger, a novel implementation in charging technology for charging batteries used in soccer robots. It was determined that there is a need for smart chargers due to the disadvantages associated with traditional charging methods. Factors (both theoretical and practical) affecting the performance of the charger were identified and exploited in order to derive maximum power transfer. The fabrication of a charger is currently under consideration to verify the theory described in the earlier sections.

7 References

[1] E. Abel and S.M. Third, "Contactless power transfer: an exercise in topology", *IEEE Transactions on Magnetics*, 20(5), pp. 1813-1815 (1984).

[2] "Fujitsu Begins Limited Sales of Service Robot "enon" for Task Support in Offices and Commercial Establishments", http://www.fujitsu.com/global/news/pr/archives/month/2005/20050913-01.html, visited on 18 Sep 2006.

[3] D.A.G. Pedder, A.D. Brown and J.A. Skinner, "A contactless electrical energy transmission system", *IEEE Transactions on Industrial Electronics*, 46(1), pp 23-30 (1999).

[4] A. Esser and H.C. Skudelny, "A new approach to power supplies to robots", *IEEE Transactions on Industry Applications*, 27(5), pp 872-875 (1991).

[5] K.W. Klontz, D.M. Divan, D.W. Novotny and R.D. Lorenz, "Contactless power delivery system for mining applications", *IEEE Transactions on Industry Applications*, 31(1), pp 27-35 (1995).

[6] G.A. Covic and J.T. Boys, "Inductively coupled power transfer: a technological review", *JSAEM Studies in Applied Electromagnetics and Mechanics*, 15, pp 9-17 (2005).

[7] C. Wang, G.A. Covic and O.H. Stielau, "Power transfer capability and bifurcation phenomena of loosely coupled inductive power transfer systems", *IEEE Transactions on Industrial Electronics*, 51(1), pp 148-157 (2004).

[8] G.H.Y. Kung, D. Kacprzac, G.A. Covic and J.T. Boys, "A study of pick-up performance for inductively coupled power transfer systems", *JSAEM Studies in Applied Electromagnetics and Mechanics*, 15, pp 143-149 (2005).

[9] A.W. Kelley and W.R. Owens, "Connectorless power supply for an aircraft-passenger entertainment system", *IEEE Transactions on Power Electronics*, 4(3), pp 348-354 (1989).

[10] H. Sakamoto and K. Harada, "A novel converter for non-contact charging with electromagnetic coupling", *IEEE Transactions on Magnetics*, 29(6), pp 3228-3230 (1993).

[11] H. Sakamoto, K. Harada, S. Washimiya and K. Takehara, "Large air-gap coupler for inductive charger", *IEEE Transactions on Magnetics*, 35(5), pp 3526-3528 (1999).

[12] H. Abe, H. Sakamoto and K. Harada, "A noncontact charger using a resonant converter with parallel capacitor of the secondary coil", *IEEE Transactions on Industry Applications*, 36(2), pp 444-451 (2000).

[13] J. Murakami, F. Sato, T. Watanabe, H. Matsuki, S. Kikuchi, K. Harakawa and T. Satoh, "Consideration on cordless power station: contactless power transmission system", *IEEE Transactions on Magnetics*, 32(5), pp 5037-5039 (1996).

[14] F. Sato, J. Murakami, H. Matsuki, S. Kikuchi, K. Harakawa and T. Satoh, "Stable energy transmission to moving loads utilizing new CLPS", *IEEE Transactions on Magnetics*, 32(5), pp 5034-5036 (1996).

[15] F. Sato, J. Murakami, T. Suzuki, H. Matsuki, S. Kikuchi, K. Harakawa, H. Osada and K. Seki, "Contactless energy transmission to mobile loads by CLPS: test driving of an EV with starter batteries", *IEEE Transactions on Magnetics*, 33(5), pp 4203-4205 (1997).

[16] C.S. Wang, O.H. Stielau and G.A. Covic, "Design considerations for a contactless electric vehicle battery charger", *IEEE Transactions on Industrial Electronics*, 52(5), pp 1308-1314 (2005).

[17] S. Suzuki, T. Katane, H. Saotome, O. Saito, "A proposal of electric power generating system for implanted medical devices", *IEEE Transactions on Magnetics*, 35(5), pp 3586-3588 (1999).

The "Long March" from Labs to Real World Applications: The Contribution Made by Robot Soccer

Norbert Jesse
Department of Computer Science,
University of Dortmund, Germany
norbert.jesse@udo.edu

Abstract

This paper outlines basic challenges of robot soccer for research and its potentials for real world applications. After some remarks about the different FIRA leagues we comment the complexity of image processing and decision making in robot soccer and appreciate the relevance for robotics applications.

Keywords: robot soccer, Federation of International Robot-soccer Association, image processing, decision support, multi-agent systems

1 Introduction

Robot soccer provides excellent research opportunities in fields such as sensor fusion, image processing, decision making in uncertain situations or multi-agent systems. The work of two globally operating organisations, the RoboCup and the Federation of International Robosoccer Association (FIRA), is dedicated to the promotion of robot soccer, aiming at bringing together skilled researchers and students from different backgrounds to foster the development of autonomous multi-agent robotic systems. Both organisations propagate robot soccer for a variety of leagues. The author has been involved in FIRA for some years and, consequently, refers to this experience to discuss the challenges of robot soccer - and the difficulties in transferring results from the labs to real world applications.

2 Great Economic Expectations

The first robots were installed in 1961 by General Motors to string and stack hot cast irons. These early robot systems were inflexible numeric control machines, but since then robots have developed into advanced sensor-controlled actuation systems capable of far more complex activities.

Today, more than 900,000 robots are employed in industry, executing clearly defined tasks with high precision and speed. As computer technology and embedded systems in particular have matured, the wide-spread use of robots has expanded beyond the automotive sector to sectors such as assembly, die casting, plastic moulding, conveyer transfer, palletising, inspection, to just name a few. However, the number of industrial robots has increased only gradually.

Experts predict that intelligent, mobile robots enter our human world sooner than later. After all, the first generation of service robots is available: lawn-mowing, vacuum-cleaning and window-cleaning robots. While 21,500 units were counted at the end of 2002, the forecasts for 2005 had estimated 719,000 units and a sales volume of 3.6 billion US$ in industrial service robots in the fields of underwater exploration, cleaning, medicine, safety and defence.

Furthermore, the market for toy and entertainment robots is expected to exceed the one million mark, most of these "robots" in the low-cost range.

Figure 1: Robonova-I - programmable and with smooth dancing capabilities [1].

All these robots still offer a significant flexibility, quality or price benefit. Robonova-I (figure 1), for instance, is equipped with a MR-C3024 micro-controller, enabling the robot to perform surprisingly elastic movements with 17 joints. But robots will only become a ubiquitous commodity if further substantial scientific progress is being made. The obvious question is: how can research in the field of robot soccer interest young people in technology and contribute to the development of future robot generations?

3 The FIRA Leagues' Challenges

The Federation of International Robosoccer Association (FIRA) comprehends six categories [2]:

- Simulated Robot Soccer Tournament (SimuroSot) - the disembodied (simulation) league concentrating on research for multi-agent and learning strategies;

- KheperaSot - small robots concentrating on navigation and self-location with proximity sensors, wheel encoders and a linear camera turret;
- Robot Soccer Tournament (RoboSot) - completely autonomous robots with local vision and processing power;
- Humanoid Robot Soccer Tournament (HuroSo) - the league for autonomous, anthropomorphic-shaped robots;
- Micro Robot Soccer Tournament (MiroSot) - 5 or 11 high-speed agents focussing on the challenges of a soccer match,
- Nano Robot Soccer Tournament (NaroSot) - miniaturised mobile platform with high speed and acceleration (>8m/s2).

There are different perspectives in classifying these robots:

- virtual vs. physical robots,
- wheeled vs. legged robots,
- global vs. local vision based robots,
- fast vs. slow robots.

All leagues apply strict rules and regulations with growing complexity (between 10 and 30 pages long) which increase the scientific challenge and move the game towards the standards of real world soccer. The scientific state-of-the-art is being published in FIRA conference proceedings, for instance in [3] [4].

3.1 The Simulation League

Obviously, the simulation league ("virtual soccer") is an excellent and inexpensive test-bed to work on learning strategies. The assumption of rather simple kinematics, the lack of special physical phenomena respectively, helps to concentrate on basic questions, but the difficulties in transferring the results to physical robots still restrict the usability of the results. The path from a "proof of concept" to the competition between real robots is paved with many obstacles [5].

This "lack of portability" corresponds to the fact that in the other FIRA leagues learning strategies, computational intelligence methods (e.g. intelligent control) or sophisticated decision-making are used in rare cases, if at all. Currently, MiroSot soccer teams, for instance, still focus their efforts on low-level problems such as movement control and prediction, changing light conditions or collision detection.

3.2 Two Legged Robots

Humanoid robots are enjoying increasing popularity - as a challenging research tool and as a marketing instrument. Impressive examples of humanoid robots are Honda's Asimo and Sony's Quiro, neither of them commercially available. Most of the robots on the market lack vision sensors and autonomy - even if they are to a certain extent user-programmable.

The institutions participating in RoboCup or FIRA competitions fill this gap by developing programmable robots equipped with vision sensors and computational power, i.e. these robots are fully autonomous. There are two ways of providing robots with computing power: adding either single-board computers or pocket PCs [6]. Independent of the solution, the overall performance of these autonomous robots has improved in recent years, especially in terms of dynamic walking and navigation abilities.

The FIRA HuroSot competition rules pose considerable challenges to the robots. The exercises include:

- forward - backward dash, i.e. the robot has to move as quickly as possible from the start to the finishing line for a series of segments;
- penalty kick, i.e. the robot has to kick the ball, a robot from a different team acting as goal keeper;
- obstacle run, i.e. the robot has to cross the field as quickly as possible, navigating through a number of obstacles;
- lift and carry, i.e. equipped with a small basket, the robot has to walk repeatedly across an uneven terrain.

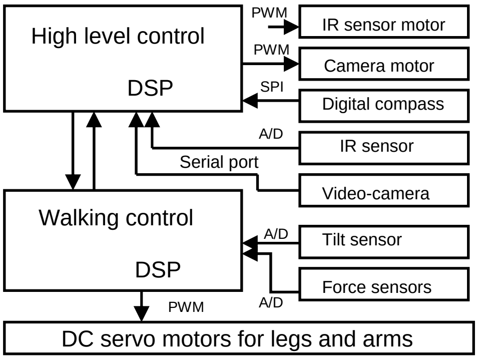


Figure 2: Control architecture of Robosapiens from Singapore.

Figure 2 illustrates the complex control architecture of Robosapiens, one of the outstanding humanoid robots participating in the FIRA competitions. Many of these exceptional robots have surprising abilities, but they still have a long way to go when it comes to the flexibility, balance control and agility of human motions. Nevertheless, RoboCup now involves more than one robot per team, and FIRA will follow soon by steering the development towards team behaviour.

Table 1: The FIRA leagues and some basic characteristics.

FIRA Category	max. Size Robots (cm)	Size Field (cm)	No. Players	Ball	Max. Speed (m/s)	Vision
MiroSot	7.5 x 7.5 x 7.5	220 x 180 or 400 x 280	5 or 11	orange golf ball	> 3.5	global
NaroSot	4 x 4 x 5,5	130 x 90	5	orange ping pong ball	2.3	global
HuroSot	150 (height) , max. 30kg	400 x 300	1	yellow tennis ball	n/a	local
RoboSot	35 x 35x height unrestricted	600 x 400	Up to 5	orange junior soccer ball	n/a	local
SimuroSot	n/a	n/a	5 or 11	n/a	n/a	n/a
KheperaSot	cylindric (diameter: 6cm)	105 x 68	1	yellow tennis ball	n/a	local

3.3 Wheeled Robots

When FIRA was founded in 1995, attention was focussed on wheeled robots. The differences between the four-wheeled FIRA leagues are considerable. KheperaSot, RoboSot, MiroSot and NaroSot differ in the number of their team members, their degree of autonomy and the speed of the players on the pitch.

Some attributes of these leagues are listed in Table 1. While KheperaSot [7]. and RoboSot robots have their own camera and full computing power on board, the MiroSot (figure 3) und NaroSot [8] leagues have opted for a setting with a camera above the pitch and a host computer beside the field. In this respect, neither MiroSot nor NaroSot robots act fully autonomously. On the other hand, this technical specification allows for speeds of more than 3.5 m/s and an acceleration of up to 8 m/sec^2. The cameras work with at least 30 frames/sec, but some teams use cameras with more than 100 frames/sec. The time between capturing and analysing a picture, deciding upon the best action and performing the wheel commands is in the 10 - 30 ms range. A special characteristic of the NaroSot robots is a considerable miniaturisation of the mobile platform. The combination of complex soccer-like rules and regulations, high speed and the number of players per team (11 in MiroSot) pose a considerable challenge on image processing and decision making.

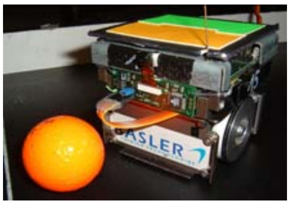


Figure 3: A MiroSot robot: prototype with a local camera.

4 What Real World Can Benefit

Many future applications for powerful robots have been discussed, ranging from autonomous agents for the production process, museum guides and rescue robots, up to companions in the household. Especially humanoid robots inspire people in terms of the assistance robots can offer in a world where the number of elderly and handicapped people is increasing dramatically. That these kinds of powerful robots will be available in the future is not a question. The questions are: when, and what are the main obstacles that must be overcome?

A white paper on the status and opportunities of the European Robotics Industry identifies the current R&D challenges with “robust perception” and “adaptation and learning” among them. With respect to perception, the authors emphasise: “A notorious problem has been a lack of robustness in advanced sensing systems, such as computational vision.” Referring to adaptation and learning, they point out: “… there is a need for an added degree of adaptivity, both in terms of control and utilisation of sensory feedback. … This calls for major new methods of adaptive control and (semi-) automatic learning of task sequences. For richer application domains, there is a significant need for the combination of sensory feedback with adaptation. In addition, there is a need for the utilisation of learning methods for the instruction of robots …” [9].

We would like to refer to some of our experiences in the fields of image processing and decision making to outline the remarkable challenges of the FIRA MiroSot league and to highlight that robot soccer is an excellent benchmark for many problems in these fields.

4.1 Image Processing

Image processing is a crucial and sometimes underestimated aspect of robot soccer. Nevertheless, the variety of colours, the speed of the game and the often unstable lighting conditions suggest that robot soccer is an excellent platform to find attractive solutions for advanced industrial applications. Due to the high speed of the game, it is essential to work with at least 30 full frames/sec. The average camera resolution is currently 640 x 480 pixels. Each team has its team colour, a second colour is used to identify the alignment, and almost all teams use separate colours to identify individual team members. Therefore, the image processing module must be able to handle 26 colours and the red ball in an 11 vs. 11 game while the design of the patches can be rather complex.

The image processing system has to meet four essential criteria: flexibility, robustness, efficiency and user-friendliness. Some keywords may exemplify the many aspects. The system should be as flexible as possible because it has to

- work under variable lightning conditions: bad illumination at the location, continued detection of changing conditions;
- support various constructions and cameras: oblique, decentralised set-up, switching to a new camera type;
- be usable in different leagues: FIRA-MiroSot-league, RoboCup-Small-League, new, unknown leagues;
- support software design changes: architectural changes, extensions and modifications, new concepts for subtasks.

The system has to be robust so that it

- delivers precise data: position, speed of all objects with minimal deviation, recognise alignment without tracking;
- is reliable with respect to data transmission: resistant to noise during the entire match, allows image interferences, occlusions, incomplete frames, colour recognition not constrained by brightness, transformation to other coordinate systems.

The system must be efficient, i.e. it has to

- share resources with the rest of the system, compute a new frame at least every 30 ms (soft real-time constraint) and require low memory consumption;
- allow a fast calibration of the systems: start-up in 5-10 minutes.

The system has to be user-friendly, i.e.

- feature a simple and intuitive user interface;
- offer intuitive parameters of the vision system;
- and show and help to understand decisions during the match.

In principle, the process follows five consecutive steps (figure 4) [10]:

1. find coherent areas of “similar” colours (pixel classification according to colour and relevance);
2. identify areas of same colours (called blobs, usually with standard region-growing algorithms);
3. find objects, i.e. assign blobs to disjointed groups (patches - defined by the colour of the team, the colour for the identification of the alignment, and in most cases a colour for the individual team member);
4. track each object instead of using a colour to identify the team member (greedy approach with local optimisation or greedy approach with movement prediction);
5. convert from a picture-based into a field-based coordinate system calculated to eliminate distortions (parallax, fisheye effect etc).

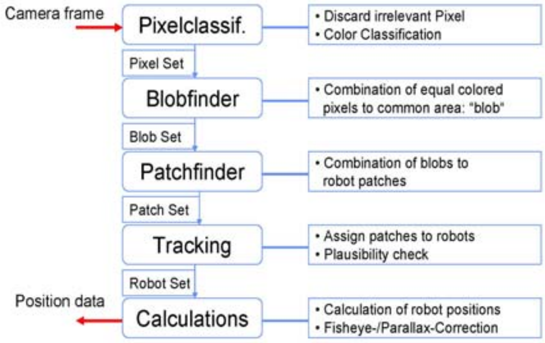

Camera frame
Pixelclassif.
• Discard irrelevant Pixel
• Color Classification
Pixel Set
Blobfinder
• Combination of equal colored pixels to common area: "blob"
Blob Set
Patchfinder
• Combination of blobs to robot patches
Patch Set
Tracking
• Assign patches to robots
• Plausibility check
Robot Set
Position data
Calculations
• Calculation of robot positions
• Fisheye-/Parallax-Correction

Figure 4: Image processing flow (simplified).

Introducing 11 vs. 11 matches on a 440 x 220 cm field poses new and demanding challenges. The cameras are mounted 2.5 m above the field And to cover the whole field with one camera would, in consequence, lead to strong fisheye effects and problems in calculating the exact robot positions. Raising the camera to a higher level is not an option since this would rule out too many locations. Therefore, the MiroSot teams playing in the 11 vs. 11 league have to revert to a multi-camera approach, where every camera records only a part of the field and the two sectors must be merged fast and reliably. Especially if the vision system shall provide tracking of objects this real-time environment poses some considerable challenges [11].

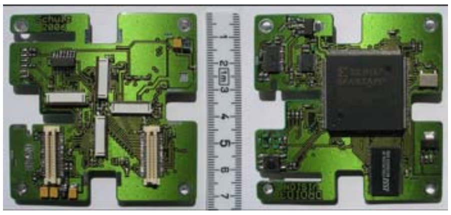


Figure 5: 4-Layer board with modules (left: bottom side/ right: upside).

Another challenging objective for MiroSot is a synergy of the global image with a local image processing [12]. A prototype of a robot with four local CMOS VGA cameras is under construction (figure 5). These cameras provide a resolution of 640 x 480 pixels and 25.5 frames/sec. The cameras are connected with a FPGA where the pre-processing takes place to disburden the DSP on the main board. Consequently, the DSP must be able to communicate with the FPGA and to process the images of the cameras [13]. In consequence, the reliable co-operation between a global and a local vision can lead to far more reliable, flexible and agile autonomous robots.

4.2 Multi-agent Architecture and Decision Making Process

The concept of multiple interacting agents has received widespread attention. An agent - derived from the Latin "agere", i.e. to do - perceives its environment through sensors and reacts to this environment through effectors. It acts on behalf of a person, but without extrinsic control. Attributes frequently regarded as constitutional are autonomy, goal orientation, collaboration, flexibility, self-starting character, mobility - to name just a few.

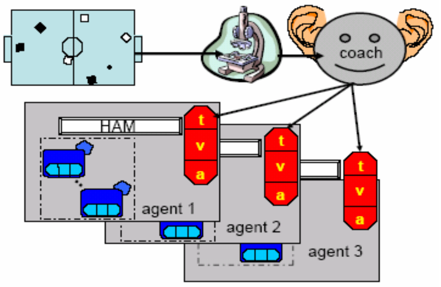


Figure 6: Robots as agents: strategy, role, action.

Members of a robot soccer team can be regarded as agents. Within the scenario of the MiroSot league, most FIRA teams follow a three-layer approach with a tactics, a role and an action layer:

- The autonomy of the robots is confined by a coach (figure 6) responsible for the global evaluation and tactical decisions. Based on the actual game situation, each tactics module calculates its own success probability. Relevant variables for this evaluation are, for example, ball position or the direction and velocity of the own team as well as of the opposing robots. The coach module decides the tactics with the highest success probability.
- The role module selects a specific role for every individual robot (goalie (t); defender (v) etc.) from a unique role set characteristic of the respective tactics. The role module re-analyses the actual situation focusing on important criteria such as "maximum driving time to reach the ball" or "position of robots and ball".
- Each role has a choice of up to 14 different actions (e.g. block_the_opponent or goal_kick). Depending on the evaluation, the most promising actions are selected. Selection criteria are "ball_at_the_border?" or "path_to_the_goal_unguarded?", among others. Finally, these actions are translated into commands for the robot's control module, where the precise target coordinates for the robot and the wheel commands are calculated.

Whilst this 3-layer approach is widely used by robot soccer teams, transparent co-operative strategies and actions such as a simple pass between two players (figure 7) are usually absent. Up to now, the main emphasis is placed on a kind of kick-and-rush soccer where speed dominates the game.

Robot soccer provides a challenging environment to explore "smart" decision-making techniques, especially those that can deal with vague and incomplete information. Fuzzy logic is predestined to cope with terms such as "near", "high speed" and "defensive", and it seems promising to fuzzify soccer rules like "if the opponent is near the penalty area and his speed is high, move towards the opponent to tackle as early as possible". Neural networks, especially self-organising maps, could be effective, e.g. in the analysis of the opponent's strategy and the evaluation of the overall situation on the field [14].

5 Transfer to the Real World

Focussing on *Image Processing* and *Decision Making* alone, there are infinite numbers of applications we can think of. Industrial image processing, for instance, is an essential driving force to reduce production costs. The overall objective in this domain is to recognise deficiencies during the production process as early as possible and to select and rework the objects at the early stage. It is obvious that many problems of robot soccer are close to this kind of real world problems. The identification of colours on the soccer pitch offers a lot of problems that are quite similar to those posed by selecting coloured objects from a fast-moving conveyor belt or by analysing colour characteristics in welding spots. Problems of image processing in robot soccer will not only motivate young students to tackle these kind of problems, at the end of the day they will lead do solutions for complex industrial problems.

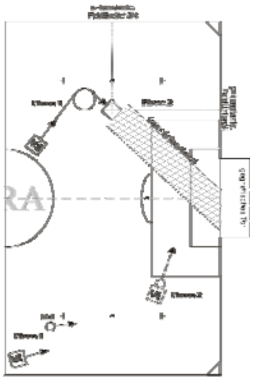


Figure 7: Co-operation between robots: a simple pass.

Similarly, the complex decision making process takes up questions that can be generalised to problems in repository optimisation or in port or plant automation. Efficient decision making software is a cutting edge technology that still today offers significant opportunities for enhanced assembly and handling technology.

The contribution of robot soccer to sustainable solutions is not immediately obvious. Two examples may exemplify the claim that FIRA robots can open up new scenarios:

- As a spin off from robot soccer the soccer team of the TU Vienna became a European Space Agency project. The robot for this project is based on the MiroSot robots platform. A robot succeeded in crawling on a net structure at an altitude of 200 km above the earth in an atmosphere of almost zero gravity [15].
- The SeT laboratory of the Université de Technologie de Belfort-Montbeliard demonstrates how MiroSot robots co-operate to move a vehicle by co-operation. A paralysed robot sends signals on its right and left sides in order to recruit mobile robots and to indicate the direction it wants to follow [16].

6 Summary

The robots of the future will be designed for tasks that take place in an unstructured, uncertain and unstable environment. Researchers have been working towards robots that have control autonomy, communication and co-operation competence as well as planning, learning and generalisation capabilities.

To achieve these ambitious goals, the functionality of robots must be extended significantly and their performance improved in subsystems like environmental perception and modelling, intelligent navigation and collision prevention, task planning and communication. Working with robot soccer systems can help fathom the feasibility of approaches for future real world solutions. But there is still a long way to go until soccer-playing robots are able to act as smart as human beings

7 References

[1] Minirobot Corp, “Home Page”, http://www. hitecrobotics.com/robonova/specs.htm, visited on 20/7/2006.

[2] Federation of International Robot-Soccer Association (FIRA), “Home Page” http://www.fira.net

[3] H. Cho, J.-H. Kim, J.-J. Lee, “Intelligent robots: vision, learning and interaction”, *Proceedings 2002 FIRA Robot World Congress*, Korea (2003)

[4] N. Weiss, N. Jesse and B. Reusch (eds), *Proceedings FIRA RoboWorld Congress 2006: Robots in Education, Entertainment and Services*, Dortmund (2006)

[5] M. Riedmiller, T. Gabel, R. Hafner, S. Lange and M. Lauer, “Die Brainstormers: Entwurfsprinzipien lernfähiger autonomer Roboter”, *Informatik-Spektrum* (2006)

[6] S. Behnke, T. Langner, J. Müller, H. Neub and M. Schreiber, “NimbRo RS: a low-cost autonomous humanoid robot for multi-agent research”, *Proceedings 27th German Conference on Artificial Intelligence,* Ulm, 2004, http://www.nimbro.net/publications/html, visited on 15/7/2006

[7] N. Keeratipranon, F. Maire and J. Sitte, “Robot soccer KheperaSot league: challenges and future directions”, *Proceedings 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AmiRE 2005),* Heidelberg, pp 141-146 (2006),

[8] P. Kopacek, M.W. Han, B. Putz, E. Schierer and M. Würzl, “NaroSot: nanotechnology in robot soccer”, *Proceedings 12th IFAC Symposium on Information Control Problems in Manufacturing,* St. Etienne, pp 193-198 (2006)

[9] H. I. Christensen et al. (eds), *European Robotic. A White Paper on the Status and Opportunities of the European Robotics Industry*”, Version 20, European Robotics Forum (IFR ERF) and European Robotics Research Network (EURON), (2003).

[10] N. Weiss, *Konzeption und Implementierung einer flexiblen und robusten Positionserkennung für den Roboterfußball*, Internal Report, University of Dortmund, Dortmund (2003).

[11] C.H. Messom, G. Sen Gupta and H.L. Sng, “Distributed real-time image processing for a dual camera system”, *Proc. Computational Intelligence, Robotics and Autonomous Systems 2001 (CIRAS 2001),* Singapore, pp 53-59 (2001).

[12] N. Weiss and N. Jesse, “Towards local vision in centralized robot soccer leagues: a robust and flexible vision system”, *Proceedings 9th FIRA RoboWorld Congress*, Korea (2004).

[13] S. Schulz, *Entwicklung eines kompakten, DSP- und FPGA-basierten Bildverarbeitungssystems mit Unterstützung mehrerer Kameras zur Navigation mobiler Roboter*, Diploma Thesis, Dortmund, work in progress.

[14] M. Reuter, “Ruling robots by the activity patterns of hierarchical SOMs”, *Proceedings ISC 2003 (Eurosis03)*, Spain (2003).

[15] Institute of Handling Devices and Robotics, http://www.ihrt.tuwien.ac.at/robyspace/, visited on 19/7/2006.

[16] S. Olivier and P. Gruer “A cooperative mobile robots architecture enabling paralyzed robots to move”, *Proceedings 12th IFAC Symposium on Information Control Problems in Manufacturing,* St. Etienne, pp 199-204 (2006).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Toward Ecological Interface Design for Human Supervision of a Robot Team

Hiroshi Furukawa
Department of Risk Engineering,
University of Tsukuba, Tsukuba, Japan
furukawa@risk.tsukuba.ac.jp

Abstract

When supervising a team of robots, the operator's task is not only the manipulation of each robot but also achievement of the top goal that is assigned to the entire team of humans and robots. It is crucial to support human operators to enable them to understand the overall progress in the work and the situation via a system-centred view. A main goal of this study is development of a design concept based on Ecological Interface Design (EID) for human supervision of a robot team, providing information about states of functions that are necessary to achieve the top goal of a human-robot system. This paper describes an experimental study conducted to reveal basic efficacy using an experimental test-bed simulation. The results suggest that the proposed interface design has the ability to enable effective human supervision, but that its usefulness closely depends on the strategies actually used in operations.

Keywords: human-robot interface, ecological interface design, functional displays, multiple robot management

1 Introduction

Multiple robots are increasingly being used in environments that are inaccessible or dangerous to humans [1][2]. Operations in these environments are numerous, and include reconnaissance, exploration, and surveillance. Regardless of their purpose of use, effective supervision and management of multiple robots poses a challenging problem because it is so labour-intensive. As a consequence, there has been increased interest in creating robot systems that allow a single human to manage multiple robots [3].

The operator's task involves not only manipulation of each robot but also achievement of the top goal that has been assigned to the entire team of humans and robots. Clearly, support of the operator’s skill-based behaviours is important. Equally, it is important to support human operators in their understanding of the overall state of a work-in-progress and the situation around it using a system-centred view.

Although humans have limitations in understanding complex system states, human-machine interfaces (HMIs) can display large amounts of complex information which risk overwhelming the operator at exactly the worst time, i.e., in an emergency situation [4]. Ecological Interface Design (EID) is a design concept based on the externalisation of the operator's mental model of the system onto the interface to reduce the cognitive workload during state comprehension [5][6][7].

One main goal of this study is the development of a design concept for a human-robot interface (HRI) based on Ecological Interface Design for human supervision of a robot team, providing information about states of functions that are necessary to achieve the top goal of a human-robot system.

Information on function is identified using the Abstraction-Decomposition Space (ADS) [8]. An ADS is a framework for representing the functional structures of work in a human-machine system that describes hierarchical relationships between the top goal and physical components with multiple viewpoints, such as abstraction and aggregation. Since the operator’s comprehension of the functional states based on the ADS is an essential view for the work, a system-centred view is necessary for operators to control a human-machine system, comprehend system states, make operational plans, and execute the plans appropriately under abnormal or unanticipated conditions [9]. Ecological Interface Design is a design concept in which the ADS of a target system is represented in such a way as to allow operators to comprehend the states of the functions intuitively [5][6][7]. A function-based HMI is designed to enable operators to develop a system-centred view. This can be thought of as an externalisation onto the HMI of the operator's mental model of the system [10].

In this study, the EID concept was used as the basic framework for implementing the information about a human-robot team work into an interface display. This paper describes an experimental study conducted to reveal the basic efficacy of EID in human-robot interactions, and the development of a design concept using a multi-level representation of functions to improve human-robot collaboration.

Several attempts have been made to apply the design concepts of the Ecological Interface Display or function-based display to HRI. Sawaragi et al. applied the EID concept to HRI to support naturalistic collaboration between a human and a robot at the skill level by joining their respective perception-action cycles via a 3D display [11]. Jin and Rothrock propose a function-based interface display which indicates the state of communication between human operators and multiple robots in which the target function is limited to one [12]. The basic concept of the HRI design proposed in this paper is a function-based display that indicates *the whole work* assigned to humans and *multiple robots*.

2 Methods

This section describes the RoboFlag simulation platform, and the application and the implementation approach to the test-bed of the proposed design concept. We then describe the experimental study that was conducted to investigate the uses of the interface during actual operations and examine the efficacy of the concept.

2.1 The RoboFlag Simulation Platform

This study uses the RoboFlag simulation, which is an experimental test-bed modelled on real robotic hardware [13]. One operator directs a team of robots to enter an opponent's territory, capture the flag, and return to their home zone without losing the flag. Defensive action takes the following form: while in the rival's territory, a rival robot can inactivate an intruding robot by bumping into it. Similarly, a rival robot will be inactivated if it is hit by a friendly robot while in friendly territory. Figure 1 shows the display used for an operator to monitor and control his or her own team of robots.

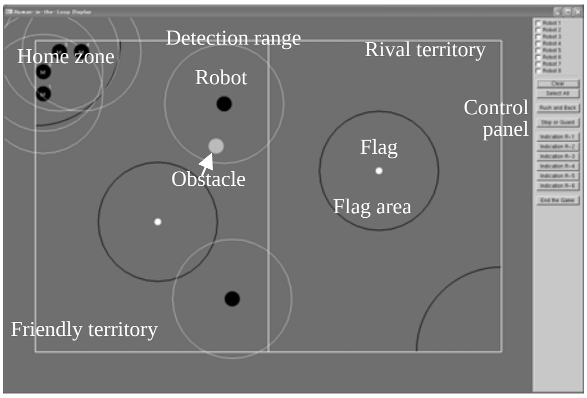


Figure 1: Operation display of the RoboFlag simulation.

The area over which a robot can detect opponent robots or obstacles is limited to the range. The robots are semi-autonomous, that is to say, they have the ability to change their own courses to avoid rival robots or obstacles.

The simulation provides two types of operations that operators can select according to their situation: manual controls and automatic controls. In manual control mode, an operator indicates a waypoint to a robot by clicking the point on a display.

Two types of automatic controls were implemented in this study. When *Rush and Back (R&B)* mode is assigned, the robot tries to reach the flag and returns home after it captures the flag. The course selected is straightforward, in that the robot heads directly to the destination. In *Stop or Guard (S/G)* mode, the robot stays in the home position until it detects an opponent. If an opponent robot comes into detection range, the robot tries to inactivate the opponent.

The chief goal of an operator' job is to take flags using home robots and to return to the home zone faster than the opponents. The basic tasks for achieving this goal are two: *Offence* and *Defence*. The former comprises two sub-tasks, which are *Capturing the flag* and *Taking it to one's own home zone*. To keep the robot active is also necessary for the sub-tasks. The sub-tasks of the latter are to prevent opponent robots from coming close to the flag and returning home with it.

Because time constraints are severe in this RoboFlag game, human operators need gain an understanding of the situation as rapidly as possible. Furthermore, it is necessary for operators to comprehend the state of entire area as well as the local area.

2.2 Implementation

This section shows the human-robot interface for the RoboFlag simulation used in this experimental study and the design process of the functional indications based on the proposed design concept.

Previous studies using the RoboFlag simulation showed that human-robot interactions depend on various contributory factors. Because of the high complexity of the interactions, the target functions indicated on the interface were limited to a part of the whole ADS in this study.

The following are descriptions of two function-based interface designs, in which one was designed to represent the state of a lower-level function under an *Offence* function, and the other under a *Defence* function.

2.2.1 A Functional Indication for Offensive Function

Figure 2 shows the outline of the ADS whose top is the *Offence* function. Two functions, *Capture flag* and *Take flag home*, depicted below the function are the means of achieving the top function. To capture the flag, the robot needs to reach the flag (*Reach flag*). At the same time, the robot should be in *active* mode (*Stay active*); and avoid opponents: *Avoid opponents* is one of the necessary functions to achieve the goal.

Figure 3 depicts the ADS below the function *Avoid opponents*. One of the means of achieving the function is *Set way-point such as not to encounter opponents*. To select an appropriate course to reach the flag, the situation along the course, especially the positions of opponents, should be understood by the decision-maker. The proposed indication was applied to the function *State comprehension near courses*, which is one of the key sub-functions included in the *Offence* function, and is allocated to the human operator. A functional indication was designed for ensuring that the operator is clearly aware of the state of the function. The indication is depicted in figure 4.

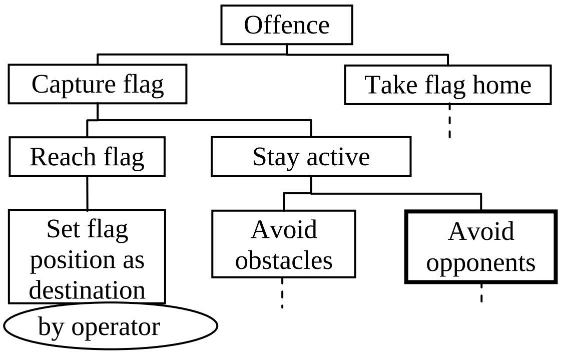


Figure 2: An ADS for the function *Offence*.

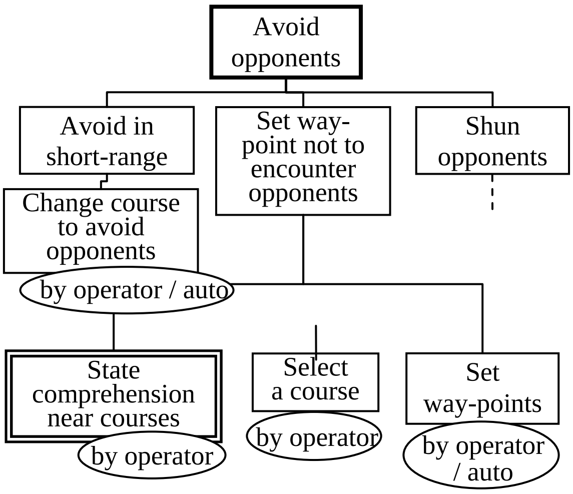


Figure 3: An ADS for the function *Avoid opponents*.

In figure 4, a robot is shown as a black circle and the flag as a white circle. The two straight lines connecting the robot and the flag show the trajectories along which the robot is going to move. A circle around a robot indicates the detection range within which the robot can detect opponents and obstacles. The two lines on the outside, which connect the detection range and the flag area, show the range in which detection becomes possible when the robot moves along the route. In other words, opponents in this area can tackle their own robots moving along the course. The display clearly indicates the *Field of play* of the target task. One of the operator's options is to send a robot as a scout to the field if there is an area where the situation is unknown.

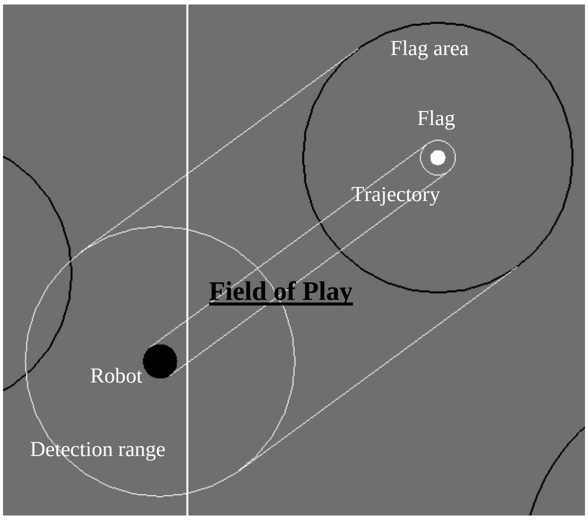


Figure 4: An interface design indicating the state of the function *State comprehension near courses*.

2.2.2 A Functional Indication for Defensive Function

The *Intercept opponents* function is an indispensable sub-function for achieving the *Defence* function. Figure 5 shows the ADS. *Cooperation between defensive robots* is a type of defensive function realised by a team of robots, and *Block opponents* is a defensive function possessed by individual robots.

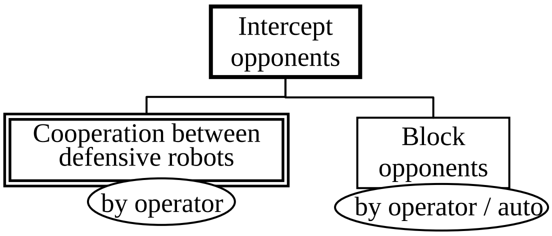


Figure 5: An ADS for the function *Cooperation between defensive robots*.

The proposed indication was applied to the function *Cooperation between defensive robots*, which is allocated to a human operator. The picture illustrated in figure 6 is the functional indication designed for enabling an operator to be clearly aware of the state of the function. A circle around a robot indicates the detection range as described in the previous section. A fan-shaped sector, a *Defensive sector* is where a robot in S/G mode has a high ability to intercept opponent robots coming through. Outside the Defensive sector, the possibility of catching opponents is lower than within the sector. An operator can use spaces between the sectors as an indication of the defensive ability of the defensive robot team in the position.

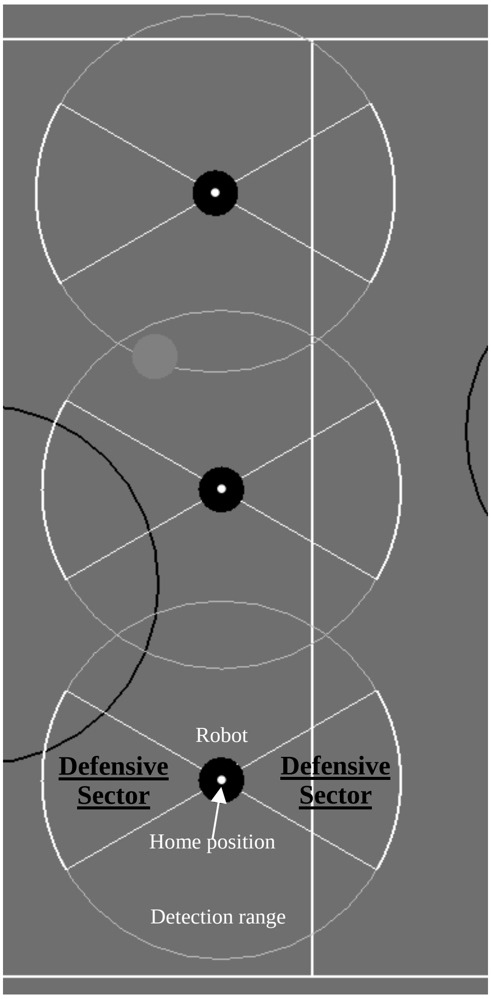


Figure 6: An interface design indicating the state of the function *Cooperation between defensive robots*.

2.3 Participants

Twenty-two paid participants (undergraduate and graduate students) took part in the experiment. All participants reported that they had normal or fully corrected vision and hearing.

2.4 Procedure

The twenty-two participants were randomly divided into two groups of eleven. One group (the *original* group; O1 – O11) used the original human-robot interface for the RoboFlag simulation, and the other group (the *modified* group; M1 – M11) used the modified interface display designed according to the proposed concept.

Offensive and defensive tasks of the rival robots were fully automated by using the two types of automatic controls implemented in this study.

The participants learned their tasks, rules of the game and the details of the assigned HRI, and mastered skills for controlling the team of robots through playing the game several times. They were asked to try it out until they found their own strategies to play the game. After they had decided on their strategies, they played the game five times as part of the main experiment. At the end of each game, they were asked to write the details of their strategies and usage of information represented on the display. The quantified data acquired in the main experiments of five games were then statistically analysed.

3 Results

3.1 Statistical Data Analysis

Averages and standard deviations of win percentages under the *original* and *modified* conditions are M = 45.4%, SD = 34.7%, and M = 63.6%, SD = 26.6%. However, a t-test shows that the difference between two conditions is not significant (t = 1.379, df =18.712, p = 0.184).

The number of flags captured was counted for every game. The averages and standard deviations of participants' captures in the *original* and *modified* conditions are M = 0.75, SD = 0.62, and M = 1.20, SD = 0.85. A repeated-measures ANOVA test indicates that the difference between two conditions is significant (F(1, 20) = 6.164, p = 0.022**).

The averages and standard deviations of opponent robots' captures in the *original* and *modified* conditions are M = 0.95, SD = 0.95, and M = 0.64, SD = 0.82. A repeated-measures ANOVA test indicates that the difference between the two conditions is not significant (F(1,20) = 2.055, p = 0.167).

In addition, the results of the statistical analysis show that there are no significant differences between the *original* and *modified* conditions for the numbers of times that participants' and opponents' robots were tagged, total elapsed times, and time before the first capture by participants' and opponents' robots.

The statistical tests show that differences in the measured performances between the groups using the modified display and the original display are not significant, except for the number of captures of the flag. This result may suggest that the modified display is effective in supporting operators in their offensive task, regardless of their ability or the strategy used for the task.

The results of averages and standard deviations indicate that the measured performances are distributed widely throughout the participants, suggesting that individual differences should be considered in data analysis on their performances.

3.2 Strategies Developed and Use of Functional Indications

The results of the statistical analysis, which are described in the previous section, may suggest that individual differences should be taken into account when designing suitable interface displays for supervising multiple robots. As a possible factor in the differences, this section illustrates the strategies developed by the eleven participants of the modified group (M1 – M11), who were assigned to the modified display, and how they used the information on functions represented on the display.

To compare the modified and original displays, the eleven participants (M1 – M11) played RoboFlag several times using the original display after they had completed the main experiments with the modified version. In an interview immediately afterwards, they were asked to explain the strategies they used during the main experiments, their usages of the indications of functions during the experiments, and the importance of the information in completing their missions. The typical strategies for offensive and defensive tasks are described in tables 1 and 2 respectively with the usages of the functional indications and the participants who used the strategies.

Table 1: Participants categorised by their strategies for offensive tasks and usage of the field of play.

Strategies	**Use of functional indication (Field of play)**	
	Used	**Did not use**
Used R&B	M1, M2, M7, M10	M11
Manual operation	none	M3, M4, M5, M6, M8, M9

Table 2: Participants, categorised by the strategies for defensive tasks and usage of the defensive sector.

Strategies	**Use of functional indication (Defensive sector)**	
	Used	**Did not use**
No operations	none	M3
Defensive operations	M1, M2, M4, M5, M7, M9, M10, M11	M6, M8

For offensive operations, the strategies that the participants used can be categorised into two varieties. Five participants mainly used the R&B automatic operation to capture the flag, where they acted as supervisors. Four of them (M1, M2, M7, and M10) tried to comprehend the state of the robots and situation around the course using the *Field of play* indication. In addition, one of them tried to decide the appropriate timing for activating R&B operation using the indication.

The other six participants chose manual controls for offensive actions, and all the waypoints and timings of the orders were fixed in advance. They developed a dynamic mental model for movement of opponent robots through the training. The indication was not necessary for them during the main experiments. In spite of this, they mentioned that the indication was useful for developing their own strategies during the trial-and-error processes in the training.

There were two types of strategies developed by the participant for defensive operations. One participant (M3) who used the manual-controlled strategy for offence decided not to take any defensive action. A swift attack was his only strategy. The *Defensive sector* display is not necessary for this strategy.

The other ten participants allocated two to four robots on a course that opponent robots followed to capture the flag. Eight of ten used the *Defensive sector* indication to decide appropriate spaces between the guarding robots at the training phase and/or the main experiments.

4 Discussion and Conclusion

The results through the qualitative analysis show the need to consider two factors to evaluate the efficacy of the displays: the one is participants' strategies developed for offensive and defensive tasks, and the other is how they used the functional indications. However, statistical tests on the performances of the participants who actually used the functional indications and who used the original display could not add new results to the previous test described in section 3.1. These results suggest that the amount of data is not sufficient for statistical analysis. It is expected that additional experiments can quantitatively reveal the relationships between the referred factors.

On the other hand, the effect on the number of captured flags was significant without regard to offensive strategies. This indicates that the *Field of play* display has clear effectiveness for offensive tasks.

It is important to be aware of the following to understand the results of this experiment. First, the purpose of this study is a verification of the design concept for human-robot interface, but not a confirmation of the effectiveness of the developed interface design itself. Second, the target functions indicated on the interface were limited to a part of the whole work, since the complexity of human-robot interactions is too high to evaluate the efficacy of the function-based display in an experimental study. Third, the results can be generalised to more complex work assigned to human and multiple robots because the target work in this study can be regarded as a typical example of a set of tasks that have multiple objectives to be achieved simultaneously.

The results of the statistical data analysis cannot reveal the efficacy of the function-based human-robot interface design. However, the findings of the strategies and subjective evaluations of the usefulness of the modified interface suggest that the proposed interface design may have the ability to provide effective human supervision of multiple autonomous robots. It is also suggested that the need for a functional display closely depends on the strategies actually used during operations. Furthermore, individual differences in the strategies may cause difficulties in confirming efficacy.

The goals of the next experiments must be to confirm the strategies developed by operators and the appropriateness of the strategies, to identify functions that operators have difficulty in comprehending, and to verify necessary information for supporting the operator's understanding of the states of the functions.

5 Acknowledgements

This work has been supported by Grants-Aid for Science Research 16500076 and 18500104 of the Japanese Ministry of Education, Science, Sports and Culture.

6 References

[1] S.A. Stoeter, P.E. Rybski, K.N. Stubbs, C.P. McMillen, M. Gini, D.F. Hougen and N. Papanikolopoulos, "A robot team for surveillance tasks: design and architecture", *Robotics and Autonomous Systems*, 40(2-3), pp 173-183 (2002).

[2] L. Chaimowicz, A. Cowley, D. Gomez-Ibanez, B. Grocholsky, M.A. Hsieh, H. Hsu, J.F. Keller, V. Kumar, R. Swaminathan and C.J. Taylor, "Deploying air-ground multi-robot team in urban environments", in L.E. Parker, F.E. Schneider and A.C. Schultz (eds), *Multi-Robot Systems: From Swarms to Intelligent Automata, Vol. III*, Springer, Dordrecht, Netherlands, pp 223-234 (2005).

[3] M.A. Goodrich, M Quigley and K. Cosenzo, "Task switching and multi-robot teams", in L.E. Parker, F.E. Schneider and A.C. Schultz (eds), *Multi-Robot Systems: From Swarms to Intelligent Automata, Vol. III*, Springer, Dordrecht, Netherlands, pp 185-195 (2005).

[4] T.B. Sheridan, "HCI in supervisory control: twelve dilemmas", in P.F. Elzer, R.H. Kluwe and B. Boussoffara (eds), *Human Error and System Design and Management*, Springer, London, pp 1-12 (2000).

[5] K.J. Vicente and J. Rasmussen, "The ecology of human-machine systems II: mediating direct perception in complex work domains", *Ecological Psychology*, 2, pp 207-249 (1990).

[6] K.J. Vicente and J. Rasmussen, "Ecological interface design: theoretical foundations", *IEEE Transactions on Systems, Man, and Cybernetics*, 22(4), pp 589-606 (1992).

[7] K.J. Vicente, "Ecological interface design: progress and challenges", *Human Factors*, 44(1), pp 62-78 (2002).

[8] J. Rasmussen, *Information Processing and Human-Machine Interaction*, Elsevier Science Publishing, New York (1986).

[9] C.A. Miller and K.J. Vicente, "Comparison of display requirements generated via hierarchical task and abstraction-decomposition space analysis techniques", *International Journal of Cognitive Ergonomics*, 5(3), pp 335-355 (2001).

[10] J. Rasmussen and A.M. Pejtersen, "Virtual ecology of work", in J. Flach, P. Hancock, J. Caird and K. Vicente (eds), *Global Perspectives on the Ecology of Human-Machine Systems*, Lawrence Erlbaum Associates, UK, pp 121-156 (1995).

[11] T. Sawaragi, T. Shiose and G. Akashi, "Foundations for designing an ecological interface for mobile robot teleoperation", *Robotics and Autonomous Systems*, 31(3), pp 193-207 (2000).

[12] J. Jin and L. Rothrock, "A visualization framework for bounding physical activities: toward a quantification of gibsonian-based fields", *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting*, Orlando, FL, pp 397-401 (2005).

[13] M. Campbell, R. D'Andrea, D. Schneider, A. Chaudhry, S. Waydo, J. Sullivan, J. Veverka and A. Klochko, "RoboFlag games using systems based hierarchical control", *Proceedings of the American Control Conference* Denver, CO, pp 661-666 (2003).

UAV Flight Controller on an FPGA

Przemyslaw Wrzos and Andrew Price
Autonomous Aircraft Research Group
Department of Electrical and Computer System Engineering
Monash University, Melbourne, Australia
andrew.price@eng.monash.edu.au

Abstract

We present a simple system for the control of a small low cost UAV under 2kg flying weight. The flight controller is constructed within an FPGA device. This approach releases the need for the controller to be in high level language and minimises the latency due to processing delay via a microprocessor. System equations are first developed and their response given the known parameters of the target UAV is evaluated. The design approach and the nature of the FPGA hardware allow the translation of this design to other UAV platforms of similar size as well as parallelisation of the algorithms for faster response. This work was conducted as part of an undergraduate thesis project by Wrzos, in the Autonomous Aircraft Research Group at Monash University.

Keywords: UAV, flight control

1 Introduction

Within this paper a design is proposed for an unmanned air vehicle (UAV) flight controller that is purely hardware based. An FPGA platform provides substantial advantage over traditional hardware solutions. Benefits of this design choice are considered and its effectiveness demonstrated. A testing rig is also proposed to sufficiently exercise the flight controller before real flight. The choice of a Personal Computer (PC) based software solution yields a cheap, flexible and highly effective test bed.

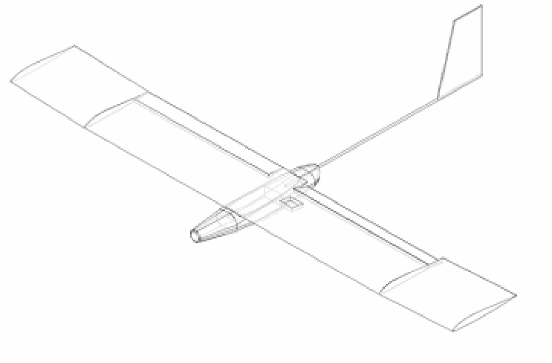


Figure 1: Target UAV.

The UAV targeted by the system has two independent control surfaces on either wing and a propeller at the front of the fuselage.

2 Design Overview

2.1 System Design

The design proposed here consists of four major sub-systems. Key to the flight controller itself are two major components: the *aerodynamic control* and the *flight planner*. Further, an external *test rig* is used to exercise the flight controller, and finally a tailored *communications interface* allows the test rig to talk seamlessly to the controller unit.

Figure 2 demonstrates the internal topology of the flight controller. Communications with the PC test rig are achieved using the *UART interface* and *packet handler*. The *test rig interface* provides an interface for the *aerodynamic stability control* making real UAV hardware indistinguishable from the equivalent used in simulated flight.

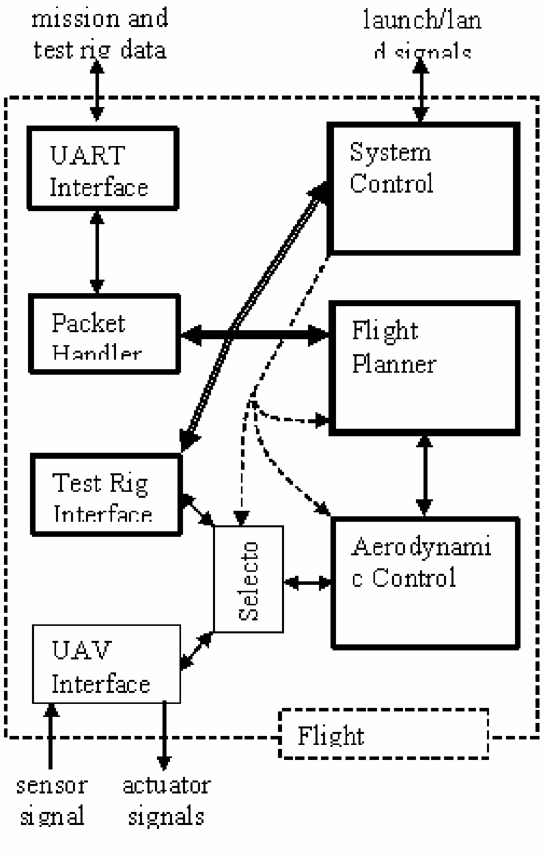


Figure 2: Flight controller block diagram.

The *aerodynamic control* block implements the stability control loops. The *flight planner* commands the aerodynamic control depending on its mission data. This was not the main focus of the initial prototype so in the first design this received only a proxy implementation, primarily forwarding data directly from the test rig.

Finally, the *system control* module coordinates the functions of the controller maintaining an awareness of the UAV's operational state. For example, different behaviour is required during takeoff, in stable flight and during a landing.

2.2 System Implementation

The controller was implemented on an Altera Cyclone FPGA, allowing hardware structures to be configured directly in silicon. All of the modules received hardware implementations. This not only made them fast but also minimised inter-module interface redundancy. While we have attempted other solutions in previous work [4], the solution presented here makes efficient use of a low cost FPGA without the need for a microprocessor, even a soft-core variant, enabling faster updates and less risk of damage to the airplane typically found when stored sequential programme architectures freeze. Because the design results in bursts of electronic activity in response to stimuli, digital switching is kept to a minimum and power is conserved - an important feature for UAVs where weight and hence available power must be kept minimal.

The hardware description language chosen was VHDL. This is a low level language; however, because the hardware structures are specified explicitly, smaller and more power-efficient designs can be created than in higher level behavioural descriptions such as Handel-C. Hardware synthesis was performed using the Quartus II 4.1 package with the Synplify compiler. This compiler can be quite aggressive in its optimisation, requiring care and use of regular structures in describing hardware.

3 Communications

A generic communications interface is designed to link the Personal Computer (PC) based test rig to the flight controller hardware. The interface is abstracted into two layers for simpler usability.

Layer 1 consists of a UART interface. It is implemented with an RS-232 connection from the PC COM port interfaced to the FPGA with a logic level shifting MAX3221ECAE chip. A state machine implementation is used for receiving and transmitting data both on the FPGA and PC.

Layer 2 implements a packet protocol. Speed is imperative so only one header character is used per packet, followed by data characters. For this reason, no error correction is performed in this layer. The header structure is outlined below:

7	6	3	2	1	0
H	SIZE		ID		-

H:	indicates if this is a header character
SIZE:	number of data characters in packet, from 0-15
ID:	source/destination module ID

The *ID* field allows clear functional separation of various data between individual handlers. Handlers can then treat data according to their individual functional requirement. Time consuming handshaking is avoided by the use of the *H* bit which is used for packet synchronisation.

A generic bus interface is used between the packet handler and all other modules for consistency.

4 Aerodynamic Stability Control

The aerodynamic stability controller has the function of keeping the UAV stable in trimmed flight and of handling external disturbances. It is given desired attitude and speed commands which it then attempts to match. These commands include bank and airspeed as well as either a pitch or an altitude command. Like other parts of the system, controller activity is triggered by incoming data and remains idle when there is no data to process.

4.1 PID Control

PID control was chosen to stabilise the UAV. This is a classical solution but is quite appropriate here as the UAV's aerodynamic derivatives do not change much during ordinary flight conditions [1]. PID control is most appropriate for control of single values and simple input to output relationships. The relevant relationships are as follows: mean aileron deflection controls pitch angle, differential aileron deflection controls bank angle, throttle controls airspeed, pitching angle controls altitude.

Because these relationships are not always linear, the PID controllers had to be augmented with saturation limiters. This adds the additional problem of integral windup when the PID output is outside of the range of the output limit. An anti-integral windup solution is used to stop integration until the integrator can contribute to returning the PID output to within the desired limits.

4.2 Digitisation

To digitise the control loops, they are first discretised by mapping from the Laplace domain to the discrete-time domain. This was achieved using the backward finite difference method [2] (equation (1)). While there are other methods, finite difference serves adequately for the task.

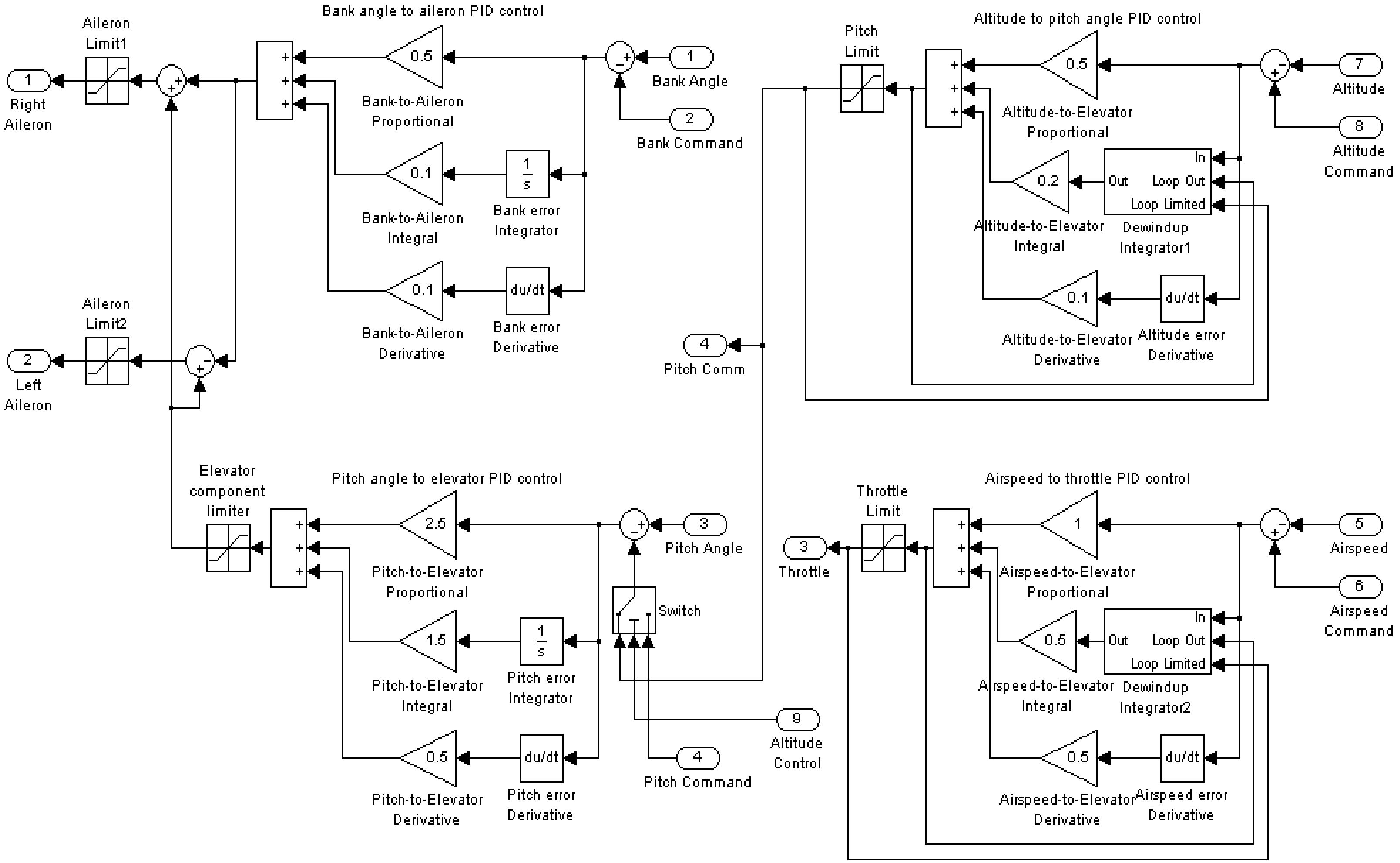

Bank angle to aileron PID control
Aileron Limit1
1
Right Aileron
0.5
Bank-to-Aileron Proportional
1
Bank Angle
2
Bank Command
0.1
Bank-to-Aileron Integral
1/s
Bank error Integrator
0.1
Bank-to-Aileron Derivative
du/dt
Bank error Derivative
Aileron Limit2
2
Left Aileron
Altitude to pitch angle PID control
Pitch Limit
0.5
Altitude-to-Elevator Proportional
7
Altitude
8
Altitude Command
0.2
Altitude-to-Elevator Integral
In
Out
Loop Out
Loop Limited
Dewindup Integrator1
0.1
Altitude-to-Elevator Derivative
du/dt
Altitude error Derivative
4
Pitch Comm
Elevator component limiter
Pitch angle to elevator PID control
2.5
Pitch-to-Elevator Proportional
3
Pitch Angle
Switch
1.5
Pitch-to-Elevator Integral
1/s
Pitch error Integrator
0.5
Pitch-to-Elevator Derivative
du/dt
Pitch error Derivative
9
Altitude Control
4
Pitch Command
Airspeed to throttle PID control
Throttle Limit
3
Throttle
1
Airspeed-to-Elevator Proportional
5
Airspeed
6
Airspeed Command
0.5
Airspeed-to-Elevator Integral
In
Out
Loop Out
Loop Limited
Dewindup Integrator2
0.5
Airspeed-to-Elevator Derivative
du/dt
Airspeed error Derivative

Figure 3: Flight control simulation schematic.

$$s \rightarrow \frac{\left(1 - z^{-1}\right)}{T_s} \tag{1}$$

Thus, applying this to the general PID transfer characteristic in (2):

$$\frac{y(s)}{e(s)} = K_p + \frac{1}{s}K_i + sK_d \tag{2}$$

we obtain the discrete time representation in (3):

$$y_k = K_p + \left(y_{k-1} + K_i T_s e_k\right) + \frac{K_d}{T_s}\left(e_k - e_{k-1}\right), \ \text{for time step } k \tag{3}$$

To make (3) suitable for a fast VHDL implementation, all calculations must be performed using fixed point arithmetic. To achieve good calculation accuracy, initial values are multiplied up and then the final result is divided back down. This can be done using shift operations which do not introduce additional logic in a hardware representation. Further, the digital representation of sensors and actuators will not correspond to the analogue values used to tune the controller in figure 3; hence, scaling must be applied. Finally all constants are grouped together so that their composite values provide for a reasonable amount of precision. A pseudo code representation is demonstrated below:

```
error = variable_value – variable_command

pout = (Kp*IN_SCALE/OUT_SCALE*2^PRECISION)*
error

iout = iout +
(Ki*Ts*IN_SCALE/OUT_SCALE*2^PRECISION)*error

dout =
(Kd/Ts*IN_SCALE/OUT_SCALE*2^PRECISION)*(error
– last_error)

last_error = error

output = (pout + iout + dout)/2^PRECISION –
OUT_OFFSET*OUT_SCALE
```

The VHDL implementation is based on the above with the addition of saturation and anti-integral windup logic achieved with basic condition statements. Explicit typecasting was required to and from the *signed* data type used for performing calculations. A real example of the power of VHDL can be seen here, where we can explicitly specify the exact number of bits for use in our variables, avoiding additional logic which might result in a waste of power.

An Excel spreadsheet was found indispensable in managing and tracking the effects of all the constants in the system. Compound constants could be calculated automatically and their effect on precision examined.

5 Test Rig

5.1 Software Solution

The design of an effective test rig was imperative to the success of the system. The controller's performance on the test rig is the only indication of how capable the controller is of flying an actual UAV. To keep costs down while still making a wide range of functionality available, a software solution was sought.

The AeroSim blockset package for Matlab/Simulink, from Unmanned Dynamics, was chosen. The blockset library models not only the dynamics of the UAV with configurable aerodynamics files, but "…also includes environment models such as standard atmosphere, background wind, turbulence, and Earth models (geoid reference, gravity and magnetic field)." [3] Combined with the power and flexibility of Simulink, this package provides both an in-depth and simple-to-use testing platform. The test rig was further augmented with an interface to flight gear which provided real-time 3D visualisation of the UAV in flight.

This platform could be utilised not only for testing but also for tuning of the controller. The controller found in figure 3 was tuned using the wide range of detailed feedback available about the behaviour of the UAV's states.

5.2 Interfacing Hardware

New functional blocks can be added to Simulink in the form of S-functions. Simulink uses a powerful callback model allowing new functions to be added in the C programming language.

This was utilised to design a block which implements both layers of the communications protocol mentioned earlier. The Win32 Comm API was utilised to implement non-blocking communications over the COM port in a way that did not hold up the aerodynamic calculations being performed.

The block also had the function of providing real-time clocking of the simulation loop. The only timer available in the Windows OS with sufficient resolution for this task is not associated with the process scheduler. Hence a polling loop was used to hold up the simulation by a sufficient amount each time through the simulation loop, effectively equalising any timing slack. This allowed the simulation loop timing to correspond to real execution time.

6 Flight Results

A test flight was set up in the test rig with parameters as per Table 1. A disturbance was also introduced in the form of a sudden gust of wind at t = 30 s. The gust is present at full strength between 30 s – 40 s and then decays to zero by 50 s. The wind is a gale force wind of 19.2 m/s (37.4 knots) with a northerly component of 15 m/s, easterly component of 11 m/s and a downward component of 5 m/s. The entire simulation runs for 180 seconds.

Table 1: Test flight parameters.

	Initial	**Commanded**
Airspeed	20 m/s	25 m/s
Altitude	1000 m	950 m
Bank Angle	0°	10°

The results are shown in the following figures.

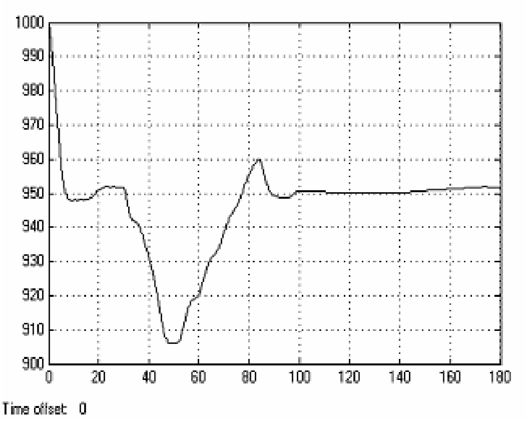


Figure 4: UAV altitude (m).

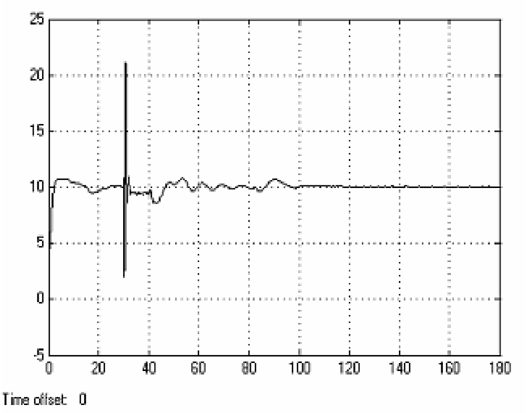


Figure 5: UAV bank angle (degrees).

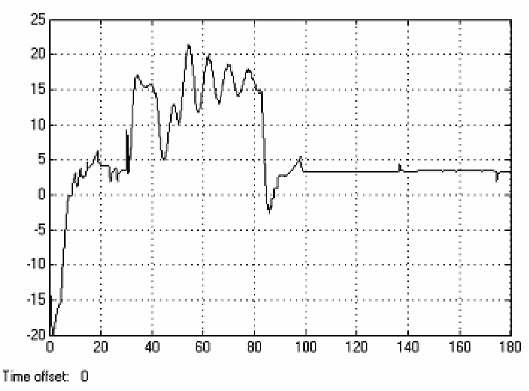


Figure 6: UAV pitch angle (degrees).

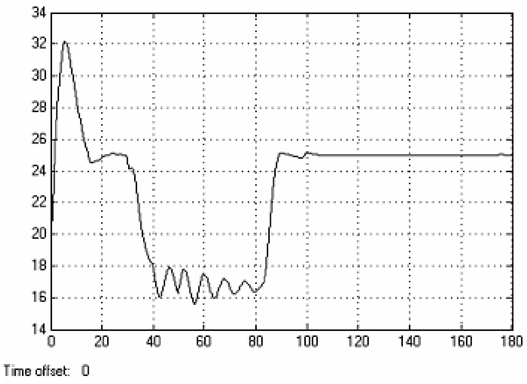


Figure 7: UAV airspeed (m/s).

We can see from figures 4 to 7 that the controller initially adjusts the actuators to reach the commanded values. During descent, airspeed remains uncontrolled to facilitate a fast descent rate. After this dive, the plane begins to level off at about 950 m and the airspeed reaches the commanded 25 m/s. The gust of wind then hits and overpowers the UAV's capability to remain at altitude. The UAV also banks sharply before the controller regains control.

As the wind dies away the UAV begins to settle out to its commanded values again. It should be noted there is a slight perturbation in altitude once 950 m is reached. This is due to the low resolution of the altitude sensor which returns a 12-bit value representing a maximum possible value of 8192 m and thus the steady state altitude will cyclically rise and fall by 2 m.

For comparison we run the same simulation without the controller to demonstrate loss of effective control (figures 8 and 9).

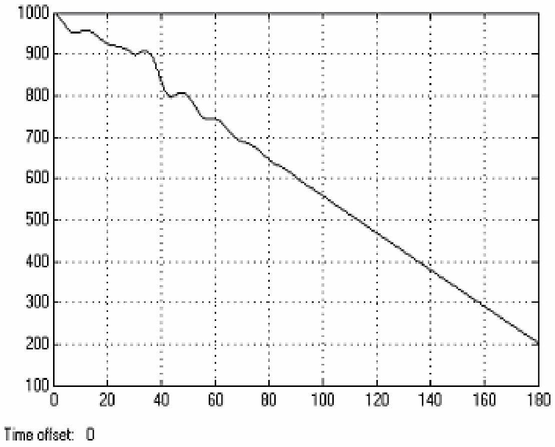


Figure 8: Effect on altitude without controller (m).

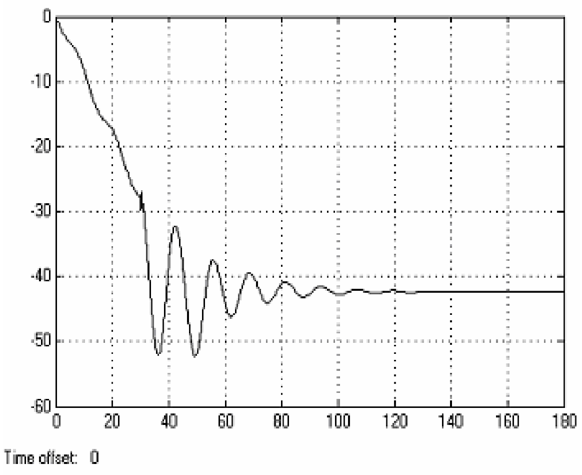


Figure 9: Uncontrolled pitching angle (degrees).

While the UAV is relatively stable and never reaches stall conditions in our experiments, without control it will obviously crash. For any other type of robot that fails statically, a malfunction in control may be acceptable; however for a UAV failure of control generally means an abrupt arrival.

7 Further Development

A platform for the development of UAV hardware flight control has been demonstrated within this paper. From here there are several avenues for further research, including more sophisticated aerodynamic control to account for the non-linearities of real world events and the development of a flight planner which can make appropriate manoeuvre decisions, for example using a way pointing system. The groundwork is in place for rapid development of these ideas on both highly capable and efficient hardware in a fraction of the time allowed by traditional development schemes.

8 Conclusion

With the known parameters of a UAV platform we have been able to demonstrate control of altitude, pitch angle and roll as well as airspeed. The system requires substantial testing and evaluation and is in the process of being implemented using the Monash University Miniflight controller. In addition, the size of the controller is sufficiently small that a low cost FPGA can be used. The performance depends upon the exact UAV configuration. This research has shown it is feasible to embed the entire control system in a UAV with the ability to adjust the parameters of the system to adapt to different airframes.

9 References

[1] R.P.G. Collinson, *Introduction to Avionics Systems*, Kluwer Academic Publishers, Boston (2003)

[2] M. Tham, *Mathematics of Sampled Data Systems: Approximation of Differentials*, Chemical Engineering and Advanced Materials, Engineering Faculty, University of Newcastle upon Tyne, http://lorien.ncl.ac.uk/ming/digicont/digimath/sampled2.htm (visited 24 November 2005).

[3] Aerosim Blockset, Unmanned Dynamics, http://www.u-dynamics.com/aerosim/default.htm (visited 24 November 2005).

[4] J. Young and A.R. Price, “FPGA based UAV flight controller”, *Proceedings Eleventh Australian International Aerospace Congress. (AIAC-11)*, Melbourne (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Evolvable Hardware Using a Neuromolecular Model for Spatiotemporal Processing

Jong-Chen Chen, Yo-Hsien Lin
Department of Information Management,
National YunLin University of Science and Technology, Taiwan, R.O.C.
jcchen@mis.yuntech.edu.tw

Abstract

Computer hardware has high processing speed; however, there is a lack of flexibility in dealing with most real world problems. Evolvable neuromolecular hardware motivated from some biological evidence is proposed. Evolvability, self-organising dynamics, and a close structure-function relationship are the three biological features captured in this hardware. The Quartus II system, a digital circuit simulation tool, is used for performing a series of experiments with this hardware. The hardware is applied to the pattern recognition problem domain. Experimental result shows that its structure and function are closely related, implying this hardware embraces an adaptive fitness landscape that facilitates evolutionary learning.

Keywords: evolutionary adaptability, pattern recognition, evolvable hardware, chips

1 Introduction

A computer system, including software and hardware, provides an effective programmability that allows us to apply it to various problem domains. However, as indicated by Conrad [1], the structures of computer programmes are brittle that a slight modification in a programme might totally change its functions. Biological systems demonstrate better adaptability than computer systems. Evolvability, self-organising dynamics, and a close structure-function relationship provide organisms with great malleability (gradual transformability) in coping with environmental changes (i.e. noisy environments) and learning new survival strategies for uncertain environments (i.e. novel environments).

Evolvable hardware (EVH) is a new approach that tries to apply evolutionary learning techniques into circuit synthesis [2,3]. The idea is to take the merits of biological systems and computer systems together and, hopefully, create hardware with high adaptability. As pointed out by Gordon and Bentley [4], EVH might have the following benefits: low hardware design costs, automatic design, innovative design, better adaptability in dealing with environmental changes, and good fault tolerance capability.

Yao [2] indicates that there are two major approaches in the field of EVH: simulated evolution and reconfigurable hardware. The former is to apply evolutionary techniques to circuit synthesis while the latter is to dynamically and autonomously reconfigure hardware architectures for specific tasks. An early attempt on EVH is to use evolutionary techniques in order to help design primitive arithmetic and logic digital gates [5,6]. Complicated circuit designs emphasising on high-level hardware functions are then proposed. These include multi-valued circuits [7], a digital string generator [8], and a digital filter [9].

The artificial neuromolecular (ANM) system developed earlier [10] has two distinguishing features. Firstly, motivated from the molecular mechanisms inside real neurons, some neurons possess complex internal dynamics. Through evolutionary learning, these neurons can be moulded as specific input-output transducers. Secondly, like other Hebbian models, it contains neurons with hierarchical control capability. The integration of these two types of neurons into a system constitutes an open evolutionary architecture with rich dynamics. In principle, this openness should allow the model to address a broader class of problems than pure connectionist models do. Through evolutionary learning, the system was applied into a number of problem domains, including robot navigation [10], pattern recognition [11], Chinese character recognition [12], and hepatitis-B differentiation [13]. However, our previous experimental results were based on a virtual machine that run on top of a serial digital computer and were therefore subject to practical computational limitations. An earlier implementation of this model on digital circuits was proposed, but only preliminary experimental results were reported [14]. In this paper, we perform a series of experiments, with emphasis on the spatiotemporal processing capability of the model.

2 Architecture

As indicated earlier, the ANM system integrates intra- and interneuronal information processing. In this section, we first explain the intraneuronal architecture that plays the role of integrating intraneuronal signals in space and time. Then, we introduce the interneuronal architecture that serves as orchestrating groups of neurons for coherent tasks.

2.1 Intraneuronal Architecture

Our implementation of the neurons with internal dynamics is to capture the signal integration capability. When an external signal impinges on the membrane of a neuron, it will initiate a signal flow along the neighbouring components of the same type. For example, in figure 1, the activation of the readin enzyme at location (2,2) will trigger a cytoskeletal signal flow on the C2 components of the second column, starting from location (2,2) to location (8,2).

An activated component will affect the state of its neighbouring components of different types when there is an MAP (microtubule associated protein) linking them together. For example, in figure 1, when the signal on the seventh column initiated by the readin enzyme at location (3,7) arrives at location (4,7), it will activate the component at location (4,8) via the MAP. The activation of the component at location (4,8) will in turn trigger a new signal on the eighth column. However, our assumption is that the interactions between two different types of neighbouring components are asymmetric. That is, the activated component at location (4,8) is not sufficient to activate the component at location (4,7).

The other assumption is that different types of components transmit signals at different speeds. For example, C1 components transmit signals at the slowest speed. On the contrary, C3 components transmit signals at the fastest speed. The transmitting speed of C2 components is intermediate between those of C1 and C3 components.

When a requisite spatiotemporal combination of cytoskeletal signals arrives at the site of a readout enzyme, the neuron will fire. For example, in figure 1, there are three possible signal flows that might arrive at location (8,3) and activate the readout enzyme at that site. The first is the signal flow transmitting along the second column, activated either by the readin enzyme at location (2,2) or by readin enzyme at location (3,2). The second is the signal flow transmitting along the third column, activated by the readin enzyme at location (4,3). The third is the signal flow transmitting along the fourth column, activated either by the readin enzyme at location (1,4) or by the enzyme at location (4,4). Any two out of the three signal flows mentioned above might activate the readout enzyme at location (8,3), which in turn cause the neuron to fire. However, the neuron might fire at different times. One reason is that signals transmit at different speeds along different types of components. The other reason is that the signals are initiated by different readin enzymes.

Each of the neurons with the above interneuronal architecture is implemented with a grid of 8x8-processing units (PUs). They are determined in a wraparound fashion. Each PU has eight neighbouring PUs. Each PU integrates signals from the receptor neurons and its neighbouring PUs, and sends its outputs to the effector neurons and neighbouring PUs. All PUs have the same architectures except the memory bits that control signal initialisation, transmission, integration, and degradation (figure 2).

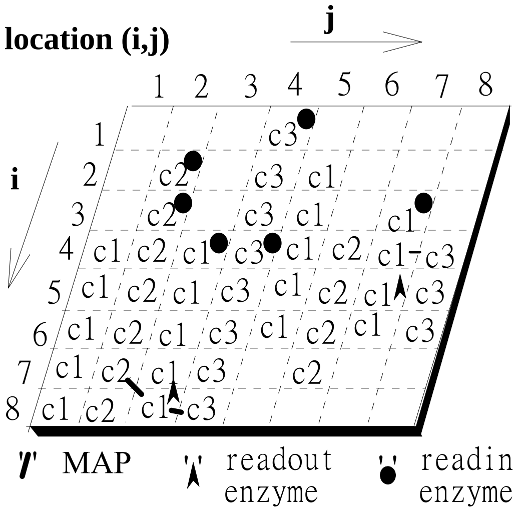


Figure 1: Cytoskeletal neurons.

Each grid location, referred to as a site, has at most one of three types of components: C1, C2, or C3. Some sites may not have any component at all. Readin enzymes could reside at the same site as any one of the above components. Readout enzymes are only allowed to reside at the site of a C1 component. Each site has eight neighbouring sites. The neighbours of an edge site are determined in a wrap-around fashion. Two neighbouring components of different types may be linked by an MAP (microtubule associated protein).

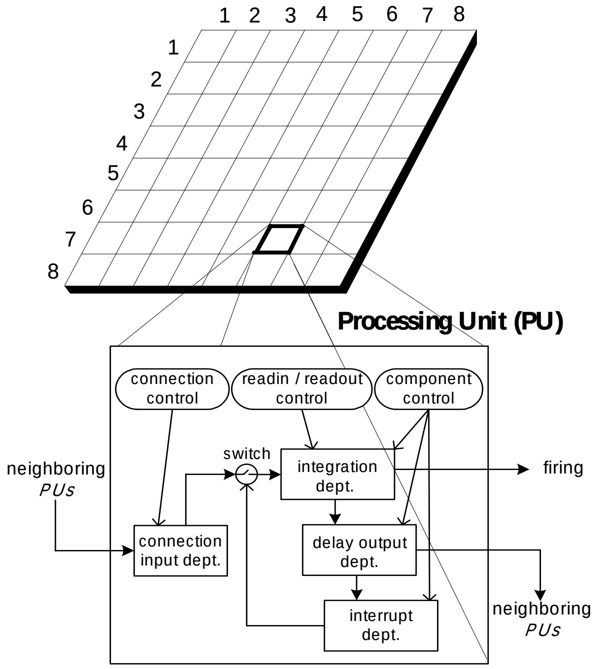


Figure 2: A PU architecture.

2.2 Interneuronal Architecture

Orchestration is an adaptive process mediated by varying neurons in the assembly that selects appropriate combinations of neurons to complete specific tasks. Currently, we use 128 cytoskeletal neurons, divided into four comparable subnets. By comparable subnets, we mean that neurons in these subnets are similar to each other in terms of their interneuronal connections and intraneuronal structures. Neurons in different subnets having similar interneuronal connections and intraneuronal structures are grouped as a bundle. Thus, these 128 cytoskeletal neurons can be grouped into 32 bundles.

Two levels of reference neurons manipulate these 32 bundles. These two levels form hierarchical control architecture, as shown in figure 3 (only two of the competing subnets are shown, each consisting of only four cytoskeletal neurons). The first is referred to as low-level reference neurons that directly control the bundles of cytoskeletal neurons. There are 32 low-level reference neurons, each of them controlling a bundle. (It is noted that only the bundles activated by the reference neurons are allowed to perform pattern transduction.) The second level is referred to as high-level reference neurons that play the role of grouping low-level reference neurons. The activation of a high-level reference neuron will fire all low-level reference neurons under its control, which in turn will activate some of the 32 bundles of cytoskeletal neurons (i.e. neurons in different subnets have similar intraneuronal structures). The connections among low-level reference neurons and cytoskeletal neurons are fixed. However, the connections between high-level reference neuron and low-level reference neuron layers are subjected to change during evolutionary learning, called the orchestral learning.

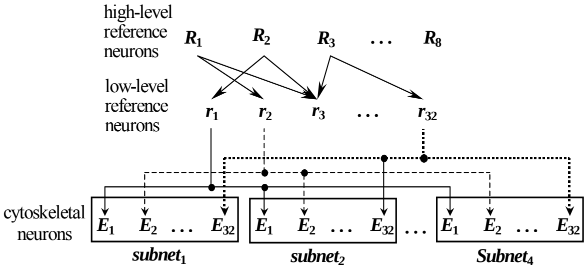


Figure 3: Hierarchical interneuronal control architecture.

3 Evolutionary Learning

One of our objectives has been to supply the digital hardware with enough structural richness for allowing variation and selection operators to act on it so as to generate a broad range of dynamic behaviours. The multilevel aspect of our system plays an important role here. Five levels of evolutionary learning are allowed in this system. Evolution can occur at the levels of initiating signal-flow-level (controlled by readin enzymes), responding to signal-flow-level (controlled by readout enzymes), transmitting signal-flow-level (controlled by cytoskeletal PUs), controlling PU-interaction level (controlled by MAPs), and selecting neuron level (controlled by high-level reference neurons).

A readin enzyme connects a PU with one or more than one external inputs (or sensors). A cytoskeletal signal is initiated when a readin enzyme receives any of these input signals. The connections between readin enzymes and external inputs are subject to change during the course of learning. Adding a readin enzyme will expand the receptive fields of a neuron whereas removing it narrows the fields. Evolution at the level of readin enzymes is implemented by copying (with mutation) the readin pattern of each neuron in best-performing subnets to all comparable neurons in lesser-performing subnets. Variation is realised by randomly adding or deleting readin enzymes during the copy procedure.

Evolutionary procedure at the level of readout enzymes is similar to that at the level of readin enzymes. Adding a readout enzyme will expand the set of patterns recognised by a neuron, whereas removing it reduces the set. Note that evolution at these levels will not modify the pattern of signal flows insides a neuron. However, it will change the patterns of inputs and outputs recognised by a neuron.

Cytoskeletal PUs are responsible for transmitting cytoskeletal signals. Different configurations of PUs represent different patterns of signal flows. Evolution at the level of cytoskeletal PUs is implemented by copying (with mutation) the PU configurations of the neurons in the best-performing subnets to those of comparable neurons in lesser-performing subnets. Variation is implemented by randomly adding or deleting part of the PU configurations during the copy procedure.

Evolutionary procedure at the level of MAPs is similar to that at the level of cytoskeletal PUs. Adding an MAP will allow a cytoskeletal signal to interact its neighbouring PUs, which in turn might initiate a new signal flow. Deleting an MAP will disable the interaction. Unlike these two levels (readin and readout), evolution will alter the pattern of signal flows inside a neuron at these levels without making any changes of the patterns of inputs and outputs.

Four competitive subnets and one reproduction subnet are used. The initial patterns of readin enzymes, readout enzymes, MAPs, and PU-types of the reproduction subnet are randomly decided. For each evolutionary learning cycle, there are three steps. Firstly, we copy (with variation) the patterns of readout enzymes, readin enzymes, MAPs, and PU-types of the reproduction subnet to each of these subnets. The variation is made for the readout pattern in the first subnet, the readin pattern in the second subnet, the MAP pattern in the third subnet, and the

PU-type pattern in the fourth subnet. Secondly, we evaluate the performance of each competitive subnet and select the best-performing subnet. Thirdly, the patterns of readout enzymes, readin enzymes, MAPs, and PU-types of the best-performing subnet are copied to those of the reproduction subnet, if the former shows better performance than the latter. The above process is repeated until learning is completed or stopped. Note that the mechanism controlling the evolutionary process does not have to be so rigid. Instead, there are several possible alternatives to train this system. In this study, we simply pick out one of these alternatives and precede our experiments. In the future, it would be interesting to investigate the impacts of varying the number of learning cycles assigned to each level and the level opening sequence on the learning.

Evolution of reference neurons is implemented by copying (with mutation) the patterns of low-level reference neuron activities loaded by the most fit high-level reference neurons to the lesser fit high-level reference neurons. The copy process is implemented by activating a most fit high-level reference neuron, which in turn reactivates the pattern of low-level reference neuron firing. This pattern is then loaded by a less fit high-level reference neuron. Variation is implemented by introducing noise into the copy process. Some low-level reference neurons activated by a most fit high-level reference neuron may fail to be loaded by a less fit high-level reference neuron. Or some low-level reference neurons that are not activated may fire and be “mistakenly” loaded by a less fit high-level reference neuron. In the present implementation, evolutionary learning at the reference neuron level is turned off, as we have not yet implemented it on digital circuits.

4 Application Domains

Our training set comprises forty binary patterns, each consisting of 25 bits. Twenty-six out of these forty patterns are capital letters (A, B, C, …, Z), ten are numeric digits (0, 1, 2, …, 9), and four are arrow signs (←, ↑, →, ↓). These forty patterns are equally divided into four groups: 1000ns, 2000ns, 3000ns, and 4000ns, representing four different types of the system’s response duration. For each of the patterns in the 1000ns group, we expect the ANM system to generate an output at 1000ns immediately after it starts processing the pattern. Similarly, we expect it to generate an output at 2000ns, 3000ns, and 4000ns after it starts processing each of the patterns in the 2000ns, 3000ns, and 4000ns groups, respectively. We note that the above setup can be pictured as different types of robotic motion controls (e.g. Higuchi’s welding robot and myoelectric prosthetic hand).

To apply the system to the training set, the input-output interface comprises 25 receptor neurons and 32 effector neurons. One receptor neuron corresponds to one of the 25 bits in a pattern. The effector neurons are divided into four groups (1000ns, 2000ns, 3000ns, 4000ns), corresponding to the four different pattern groups in the training set. In the current implementation, only the effector neurons in the same group as its input pattern are taken into account. The fitness of the system for a pattern is defined as the difference (the error value) between its expected firing time and the firing time of the effector neuron that is most close to the former. The difference is set at 5000ns if there is not any effector neuron firing before then. The sum of the error values for each pattern represents the fitness of the system for the training set. The smaller the value of the sum, the greater fitness the system has.

5 Experimental Results

Two types of experiments were performed with the ANM system. The first investigated its learning capability and the second experiment examined its noise tolerance capability in dealing with spatiotemporal input changes.

5.1 Learning Capability

The system took slightly more than half a day to process the forty patterns in the training set for a learning cycle. We ran the system for four months and terminated it at cycle 125. Nevertheless, we can let the system run in an indefinite amount of time to receive better results. However, at this point, we would prefer not to continue any further so that we can switch to other experiments.

As shown in table 1, the firing times of seventeen out of the forty patterns are exactly the same as their expected times. In other words, the system is able to perform input/output transformation correctly. And there are six, one, and eleven out of the forty patterns whose firing times are different from their expected times by one (20ns), two (40ns), and three clocks (60ns), respectively. The error is very small upon comparing to the performance at the initial stage (3266ns per pattern on average). Among the forty patterns, the firing times of only five (#2, 10, 13, 31, 39) are significantly different from their expected time, but except pattern 39 the errors are still within acceptable range. In other words, the system could not perform input/output transformation nicely for one out of the forty patterns.

5.2 Noise Tolerance Capability

The following experiment is to examine the system’s noise tolerance capability in dealing spatiotemporal input changes. The system’s structure in the previous section (the one after 125 learning cycles) is used. We hold the system’s structures unchanged, but gradually vary its input patterns in space and time to obverse its corresponding outputs. If the performance of the system degrades gracefully as the extent of variations in inputs increases, this in part supports the system's structures embracing some degrees of gradual transformability.

Table 1: Error for each pattern.

Pattern #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Symbol	0	4	8	C	G	K	O	S	W	↑	1	5	9	D	H	L	P	T	X	↓
Errors (clocks)	0	17	0	0	0	0	0	0	0	18	0	3	35	3	0	0	3	3	0	2
Pattern #	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Symbol	2	6	A	E	I	M	Q	U	Y	←	3	7	B	F	J	N	R	V	Z	→
Errors (clocks)	0	1	1	0	1	3	1	3	3	1	29	0	3	3	1	0	3	0	149	3

5.2.1 Noise in Space

To generate the test sets that vary in space, we make a copy of the training set but alter some bits during the copy process (change a bit into “1” if it was “0” and into “0” if it was “1”). Five levels of variations (noises) are imposed during the copy process: 5%, 7.5%, 10%, 20%, and 30%. For example, at the 5% degree of variation, we mean that each bit has 5% of possibility that might be altered (one or two bits changes per pattern on average). For each degree of variation, three test sets are generated (each with a different random seed).

Experimental result showed that the error increases smoothly as we increase the degrees of variations. That is, the system’s performance degrades gracefully as the degree of noises increases. As shown in figure 4, the error increases only slightly as we impose one or two bit changes on each pattern. Even when we increase the degree of variation to 20% (i.e. five bits changes per pattern on average), the system still demonstrates acceptable results. The error increases significantly (almost double) when the level of variation is set at 30% (i.e. one third of these bits in each pattern are altered). The above implies that the system has good noise tolerance capability in dealing with input changes pertaining to noises in space.

5.2.2 Noise in Time

In the previous experiments, we assume that all signals sent to cytoskeletal neurons arrive simultaneously. In this experiment, the arriving time of each signal is not necessarily the same. Some might arrive later than others by one, two, three, four, and five clocks, corresponding to 20ns, 40ns, 60ns, 80ns, and 100ns, respectively.

Six levels of delay experiments are performed: 2.5%, 5%, 7.5%, 10%, 20%, and 30%. For each level of delay, we run three times (each with a different random seed). At the 2.5% level of delay, we mean that each signal has 2.5% of possibility that might arrive with a delay. If delays, its actual delay time is randomly chosen from 1 to 5 clocks.

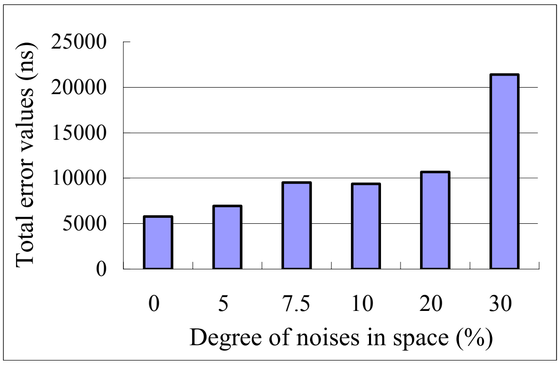


Figure 4: System performance for different degrees of noise in space.

An interesting result is that the system’s outputs do not change accordingly as we increase the levels of delay. Roughly speaking, the error trend exhibits a “Mexican-hat”-shaped curve (i.e. it moves upward at comparatively low delay levels, downward at intermediate delay levels, and in a flat manner at high delay levels) as we increase the delay level. For example, at the 2.5% delay level, the system’s error value is roughly the same as the case with no delay. As we increase the delay level from 2.5% to 7.5%, the change in errors becomes more obvious but is still within the acceptable range, when compared to the error value at the initial stage.

Somewhat out of our expectation is that the system’s error value does not go up as we increase the delay to the 10% level (we note that the error value in this case is almost the same as the case with no delay). To look into this issue further, we perform two more experiments by increasing the maximum delay time from 100ns to 300ns, and then to 500ns. As shown in figure 5, the results illustrated that the general trend in the 300ns case is roughly the same as that in the 100ns case. In the 500ns case, the system’s output changes slightly greater than the previous two cases, but basically exhibits the same trend.

We note that, to fire a neuron, it requires two or more signals to integrate together within a short period of time. Postponing any one of these signals would make signal integration impossible. However, signal integration becomes possible if we simultaneously postpone two signals, as they might integrate at a later time. This might explain why the system’s outputs do not change significantly when we increase the delay level to 10%, 20%, and even to 30%. We note that in some cases the system error (e.g. the 2.5%, 10%, and 30% levels) is slightly lower (better) than that in the case with no delay. The result might be due to the luck that postponing some input signals makes it feasible to adjust the timing of signals inside a neuron, which provided better fit to its assigned tasks. The above result implies that the system has good generalisation capability in dealing with temporal changes in inputs. In other words, not only does the system recognise a set of patterns in its training phase, but also includes its family sets, the sets of patterns that vary in space and time.

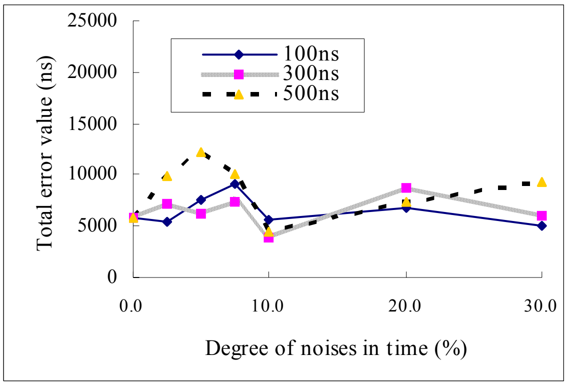


Figure 5: Comparison of total errors for different degrees of noise in time.

6 Conclusions

Our results show that the system demonstrates good noise tolerance capability. That is, its outputs degrade gracefully as we increase the degrees of varying its inputs in space. An interesting finding is that the system's outputs do not change accordingly as we continue increasing the degrees of delaying its inputs. This is because that delaying some inputs may alter a neuronal firing activity while delaying several inputs may not alter it as two delayed signals may integrate at a later time (without doubt, this will change its firing timing). The above results demonstrate that this system has good noise tolerance capability in dealing with spatiotemporal changes in inputs, implying that it possesses an adaptive surface that facilitates evolutionary learning.

7 Acknowledgements

This research was supported in part by the R.O.C. National Science Council (Grant NSC 95-3113-P-224-003).

References

[1] M. Conrad, "Bootstrapping on the adaptive landscape", *BioSystems*, 11, pp 167-182 (1979).

[2] X. Yao, "Following the path of evolvable hardware", *Comm. of the ACM*, 42(4), pp 47-49 (1999).

[3] M. Sipper and E.M.A. Ronald, "A new species of hardware", *IEEE Spectrum*, 37(3), pp 59-64 (2000).

[4] T.W. Gordon and P.J. Bentley, "Towards development in evolvable hardware", *Proc. NASA/DoD Conference on Evolvable Hardware*, Alexandria, VA (2002).

[5] H. Hemmi and J.S. Mizoguchi, "Development and evolution of hardware behaviors", *Lecture Notes in Computer Science: Towards Evolvable Hardware*, 1062, pp 250-265, Springer-Verlag (1996).

[6] J.F. Miller, D. Job and V.K. Vassilev, "Principles in the evolutionary design of digital circuits Part 1", *Journal of Genetic Programming and Evolvable Machines*, 1(1), pp 8-35 (2000).

[7] T. Kalganova and J.F. Miller, "Evolving more efficient digital circuits by allowing circuit layout evolution and multi-objective fitness", *Proc. 1st NASA/DoD Workshop on Evolvable Hardware*, Pasadena, CA (1999).

[8] R. Zebulum, M.A. Pacheco, and M. Vellasco, "Evolvable systems in hardware design: taxonomy, survey and applications", *Proc. International Conference on Evolvable Systems: From Biology to Hardware*, Tsukuba, Japan, pp 250-265 (1996).

[9] J.F. Miller, "On the filtering properties of evolved gate arrays", *Proc. 1st NASA/DoD Workshop on Evolvable Hardware*, Pasadena, CA (1999).

[10] J.-C. Chen and M. Conrad, "Learning synergy in a multilevel neuronal architecture", *BioSystems*, 32, pp 111-142 (1994).

[11] J.-C. Chen and M. Conrad, "Pattern categorization and generalization with a virtual neuromolecular architecture", *Neural Networks*, 10(1), pp 111-123 (1997).

[12] J.-C. Chen and M. Conrad, "Evolutionary learning with a neuromolecular architecture: a biologically motivated approach to computational adaptability", *Soft Computing*, 1(1), pp 19-34 (1997).

[13] J.-C. Chen, "Data differentiation and parameter analysis of a chronic hepatitis B database with an artificial neuromolecular system", *BioSystems*, 57, pp 23-36 (2000).

[14] J.-C. Chen and R.-D. Chen, "Toward an evolvable neuromolecular hardware: a hardware design for a multilevel artificial brain with digital circuits", *Neurocomputing*, 42, pp 9-34 (2002).

Detection and Isolation Method for Operator Failure by Unknown Input Observer

Hwan-Seong Kim[1], Ngoc Hoang Son Tran[1], Seung-Min Kim[2]
[1]Department of Logistics,
Korea Maritime University, Busan, Korea
[2]Department of Mechatronics Examination,
The Korean Intellectual Property Office, Daejeon, Korea
kimhs@hhu.ac.kr, kipo0070@kipo.go.kr

Abstract

In this paper, we suggest a fault detection and isolation method for detecting the operator failures by using observation technique. The suggested algorithm is derived by conventional sensor/actuator fault detection method. Firstly we assume that operator failures can effect to the human work as an external input term. With the assumption, we suggest a human work model with operator failures. Secondly, we introduce an unknown input observer with proportional and integral gains. The characteristic of observer which estimating the external disturbance signal without exact input information is shown and the condition of realisation is proposed for detecting the operator failure. Lastly by using simulation of gantry crane operation, we verify that the observer can detect the operator failure, and isolate the failure by using the estimated fault magnitude, which induced by the observer's internal signal.

Keywords: operator failure, observer, model, fault detection, isolation, gantry crane

1 Introduction

Nowadays, it is increasing numbers of industrial systems with high techniques and automation, especially in modern port systems and container terminal systems. Generally in the systems, almost of all tasks have still been operated by operator who can manage and execute the work. However during the works, the operator might cause some mistakes, and if they take longer then there will occur an accident. So it is necessary to develop the monitoring and the diagnosing systems for human operation during work process.

For generic approach for assessment of human error, there are three goals, namely: the human error identification is first step to reduce human error in systems. The second step can provide the quantification of errors that might be needed to construct a safety case. The final step will develop error databases to reduce human error.

These steps are related on the identification of model of human behavioural by analysing the neuro-muscular characteristics of the human operator to design satisfactory manually controlled systems. Some of the works were mentioned by Shinners [1], Malek et al. [2], Yehia [3], Martens [4]. Their approach was to consider the human as an inanimate servomechanism with a well-defined input and output.

Over the years, the evolution of the control-theory paradigm for the human controller or operator parallelled the development of new synthesis techniques in feedback control. Thus, "optimal control models (OCMs)" of the human operator appeared as linear quadratic Gaussian (LQG) control system design techniques were being developed. "Fuzzy controller" models and "H infinity" models of the human operator closely followed the appearance of these design techniques [5]-[10].

These models seek to account for many of subsystem characteristics of the human operator by assigning transfer functions to the different subsystems involved. These subsystems and their interconnections are postulated on the basis of physiologically isomorphic considerations.

Especially, the application of the time-series analysis to this problem was first introduced by Shinners who developed autoregressive moving-average (ARMA) models of data collected from human operators involved in compensatory tracking experiments using band-limited white noise inputs (bandwidth=1.5Hz). His results showed that all operators exhibited a time delay 0.2s and the discrete transfer functions that represent their dynamics had one zero and two poles. Based on the analysis of model residuals, he concluded that the human operator is a generator of seasonal (rhythmic) characteristics during tracking of random inputs.

Generally, an importance step in the modelling of human operator (HO) dynamics using the time-series approach is to identify the model order (i.e., the number of poles and zeros in the model). Several model order determination criteria, commonly used in systems identification were critically evaluated in their ability to estimate the order of a simulated autoregressive model (AR) with parameters obtained from input-output data of HO during a compensatory tracking experiment [11]-[14].

On the other hand, the detection and the isolation of actuator and/or sensor failures have received considerable attention during the last two decades in both theoretical researches and practical applications [15]-[18]. Although several works are studied on fault detection and isolation (FDI) observers [19]-[21] and residual generators [17], those are concerned with only the partial detection and isolation of actuator failures or sensor failures. In case of FDI observer, multiple observers have been used to determine the localisation of faulty elements of the process, where unknown-input observers are constructed as many as the number of inputs [19].

Recently, a fault diagnosis method to detect and isolate the actuator and sensor failures including the magnitude of fault was proposed by using multiple proportional integral (PI) observers [22]. It was based on intensive use of knowledge on the characteristics of PI observer, which estimates and cancels the step actuator failures.

In this paper, we suggest a fault detection method for operator failures by using observation technique. The suggested algorithm is extended by conventional sensor/actuator fault detection method. Firstly we assume that the operator failure effects to the human work operation as an external input signal. With assumption, we suggest a human work model with operator failure.

Secondly, we introduce an unknown input observer with proportional and integral gains. And the characteristic of observer which estimating the external signal without exact input is shown and the realisation condition is proposed for detecting the operator failure.

Lastly by the simulation for gantry crane operation, we verify that the observer can detect the operator failure and estimate its magnitude from given internal signal perfectly, where the human model obtained previous research works [23] is used.

2 Design of Observer

2.1 Problem Statements

We assume that an operator work model is described by linear time-invariant model as:

$$\dot{x}(t) = Ax(t) + Bu(t) + Dd(t) \tag{1a}$$

$$y(t) = Cx(t) \tag{1b}$$

where $x(t) \in \mathrm{R}^n$ denotes state vector, $u(t) \in \mathrm{R}^m$ input vector, $y(t) \in \mathrm{R}^p$ output vector, and $d(t) \in \mathrm{R}^q$ disturbance vector, respectively. Matrices $A, B, C,$ and D are appropriate dimensions and fair (C, A) is observable.

In (1), the input $u(t)$ includes the all of control information that recognised by operator during the operations. The output $y(t)$ denotes the operation outputs such as the joystick and/or button signals by operator's control resulted by learning, experiments, knowledge, know-how, etc. And the disturbance $d(t)$ includes the uncertainties for work environment change, changes of operator's work condition, and changes of weather.

In this paper, we suggest that an operator have a mistake from wrong information and it makes the operator failure to be taken. From the mistake, the systems would not work normally. In operator work model, the wrong information will be added into the input term. For this, we define $u(t)$ is normal condition input and $\bar{u}(t)$ is abnormal condition input which include the wrong information. From the definition, the input term in (1a) will be modified as

$$\bar{u}(t) = u(t) + a(t) \tag{2}$$

where $a(t) \in \mathrm{R}^m$ denotes the wrong information which make the mistake by operator and it will be added to input term in (1a).

By considering (2), (1) will be changed as the following model.

$$\dot{x}(t) = Ax(t) + Bu(t) + Ba(t) + Dd(t) \tag{3a}$$

$$y(t) = Cx(t) \tag{3b}$$

The aim of this paper is to detect and isolate the operator failures that included into the input term such as $a(t)$ by using observer techniques perfectly.

2.2 Structure of Observer

In this paper, we deal with an observer included proportional and integral gains. The structure of observer is given as follows:

$$\dot{z}(t) = \hat{A}z(t) + \hat{B}y(t) + \hat{J}u(t) + \hat{H}\omega(t) \tag{4a}$$

$$\hat{x}(t) = \hat{C}z(t) + \hat{D}y(t) \tag{4b}$$

$$\dot{\omega}(t) = y(t) - C\hat{x}(t) \tag{4c}$$

where $\hat{x}(t) \in \mathrm{R}^n$ denotes the estimated state vector, $\omega(t) \in \mathrm{R}^p$ output error, and matrices $\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{H}$, and $\hat{J}$ are appropriate dimension for observer.

Definition 1: For operator work system in (1) and/or the system with failure in (3), the system (4) is said to be an observer if and only if

$$\lim_{t\to\infty} e(t) = 0, \quad \forall x(0), z(0), u(\cdot)$$

$$\lim_{t\to\infty} \omega(t) = 0, \quad \forall \omega(0)$$

where $e(t) = \hat{x}(t) - x(t)$ denotes the estimated state error.

Under the Definition 1, we have the following lemma which describes the relation between the system and the observer.

Lemma 1: The system (4) is an observer for the system (3) with an unknown external input if

$$\operatorname{Re} \lambda_i \begin{bmatrix} \hat{A} & \hat{H} \\ -C\hat{C} & 0 \end{bmatrix} < 0, \quad i = 1, \cdots, n+p \tag{5}$$

and there exist a matrix $U \in \mathrm{R}^{n\times n}$ such that

$$\hat{A}U + \hat{B}C = UA \tag{6}$$

$$\hat{J} = UB \tag{7}$$

$$UD = 0 \tag{8}$$

$$\hat{C}U + \hat{D}C = I_n \tag{9}$$

where $\lambda_i[\cdot]$ denotes the eigen values of the matrix.

The problem of designing the observer for the operator work model with unknown disturbance input is reduced to find the matrices $\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{H}, \hat{J}$ and U such that the conditions of Lemma 1 are satisfied.

In the next section, we will show the fault detection characteristic and the realisation condition for the observer.

3 Detection of Operator Failure

3.1 Characteristics of Operator Failure Detection and Isolation

Consider the dynamics of system with the operator failure and interconnected unknown disturbance. By using the estimation error, we have the following error dynamics from (3) and (4a).

Let us define an estimation error $\varsigma(t)$ as

$$\varsigma(t) = z(t) - Ux(t) \tag{10}$$

Then, the dynamics of this error (10) obeys

$$\dot{\varsigma}(t) = \hat{A}\varsigma(t) + (\hat{A}U + \hat{B}C - UA)x(t)$$

$$+UDd(t) + (\hat{J} - UB)u(t) + \hat{H}\omega(t) + UBa(t) \tag{11}$$

Also, (4b) leads to

$$\hat{x}(t) = \hat{C}\varsigma(t) + (\hat{C}U + \hat{D}C)x(t) \tag{12a}$$

$$\dot{\omega}(t) = C(x(t) - \hat{x}(t)) \tag{12b}$$

Also by applying Lemma 1, the observer leads to

$$\dot{\varsigma}(t) = \hat{A}\varsigma(t) + \hat{H}\omega(t) + UBa(t) \tag{13}$$

$$\hat{x}(t) = \hat{C}\varsigma(t) + x(t) \tag{14}$$

$$\dot{\omega}(t) = -C\hat{C}\varsigma(t) \tag{15}$$

Let $\xi(t)$ be defined as

$$\xi(t) = a(t) - \omega(t) \tag{16}$$

Under the assumption of step operator failures (because the operator failure does not change frequently) $\dot{a}(t) = 0$, we obtain

$$\dot{\xi}(t) = -\dot{\omega}(t) = C\hat{C}\varsigma(t) \tag{17}$$

If we design the matrix $\hat{H}$ as

$$\hat{H} = UB \tag{18}$$

Then, we get

$$\dot{\varsigma}(t) = \hat{A}\varsigma(t) + \hat{H}\xi(t) \tag{19}$$

$$\dot{\xi}(t) = C\hat{C}\varsigma(t) \tag{20}$$

or

$$\begin{bmatrix} \dot{\varsigma}(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{H} \\ -C\hat{C} & 0 \end{bmatrix} \begin{bmatrix} \varsigma(t) \\ \xi(t) \end{bmatrix} \tag{21}$$

Under the condition, $\omega(t) \to 0$ and $\varsigma(t) \to 0 (t \to \infty)$. Therefore, the operator failure $a(t)$ can be estimated as

$$\hat{a}(t) = \omega(t) \tag{22}$$

From the above statements, the magnitude of operator failure can be estimated and the failure is effectively isolated in the system with unknown external disturbance. Based on the above statement, we get the following theorem.

Theorem 1: The operator failure is detected and isolated effectively by using observer (4) for operator work model with unknown external disturbance (3), if the condition of Lemma 1 is satisfied.

3.2 Realisation of Observer

To realise the observer, we let $\hat{C} = I_n$ for simplicity. Then, from (9), we obtain

$$U = I_n - \hat{D}C \tag{23}$$

By substitution of (23) into (6), we have

$$\hat{A} = UA - KC \tag{24}$$

$$\hat{B}=\hat{A}\;\hat{B}+K \tag{25}$$

where $K=\hat{B}-\hat{A}\;\hat{D}$.

Substituting (23) into (8), we have

$$\hat{D}CD=D \tag{26}$$

In order to guarantee the matrix $\hat{D}$ satisfying (26), the following condition should be hold.

$$\text{rank } CD=\text{rank } D=q \tag{27}$$

The condition (26) requires that $p>q$, i.e., the number of measured output must be greater than or equal to that of the external disturbance input.

The general solution (26) can be obtained as

$$\hat{D}=D(CD)^{+}+G(I_p-CD(CD)^{+}) \tag{28}$$

where the superscript + indicates the generalised inversion and G is an arbitrary matrix.

By substituting (28) into (23), we can get

$$U=(I_n-GC)\left\{I_n-D(CD)^{+}C\right\} \tag{29}$$

From the above equation, there exists a matrix G, which makes (I_n-GC) non-singular, and then the rank $U=n-q$.

Since rank $D=q$, there exists the left inverse of matrix D, i.e. $D^{+}D=I_q$.

Under the condition of rank $U=n-q$, we have $\text{Ker}U\cap\text{Ker}D^{+}=0$, i.e.

$$\text{rank}\begin{bmatrix}U\\ D^{+}\end{bmatrix}=n \tag{30}$$

Then, we have the following relation

$$\text{rank}\begin{bmatrix}sI_n-A & D\\ C & 0\end{bmatrix}=\text{rank}\begin{bmatrix}sI_n-UA\\ C\end{bmatrix}+q \tag{31}$$

Consequently, for $\forall s\in \mathrm{C}$, where C denotes the complex space,

$$\text{rank}\begin{bmatrix}sI_n-A & D\\ C & 0\end{bmatrix}=n+q,\quad \forall\in \mathrm{C} \tag{32}$$

which means that the invariant zeros of the system $(A,D,C,0)$ must be stable. From the above statements, we summarise the following theorem.

Theorem 2: The observer (4) for the operator work model (3) can be realised if

$$\text{(i) rank } CD=\text{rank } D=q \tag{33}$$

$$\text{(ii) rank}\begin{bmatrix}sI_n-A & D\\ C & 0\end{bmatrix}=n+q,\quad \forall\in \mathrm{C} \tag{34}$$

Remark 1: The eigen values of the observer depend upon the arbitrary matrix G and the matrix K significantly.

4 Simulation and Results

4.1 Operator Work Model

In simulation, we use the previous research works [23] for obtain the operator work model, where the gantry crane simulator is used. To obtain the operator work model, we have to construct the human operation model for gantry crane system. Figure 1 shows the basic human work model that represents the container transportation from the initial position to the reference position, where the container should be avoid the given obstacles.

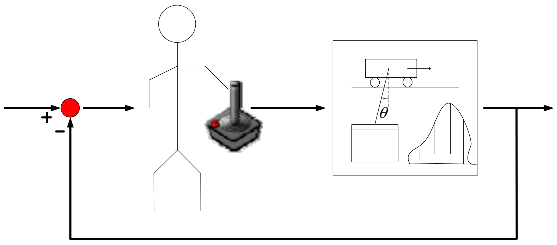


Figure 1: Representation of human operation for gantry crane system.

In figure 1, the operator operates the joysticks to control the trolley position and hoist cable length by through the motor inverters, in respectively. In this case, the operator can watch the container movement and its sway angle. If operator recognises error between desired and actual position of container, then the operator decide to control by using joystick for compensating the error. The signals from joystick, horizontal direction and vertical direction, are proportional rate of change of the forces on trolley and cable, in respectively.

By using the gantry crane operation, we have to achieve the human operation data such as joystick angular, gantry crane operations such as trolley position, container position, sway angle, cable length etc., and desired container transportation trajectory, respectively. For this, we can make a block diagram for human works model as in figure 2.

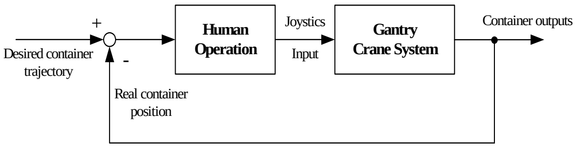


Figure 2: A block diagram for human work model.

For identify the human work model for gantry crane system, we have to divide the input and output factors from the above data. In here, we assume the desired container trajectory will be given by using optimal control method or other procedures. For human operation, the input term is given as desired container trajectory, real container position, trolley position, sway angle, and container cable length. And the joystick angles for trolley and hoist control are given as in input terms.

In simulation, we make three work trajectory paths for one person and take a human operation data with 80ms sampling time.

4.2 Results and Considerations

In simulation, the response results for the gantry crane system are shown in figure 3 for two cycle of work's operation.

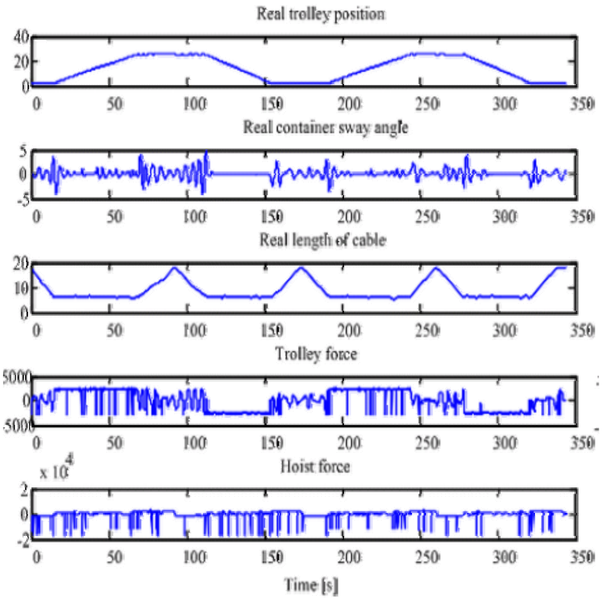


Figure 3: Reponses of container crane in normal case.

For verify the suggested observer to estimate the operator failure for detection and isolation, we assume that the operator have a mistake a trolley joystick for 5 second at 50[s]. In this case the simulation results are shown in figure 4, where the upper and down figures show the simulation results for trolley and hoist operation, respectively.

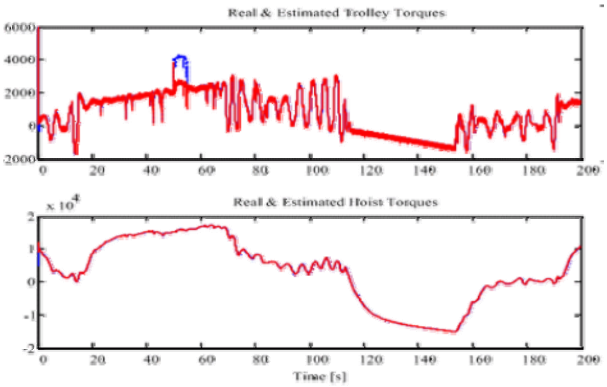


Figure 4: Reponses of container crane with human faults.

As shown in figure 4, the hoist values are almost same, but the trolley values are not matched especially at 50[s]. It shows that operator failure is appeared at 50[s].

To detect the exact operator failure, we take the observer signal $\varpi(t)$ in (22) and show it in figure 5.

The estimated operator failures, trolley and hoist, are shown in figure 5. As results, we know that the operator failure is detected for trolley part at 50[s].

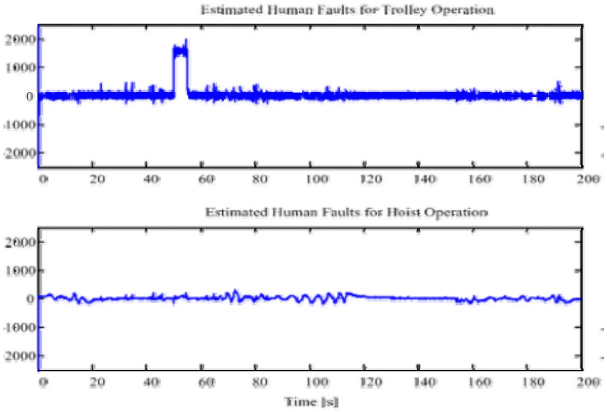


Figure 5: Estimated human faults for trolley and hoist.

Therefore from the results, we verified that the suggested observer have an effect on detection and isolation for the operator failure.

5 Conclusions

In this paper, we have suggested a fault detection and isolation method for operator failures by using observation technique. The suggested algorithm is derived by previous works result for sensor/actuator fault detection method.

Firstly we assumed that the operator failure effects to the human work operation as an external input signal. With assumption, we suggested a human work model with operator failure.

Secondly, we introduced an unknown input observer with proportional and integral gains. And the characteristic of observer which estimating the external signal without exact input is shown and the realisation condition is proposed for detecting the operator failure.

Lastly by the simulation for gantry crane operation, we verified that the observer can detect the operator failure and estimated its magnitude from given internal signal perfectly.

In the future work, the detail operator failure should be researched such as emotion and stress that includes environment changes.

6 Acknowledgements

The first author would like to acknowledge the financial assistance provided by Korea Research Foundation for the 21st Century.

7 References

[1] S.M. Shinners, "Modelling of human operator performance utilizing time series analysis", *IEEE Trans. Syst. Man Cybern.*, SMC-4(5), pp 446-458 (1947).

[2] A.A. Malek and V.Z. Marmarelis, "Modelling of task-dependent characteristics of human operator dynamics pursuit manual tracking", *IEEE Trans. Syst., Man, and Cybernetics*, 18(1), pp 163-172 (1988).

[3] M.E. Yehia, "Human operator behaviour modelling using non-linear identification techniques", *Proceeding of Instrumentation and Measurement Technology Conference*, pp 211-216, Massachusetts (1995).

[4] D. Martens, "Neural networks as a tool for the assessment of human pilot behaviour in wind shear", *Aerospace Science and Technology*, 3(1), pp 39-48 (1999).

[5] A.E. Bryson and J.Y.C. Ho, *Applied Optimal Control: Optimisation, Estimation and Control*, John Wiley & Sons, New York (1975).

[6] G.F. Franklin, J.D. Powell and M. Workman, *Digital Control of Dynamic Systems*, Addison Wesley, California (1998).

[7] W.S. Levine, *The Control Handbook, Vol. II*, IEEE Press (1996).

[8] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control, Analysis and Design*, John Wiley & Sons, New York (1996).

[9] P.J. Antsaklis and N.M. Anthony, *Linear Systems*, McGraw-Hill, Singapore (1997).

[10] K. Zhou and J.C. Doyle, *Essentials of Robust Control*, Prentice Hall , New Jersey (1998).

[11] N.K. Sinha and B. Kuszta, *Modeling and Identification of Dynamic System*s, Van Nostrand Reinhold, New York (1983).

[12] B.J. Glass and C.M. Wong, "A knowledge-based approach to identification and adaptation in dynamical systems control", *Proceedings 27th IEEE Conference on Decision and Control*, Austin, TX, pp 881-886 (1988).

[13] M.O. Efe and O. Kaynak, "Neural-fuzzy approaches for identification and control of non-linear systems", *Proceedings of the IEEE International Symposium on Industrial Electronics*, Slovenia, pp TU2-TU11 (1999).

[14] L. Ljung, *System Identification: Theory for Users*, Prentice Hall PTR, New Jersey, (1999).

[15] R.N. Clark, "Instrument fault detection", *IEEE Trans. Aero. and Electr. Syst.*, AES-14, pp 456-465 (1978).

[16] P.M. Frank and L. Keller, "Sensitivity discriminating observer design for instrument failure detection", *IEEE Trans. Aero. and Elect. Syst.*, AES-16, pp 456–465, (1980).

[17] E.A. Garcia and P.M. Frank, "A novel design of structured observer-based residuals for FDI", *Proceedings of the American Control Conference*, San Diego, pp 1341-1345 (1999).

[18] J.P. Park and C.W. Lee, "Diagnosis of faults in rolling element bearings by using directional spectra of vibration signals", *KSME Int. Journal*, 13(1), pp 63-73 (1999).

[19] N. Viswanadham and R. Srichander, "Fault detection using unknown-input observer", *Control Theory and Advanced Technology*, 3(2), pp 91-101 (1987).

[20] M. Hou and P.C. Muller, "Fault detection and isolation observers", *Int. J. Control*, 60(5), pp 827-846 (1994).

[21] H. Wang and S. Daley, "Actuator fault diagnosis: an adaptive observer based technique", *IEEE Trans. on Automatic Control*, 41(7), pp 1073-1078 (1996).

[22] H.S. Kim, S.B. Kim and S. Kawaji, "Fault detection and isolation of system using multiple PI observers", *Journal of the Korean Society of Precision Engineering*, 14(2), pp 41-47 (1997).

[23] H.S. Kim, N.H.S. Tran and S.H. Kim, "Design of human works model for gantry crane system", *International Journal of Navigation and Port Research*, 29(2), pp 135-140 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Motion Control of a Chewing Robot of 6RSS Parallel Mechanism

J. D. Torrance, J.-S. Pap, W.L. Xu, J.E. Bronlund and K.D. Foster
School of Engineering and Technology, Massey University
Palmerston North/Auckland/Wellington, New Zealand
j.d.torrance@massey.ac.nz

Abstract

Human mastication is a complex series of movements involving 6 degrees of freedom. A robotic device to mimic the human mastication action is of interest to the food industry to understand masticatory function-food interactions and as an aid in helping to develop new foods. A prototype chewing robot of 6-RSS parallel mechanism has been constructed, and is a physically functioning robot. Real human mastication motion has been tracked and served as a target for the robot to reproduce. A control system based around a Galil motion control card has been formed with the capabilities needed to achieve the motion of human mastication. A closed-form solution to the inverse kinematics for the robot has been found and trajectory tracking motion control has been attempted on the physical robot.

Keywords: mastication, chewing robot, parallel mechanism, motion control, trajectory tracking

1 Introduction

Assessing properties of foods is of great importance to the food industry. For foods such as pasta and rice, texture perception is particularly important, and understanding the relationship between food structure and food texture perception is of increasing importance for companies wishing to produce textually attractive food products [1]. Rheological tests on foods can be used to assess food texture; however they are difficult to correlate with sensory perception [2]. These methods can not be used to understand how food texture changes during mastication. Tests that better mimic the forces and conditions existing in the mouth have the greatest potential to provide useful information concerning food texture [2].

A chewing robot made with the aim of assessing masticatory efficiency has been produced at Waseda University, Japan. The WJ-3 robot used nine artificial muscle actuators which consisted of nine DC motors together with encoder, tacho and force sensors [3]. The WJ-3 has three degrees of freedom and implements artificially produced trajectories for clenching and grinding. The robot employs a hybrid control with the position control during non-occluding phases and the force control during the occluding phase when the jaw is in contact with the food. Control is carried out at two levels, with the lower being concerned with position and force control of each actuator and the higher level control supervising and passing the motion pattern of the mandible to the lower level control [3].

This robot successfully implemented predefined chewing trajectories by specifying two of the three degrees of freedom, and controlling force in one degree of freedom during the occlusion phase. The control scheme using force control on occluding and position on non-occluding is well suited to food chewing applications, but more degrees of freedom must be controlled in order to mimic the forces associated with human mastication.

Another jaw robot from Waseda University is the WY-5R. This robot was designed for use as a mouth opening and closing trainer. It has six degrees of freedom so can conform to all human jaw motion [4]. The robot uses six linear actuators so does not have any mechanical redundancy. While the robot is able to move in six degrees of freedom, only two degrees of freedom are controlled; more controlled degrees of freedom would be needed in order to recreate the forces involved in mastication. The location of the actuators is unrelated to the biological location of the muscles of mastication, so work done by each actuator can not be related to the biological model and physiological measures during human mastication can not be compared.

A jaw simulator, JSN/2A, was created at Niigata University Japan to produce repeatable jaw movements. The robot uses six wire actuators which are located in similar locations to the biological muscles of mastication. Actuators are activated according to EMG data recorded during open-close movements [5]. Accurate trajectory tracking by this means is very difficult, and while a chewing like motion was able to be produced, the robot does not have the capabilities to chew foods.

At Massey University, New Zealand, the design and construction of a chewing robot of six degrees of freedom have been attempted according to the biological mechanism [6,7,8]. Each masticatory muscle was replaced by a linear actuator; the actuators’ attachment points on mandible and skull were found from biomechanics; and the temporal-mandibular joint (TMJ) was relaxed to move freely. By accurately reproducing human chewing behaviour is it hoped that we may better relate sensory, psychophysiological and instrumental measures of food texture properties.

This paper describes a robot of 6 RSS parallel mechanism newly constructed at Massey University, where each key muscle group involved in mastication is modelled by a RSS spatial linkage [9]. The mechanisms inverse kinematics and motion control are presented. Matlab simulations were completed for inverse kinematics with respect to real chewing trajectories. A control system has been assembled and motion playback has been implemented together with passive force control for a given human chewing trajectory.

The first section of this paper reviewed the state of the art of chewing robotics, and defined the purpose of this study. A full description of the built chewing robot is given in Section 2, where typical chewing motions are outlined, followed by sections on inverse kinematics, motion control system with some trajectory tracking results. Section 6 concludes the paper.

2 The Chewing Robot

A chewing robot of 6 RSS mechanism was constructed recently at Massey University, as shown in figure 1. The actuators that provide motion to the robot are six DC servo motors attached to a reduction gearbox and encoder. A key consideration for its design was that the six actuators were located to replicate the way the biological jaw works by using similar attachment points for the actuators as the human jaw has for its muscles of mastication. Specifications for the design were based on data from tracked human jaw trajectories during chewing and data from the dental industry relating to the maximum force of the human jaw [8].

2.1 The Robotic Linkage

Figure 2 shows a kinematic model of the robot. Because of the need to keep actuators positioned in a similar way to the muscles of the biological model, there was only limited room to fit the actuators. The solution to fit an actuator that could supply enough force was to use six linkage mechanisms seen in figure 2.

The robotic jaw is moved by 6 RSS type actuators, each actuator consisting of a driver unit (Di, i = 1,2,…6), crank (Ci, i = 1,2,…6), and coupler link (Li, i = 1,2,…6). The three joints are a revolute at the actuators ground point where the driving shaft connects to the crank (Gi, i = 1,2,…6), a spherical joint where the crank joins to the coupler linkage (Si, i = 1,2,…6), and a final spherical joint where the coupler linkage joins to the mandible (Mi, i = 1,2,…6). Each driver unit consists of a 60W DC motor, a 66:1 reduction gearbox, and a 500 count encoder.

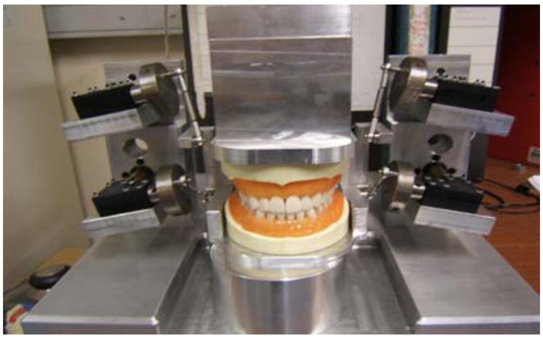


Figure 1: Prototype chewing robot at Massey University, New Zealand.

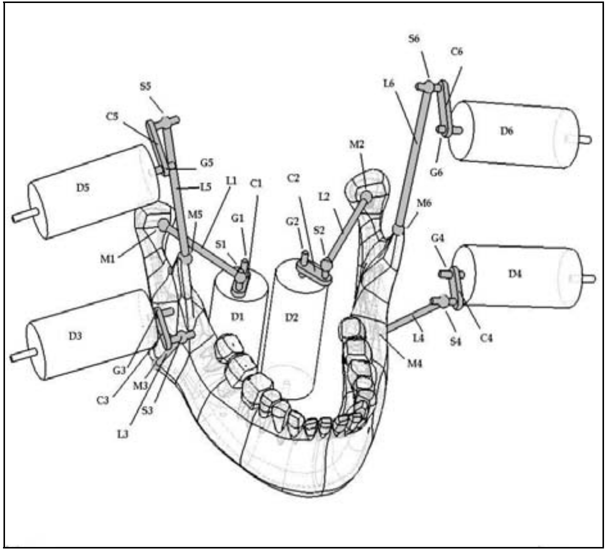


Figure 2: Kinematic model of the chewing robot.

The positioning of actuators to mimic the biological make up of the human jaw allows comparisons to be made between the robot and human chewing actions, for example actuator work compared with EMG data recorded in humans during chewing. It also presents a unique motion control problem where the specially placed actuators need to be co-ordinated and controlled in an analogous way to how the major muscles of mastication are controlled in the biological model. By using six actuators to replace the many muscles of mastication the control problem has been simplified, while still ensuring the robot is capable of performing with six degrees of freedom so that the full range of human jaw motion can be implemented.

2.2 The Physical Design

The chewing robot was designed to meet the kinematic and force requirements of human mastication. The Posselt envelope [9] which describes the shape of the maximum jaw extensions possible in humans was used as a guide together with data gathered from human jaw tracking experiments, which will be described in more detail in the next section. While measuring forces between teeth as humans chew is difficult, some data exists and from it a specification for the robot of 260N across the molar contact point was established [9].

Figure 3 shows the final design in SolidWorks. The frame holding the robot components was made from aluminium to provide adequate strength and stability while not making the whole assembly too heavy. Bevel gearboxes were used to enable compact positioning of motors behind the assembly.

Real denture morphology was necessary to make comparisons to the human system possible. For this a basic study model from Trimut Corporation, Kyoto, Japan, was used to provide realistic morphology with the important 2:1 tooth relationship between mandible and maxilla, and reliable contact point [10]. The teeth were made out of a type of plastic called melamine which is hard enough for initial testing.

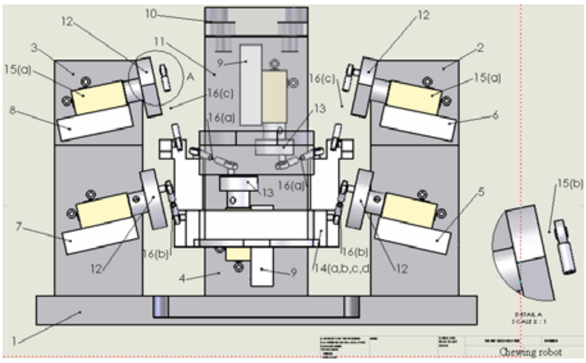


Figure 3: The chewing robot CAD assembly.

2.3 Typical Chewing Motion

Defining a target motion for the chewing robot is important, and an early aim was to track a recorded human chewing trajectory in three-dimensional space.

The Institut National de la Recherche Agronomique (INRA), Clermont-Ferrand, France, measured trajectories with a Carstens 2D Articulograph in two-dimensions [11]. The two dimensions were the horizontal and vertical displacements of the incisor teeth as seen in the frontal plane of the jaw, as shown in figure 4.

More recently, at the Bioengineering Institute at the University of Auckland, New Zealand, a project was undertaken to capture human chewing motion within three-dimensional space. A custom made brace (figure 5) and a motion capturing system, VICON MX, were used to successfully capture chewing data with six degrees of freedom [12].

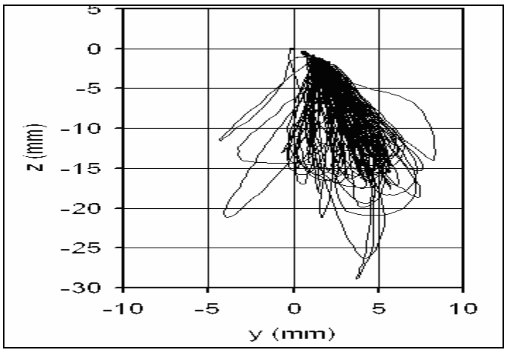


Figure 4: Mastication movement results in frontal plane.

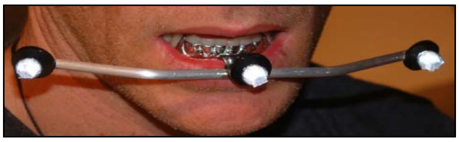


Figure 5: Custom made brace for jaw tracking with six degrees of freedom.

The trajectories recorded with this method were well suited for the chewing robot application as all the three-dimensional movements were specified.

2.4 Motion Compensation for Physiological Difference in Jaws

Jaw size of humans differs from person to person, and the Massey robotic jaw will always differ from the subject's jaw in which the implemented trajectory was tracked. This makes it difficult for the robot to reproduce chewing behaviours of different subjects. In this study, a human chewing trajectory is specified by three reference points on the mandible, which serves as a target motion for the robot. Due to the physiological differences between the human subject and the robot, the lower and upper teeth of the robot could not come in correct contact for the chewing of foods, even though the movement of the lower jaw is precisely followed.

To correct the mismatching of the two sets of jaws, a compensation technique, among many possible methods, could be applied. By this technique, the mismatching, measured by the coordinates of the three reference points in three dimensions, is compensated by adjustment of the six actuations in joint space. This can be expressed by:

$$\Delta\theta = J^{-1} \bullet \Delta p \tag{1}$$

in which Δp is a vector of six dimensional difference in mandibular position and orientation, which can be found from the measured coordinates of the three reference points; $\Delta\theta$ is a vector of six adjustments in the actuation; and J is the Jacobian matrix for differential kinematics of the robot.

The measurement of the mismatching in terms of the three points can be performed in virtual operation of the robot in SolidWorks/CosmosMotion. The fact that the robot was built precisely according to the SolidWorks model makes this method possible.

3 Inverse Kinematics

Having established a target trajectory, the next problem to be solved was the inverse kinematics. Inverse kinematics involves solving the six actuation angles of the six cranks for a given mandible position. The six crank angles as a function of time are required for the low-level closed-loop control of each actuator when implementing trajectory tracking control. This will be explained in more detail in the next section.

The inverse kinematics of the chewing robot requires six calculations for each point recorded in the trajectory of the mandible. The robot being a parallel based mechanism means inverse kinematics are relatively straight forward, and the six calculations for a given mandible pose can be calculated in parallel.

For a given crank the inverse kinematic solution is explicitly expressed as,

$$\gamma = 2 . \tan^{-1}\left[\frac{-i \pm \sqrt{i^2 - 4hj}}{2h}\right] \qquad (2)$$

where γ is the actuation angle of the crank relative to the reference actuation at jaw closed position; i, h and j are constants derived from link mechanism parameters and the mandible pose.

M-files in Matlab where made to find the three constants in (2) for a given mandible pose, and then to solve (2) for each of the six actuators. A final M-file called ‘inversesolve.m’ was written to take a trajectory of points verses time, each containing three translational components and three rotational components, and outputting six angular trajectories, one for each actuator.

Matlab solutions to the inverse kinematics were compared to the trajectories formed from a simulation of the target mandible trajectory in SolidWorks/CosmosMotion, and results were in agreement. With the closed-form solution now found, crank angles can be calculated in real time.

4 Motion Control

4.1 System Architecture

A six axis motion control card, DMC-1860 by Galil, was purchased which is capable of simultaneous PID control of all six actuators. Figure 6 shows the control system set up.

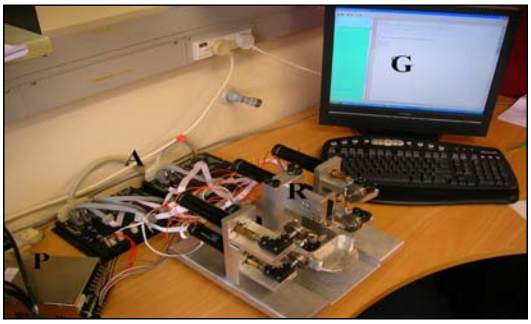


Figure 6: Massey robotic jaw control system.

Encoders with the DC motors and gearbox’s were from Maxon, and are fitted direct to the motor shaft, with a resolution of 0.003 degrees.

The Galil DMC Smart Terminal programme (G in figure 6) provides a means of programming motion to the control card. Commands are specific to the Galil motion control card and are outlined in a lengthy command reference [13].

The amplifiers (A in figure 6) are capable of driving four motors each and use pulse width modulation to output current to the motors proportional to the control signal from the controller. The amplifiers are capable of supplying 10 A per motor, and are powered by a 20 A, 500 W power supply (P in figure 6) which is sufficient for the 60 W motors used.

The motion control card outputs a control signal for each actuator that varies between -10 V and 10 V.

4.2 Motion Control

The first aim for the motion control of the chewing robot is to follow a set trajectory as determined by human jaw tracking data [12]. Closed loop force control during the occluding phase is proposed as the next step, and will require force sensors for feedback.

An alternative to closed-loop force control is passive complaint control. Galil provides a torque limiting function where each actuator can have a maximum torque specified. Using this method a maximum chew force can be specified and mandible motion altered so that specified force between teeth is not exceeded. Details of the implementation of toque limiting to provide passive force control are seen in the next section.

Low level motion control of the robot involves individual control of each of the six actuators. PID control using Galil allows PID gains to be set for each actuator which act on differences between commanded position and position feed back by the encoders. Galil offers PID tuning software, WSDK, which has a variety of tuning methods to find the PID values that give best control for different applications.

Simple PID control is expected to work well on the robot due to the 66:1 gearbox reduction between motor and cranks. This high reduction will minimise the effects of dynamic coupling between cranks and may allow this linear control approach to be taken.

4.3 Trajectory Tracking Control for Chewing

Given a human chewing trajectory (figure 4), the inverse kinematics was used to find individual trajectories of actuators. Low level control of each actuator using PID control in Galil can now be used to implement the 6 actuator trajectories.

4.3.1 PID tuning with WSDK

Galil's servo design kit, WSDK, has a number of automated tuning methods that find the best PID values for a given application. WSDK tunes automatically by trail and error. It sets initial PID values then commands motion depending on the application it is tuning for (usually a step response), analyses the response, then increments to a new PID value. The process takes a number of minutes, and after completion the PID values giving the best response are shown.

Tuning methods were applied to an individual actuator unattached to the mandible (zero load) and point-to-point tuning provided the PID values which lead to least error between commanded and actual position once the trajectory was run.

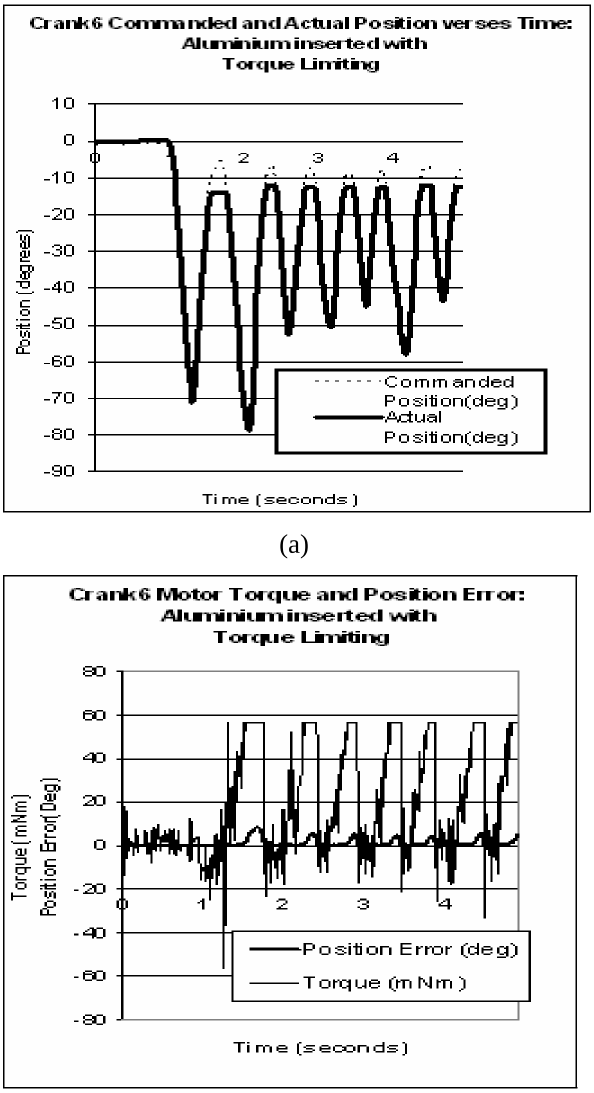
4.3.2 Programming Motion

The trajectory data for each actuator is in the form of a series of points versus time. The time gap between points is 0.01 s. This type of motion suited a mode of motion control called contour mode used by the Galil controller and implemented via the Galil Smart Terminal programme. To programme motion, two arrays must be defined, one for the absolute position data, the other for the difference between consecutive points. The absolute position data is imported explicitly from the output of the inversesolve.m Matlab programme. The difference data array is then calculated in the terminal programme. A constant time between consecutive points is also defined.

Contour mode works by moving the specified difference (in encoder counts) over the fixed time interval. Using this method, commanded synchronised motion of all 6 actuators is possible, giving movement to the robotic jaw.

Figure 7 below shows crank 6 during the chewing of a 6mm aluminium plate. Torque limiting is used on all actuators to limit the force the jaw applies to the plate.

Torque was limited to 55 mNm at the motor, as this was the maximum required torque to allow error free tracking of the jaw during the non-occlusion phases of the chewing. 55 mNm at the motor equates to 3.6 Nm at the crank, and approximately 250 N of force at the link attaching to the jaw, which is similar to reported forces during human chewing [14].



(a)

(b)

Figure 7: Crank 6 chewing aluminium: (a) actual position and commanded position; (b) motor torque and position error of crank 6.

A position error can be seen during occlusion as torque is limited and jaw locks up, unable to follow the commanded trajectory.

5 Conclusion

In this paper a robot designed to implement human mastication has been described. The inverse kinematics for the robot, needed to implement low level control, have been solved and a control system formed. The PID controllers were tuned with WSDK and the trajectory tacking using contour mode and torque limiting in Galil is seen as a promising way of implementing recorded human chewing patterns on food samples.

Future work is to be done on the dynamics of the robot, and calculations formed to find torque limits for a given force at the jaw so different foods can be chewed at different known forces. Performance assessment and advanced PID also need to be looked at.

6 Acknowledgement

The work in this paper is supported by the Foundation of Research, Science and Technology (FRST) grant, New Zealand, under contract number UOAX0406. Thanks are given to O. Rohrle and A. Pullan of the Bioengineering Institute at the University of Auckland, New Zealand, for provision of the mathematical model and chewing data of the mastication system, and I. Ichim and J. Kieser of the School of Dentistry at University of Otago, New Zealand, for providing the accurate replica of a human jaw.

7 References

[1] C. Wilinson, G.B. Dijksterhuis and M. Minekus, “From food structure to texture”, *Trends in Food Science and Technology*, 11, pp 442-450 (2000).

[2] A.V. Cardello, “Sensory-instrument research”, *Cereal Foods Works*, 39, pp 567-569 (1994).

[3] H. Takanobu, T. Yajima, M. Nakazawa, A. Takanishi, K. Ohtsuki and M Ohnishi, “Quantification of masticatory efficiency with a mastication robot”, *Proceedings IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp 1635-1640 (1998).

[4] H. Takanobu, R. Soyama, A. Takanishi, K. Ohtsuki, D. Ozawa and M. Ohnishi, “Remote treatment with mouth opening and closing training robot”, *IEEE Intelligent Robots and Systems*, 1, pp 553-558 (2000).

[5] T. Hayashi, S. Kato, S. Nakajima, Y. Yamada and H. Kobayashi, “Physiological control scheme of jaw simulator JSN/2A for improving reproducibility of open-close movement”, *Proceedings 1st Joint BMES/EMBS Conference: Serving Humanity, Advancing Technology*, Atlanta, p564 (1999).

[6] B. Daumas, W.L Xu and J. Bronlund, “Jaw mechanism modeling and simulation”, *Mechanisms and Machine Theory*, 40(7), pp 821-833 (2005).

[7] W.L. Xu, J. Bronlund and J. Kieser, “A robotic model of human masticatory system for reproducing chewing behaviours”, *IEEE Robotics and Automation Magazine*, 12(2), pp 90-98 (2005).

[8] J.S. Pap, W.L. Xu and J.E. Bronlund, “Design of a biologically inspired robot for foods chewing”, *IEEE Transactions on Industrial Electronics*, in press (2006).

[9] J.S. Pap, *A Chewing Robot Based on Parallel Mechanism: Analyze and Design*, Masters Thesis, Massey University, New Zealand (2006).

[10] Trimunt, *Catalog*, Trimunt Corporation, Kyoto (2001).

[11] K.D. Foster, A. Woda and M.A. Peyron, “Effect of texture of plastic and elastic model foods on the parameters of mastication”, *Journal of Neurophysiology*, 95, pp 3469-3479 (2006).

[12] O. Rohrle, I.A. Anderson and A.J. Pullan, “From jaw tracking towards dynamic computer models of human mastication”, *Proceedings 12th International Conference on Biomedical Engineering*, Singapore (2005).

[13] Galil Motion Control, *Command Reference, Manual Rev. 1.0m*, Galil Motion Control, Inc., Rocklin, California (2003).

[14] C.H. Gibbs, P.E. Mahan, H.C. Lundeen, K. Brehnan, F.K. Walsh, S.L. Sinkewiz and S.B. Ginsberg, “Occlusal forces during chewing: influences of biting strength and food consistency”, *Journal of Prosthetic Dentistry*, 46, pp 561-567 (1981).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Reconfigurable Hierarchical Finite State Machines

Valery Sklyarov, Iouliia Skliarova
Department of Electronics, Telecommunications and Informatics, IEETA
University of Aveiro, Aveiro, Portugal
skl@det.ua.pt, iouliia@det.ua.pt

Abstract

The paper suggests design methods for reconfigurable hierarchical finite state machines (RHFSM), which possess two following important features: 1) they enable the control algorithm to be divided in modules providing direct support for “divide and conquer” strategy; 2) they can be reconfigured both statically and dynamically. Run-time reconfiguration permits virtual control systems to be constructed, including systems that are more complex than capabilities of available hardware. It is shown that RHFSM can be synthesised from specification in form of hierarchical graph-schemes with the aid of the considered in the paper VHDL templates. The results of experiments show correctness of the proposed methods and their applicability for the design of engineering systems.

Keywords: control systems, hierarchical specification, finite state machines, VHDL templates

1 Introduction

Divide and conquer is a challenging design technique for any engineering system, including robotics and embedded applications [1]. Another demanded feature is reconfigurability whose importance was reported in a number of publications [1-5].

Finite state machines (FSM) are probably the most widely used components in digital systems. For many practical applications it is desirable to provide FSM with virtual capabilities, in general, and modifiability, in particular [1]. The objectives of these are very different. For one kind of applications we might want to supply capabilities similar to those provided for general-purpose processors which support the virtual memory mechanism. In particular, this feature allows a device to be constructed on a microchip that does not have sufficient hardware resources to accommodate all the functionality of the device. For another kind of applications it may be desirable to be able to alter the behaviour depending on external events that cannot be predicted in advance (this is very important for self-reconfigurable robots [2,3] and adaptable systems [4]). In some cases we need to provide sufficient flexibility to allow for changes during the debugging stage [1] (for example, to verify alternative or competitive algorithms), etc.

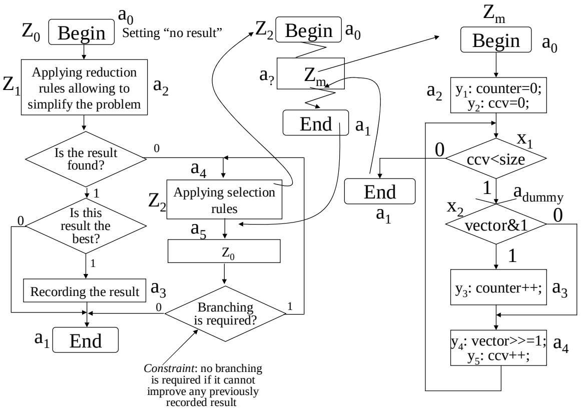
This paper suggests methods of FSM synthesis (with primary emphasis on reconfiguration) that possess two following very important characteristics: 1) supporting the hierarchy and the strategy “divide and conquer”; and 2) permitting statically and dynamically reconfigurable control systems to be developed.

The main distinctive features of the paper comparing with [5-7] are listed below:

- Revision of known results (namely the modular specification and synthesis of hierarchical FSMs) and explicit definition of design templates supporting the strategy “divide and conquer”;
- Structure of RHFSM and reconfiguration methods;
- Advanced techniques for stack operations in RHFSM.

2 Modular Specification of RHFSM

Modular specification can be presented in the form of hierarchical graph-schemes (HGS) [5]. Figure 1 shows an example. Module Z_0 in figure 1 represents a top-level recursive algorithm for solving different optimisation problems over binary and ternary matrices, such as discovering the minimal cover of a binary matrix [8], Boolean satisfiability [9], graph colouring, etc. Such algorithms are required for many engineering problems, including those appearing in robotics and embedded systems [10].



$a_0,a_1,\ldots,a_5$ are HFSM states

Figure 1: Example of modular specification.

One of the tasks of module Z_2 is executing operations over binary and ternary vectors. Examples of such operations are calculating the maximum number of successive 1s (0s) in a given binary vector, testing for orthogonality between two ternary vectors, comparing two vectors, etc. As an example, module Z_m in figure 1 describes a trivial algorithm for counting the number of ones in a given binary vector.

Note that for numerous optimisation tasks over matrices different algorithms have to be executed. For example, for the covering problem one of the required operations is counting 1s [8] and for the satisfiability problem "counting 1s" is not needed but checking for orthogonality of ternary vectors is required [8]. Thus, a possible approach to provide adaptability of the module Z_0 for solving different problems is to reconfigure the module Z_m.

We will divide the considered task into the following two subtasks: static hardware implementation of the modules in figure 1 (see section 3) and dynamic reconfiguration of the module Z_m (see section 4). Finally, we will show how to design hardware, which joins these subtasks and provides advanced operations for modules' collaboration.

3 Synthesis of HFSMs

Methods of HFSM synthesis from HGS specification were proposed in [5] and we will demonstrate these methods just on an example. The following VHDL code describes the stacks (called *M_stack* and *FSM_stack* [5]) that are used as HFSM memory:

```
process(clock,reset)                                    (1)
begin
  if reset = '1' then     -- initialising
  elsif rising_edge(clock) then
        if inc = '1' then
           if -- test for possible errors
           else    sp <= sp + 1; -- sp – stack pointer
                   FSM_stack(sp+1) <= a0;    -- ref1
                   FSM_stack(sp) <= NS;  -- next state
                   M_stack(sp+1) <= NM; -- next module
           end if;
        elsif dec = '1' then  sp <= sp – 1;      -- ref2
         else     FSM_stack(sp) <= NS;
         end if;
  end if;
end process;
```

Here, the signal *inc* is generated in any rectangular node of HGS, which calls another module; the signal *dec* is generated in the *End* node of any module.

The following VHDL fragment describes a template (skeletal code) for combinational circuit of HFSM and the module Z_m from figure 1 is completely specified.

```
process (CM,CS,X) -- current module (CM), state (CS)
begin              -- X = {x1,…,xL} - input signals
  case M_stack(sp) is                                   (2)
    when Z1 =>
            case FSM_stack(sp) is  -- state transitions
              -- and output generation in the module Z1
            end case;
  -- ……………………………………………………
  when Zm =>     -- below the complete code is given
            case FSM_stack(sp) is
                when a0 => Y <= (others => '0');
                      inc <= '0'; dec <= '0'; NS <= a2;
                when a1 => Y <= (others => '0');
                      inc <= '0'; NS <= a1;
                      if sp > 0 then dec <= '1';
                      else dec <= '0';
                when a2 => Y <= "11000"; -- y1 and y2
                       dec <= '0'; inc <= '0';
                       if x1='0' then NS <= a1;
                       elsif x2='0' then NS <= a4;
                       else NS <= a3;
                       end if;
                 when a3 => Y <= "00100"; -- y3
                       dec <= '0'; inc <= '0';
                       NS <= a4;
                 when a4 => Y <= "00011"; -- y4 and y5
                        dec <= '0'; inc <= '0';
                         if x1='0' then NS <= a1;
                         elsif x2='0' then NS <= a4;
                         else NS <= a3;
                         end if;
                  when others => null;
              end case;
    -- repeating for all modules, which might exist
    end case;
  end process;
```

The following VHDL code gives an example of hierarchical module call in the module Z_0:

```
when a2 => dec <= '0'; inc <= '1'; NM <= Z1;
```

4 General Structure of Reconfigurable FSMs

Methods for synthesis of non-hierarchical reconfigurable FSMs (RFSM) were suggested in [6]. They permit to implement reconfigurable modules (such as Z_m) on the basis of RAM blocks in such a way that reloading the contents of RAM blocks enables us to reconfigure the FSM.

Let us consider the basic ideas of the cascaded RFSM model [6], which is composed of RAM blocks, programmable multiplexers (PM) and a register. By modifying the contents of the RAM blocks we can implement any desired behaviour within the scope of predefined constraints [1,6], which are the size R of the RFSM register, the number of RFSM inputs/outputs L/N and the number of reprogrammable levels G [6]. Reprogrammable levels are the primary building blocks of the RFSM combinational circuit. All the required parameterisation can be provided through VHDL *generic* and *generate* statements.

Figure 2 depicts the basic building blocks of RFSM. The PM is composed of a RAM and a multiplexer and is used for selecting appropriate input variables dependently on FSM states. VHDL code for the circuit in figure 2 was presented in [11].

Figure 3 shows a trivial RFSM [6]. It is composed of three blocks: an FSM memory, a state transition RAM (STRAM) and a programmable multiplexer (PM), which permits any input $x_l \in \{x_1,...,x_L\}$ of the PM to be selected in any RFSM state. Clearly we can provide any correspondence between states $a_0,...,a_{M-1}$ and inputs $x_1,...,x_L$. STRAM enables us to generate codes for the next states and outputs. Obviously, any subset of output signals $y_1,...,y_N$ can be generated in any state transition.

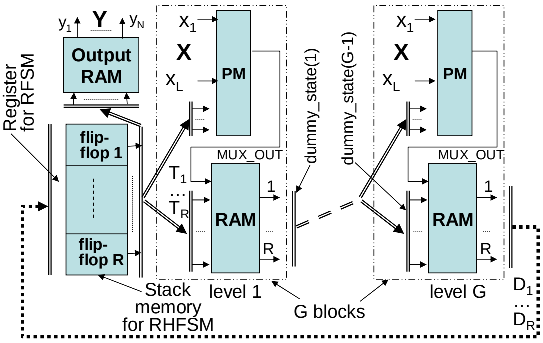


Figure 2: Basic structure of RFSM.

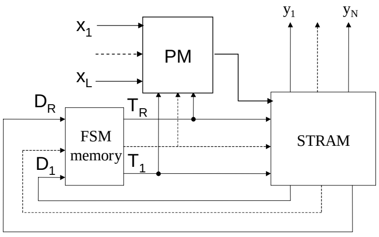


Figure 3: Trivial RFSM with limited capabilities.

By programming PM and STRAM we can implement any desired FSM behaviour within the scope of the predefined constraints. However the circuit depicted in figure 3 has a very significant limitation: any state transition can only be affected by a single input variable. An arbitrary state transition graph can be altered in such a way that allows this constraint to be satisfied by splitting state transitions and inserting dummy states as shown in figure 1 (see the state a_{dummy}). However in the general case this changes the behaviour of the FSM, increases the number M of states, and reduces the speed of state transitions. The solution proposed in [6] permits to overcome the problem. The blocks PM and STRAM are repeated for the required number of levels G, where G is the maximum number of input variables that have to be tested in any state. Dummy states can appear just in between levels (see figure 2), but they are not stored in any register and are considered as an intermediate signals between levels. Thus, any state transition can be executed during a single clock cycle.

5 Structure of RHFSM

Figure 4 depicts the basic structure of RHFSM, which permits to reconfigure the functionality of CC through reloading the RAM blocks. Changes to the modules can be provided through re-switching segments of the same RAM using the most significant bits (MSB) of RAM address space. Thus, MSB in figure 4 permits to select the current module (CM), and the less significant bits (LSB) of RAM address space are used in the same manner as in [11].

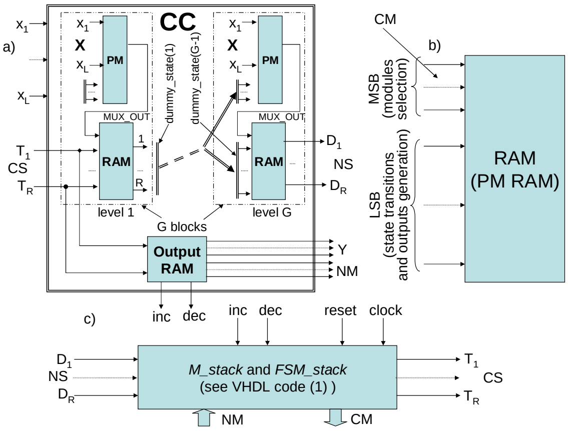


Figure 4: (a) CC of RHFSM, (b) dual-port RAM block and (c) RHFSM stack memories.

Note, that RHFSM in figure 4 can be reconfigured just statically. To provide for dynamic reconfigurability it is necessary to supply additional functionality, which would allow reloading RAM blocks during execution time.

Figure 5 presents a circuit, which implements such functionality. To reconfigure any module it is necessary to reload 2G memory blocks. Indeed, to modify functionality of any level in figure 2 it is necessary to reload the RAM block for PM and the RAM block responsible for state transitions (this block is designated as RAM in figures 2, 4). Activating the proper RAM block is achieved through the respective enable signals (see figure 5). In order to simplify reloading, dual port RAM blocks have been used in such a way that the first port provides a normal functionality and the second port enables the controller shown in figure 5 to change the contents of RAM. Two types of feasible reconfigurations have been considered. The first type, which is a wired reconfiguration, has been implemented and tested.

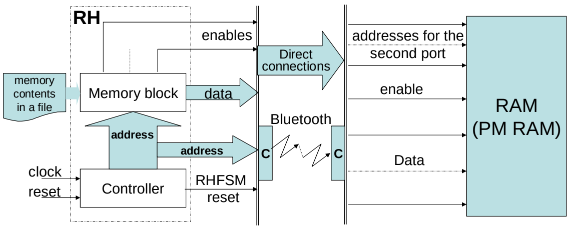


Figure 5: Dynamic reconfiguration of RHFSM.

The reconfiguration handler (RH) in figure 5 is composed of a source for data and a controller. The controller generates memory addresses and copies data from the source to the RAM blocks activated by the appropriate enable signals. When all the blocks are programmed the controller resets the RHFSM and the latter is set into a working mode.

6 Advanced Techniques for Stack Operations in RHFSM

This section shows that the efficiency of the RHFSMs can be significantly improved through the use of the following methods [12]: 1) supporting multiple entry points to sub-algorithms; 2) employing fast unwinding procedure for stacks used as an HFSM memory in case of recursive module invocations; 3) establishing flexible hierarchical returns based on two alternative approaches, which can be chosen depending on the functionality required; 4) the rational use of embedded memory blocks for the design of RHFSM stacks.

6.1 Providing Multiple Entry Points to Sub-algorithms

Figure 6 demonstrates a fragment of a recursive message ordering algorithm. As we can see from figure 6 any hierarchical module invocation, such as that is done in the node a_2, activates the same algorithm once again, starting from the node *Begin* (a_0). Skipping the node a_0 removes one clock cycle from any hierarchical call. However in this case the algorithm in figure 6 must have multiple entry points and a particular entry point will be chosen by the group of rhomboidal nodes enclosed in an ellipse. This possibility is provided by the additional tests performed in the nodes with hierarchical calls (such as a_2 and a_3 in figure 6). The following fragment demonstrates how these tests can be coded in VHDL for the state a_2.

```
when $a_2$ => -- generating outputs and the signal inc
    NM<=$z_1$; -- NM is the next module
    if $x_3$='1' then  NM_FS <= $a_1$;  -- this is because
        -- $x_1$ cannot be equal to 1 after the state $a_2$
    elsif $x_2$ ='0' then NM_FS <= $a_5$;
    elsif $x_4$ ='0' then NM_FS <= $a_2$;
    else NM_FS <= $a_3$;
    end if;
```

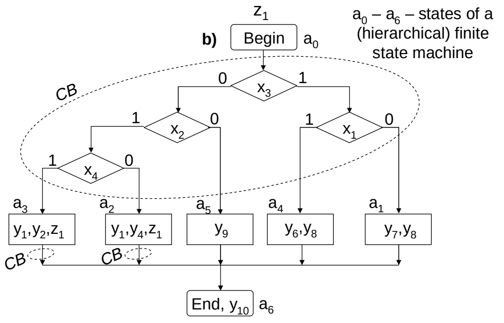


Figure 6: Message ordering algorithm.

Here *NM_FS* is the first state of the next module. The line *ref1* in (1) has to be changed as follows:

```
FSM_stack(sp+1) <= NM_FS;
```

6.2 Fast Stack Unwinding

Since there is just the node *End* after a_2 and a_3, hierarchical activation of any one of the nodes a_1, a_4, a_5 (see figure 6) leads to termination of the module. To implement this termination in [7] the line in (1)

```
if dec = '1' then sp <= sp – 1;
```

is executed repeatedly until the pointer *sp* receives the value assigned at the beginning. In the general case, this value is assigned during the first call of the respective module (such as that depicted in figure 6) following by subsequent recursive invocations of the module. Repeated execution of the line *sp <= sp – 1;* requires multiple additional clock cycles. To eliminate these redundant clock cycles the proposed method of fast stack unwinding is employed. The line *ref2* in (1) is changed as follows:

```
sp <= sp – unwinding;
```

where the signal *unwinding* is calculated as

```
unwinding <= sp - saved_sp + 1;
```

and *saved_sp <= sp* at the first invocation of the module. Thus, redundant clock cycles for hierarchical returns will be avoided.

6.3 Execution of Hierarchical Returns

Hierarchical calls in (1) are carried out as follows:

```
if inc = '1' then                                         (3)
     -- error handling
    sp <= sp + 1;
    FSM_stack(sp+1) <= $a_0$;
    FSM_stack(sp) <= NS;          -- ***
    M_stack(sp+1) <= NM;
```

The line in (3) marked with asterisks sets the code of the next state NS during a hierarchical call. As a result, after a hierarchical return the top register of the *FSM_stack* contains the code of the proper HFSM state (i.e. no additional clock cycle is required). Since the NS is determined before the invocation of a module, the latter cannot affect the state transition, i.e. any possible change of the conditions $x_1,\ldots,x_4$ (see figure 6) in the module cannot alter the previously defined next state. Very often this does not create a problem. However, for some practical applications it is a problem and it must be resolved. The following code gives one possible solution:

```
if rising_edge(clock) then                                (4)
    if inc = '1' then
        -- error handling
          sp <= sp + 1;
          FSM_stack(sp+1) <= NM_FS;
          M_stack(sp+1) <= NM;
```

After a hierarchical return from *NM*, the code (4) sets *FSM_stack* to the state where the hierarchical call of the *NM* was executed. This enables us to provide correct transitions to the next state because all logic conditions that might be changed in the called module

NM have already received the proper values. However, this gives rise to another problem; namely it is necessary to avoid repeating invocation of the same module *NM* and iterant output signals. The following code overcomes the problem:

```
if rising_edge(clock) then                                    (5)
      if inc = '1' then
         -- error handling
            sp <= sp + 1;
            FSM_stack(sp+1) <= NM_FS;
            M_stack(sp+1) <= NM;
      elsif dec = '1' then
         sp <= sp - 1;
         return_flag <= '1';
      else     FSM_stack(sp) <= NS;
               return_flag <= '0';
end if;
```

The signal *return_flag* permits module invocation and output operations to be activated during a hierarchical call and to be avoided during a hierarchical return. Indeed, the *return_flag* is equal to 1 only in a clock cycle when the signal *sp* is decremented (see the code (5) above). As soon as the currently active module is being terminated, the control flow will be returned to the point from which this module was called. Thus, the top of the *M_stack* will contain the code of the calling module and the top of the *FSM_stack* will store the code of the calling state. The *return_flag* enables us to eliminate the second call of the same module. This is achieved with the aid of the following lines that have to be inserted in the code (2):

```
when state_with_module_call => NS <=
         -- testing the conditions and
         -- computation of the next state
   if return_flag = '0' then  inc <= '1';
         -- specifying outputs
         NM<= -- assigning the next module
   else
         inc <= '0';
         Y <= (others => '0');
   end if;
```

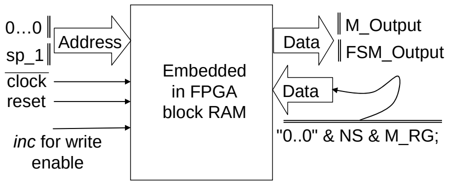
6.4 Using Embedded Memory Blocks

Synthesis of RHFSMs has shown that stack memories (1) are very resource consuming. However, for implementing the functionality (1) embedded memory blocks can be used (such as that are available for FPGAs and shown in figure 7).

Embedded RAM was constructed using VHDL structural specification (for experiments Xilinx library component RAMB4_S8_S8 was used). Signal *sp_1* forms the RAM address and it is defined in VHDL architecture as follows:

```
sp_1 <= (others => '0') when reset = '1'
       else sp - unwinding when dec = '1'
       else sp;
```

As a result, the signal *sp_1* provides for fast stack unwinding if required. Note that in figure 7 just a small part of the available memory block was used. That is why unnecessary address inputs were set to 0 and some outputs were left unconnected.



```
process(clk,rst)
begin
if reset = '1' then  -- initialization
elsif rising_edge(clock) then
     if inc = '1' then
            -- error handling
            sp <= sp + 1; FSM_RG <= NM_FS; M_RG <= NM;
     elsif dec = '1' then
            sp <= sp - unwinding; M_RG <= M_Output;
            FSM_RG <= FSM_Output;
     else   FSM_RG <= NS;
     end if;
end if;
end process;
```

Figure 7: Embedded memory block.

Since the stack pointer is common to both stacks, two segments of the RAM data bus are used for the *FSM_stack* and the *M_stack*, respectively. Two signals *FSM_RG* and *M_RG* enable the process shown in figure 7 to function at a single level. Switching to different levels of stack memories is provided through copying the RAM block outputs to the *FSM_RG*/*M_RG*, which is needed just for hierarchical returns. This permits very fast HFSMs to be constructed. The same approach can be used for distributed memory available for Xilinx FPGAs. The following code demonstrates how to use the distributed library component RAM16X1S:

```
RAM16X1S_instM0 : RAM16X1S
  generic map (
    INIT => X"0000")
  port map (
    M_Output(0),
    sp_1(0), sp_1(1), sp_1(2), sp_1(3), M_RG(0),
    inverted_clk,              -- inverted_clk <= not clock;
     inc);
```

N-bit stacks (N>1) can be built from N components shown above.

Note that similar FPGA RAM blocks and distributed memory blocks can be used for all memories shown in figure 4. Thus, the proposed RHFSM is very well suited for implementation in commercially available FPGAs.

7 Experiments

The primary goal of the experiments was to prove on arbitrary selected examples that the results presented in the paper are correct and can be used for practical applications. The RHFSMs were implemented in FPGA of Spartan-IIE family of Xilinx and were tested using virtual execution units described in [13]. All the experiments were performed using the stand-alone board TE-XC2Se [14]. For dynamic modifications we used dual port RAM library components, such as RAMB4_S8_S8 of Xilinx. Thus, the first port took part in the RHFSM state transitions whilst the second port was used to reprogram the RAM from an external source. Consequently, FSMs with dynamically alterable behaviour were implemented on statically configurable FPGAs. The results of experiments have shown that the proposed methods and tools are correct and can be used for practical applications.

The experiments have also shown that the considered advanced techniques for stack operations in RHFSM permit to reduce the number of clock cycles (on average by 9%) and the required hardware resources (on average by 14%). The latter is basically achieved with the aid of replacement of an arbitrary logic with embedded in FPGA memory blocks.

8 Conclusion

The paper presents novel methods for the design of RHFSMs from hierarchical specifications. All the suggested structures are supported by VHDL templates (skeletal code, which can directly be used in engineering practice). The majority of the proposed methods have been verified in commercial hardware (in Xilinx FPGA). The proposed RHFSMs have two very important features: 1) they support the design hierarchy (including possible recursive calls, if required); and 2) they are statically and dynamically reconfigurable, which is very important for adaptable systems.

9 References

[1] V. Sklyarov, A.A. da Rocha and A.B. Ferrari, “Synthesis of reconfigurable control devices based on object-oriented specifications”, in J.C. López, R. Hermida and W. Geisselhardt (eds), *Advanced Techniques for Embedded Systems Design and Test*, Kluwer Academic Publishers, pp 151-177 (1998).

[2] W. Xu, S.G. Wang, A.L. Wang and G.B. Wang, “Toward an efficient self-organizing reconfiguration method for self-reconfigurable robots”, *Journal of Intelligent and Robotic Systems*, 37, pp 415-425 (2003).

[3] H. Bojinov, A. Casal and T. Hogg, “Emergent structures in modular self-reconfigurable robots”, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, USA, pp 1734-1741 (2000).

[4] Y. Meng, “A dynamic self-reconfigurable mobile robot navigation system”, *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, USA, pp 1541-1546 (2005).

[5] V. Sklyarov, “Hierarchical finite-state machines and their use for digital control”, *IEEE Transactions on VLSI Systems*, 7(2), pp 222-228 (1999).

[6] V. Sklyarov, “Reconfigurable models of finite state machines and their implementation in FPGAs”, *Journal of Systems Architecture*, 47, pp 1043-1064 (2002).

[7] V. Sklyarov, “FPGA-based implementation of recursive algorithms”, *Microprocessors and Microsystems*, *Special Issue on FPGAs: Applications and Designs*, 28(5-6), pp 197-211 (2004).

[8] I. Skliarova and A.B. Ferrari, “The design and implementation of a reconfigurable processor for problems of combinatorial computation”, *Journal of Systems Architecture, Special Issue on Reconfigurable Systems*, 49(4-6), pp 211-226 (2003).

[9] I. Skliarova and A.B. Ferrari, “A software/ reconfigurable hardware SAT solver”, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(4), pp 408-419 (2004).

[10] R. Feldman, C. Haubelt, B. Monien and J. Teich, “Fault tolerance analysis of distributed reconfigurable systems using SAT-based techniques”, *Proceedings of FPL*, Portugal, pp 478-487 (2003).

[11] V. Sklyarov, I. Skliarova, “Design of digital circuits on the basis of hardware templates”, *Proceedings of International Conference on Embedded Systems and Applications (ESA’03)*, USA, pp 56-62 (2003).

[12] V. Sklyarov and I. Skliarova, “Recursive and iterative algorithms for N-ary search problems”. *Proceedings 19th World Computer Congress, Professional Practice in Artificial Intelligence*, Santiago, Chile, pp. 81-90 (2006).

[13] V. Sklyarov, “Hardware/software modeling of FPGA-based systems”, *Parallel Algorithms and Applications*, 17(1), pp 19-39 (2002).

[14] Trenz Electronic, “Spartan-IIE development platform”, http://www.trenz-electronic.de, visited on 10/10/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Balancing Sociality in Meta-agent Approach

Oomiya Kenta[1], Miyanishi Keiji[2] and Suzuki Keiji[1,3]
[1]Institute of System Information Science,
Future-University Hakodate, Hokkaido, Japan
[2]Information Science Research Center, Japan
[3]Core Research for Evolutional Science and Technology (CREST, JST), Japan
g2105004@fun.ac.jp

Abstract

The tragedy of the commons is one of the well-known game problems that involve sharing limited common resources. The aims of this study are (1) to solve the tragedy of the commons using a social agent-based approach, and (2) to examine the effectiveness of the approach, which treats heterogeneous agents who have different purposes and roles. These agents dynamically organise a hierarchy according to the situation. Furthermore, to control the usage of a common resource, a levy-based control strategy is proposed. Even if the agents obey individual rationality, this strategy can avoid dilemma situations. To adjust the charging plans, a Genetic Algorithm (GA) is applied. The properties of the charging plan depend on the evaluation of the GA. For the evaluation of the GA, the ratios of altruism and selfishness and the importance of roles are considered. Through an experiment, features of these ratios are examined. Furthermore, we introduce a mechanism that adjusts the parameters in the evaluation of the GA to the game simulation, and examines the effectiveness of the introduced mechanism. The proposed approach avoids the tragedy situation and indicates the optimum values of the parameters in the evaluation of the GA.

Keywords: meta-agent, tragedy of the commons, levy-based control

1 Introduction

Agent-based social behaviour simulation is a research field that treats complex game situations and examines artificial intelligence [1, 3]. Social dilemmas are complex game situations and suites that examine the intelligence of agents [6]. In this paper, the tragedy of the commons [2], which is a social dilemma game problem, is considered in an agent-based simulation. In this game, players use common limited resources to gain rewards. If players behave based on individual rationality, all players will face tragedies by losing higher payoffs. To avoid such tragedies, players have to make higher payoffs and foster their relationship with other agents to prevent selfish behaviour or a change in the problem structure, such as a change in the payoff functions. Our previous studies employed a levy-based control strategy, which involves a change in the payoff function, and thus avoids the tragedy situation. In this strategy, the meta-agent has a charging plan to acquire an incoming levy from the agents. This approach can change the activities the agents select without changing the individual rationality. Furthermore, we introduced dynamic selection roles in the strategy. Thus, the agents dynamically organise a hierarchy.

[1] is the previous study that proposed changing the rules of agents and the equation that evaluates the charging plans. In this paper, we extend the equation that evaluates the charging plans. The importance of the roles of agents and the ratio of sociality, which indicates if the agents place emphasis on individual gain or total gain are considered. As a result, effective ratios are provided by the new evaluation function.

The properties of the charging plans depend on the evaluation of the GA. That is, the parameters in the evaluation of the GA should be appropriately adjusted by a mechanism. Accordingly, we introduce the Tabu Search [7] to the game simulation to adjust the parameters. Furthermore, we examine the effectiveness of introducing the Tabu Search.

2 The Tragedy of the Commons

The tragedy of the commons [2] is a well-known game problem, which is one of the *n*-person social dilemmas [5]. This game enables us to analyse the behaviour of players who share common limited resources. As common resources are limited, the higher activities of agents to gain higher payoffs will result in lower payoffs. One of the general forms of the payoff function in the tragedy of the commons is as follows:

$$Payoff(a^i, TA) = a^i(|A| \times N - TA) - 2a^i \tag{1}$$

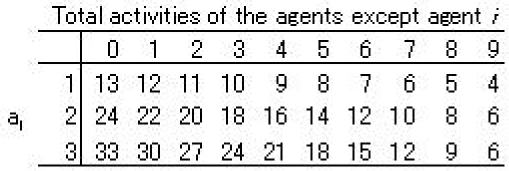
where $Payoff(a^i, TA)$ is the payoff of agent i; a^i denotes the degree of activity of agent i; TA is the total activity; N is the number of agents; A is the set of activities; $|A|$ is the number of activities.

An example of the payoff function is as follows:

$$Payoff(a^i, TA) = a^i(16 - a^1 + a^2 + a^3 + a^4) - 2a^i \tag{2}$$

Here, four agents participate and add four degrees of activity, a^i□{0, 1, 2, 3}. The payoff values based on the function are as shown in table 1. If the agents act on individual rationality and select their own activity, TA becomes 12. Therefore, the agents can only obtain the smallest reward. This situation is called the tragedy situation.

Table1: Example payoff table of the tragedy of the commons.



		Total activities of the agents except agent *i*									
		0	1	2	3	4	5	6	7	8	9
	1	13	12	11	10	9	8	7	6	5	4
a_i	2	24	22	20	18	16	14	12	10	8	6
	3	33	30	27	24	21	18	15	12	9	6

A characteristic of this game is that no known technical solution exists. Therefore, to solve this game, players should change individual rationality to other types of rationality or change the problem structure by changing the payoff function. One of the objectives of agent-based simulations is to examine the different kinds of rationalities and extended problem structures that can avoid social dilemmas such as the tragedy situation. In this paper, the proposed simulation method involves extension of the problem structure. A meta-agent is introduced in order to prevent the agents from causing the tragedy situation. The detail of the proposed approach is described in the next section.

3 Approach

To solve the social dilemma, we propose an agent-based simulation process. Figure 1 shows a schematic view of the process.

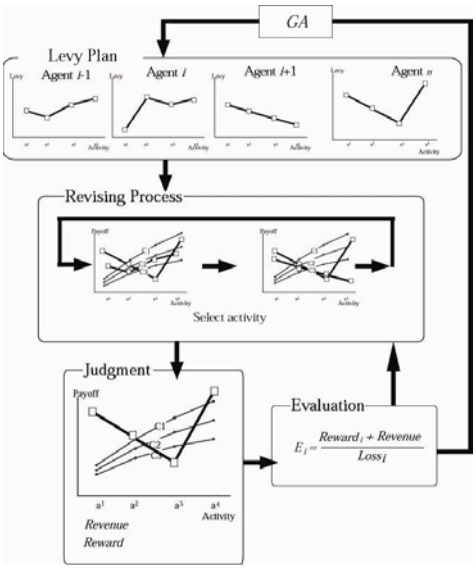


Figure 1: Proposed agent-based simulation process.

In this process, the social agent-based approach expands the levy-based control strategy. Each agent has a charging plan, which consists of levies for each activity. In the revised process, the agents in turn decide on their own role and activity to satisfy individual rationality. This process ends when all agents decide their activities. In the judgement process, the agents calculate their own gain. In the evaluation process, charging plans are evaluated, to calculate gains of agents. In the final step, new charging plans are evolutionally generated by the Genetic Algorithm.

In the following subsections, we explain the details of the simulation processes. First, we explain the architecture of the agents. Then, the levy-based control strategy, the charging process and decision making are explained. Finally, we explain the evolution of charging plans and the extended equation.

3.1 Architecture of the Agent

The group of agents is represented by N. $N = \{1, 2, \ldots, n\}$. $A = \{a_1, a_2, \ldots, a_k, a_{\mathrm{meta}}\}$ denotes the activities of the agents. It is assumed that a_{i+1} is a higher activity than a_i in the tragedy of the commons game. The element of the activities, a_{meta}, means the agent plays the role of a meta-agent. In other words, the agent can select the role of a meta-agent as one of the activities. An agent who does not select a_{meta} is called a player. The aim of these agents is to maximise their own gain. Each agent has a charging plan and decides on his own activity by referring to the payoff function [Remark 3] and the charging plans.

3.2 The Levy-based Control Strategy

In our previous studies [1, 4], the levy-based control strategy was employed in order to avoid the tragedy situation. As a result, the meta-agent and the other players were able to avoid the tragedy situation. However, the fixed roles lacked robustness. In this paper, we introduce the social agent-based approach to the strategy. In addition, we assume that the agents can select their own role according to situations and dynamically organise the hierarchy of the agents.

Figure 2 shows a schematic view of the levy-based control strategy. This strategy changes the problem structure, for example, the payoff function. In this strategy, the agent selected randomly decides his own activity, to be a meta-agent or a player, by referring to other agents. The meta-agents can change the payoff function by charging levies for activities of the players. When an agent selects the role of a meta-agent, charging plans are broadcasted to the others. If more than two charging plans exist, a new charging plan, the social charging plan, is generated, which is a set of minimum levies against each activity. The agents calculate the expected reward and the revenue. The agents select the activity that maximises the expected reward and revenue. The details of the charging process are mentioned in subsection 3.3.

In this strategy, the property of the charging plan is important. Apparently, it is possible to fix a suitable charging plan by analysing the game properties. Then the plan can be embedded in the simulation beforehand. However, one of the purposes of agent-based simulations is to examine the autonomous property of the agents in a social environment. Accordingly, the autonomous acquisition of a suitable charging plan is desirable. To search for a suitable charging plan, the genetic algorithm is applied. Details of the evolution of the charging plan are mentioned in subsection 3.5.

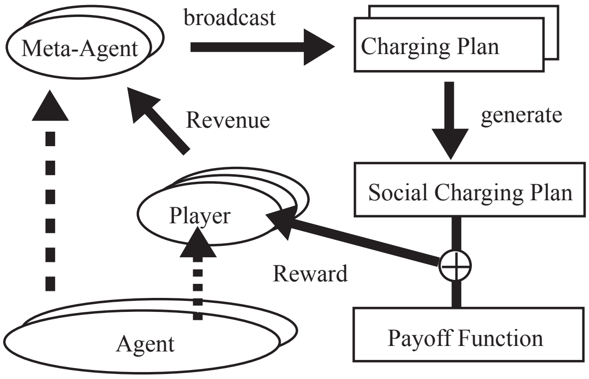


Figure 2: A schematic view of the levy-based control strategy.

3.3 Charging Process

The charging plan consists of values of levies, $\{Lv_1, Lv_2, \ldots, Lv_k\}$, corresponding to the activities, $\{a_1, a_2, \ldots, a_k\}$. That is, levy Lv_i charges to the payoff of the activity a_i. Figure 3 shows the relationship between the payoff function and the charging plan. The vertical axis indicates gain and the horizontal axis indicates the activities. The thin lines are the payoff functions. In this example, line C2 is used. A bold line indicates a social charging plan, which is a set of minimum levies of the charging plan.

The reward is the difference between a payoff value and a levy. The reward of an agent-*i* who selects an activity a_j is represented by the following equation:

$$Reward(a_j^i) = Payoff(a_j^i, TA) - Lv_j^{min} \tag{3}$$

where Lv_j^{min} is the minimum levy among the broadcasted levies. If a levy value is greater than the corresponding payoff value, the levy value is treated as 0. Thus, the selfish behaviour of the meta-agents is brought under control.

The revenue of the meta-agent is a set of charges which are not greater the corresponding payoff values. Revenue is shown by the following equation:

$$Revenue_i = \sum_{j=1}^{k} L'v_j^i \left| A^j \right| \qquad where \quad L'v_j^i = \begin{cases} Lv_j^i & if\ Lv_j^i = Lv_j^{min} \\ 0 & otherwise \end{cases} \tag{4}$$

In the above equation, $|A^j|$ means the number of agents selecting the activity a_j. Only the meta-agent who broadcasts the minimum levy can obtain revenue from the corresponding activity. Accordingly, the social charging plan can control the meta-agents' selfish behaviour.

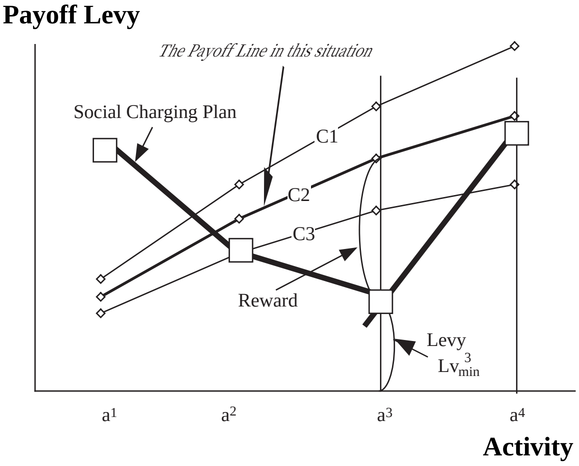


Figure 3: Relation between the payoff function and a charging plan concerning reward.

3.4 Decision Making and Revising Process

In this strategy, the agents decide their activities based on individual rationality, which is a policy to maximise their gain. That is, the agents select their own activity in order to maximise their gain according to the situation. The randomly selected agent in turn decides on his own activity. However, the first agent cannot decide on an activity according to the situation, because the initial activities of the agents are configured as a_1. To correct this inequality, this process is repeated several times. These processes are called revising processes. After these processes, the judgement process is initiated and the gains of the agents are calculated in order to evaluate the charging plans.

3.5 Evolution of Charging Plans

To obtain a suitable charging plan, the genetic algorithm is applied to each agent. Each agent has a set of chromosomes. The charging plan decoded from one chromosome is applied to the game. After the game, agents to whom the charging plan was applied obtain the reward and the revenue. The evaluation of the charging plan is based on the reward and the revenue. For an agent *i*, the evaluation of the charging plan is defined as follows:

$$E_i = Reward(a^i) + Revenue_i \tag{5}$$

This game has a dynamic feature. That is, an evaluation value of a charging plan is not constant. In this simulation, the game is iterated several times against the same charging plan, because the averaged evaluation value should be considered. Furthermore, we evaluate the charging plan by using other charging plans of other agents, because the evaluation depends on other charging plans. Consequently, an evaluation value of a charging plan is average of these evaluations. After the evaluation values are fixed for all chromosomes, the genetic operations are applied. That is, the crossover, the mutation and reproduction are applied.

3.6 Extended Equation Evaluating a Charging Plan

In this paper, we have extended (5). The ratios of the roles of the agents and the sociality of the agents are considered. We show the extended equation for an agent-pi as follows:

$$E_{pi}^{aj} = \left(\frac{\alpha \cdot Revenue_{pi} + \beta \cdot \mathrm{Re}ward_{pi}}{Reward_{pj_{worst}}}\right)^{a} + \left(\frac{\sum_{j=1}^{N} Reward_{pj}}{\sum_{j=1}^{N} Reward_{pj_{worst}}}\right)^{b} \quad (6)$$

where *α*, *β*, *a*, and *b* are the weighting coefficients. *α* concerns the role of the meta-agent; *β* is the weighting coefficient of the role of the players; *a* is the corresponding charges of the meta-agents; *b* represents the weighting coefficient of the rewards of the players. The first term in the equation represents the gain of all agents and the second term denotes the gain of the players. $Reward_{\mathrm{pj\ worst}}$ is the gain in a tragedy situation.

3.7 Adjustment of Parameters in the Extended Equation

The parameters in (6) influence the simulation results. This is detailed in the next section. In this subsection, the procedure to adjust the parameters is described.

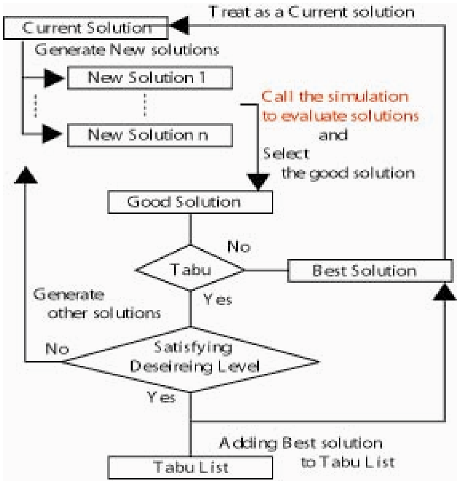

Current Solution
Treat as a Current solution
Generate New solutions
New Solution 1
New Solution n
Call the simulation to evaluate solutions and Select the good solution
Good Solution
No
Tabu
Best Solution
Generate other solutions
Yes
No
Satisfying Deseireing Level
Yes
Adding Best solution to Tabu List
Tabu List

Figure 4: Schematic view of simulation using TS.

To adjust the parameters, the tabu search (TS) [7] method is used. TS is a meta heuristic search method, which uses a record of solutions. TS is based on the assumption that the current solution can transit to proximity solutions by a neighbourhood search. Figure 4 shows a schematic view of the proposed simulation using TS. The TS process is used as a framework. When new solutions are evaluated, the proposed simulation of the tragedy of the commons is used.

4 Simulations and Results

In this paper, two types of simulation are executed. The first is the proposed simulation of the tragedy of the commons. The aim of this simulation is to examine the effectiveness of the proposed approach. The second simulation is to examine the effectiveness of autonomous adjustment of the parameters in the evaluation of the GA. To adjust the parameters, the game simulation is built into the framework of a Tabu Search.

4.1 A Game Simulation

In this section, in order to examine the effectiveness of the proposed methods in avoiding the tragedy situation, a game simulation is executed. The payoff function is set as follows:

$$Payoff_i = a^i\left(\left|A\right| \times N' - \sum\nolimits_{j}^{N'} a^j\right) - 2a^i \quad (7)$$

where *N*′ is the number of agents, excluding the meta-agents.

The parameters in the simulation are as follows: the number of agents, *N*, is 12. The activities for each agent are 7, a_j□$\{a_1, a_2, \ldots, a_5, a_{\mathrm{meta}}\}$. The initial activity is a_1. Each agent has 30 chromosomes. The length of the revising step is 12. To evaluate a chromosome, the game is iterated four times using the same chromosome set. Furthermore, this iterated part is iterated three times using other chromosome sets of other agents. So the game is iterated 12 times to evaluate a charging plan. Crossover and mutation are applied to the chromosomes. The crossover rate is 1.0 and the mutation rate is 0.05. Under these parameters, the evolution of the charging plans of the agents proceeds until 50 generations.

In this paper, to confirm the effectiveness of the extended evaluation, the simulation is iterated 81(=9 times 9) times using sets of *α*, *β*, *a* and *b*. Parameter sets of *α* and *β* are { (*α* = 3, *β* = 0), (*α* = 2, *β* = 0), (*α* = 3, *β* = 1), (*α* = 2, *β* = 1), (*α* = 2, *β* = 2), (*α* = 1, *β* = 2), (*α* = 1, *β* = 3), (*α* = 0, *β* = 2), (*α* = 0, *β* = 3) }. Parameter sets of *a* and *b* are { (*a* = 3, *b* = 0), (*a* = 2, *b* = 0), (*a* = 3, *b* = 1), (*a* = 2, *b* = 1), (*a* = 2, *b* = 2), (*a* = 1, *b* = 2), (*a* = 1, *b* = 3), (*a* = 0, *b* = 2), (*a* = 0, *b* = 3) }. These two sets of nine patterns are used.

4.1.1 Results of the Simulation

Gains of the meta-agents are shown in figure 5. Gains of the players, who are agents but not meta-agents, are shown in figure 6. Figure 8 shows the average of the agents' activity. These figures are the results of 46-50 generations. Therefore, these figures are conclusive results of the proposed simulation. The vertical axis represents the ratio of sociality. More selfishness is represented by low values on the vertical axis. More altruism is represented by higher values on the vertical axis. The horizontal axis represents the importance of the roles of the agents. The left side of the horizontal axis represents the case where the role of the meta-agent is more important than the role of the player. The right side of the horizontal axis represents the case where the role of the player is more important than the role of the meta-agent. In figures 5 and 6, deep colours represent high gains of agents. However, in figure 7, deep colours represent low value of agents' activity. As in figure 6, in the tragedy situation, which is the game without meta-agents, the worst gain is 35-45. Therefore, meta-agents exist in areas excluding the 35-45 area.

4.1.2 Features in Evaluation of the Role of the Agents

In the horizontal axis of figures 5 to 7, the colour on the right side is deeper than on the left side. So these figures show that the role of the player is more important than the role of the meta-agent. This case not only improves the gains of the players but also the gains of the meta-agents, because the charging plans are not under the control activities of the agents. Figure 8a shows that a few such charging plans exist.

The area of (α = 0, β = 2) is an exception. The activities of the players come under control in this area. However, the gains of players are not high. On the other hand, the gains of meta-agents are high. Because the parameter α is 0 in this area, the role of the meta-agent is not evaluated. Therefore, the charging plans have low values. Figure 9b shows these charging plans.

4.1.3 Features in the Ratio of Sociality and Individuality

The vertical axis shows that the gains of the players are high in the area of (a = 0, b = 2). This parameter set shows that evaluation of the charging plan does not depend on α and β. Therefore, in this case, the charging plan is evaluated by the gains of all players. Therefore, to eliminate the situation where the charging plan cannot improve the gains of the game, the gains of the players are improved. By examining the charging plans that appear in the simulation, we can confirm the tendency that charging plans come under control activities. Figure 8c shows these charging plans.

4.2 Adjustment of Parameters in the Extended Equation

A result of the proposed simulation that is introduced by the Tabu Search is as follows. A schematic view of this simulation is shown in figure 6. In this simulation, in order to evaluate new solutions, the game simulation is executed under the same conditions as in section 4.1.

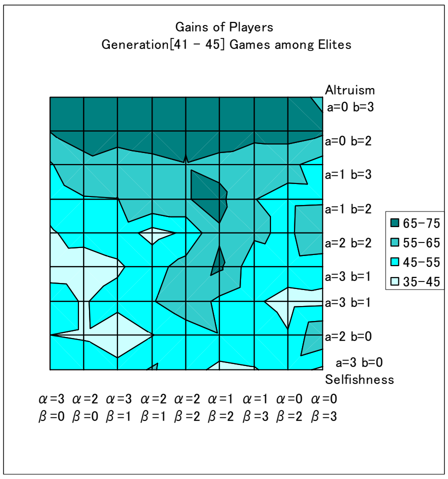


Figure 5: Map of meta-agents' revenue.

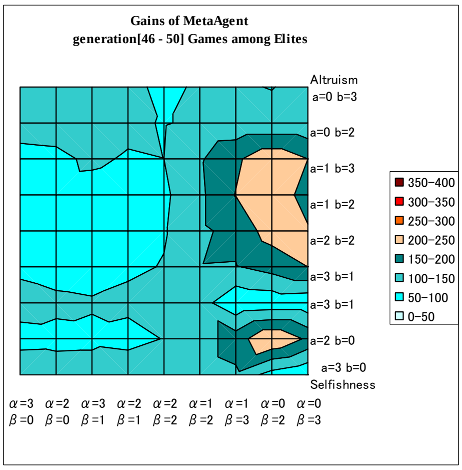


Figure 6: Map of players' reward.

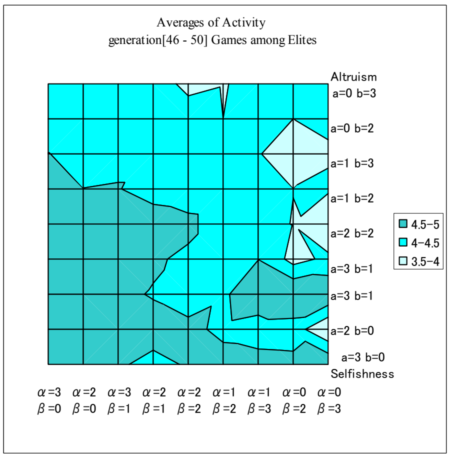
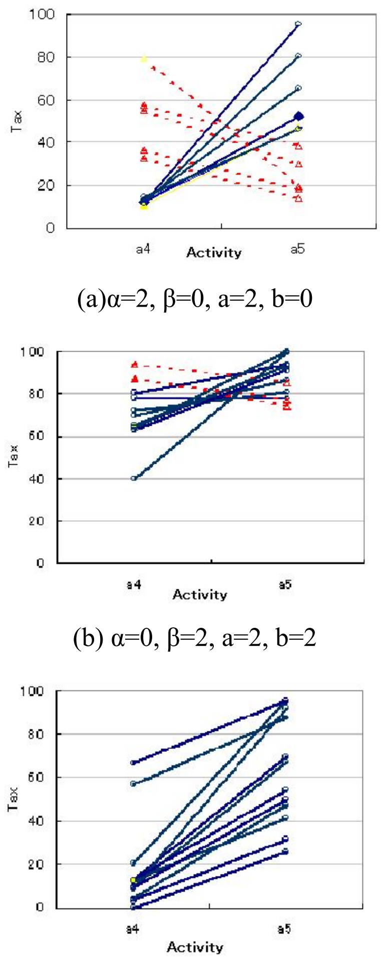


Figure 7: Map of average players' activities.



(a)α=2, β=0, a=2, b=0

(b) α=0, β=2, a=2, b=2

(c) α=1, β=2, a=0, b=2

Figure 8: Example of acquired charging plans.

A result of the simulation conducted to adjust the parameters is shown in figure 9. This figure shows the transition of the evaluated value and its parameters. the evaluated value increases stepwise with transition of the parameters. A feature of the stepwise increase shows a feature of the Tabu Search method. The adjusted parameters transit to the parameters indicated in section 4.1, which emphasises the importance of the role of the players and altruism. Therefore, the parameters in the extended equation evaluating the charging plans are adjusted by the proposed simulation.

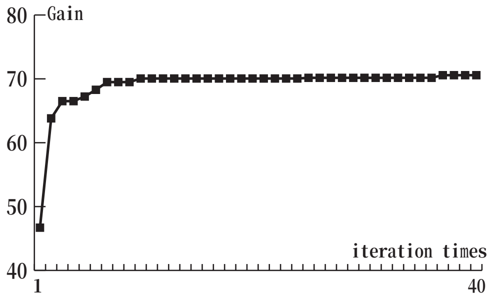


Figure 9: Transition of the parameters and profits.

5 Conclusion

This work addresses the game of tragedy of the common by a social agent-based approach with a levy-based control strategy. This paper evaluates the importance of the role of the agents and the ratio of sociality in the coevolution. Throughout the experiment, the proposed approach performed well and avoided the tragedy situation. Furthermore, we indicated the optimum values of the parameters in the evaluation of the GA. The features of these parameters emphasise the role of players and altruism. Finally, we introduce a mechanism that adjusts the parameters in the evaluation of the GA in order to avoid the tragedy situation.

The characteristics of the acquired charging plans must be analysed. In particular, the relationship with the Nash equilibrium should be clarified. These points will be considered in a future work.

6 Acknowledgement

This work is supported by Core Research for Evolutional Science and Technology, Japan Science and Technology Agency (CREST, JST).

7 References

[1] K. Suzuki, “Dynamics of autonomous changing roles in social dilemma games”, *Proceedings IJCIA-03 Workshop on Multiagent for Mass User Support*, pp 19-25 (2003).

[2] G. Hardin, “The tragedy of the commons”, *Science*, 162, pp 1234-1248 (1968).

[3] R. Suleiman, K. G. Troitzsch and N. Gilbert (eds), *Tools and Techniques for Social Science Simulation*, Springer (2000)

[4] T. Yamashita, K. Suzuki and A. Ohuchi, “Distributed social dilemma with competitive meta-players”, *Int. Trans. in Operational Research*, 8(1), pp 75-88 (2001).

[5] X. Yao, “Evolutionary stability in the N-person prisoner’s dilemma”, *Biosystems*, 37, pp 189-197 (1996).

[6] R. Conte, R. Hegselmann and P. Terna (eds), *Simulating Social Phenomena*, Springer (1997).

[7] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic (1998).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Fast Colour Classification for Real-time Colour Object Identification: AdaBoost Training of Classifiers

Napoleon H. Reyes, Andre L. Barczak and Chris H. Messom
Computer Science, Institute of Information and Mathematical Sciences,
Massey University, Auckland, New Zealand
n.h.reyes@massey.ac.nz, a.l.barczak@massey.ac.nz, c.h.messom@massey.ac.nz

Abstract

This paper presents an AdaBoost learning based Fuzzy Colour Contrast Fusion colour classification algorithm. The system is able to learn colour discrimination knowledge in a matter of minutes during training that accounts for hue and saturation drift of target colours. The performance of the system is compared to a pure fuzzy system using colour contrast rules, a pure AdaBoost system, as well as to a hue-saturation pie-slice decision region approach. The Fuzzy-AdaBoost system is shown to be superior to the alternate approaches studied, while maintaining real-time performance during colour identification.

Keywords: fuzzy logic, fuzzy colour contrast rules, AdaBoost, colour locus constancy, colour constancy, robot soccer, colour object recognition.

1 Introduction

Colour calibration is still a challenging problem in machine vision particularly for poorly lit or varying environments [1]. A robust algorithm should handle small errors in lens focus, zoom and aperture setting, shadow, change in the spectral colour of the source illuminant, varying illumination intensities across the exploration field, object rotation, quantum electrical effects in the imaging device, etc. Research teams involved in robotic competitions employ many different object tracking techniques [2] to compensate for the shortcomings of their vision system.

The inaccuracy in colour object tracking can be traced to the first step of colour object recognition; that is, in the colour classification step. Normally colour classification algorithms [1, 3-11] require a distinct set of colour descriptors to allow for colour discrimination of one object from another. A set of colour descriptors must be specified that are constant under changing illumination intensities, while the object traverses the playing field. The vision system should be adaptable for different environments and illumination settings, have the ability to discriminate similar colours and automate the colour calibration task while ensuring accuracy in robot position and orientation calculation within given tolerances, for Robot Soccer approximately ±5mm and ±5 degrees respectively are reasonable targets. In addition the vision algorithm should guarantee that the calibration task can be accomplished within some real time constraint as most of object tracking algorithms are used to identify robots and control their motion in tight control loops. Errors and inaccuracies in the object tracking system as well as delays in object identification results in poor motion control.

1.1 Experiment Test Bed

All experiments were performed using the robot soccer system (FIRA); however, under extreme illumination conditions, worse than expected in normal competition environments. As can be viewed in figure 1, the scene has a noticeable uneven light distribution (as captured by an overhead camera), causing two elliptically-shaped highlights in the centre of the field. The subjects of classification are colour patches distributed across the platform under varying illumination intensities (6 colours: yellow, green, light blue, pink, violet and purple, with 6 representative sample patches each, plus 4 additional small patches). There are several similar colours present in the scene. Under strong white light illumination, yellow and green, as captured by the camera tend to appear very similar to each other that even the human visual system finds it difficult to differentiate them. All 40 colour objects were used as targets for classification, and the empirical results demonstrate that the 6 different colour objects' classes could be identified accurately by the Fuzzy-AdaBoost systems, while there are cases where the pie-slice algorithm fails.

2 Colour Classifiers

The system is comprised of the fuzzy colour contrast fusion system and the AdaBoost classifier. This section discusses first the Fuzzy colour contrast fusion architecture and the AdaBoost system in isolation, then, gives the details of the integrated architecture. It is seen that AdaBoost training can be applied at various stages of the processing and experiments show that a multiple AdaBoost system combined with colour contrast fusion is comparable to a hand-calibrated fuzzy approach and superior to alternate approaches.

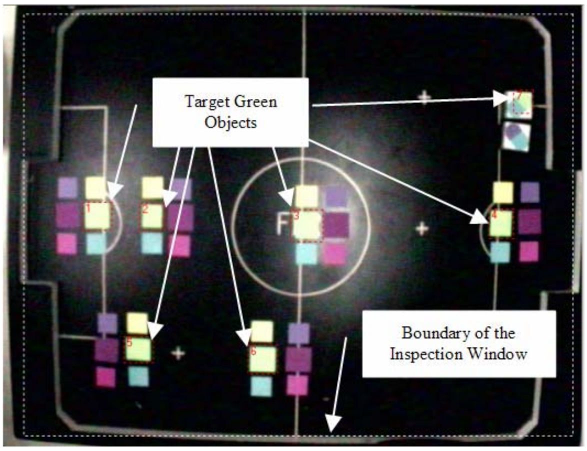


Figure 1: Tracked colour objects (yellow, green, light blue, violet, purple and pink) with 6 representative patches for each (plus four extra small patches on the right hand corner); subjected under relatively bright, dim and dark regions in the exploratory space.

2.1 Learning Colour Classification

Ideally, intelligence is best imparted into a system by allowing it to learn both from rules and from examples. In previous research [3,4,5,6,7], a fusion of fuzzy colour contrast rules has been applied to compensate for many colour distortions found in the scene. The colour corrections are made prior to classifying the colour sensed. This can be viewed in figure 2 which shows the architecture of the fuzzy colour contrast fusion algorithm.

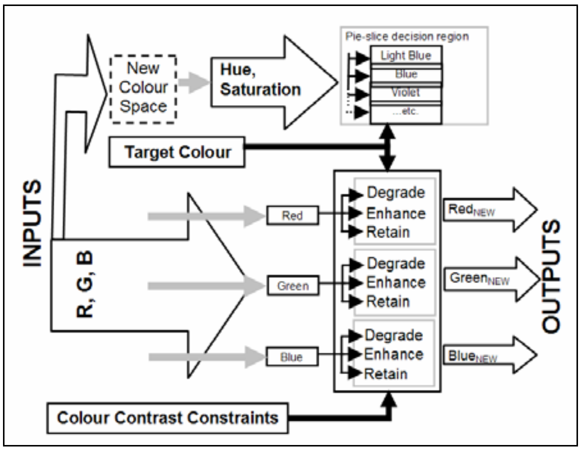


Figure 2: Colour contrast fusion architecture. This technique compensates for any pixel values affected by illumination variations in the environment through the application of a colour contrast enhancement or degradation operation on each colour channel.

Contrast fusion is applied based on the applicability of the rules assigned to the colour being examined at hand. The system is calibrated by measuring the true positive and false positive proportions for each rule combination tested and choosing appropriate fuzzy colour contrast rules. In general, the fuzzy colour contrast rules extracted for one environment can be used for any other indoor environment with relatively different illumination intensity distribution, provided that same source illuminants are used (e.g. fluorescent lamps.

The fuzzy colour contrast fusion technique embraces a new concept that we call colour locus constancy [4], which corresponds to the algorithm's ability to constrain a locus of colours (in a colour space) representing a dynamic object, despite the object's traversal of a space of varying illumination intensities. Precise colour classification has its price, in terms of extracting the colour descriptors by human visual inspection. The hand calibrated colour descriptors for the given target colours are shown in table 1. It can be observed that the descriptors may vary depending on whether or not fuzzy rules are utilised or not.

2.2 Learning from Rules and Training Patterns

Although colour correction and colour classification have been successful [4], the challenge of automatically extracting the colour descriptors remains. In the following subsections, we provide a discussion of the AdaBoost system which learns from examples, then, we present the architecture that integrates fuzzy and AdaBoost systems together to allow for automating the colour calibration task.

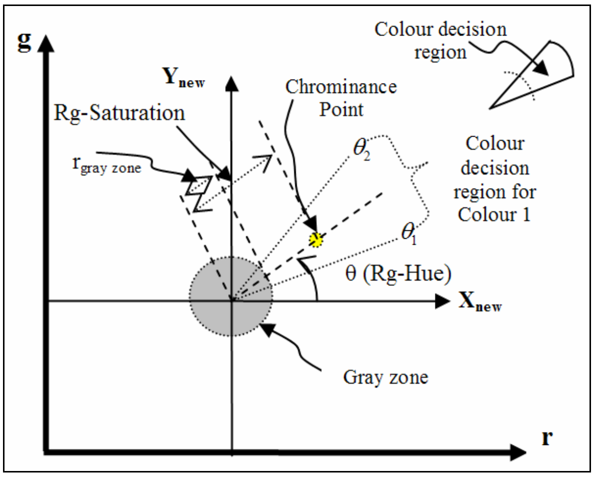


Figure 3: The pie-slice decision region makes use of bounding angles and radius for colour classification. Colours are classified by inspecting their hue (angle) and saturation (radius) values, whether or not they fall within a region associated with a colour name such as pink, violet, orange, etc.

2.3 AdaBoost Training

AdaBoost is an algorithm that was originally proposed by Freund and Schapire [13]. The algorithm is particularly successful as compared to earlier boosting methods. The basic idea behind boosting is to combine several weak classifiers, which would individually classify a set of marked examples very roughly, into a strong classifier that accurately predicts a rule for that set.

Table 1: Colour descriptors and fuzzy colour contrast rules for all 6 target colours. The '+' label alongside each algorithm name indicates utilisation of fuzzy colour contrast fusion, while the lack of a '+' label indicates non-utilisation of fuzzy rules. The 'X' label denotes non-applicability.

	Saturation		Hue		Contrast Constraints		Colour Contrast Fusion Rules			Level		
Algorithm	Min. Angle	Max. Angle	Min Radius	Max Radius	Min. Angle	Max. Angle	R	G	B	R	G	B
Yellow	163.8	171	7.344	40.194	X	X	No_Op	No_Op	No_Op	X	X	X
Yellow+	161.64	170.28	13.968	66.402	162.612	172.332	No_Op	Degrade	Degrade	0	1	1
Green	176.68	211.68	11.304	62.91	X	X	No_Op	No_Op	No_Op	X	X	X
Green+	171.072	226.08	15.48	62.91	171.612	228.132	Degrade	Enhance	Degrade	2	2	2
Light_Blue	269.604	310.392	22.356	44.514	X	X	No_Op	No_Op	No_Op	X	X	X
Light_Blue+	240.444	310.284	10.17	56.268	240.192	310.896	Degrade	Enhance	Degrade	1	1	1
Pink	70.164	124.128	22.464	69.678	X	X	No_Op	No_Op	No_Op	X	X	X
Pink+	36	98.424	54.198	180	39.204	93.924	Enhance	Degrade	Enhance	1	1	1
Purple	39.924	70.236	22.464	69.678	X	X	No_Op	No_Op	No_Op	X	X	X
Purple+	38.16	55.224	56.808	73.494	45.144	69.768	Degrade	Enhance	Degrade	1	2	1
Violet	0.54	44.964	0	40.644	X	X	No_Op	No_Op	No_Op	X	X	X
Violet+	0.54	44.964	0	70.218	0	52.884	Enhance	Degrade	Enhance	1	3	1

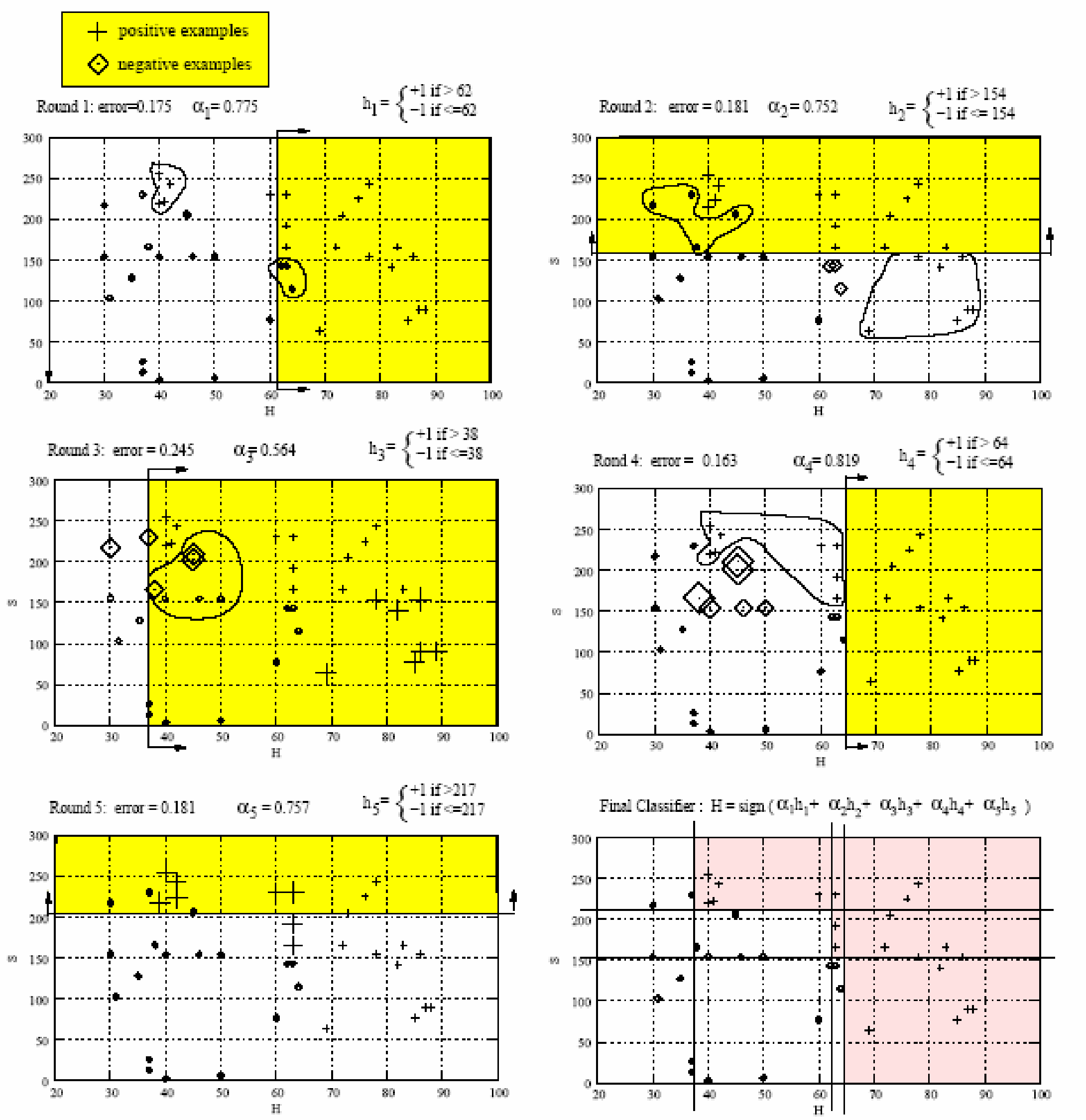


Figure 4: AdaBoost applied to a simple colour classification problem.

The so-called weak classifiers could be any classification method that, for a given distribution, is just slightly better than chance. One could use for example simple Bayesian rules, simple histogram based threshold, decision trees etc. It has been argued that due to the difficulty in modelling distributions, a rule based approach can be more accurate and effective than a model based approach [14]. AdaBoost uses the original distribution to compute rules that will try to fit the data to a given hit ratio and miss ratio. AdaBoost has two strong advantages over similar training methods. The first is the fact that Freund and Shapire proved that the training error tends to zero and the second is the fact that the algorithm is very simple to implement.

For each round of AdaBoost, instead of computing one hypothesis we get one hypothesis for each dimension and choose the one with the smallest error. Sets with a very large number of dimensions can be successfully trained using AdaBoost because it performs a greedy selection of features [15]. The process goes on until the required training error is achieved. When using simple thresholds as weak classifiers, one could look at AdaBoost as a method that splits the space into several positive and negative regions. These regions would be rectangular in 2D and rectanguloid in 3D. Figure 4 shows an example of colour classification using AdaBoost. A number of pixels are represented by their values in HSV colour space, of which we assume that there is no correlation with V (so only two dimensions will be used for training). Each round of AdaBoost increases the weights of incorrectly classified examples. Thresholds are found for each new distribution so the error is minimised for both H and S dimensions. The best threshold is chosen, and the re-weighting process is carried out once again. After 5 rounds the final classifier divides the HxS space in 12 rectangles. Each rectangle only contains either positive or negative examples, and therefore the training error is zero for this simple case. For comparison purposes, we have implemented a pure AdaBoost colour classifier that is trained with both positive and negative examples extracted from the unaltered image in figure 1. Next, we combined the fuzzy and AdaBoost algorithms in two different ways as described below.

2.4 Fuzzy-AdaBoost Colour Classifier System

In this integrated architecture, the AdaBoost system is trained and tested for colour classification using the fuzzified colours returned by the fuzzy colour contrast fusion technique. This system would require that all colour descriptors (minimum and maximum angles, radius and contrast constraints) and fuzzy colour contrast rules are known, as determined through human visual inspection. The main purpose is to investigate whether or not this Fuzzy-AdaBoost architecture would perform better than the pure fuzzy system, considering the possibility that the Fuzzy-AdaBoost system could learn some patterns not accounted for by the pure fuzzy approach [16].

2.5 Fuzzy-Multiple-AdaBoost Colour Classifier System

This system architecture aims to eliminate the task of determining 6 colour descriptors (minimum and maximum angles, radius and contrast constraint angles) required by the fuzzy colour contrast fusion technique. The only prerequisite for this architecture is that the colour contrast rules (right hand side of table 1) should be known. The advantage of such an approach is that we can allow the system to determine which colour pixels need to be fuzzified in the first place so that the colour contrast rules can be immediately employed.

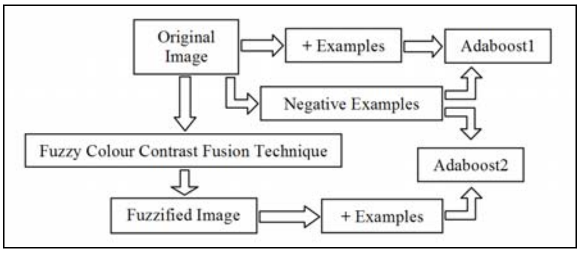


Figure 5: Training system for the fuzzy-multiple-AdaBoost system.

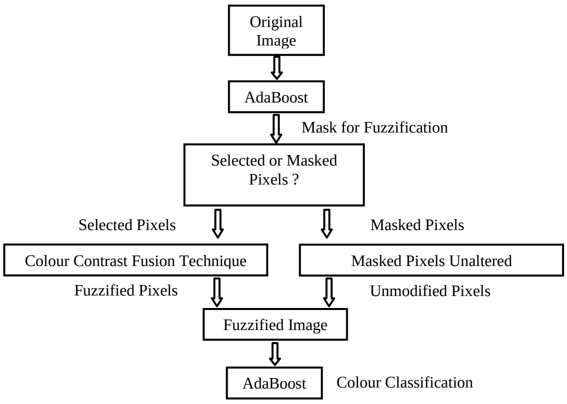


Figure 6: Colour classification system for the fuzzy-multiple-AdaBoost system.

As can be seen in figure 5, the system comprises two AdaBoost systems and one fuzzy colour contrast fusion component. AdaBoost1 is trained using the original, unaltered colour pixels. The system is fed with both positive and negative examples, and does not require any colour descriptors to be calibrated. The main purpose of this system is to identify which among the colours in the scene are the candidate pixels for fuzzification, providing a mask for fuzzification (figure 6). On the other hand, we have AdaBoost2 that is trained on the fuzzified colours of the target objects (positive examples), and original colours of the background and other colour objects as

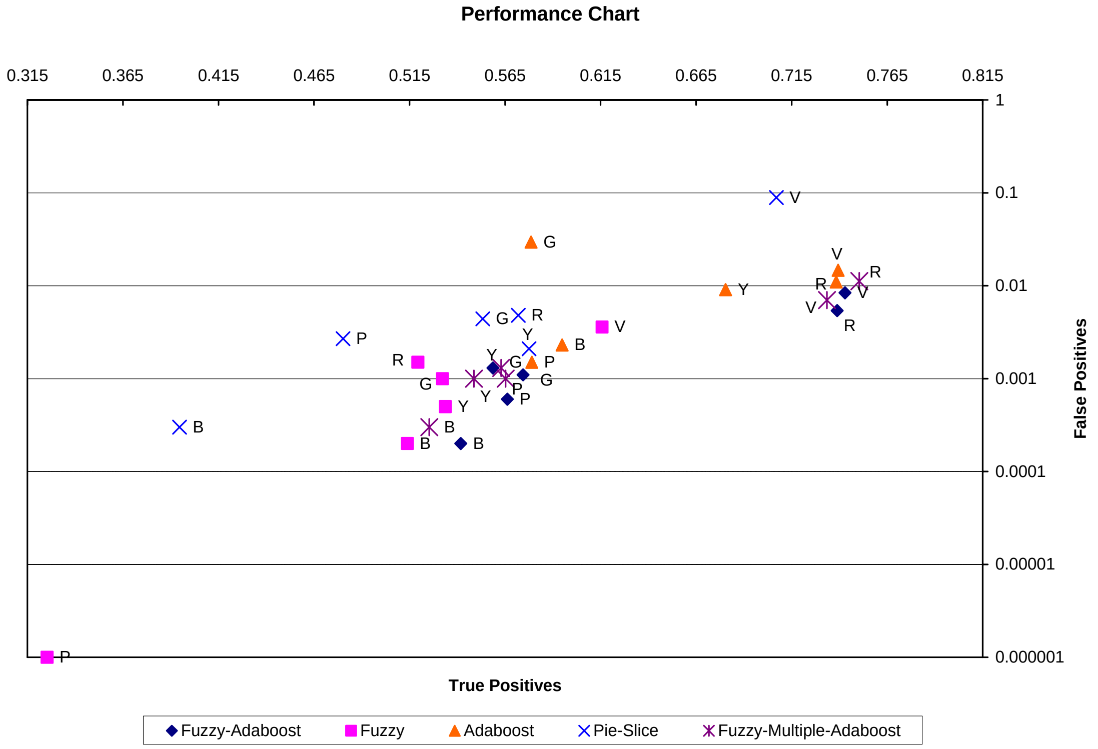


Figure 7: Performance chart of the colour classification algorithms (Fuzzy, Fuzzy-AdaBoost, Fuzzy-Multiple-AdaBoost and Pie-Slice) when tested to recognise 40 colour patches grouped by 6 colours under varying illumination intensities. (P-pink, Y-Yellow, G-Green, V-Violet, B-Light Blue and R-purple).

negative examples. This AdaBoost system would later be used for the final colour classification task.

3 Experiments

The experiments discussed in this section investigate the effects of the utilisation of the rules during training and colour classification. A performance comparison chart of all the algorithms presented in the previous section is given in figure 7 to identify their relative accuracy.

3.1 Performance Analysis of the Colour Classification Algorithms

All classification algorithms presented in the previous sections were tested to identify the colour targets (6 colours, with 6 representative patches each) under relatively dim, dark and bright regions. As shown in figure 7, the hand-calibrated fuzzy approach returned the lowest false positive values. This approach however entails manual calibration of the colour descriptors, which we aim to automate in this work. On the other hand, the pie-slice decision region performed relatively well in recognising 5 colour objects, but failed in identifying the violet colour objects, making it the worst algorithm tested. With regards to the AdaBoost-based systems, all the fuzzy-AdaBoost architectures performed better than the pure AdaBoost system. The Fuzzy-Multiple-AdaBoost system performed well enough for the task of colour identification, considering that it required the least amount of human assistance for colour calibration. Lastly, the results further suggest that the utilisation of colour contrast rules is vital to improving all of the system's performance.

4 Conclusions

This work has presented a Fuzzy-AdaBoost algorithm and its variants that learn both from fuzzy colour contrast rules and examples to address the problems of colour classification. The system allows for colour classification that is better than the pie-slice decision region technique and the pure AdaBoost one, and is about as good as the pure hand-calibrated fuzzy colour contrast fusion approach. It is also suitable for real-time colour object recognition and contributes to more accurate calculation of object position and orientation for the robot soccer game application.

The system provides a good alternative for colour classification in that it alleviates the colour calibration task by employing AdaBoost training instead of hand-calibrating the entire colour descriptors. Results have shown that the algorithm works under extreme illumination conditions, and even with the presence of similar colours in the scene. Moreover, the experiments performed have tested the limits of the algorithms presented and have demonstrated cases that can be solved using the architecture where other algorithms fail.

5 References

[1] J.H. Kim, *Robot Soccer System Manual*, Intelligent Control Lab, KAIST, Korea (2000).

[2] N. Weiss and N. Jesse, "Towards local vision in centralized robot soccer leagues: a robust and flexible vision system also allowing varying degrees of robot autonomy", *Proceedings 5th Intl Conf on Simulated Evolution and Learning (SEAL04) and 2004 FIRA Robot World Congress*, Busan, Korea (2004).

[3] N.H. Reyes and C.H. Messom, "Identifying colour objects with fuzzy colour contrast fusion", *Proceedings 3rd Intl Conf on Computational Intelligence, Robotics and Autonomous Systems, and FIRA RoboWorld Congress (CIRAS 2005)*, Singapore (2005).

[4] N.H. Reyes, *Color-Based Object Recognition: Analysis and Application*, PhD Dissertation, De La Salle University, Manila, Philippines (2004).

[5] N.H. Reyes and E.P. Dadios, "Dynamic color object recognition", *Journal of Advanced Computational Intelligence*, 8(1), pp 29-38, Tokyo, Japan (2004).

[6] N.H. Reyes and E.P. Dadios, "Machine vision using fuzzy logic", *Proceedings 2003 Int'l Conf on Instrumentation and Control*, Manila (2003).

[7] N.H. Reyes and E.P. Dadios, "Combining fuzzy techniques in color object detection", *Proceedings 1st Int'l Conf on Humanoid, Nanotechnology, Information Technology, Communication and Control Environment and Management (HNICEM)*, Manila (2003).

[8] G. Kwei, K. Barnard and B.V. Funt, "Spotting colours", *Proceedings Colour Image Science 2000*, Derby, UK (2000).

[9] P.J. Thomas, R.J. Stonier and P.J. Wolfs, "Robustness of color detection for robot soccer", *Proceedings 7th Intl Conf on Control, Automation, Robotics and Vision (ICARCV 2002*), Singapore, pp 1245-1249 (2002).

[10] C.H. Messom, G. Sen Gupta and S. Demidenko, "Hough transform run length encoding for real-time image processing", *IEEE Transactions on Instrumentation and Measurement* (forthcoming).

[11] G. Sen Gupta, C.H. Messom and S. Demidenko, "Real-time identification and predictive control of fast mobile robots using global vision sensing", *IEEE Transactions on Instrumentation and Measurement*, 54(1), pp 200-214 (2005).

[12] N. Weiss and L. Hildebrand, "An exemplary robot soccer vision system", *Proceedings CLAWAR / EURON / IARP Workshop on Robots in Entertainment, Leisure and Hobby (ELH'04)*, Vienna, Austria (2004).

[13] Y. Freund and R.E. Schapire, "Experiments with a new boosting algorithm", *Proceedings 13th Intl Conf in Machine Learning*, San Francisco, pp 148-156 (1996).

[14] Y. Freund, "An introduction to boosting based classification", *Proceedings AT&T Conference on Quantitative Analysis*, San Francisco (1998).

[15] Y. Freund and R.E. Schapire, "A short introduction to boosting", *Journal of Japan Society for Artificial Intelligence*, 14(5), pp 771-780 (1999).

[16] T. Ross, *Fuzzy Logic with Engineering Applications*, McGraw-Hill, Singapore (1997).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Working Towards Self-assembling Robots at All Scales

Navneet Bhalla[1], Peter J. Bentley[2]
[1]Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
[2]Department of Computer Science, University College London, London, UK
bhalla@cpsc.ucalgary.ca, p.bentley@cs.ucl.ac.uk

Abstract

From crystals to galaxies, self-assembly is evident throughout nature at all scales. Self-assembly in natural systems is primarily dictated by the morphology of the components within a system and the environmental conditions they are subjected to, as well as their component and environment physical and chemical properties. Several experiments are discussed that demonstrate how to harness these principles of self-assembly to create two-dimensional, physical, geometric mesoscale structures. Primarily, these experiments demonstrate how the relationship between component shape and an assembly protocol can be used to create defined entities of varying form, and could be used as a method for creating more sophisticated netted systems by leveraging the physical attributes of a system. Based on the successful results of these experiments, an evolutionary computation model for applying the principles of self-assembly in nature is also presented, as an approach for working towards the creation of self-assembling robots from the macroscale to the nanoscale.

Keywords: self-assembly, evolutionary computation, nanorobotics, swarm robotics, modular robotics

1 Introduction

Self-assembly (the autonomous construction of a device by itself) is a dream of robotics engineers. A payload of self-assembling components would be easier to transport to hazardous and distant locations compared to complete robots. A device that can self-assemble also has the ability to self-repair or regenerate damaged parts of itself, given replacement components. But, the creation of self-assembling devices is a highly challenging problem.

The concept of self-assembling robots has been a popular theme in science fiction for many years. Only recently have robots been developed that display self-assembly characteristics. These robots are examples of netted systems [1], consisting of sensors and controllers that interact and self-assemble through data communication. These robots demonstrate the synthetic realisation of templated self-assembly [2, 3], biological self-assembly [4], and self-reconfiguration [2, 5, 6 , 7, 8], as examples from the disciplines of modular robotics and swarm robotics. However, such disciplines do not provide a generic methodology to creating self-assembling robots at all scales. This is largely due to scalability issues in relation to their respective methods of communication and assembly between modules or robotic-units.

Here, we refine the term self-assembly and suggest that it should be used to describe a reversible processes that can be controlled by an appropriate design of pre-existing components that interact in order to create emergent aggregate forms [1]. This view of self-assembly is used to link the principles of self-assembly from nature to previous work in robotics and design. Based on this, a general framework is presented that describes the necessary attributes to design a self-assembling system. Experiments and results involving the creation of two-dimensional, physical, geometric mesoscale structures are discussed, to demonstrate how to leverage the self-assembly framework. In addition, an evolutionary computation model [9] is presented as a method to illustrate how harnessing the principles of self-assembly in nature can be used as a design process for working towards self-assembling robots at all scales.

Applying the principles of self-assembly to robotics has tremendous potential. This is especially true at the microscale and nanoscale, where self-assembly is viewed as the only viable means of fabrication [1].

2 Background

L.S. Penrose and R. Penrose were the first to show a mechanical analogue to natural self-assembly, specifically self-reproduction in the form of templated self-assembly [10]. They created two component shapes, labelled A and B, that connected in either an AB or BA configuration. Multiples of these A and B components were confined to a track in a random ordering, that when shaken, allowed components to move horizontally and interact with one another. By placing either an AB or a BA seed complex on the track, it would cause neighbouring A and B or B and A components to self-assemble into AB and BA complexes respectively.

This example of templated self-assembly has recently been extended to robotics [3]. In this case, square-shaped programmed electromechanical components move randomly in two-dimensions on a cushion of air. When components collide, they communicate and latch and unlatch accordingly. Again, by initially placing a seed complex, free components can self-assemble and construct replicas of the seed complex [3].

In these two examples, templates are used to direct the self-assembly process of decentralised components. In contrast, swarm robotics uses swarm intelligence to direct the self-assembly process of decentralised robotic units, in a form of biological self-assembly. Of the robots produced in this discipline, Swarm-bot has shown successful results in mimicking self-assembling formations of social insects (e.g. the formation of living bridges by *Oecophylla longinoda* worker ants) [4]. Swarm-bot is the collective name to the set of cube-shaped mobile robotic units, named s-bots, which are capable of physically linking together. For example, s-bots can self-assemble into aggregate structures to move across terrain, otherwise not possible by an s-bot solely.

The discipline of modular robotics has produced self-reconfigurable robots using both decentralised and centralised control systems [5]. These robots posses the ability to self-reconfigure a pre-exisiting set of modules that are physically connected together, and that move and attach/detach in terms of the degrees of freedom allowed by the components.

Using a hormone-inspired communication method between robotic units, CONRO is an example of a modular robotic system using a decetralised control system [8]. This system facilitates communication betwwen subsets of robotic units, allowing for more roboust self-reconfiguration and locomotion capabilities.

Two of the most successful centralised modular robot implementations to date include PolyBot [6] and MTRAN [5]. PolyBot (a precursor to CONRO) uses cube-shaped modules with one axis of rotation, which are capable of self-reconfiguring into various forms with movement such as in a loop, and in a snake-like and spider-like fashion [6]. MTRAN modules consist of two semi-cylindrical parts connected by a link, with each part being able to rotate one hundred and eighty degrees about its axis. Each semi-cylindrical part has four permanent magents, on its three surfaces, allowing modules to attach and detach from one another. These modules allow MTRAN to self-reconfigure into forms with one type of crawler and two types of quadruped movement [5].

The use of magnetism as an assembly mechanism to facilitate self-reconfiguration of robotic units has also been leveraged to create modular robotic systems capable of templated self-assembly [2] and self-reproduction [7]. These two examples incorporate self-reconfiguration as intermediate steps, to increase efficiency, in their procedures of self-reproduction and templated self-assembly.

These robots are all implementations of subsets of self-assembly, in the form of netted systems. In nature, self-assembly is primarily dictated by the design of the components within a system and the environmental conditions they are subjected to, as well as their component and environment physical and chemical properties [11, 12]. The following section describes a general framework for self-assembling system, which covers the above mentioned types of self-assembly currently used in practice (templated self-assembly, biological self-assembly, and self-reconfiguration) and the potential to create self-assembling robots in the future, particularly at the nanoscale.

3 Framework

For the purposes of creating an artificial self-assembling system, the natural principles of self-assembly can be abstracted to four items:

- Components
- Environment
- Assembly Protocol
- Energy

Components are defined by their properties. Such properties include, but are not limited to, shape, scale, material properties, as well as communication methods and interaction methods between components and/or their environment.

The environment in which components are subjected to can provide various functionalities, such as a boundary to which components are confined to. The physical and chemical properties of the environment will also influence the nature in which components interact with one another, as well as the way in which components self-assemble.

An assembly protocol defines the methods in which components can self-assemble (e.g. methods of attraction and repulsion). These methods are highly dependent on the scale of the system, as well as the physical and chemical properties of the components and the environment.

In order for the components to self-assemble, the components need to be mobile in their environment. This requires the components to have energy. This can either be available internally or transferred to components, for example, by the environment.

This self-assembly framework should be considered from the viewpoint of specific self-assembling systems. Physical constraints are normal in such systems, as we can observe in nature. A sand dune will only form in specific circumstances; if the wind force is not sufficient, it will not form. However, by continuing to gain a deeper understanding of self-assembly in nature, it can be leveraged for the purposes of design. This of course can be utilised by robotics, and the creation of simple self-assembling mechanical structures (e.g. pivots, joints, and levers) would be a fundamental next step.

4 Experiments

One possible solution to creating simple mechanical structures is to utilise the relationship between component shape and an assembly protocol. Here, the relationship is investigated in the context of creating two-dimensional, physical, geometric mesoscale self-assembling structures.

Experiments were conducted to investigate whether a set of two-dimensional components (with concave and/or convex polygon shapes) could self-assemble into a desired shape. The assembly process is initiated by placing components on a tray, which is shaken in parallel to the surface of the tray. In this way energy is transferred to the components in the form of vibration, causing the components to move around and interact with one another; and magnetism is used to enable the components to attract and repel one another.

In this context, a component must have two essential properties; the first being the ability to fit together to form the desired shape and the second being the ability to join selectively to corresponding components or not to conflicting components. To achieve the first point, a set of components must include both concave and convex component shapes. By the components' shapes being both concave and convex, components are able to create stronger joints, leading to more stable structures overall, and less likely to break apart when colliding with other components or the sidewalls of the tray, compared to if components' shape were restricted to convex forms only. The second point is achieved by placing a magnet in the interior of a nonmagnetic material. The magnets allow components with opposite polarity to attract and assemble together, whereas components having similar polarity will repel each other and therefore not assemble together. The nonmagnetic material is used to determine the polygon form of the components. By not allowing the magnets in the components to join directly together, the components have a higher degree of freedom to move around in the given space and interact with one another. Figure 1 shows the principles behind the component design.

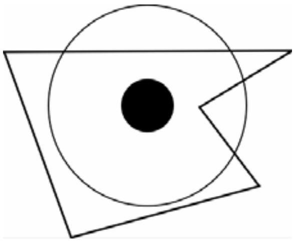


Figure 1: Component design. Solid black circle represents magnetic disc. Outer circle represents area of the magnetic field. Irregular pentagon represents non-magnetic material that defines component's shape (left and bottom of component are areas not affected by the force of magnetism).

The components are placed on a tray, which allows a space in which the component shapes can move around and interact with one another. Movement of the components and their interaction is dictated by two-dimensional rigid body dynamics and magnetism. Figure 2 shows the three stable two-dimensional formations of magnetic discs.

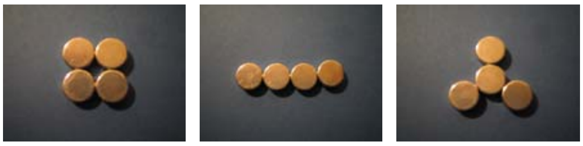


Figure 2: The three stable two-dimensional formations of magnetic discs: grid (left); chain (centre); and triangular (right).

To test the validity of this design, five experiments were conducted. Components were constructed out of foam board, magnetic discs, and scotch tape. The tray was constructed out of foam board, pushpins, and general purpose adhesive. Each of the five experiments had a different number of components and different desired final forms. Symmetric and non-symmetric component shapes, along with the three stable two-dimensional magnetic disc formations, were also tested to see if they had an effect on the self-assembly process. Table 1 summarises the design and purposes of each of the five experiments.

Table 1: Experiments.

Experiment	No. of component shapes	Symmetric vs. non-symmetric component shapes	Magnetic formations
1. Triangle	4	symmetric	triangular
2. Square	4	non-symmetric	grid
3. Parallelogram	6	3 sets of 2 of symmetric shapes	grid and chain
4. Irregular octagon	7	1 set of 4 symmetric shapes, and 3 non-symmetric shapes	chain and triangular
5. 16-sided polygon	10	non-symmetric	grid, chain and triangular

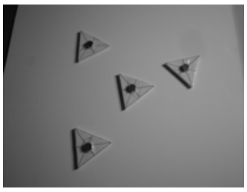
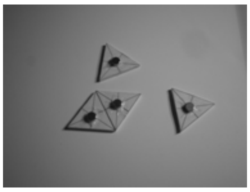
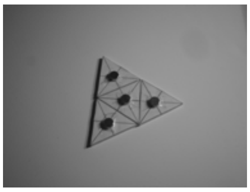
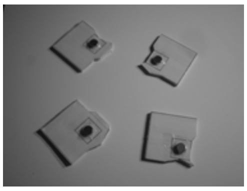
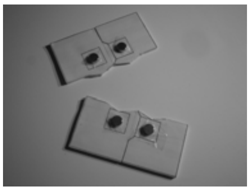
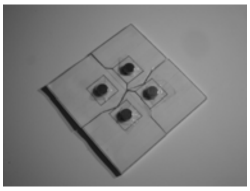
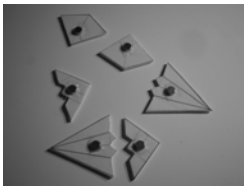
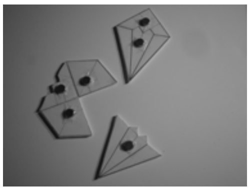
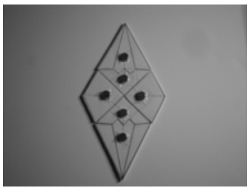
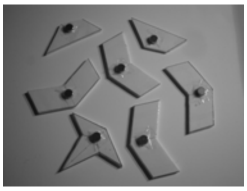
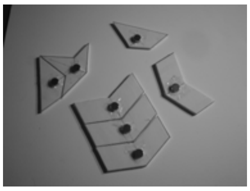
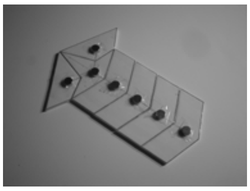
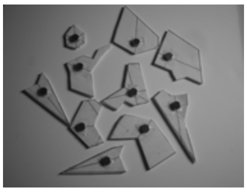
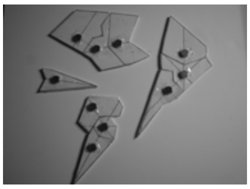
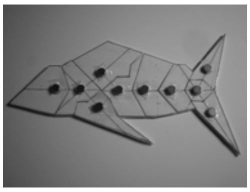
5 Results

Each of the five experiments were successful in having their set of components self-assemble into their corresponding desired final form. Symmetric systems and systems with a lower number of components were able to self-assemble faster in general, in comparison to non-symmetric systems or systems with a large number of components. Table 2 shows the results of the five experiments.

These results demonstrate how the relationship between component shape and an assembly protocol can be used to create self-assembling entities of varying form. Although this combination of shape and an assembly protocol (magnetism) does not apply to all scales, it does however suggest that physical (as well as chemical) properties of a system can be leveraged to aide in creating netted systems.

This relationship of component shape and an assembly protocol allowed for a larger set of feasible self-assembling entities (in the context of the experiment setup and design). In particular, this combination allowed for the exploitation of an effective magnetic force (regions of a component in which the effects of magnetism were subjected to neighbouring components) to create closed self-assembled forms. These forms, in contrast to open forms, do not allow for free components to self-assemble to the entity, when it reaches its target end state. This emergent property is achieved by the way in which component shape is utilised. By allowing a component's shape to be larger than the effective

Table 2: Experiment results.

	Configuration		
Experiment	**Initial**	**Intermediate**	**Final**
1. Triangle			
2. Square			
3. Parallelogram			
4. Irregular octagon			
5. 16-sided polygon			

magentic force region, components of opposite magentic polarity are not able to join together, because the magnetic force is not greater than the force of friction between the components and the surface of the tray. Closed self-assembled forms are of particular interest to self-assembling robotics.

Another application of these results is that they demonstrate that physical properties, in this case shape and magnetic attraction/repulsion, can be used as a physical encoding, and as a communication mechanism between components. This concept could also be extended to chemical properties. Physical and chemical properties could be used to replace, simplify, or enhance communication and interaction mechanisms between modules or robotic-units, in self-assembling robots.

Understanding and utilising the principles of self-assembly in nature, could be used for the realisation of nanorobots. At the nanoscale, self-assembly is considered as the only viable means of fabricating entities [1]. At this scale, as well as all others, numerous variables affect the process of self-assembly. Optimisation algorithms can be used to generate the specifications of the components and environment to create self-assembling systems [9].

6 Evolutionary Computation Model

With its ability to navigate through complex problem spaces, evolutionary computation has poven to be an extremely useful approach to solving optimisation problems. As well, evolutionary computation can be used as a creative tool. This is most notabily seen in its ability to generate novel designs [13]. This duality of evolutionary computation makes it a prime candidate to be incorporated into a process for designing and physically creating self-assembling systems [9].

One embodiment of this process [9] can be described as an eight-step process. These steps include:

1. Define the properties of the desired self-assembling entity.
2. Encapsulate the component design, environment design, and/or construction process (refering to the methodology in which the components and/or environment would physically be created, e.g. using rapid prototyping techniques). This encapsulated information would be encoded into the genotype and phenotype representations of the components and environment.
3. Define the translation process in which the computer generated designs can be used to physically create the components and/or environment (e.g. translating the software representations of the components and/or environment to CAD files).
4. Create software that incorporates a computer model using evolutionary computaion to virtually design and test the candidate components and/or environment, to allow for self-assembly of the components into the desired entity.
5. Execute the software to generate the designs of the components and/or environment.
6. Execute the translation process of the computer generated component/and or enviroenment designs, to a form that can be used for physical fabrication.
7. Build the components and/or environment.
8. Place the components in their environment to allow for the components to self-assemble into the desired entity.

In this evolutionary computation model, the notion of a design space (the set of buildable designs of components and/or environment) is crucial. If this space is ill-defined, it will greatly affect the performance of the software, as well as inhibit the creation of the self-assembling system.

The encapsulation of a design space is a complicated task. However, it is of great importance, especially in using this process [9] for creating physical system, such as self-assembling robots. Using this process [9] to create simple self-assembling entities with features of simple mechanical machines (e.g. pivots, joints, and levers) would be an important next step, with benefits from the macroscale to the nanoscale. Preliminary results from an implementation of this embodiment, which used a genetic algorithm to evolve shapes and a simulator to model their interaction are described in [14]. Work on this area is ongoing by the authors.

7 Conclusions

The principles of self-assembly in nature should be considered when creating self-assembling robots. The general framework (consisting of a set of components, an environment, an assembly protocol, and energy) provides a method for understanding the requirements of a specific self-assembling system.

Experimental results showed that this framework yielded two-dimensional geometric self-assembling mesoscale structures. This was achieved through understanding the relationship between component shape and an assembly protocol. This relationship allowed for the creation of closed self-assembled forms, as well as insight into the use of this relationship to physically encode communication and interaction mechanisms. This could be used to replace, simplify, or enhance communication and interaction mechanisms between modules or robotic-units in self-assembling robots.

As well, an evolutionary computation model was outlined, as a method for working with the numerous variables affecting the process of self-assembly. The process described [9] could be used in working towards self-assembling robots at all scales.

8 References

[1] G.M. Whitesides and B. Grzybowski, “Self-assembly at all scales”, *Science*, 295(5564), pp 2418-2421 (2002).

[2] P. White, V. Zykov, J. Bongard and H. Lipson, “Three dimensional stochastic reconfiguration of modular robots”, *Proceedings of Robotics Science and Systems*, Cambridge (2005).

[3] S. Grifth, D. Goldwater and J. Jacobson, “Robotics: self-replication from random parts”, *Nature*, 437(7059), pp 636 (2005).

[4] R. Gross, M. Dorigo and M. Yamakita, “Self-assembly of mobile robots: from swarm-bot to super-mechano colony”, *Proceedings 9th Intl. Conf. on Intelligent Autonomous Systems (IAS-9)*, Tokyo, pp 487-496 (2006).

[5] A. Kamimura, S. Murata, E. Yoshida, H. Kurokawa, K. Tomita and S. Kokaji, “Self-reconfigurable modular robot: experiments on reconfiguration and locomotion”, *Proceedings 2001 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS ’01)*, Maui, pp 606-612 (2001).

[6] D.G. Duff, M. Yim and K. Roufas, “Evolution of PolyBot: a modular reconfigurable robot”, *Proceedings Harmonic Drive Intl. Symposium*, Nagano (2001).

[7] V. Zykov, E. Mytilinaios, B. Adams and H. Lipson, “Self-reproducing machines”, *Nature*, 435(7038), pp 163-164 (2005).

[8] W.-M. Shen, B. Salemi and P. Will, “Hormone inspired adaptive communication and distributed control for CONRO self-reconfigurable robots”, *IEEE/ASME Trans. on Robotics and Automation*, 18(5), pp 700-712 (2002).

[9] N. Bhalla and P.J. Bentley, *Self-assembling Systems*, UK Patent 0614400.0 (2006).

[10] L.S. Penrose and R. Penrose, “A self-reproducing analogue”, *Nature*, 179(1183) (1957).

[11] D. W. Thompson, *On Growth and Form*, Cambridge University Press, Cambridge (1917).

[12] P. Ball, *The Self-made Tapestry: Pattern Formation in Nature*, Oxford University Press, Oxford (1999).

[13] P.J. Bentley, “The revolution of evolution in design: from coffee tables to hospitals”, *Proceedings Recent Advances in Soft Computing ’98*, Leicester, pp 172-182 (1998).

[14] N. Bhalla, *Self-Assembling Systems in Two Dimensions*, MSc Project Dissertation, Department of Computer Science, University College London (2004).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Hierarchical Specification and Design of Control Systems in Robotics

Valery Sklyarov, Iouliia Skliarova
Department of Electronics, Telecommunications and Informatics, IEETA
University of Aveiro, Aveiro, Portugal
skl@det.ua.pt, iouliia@det.ua.pt

Abstract

Control systems that manage the behaviour of collaborative robots are in general hierarchical and parallel. Hierarchy permits the required functionality to be specified at different levels of abstraction. Higher levels of abstraction include more general operations that are described in detail at lower levels. Parallelism allows several robot components to be managed simultaneously. Control systems collaborate with execution systems (that are composed of mechanical blocks) in analysing signals from sensors and affecting actuators. The paper suggests a method for the specification of hierarchical control systems and for the design of such systems based on the model of a hierarchical finite state machine. The latter has been simulated in software using the C++ language and implemented in hardware (in FPGA, in particular) based on the developed VHDL templates. The results of experiments demonstrate that the proposed technique offers a number of advantages.

Keywords: control systems, hierarchical and parallel specification, finite state machines, VHDL templates

1 Introduction

In collaborative systems, robots interact with each other to achieve an integrated behaviour aimed at the solution of a common problem. A good example of such a system was considered in [1], which describes collaborative robot soccer with possible dynamic allocation of the roles (namely, goalkeeping, defence and offence). Agent behaviour was implemented using traditional state transition based control and one of the conclusions was that "the complexity of the system can be greatly simplified and easily managed by building a hierarchy of state transition diagrams". A number of examples have been shown in [1], which prove that hierarchy is a very important technique in a system of collaborative robots.

It is known that modular multilevel specification can be provided using a language called hierarchical graph-schemes (HGS) [2], which permits modules (sub-algorithms) and interactions between modules to be described. Consequently, HGS offer direct support for the hierarchical specification of algorithms, for example, for the hierarchical description of the state transition diagrams considered in [1]. In addition, HGS possess another very important characteristic. Since any HGS is considered to be a collaborative set of modules, which can be changed/altered if required, direct support for reconfiguration can easily be provided and this capability will be shown in the paper. Reconfiguration is very important for many practical applications, for example, for robot navigation systems [3]. Indeed, very few mobile robots can move from one environment to another without losing some performance or capabilities [3]. The technique proposed makes it possible to modify the functionality of control systems (within the same hardware resources) easily, just by replacing modules. In fact, just an HGS module call has to be altered to make a given change.

The technique proposed in this paper provides direct support for behaviour-based robotics with the possibility of dynamic changes. A modular specification permits a collection of behaviours (modules) to be developed and used that achieve or maintain a given set of goals. Another very important aspect of the specification method proposed is that it is directly supported by an implementation model, which is a hierarchical finite state machine [2].

The remainder of this paper is organised in six sections. Section 2 describes the hierarchical specification of control systems. Section 3 characterises hierarchical finite state machines that allow hierarchical specifications to be implemented in hardware. Section 4 discusses software modelling of hierarchical algorithms and presents some additional details of the hardware implementation. Section 5 shows how parallel algorithms can be described and implemented. Section 6 is devoted to experiments. The conclusion is in section 7.

2 Hierarchical Specification of Control Systems

Hierarchical algorithms described in HGS are constructed from modules with the aid of the method [2]. They are very useful for a number of practical applications (see, for example, [1]). Since all the required formalities are presented in [2] we will consider just an example taken from the scope of embedded systems. Suppose a system receives messages from an external source. The messages have to be buffered and processed sequentially according to their priority. Each incoming message changes the message sequence, because it has to be inserted in the sequence in a position that is in accordance with its priority. Let us assume that the following strategy is applied:

1. All incoming messages are stored in a binary tree, whose nodes contain four fields: a pointer to the left child node, a pointer to the right child node, a counter, and a message. The nodes are maintained so that at any node, the left sub-tree contains only messages that are of a lower priority than the message at the node, and the right sub-tree contains only messages that are of a higher priority. The counter indicates the number of occurrences of the message associated with the current node (this is because some messages can arrive more than once). It is known that such a tree can be constructed and used for sorting various types of data [4]. In order to build this tree for a given set of messages, we have to find the appropriate place for each incoming node in the current tree. In order to sort the data, we can apply a special technique [4] using forward and backward propagation steps that are exactly the same for each node. Thus, a modular (and moreover recursive) procedure is very efficient.

2. As soon as a processing unit is free, a message from the binary tree with the highest priority is selected and processed.

Figure 1 depicts the HGS specification for the example, which is very similar to the sorting algorithm described in [5]. The same model can be employed if there is more than one processing unit and several units are working in parallel.

The HGS Z_0 first constructs the tree incrementally (calling HGS z_1) while input data are available. The latter is indicated by the logic condition x_1 (x_1=0 if data are available and x_1=1 if the flow of incoming data has finished). Then the HGS Z_0 sorts the data with the aid of the HGS z_2.

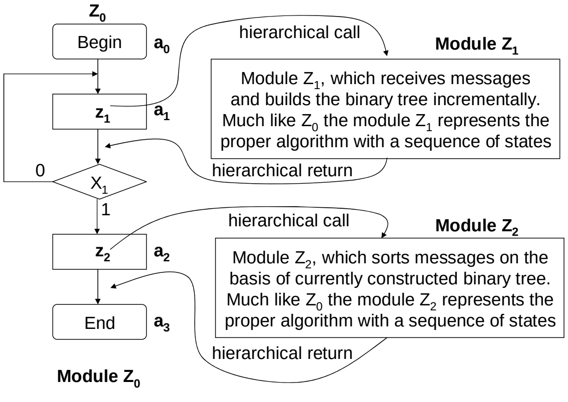

Z_0
Begin
a_0
hierarchical call
Module Z_1
Module Z_1, which receives messages and builds the binary tree incrementally. Much like Z_0 the module Z_1 represents the proper algorithm with a sequence of states
z_1
a_1
hierarchical return
0
X_1
1
hierarchical call
Module Z_2
z_2
a_2
Module Z_2, which sorts messages on the basis of currently constructed binary tree. Much like Z_0 the module Z_2 represents the proper algorithm with a sequence of states
End
a_3
hierarchical return
Module Z_0

Figure 1: HGS specification of a message processing system.

Formally an HGS is a directed connected graph containing rectangular and rhomboidal nodes. Each HGS has one or more entry points (see the node *Begin* in figure 1) and one or more exit points (see the node *End* in figure 1). Other rectangular nodes contain either *micro instructions* or *macro instructions*, or both. Any *micro instruction* Y_j, includes a subset of *micro operations* from the set Y={y_1,...,y_N}. A *micro operation* is an output signal, which causes a simple action in the execution unit. Any *macro instruction* incorporates a subset of *macro operations* from the set Z={z_1,...,z_Q}. Each *macro operation* is described by another HGS of a lower level. Each rhomboidal node contains one element from the set X, where X={x_1,...,x_L} is the set of *logic conditions*, i.e. input signals, which communicate the result of a test. The sets Y and X are associated with actuators and sensors of the respective execution unit. Directed lines (arcs) connect the inputs and outputs of the nodes in the same manner as for a graph-scheme (GS) [6].

Any HGS can be formally converted to a finite state machine with stack memory [2]. Figure 1 shows the states a_0,…,a_3 of the module Z_0 for such machine.

As we can see, in general an HGS is composed of modules. There are two acceptable and reasonable ways in which the binding between modules can be established [2]. These are before runtime and during runtime. The first kind of binding is called static because it is established outside the scope of dynamic physical control. Dynamic binding sets up links during runtime. This means that while one HGS is executing, another HGS might be either being modified or replaced. Any two HGS can be swapped by setting a new entry point in the appropriate module call. This feature provides direct support for static and dynamic reconfiguration. In addition, we can create a library of reusable modules.

3 Hierarchical Finite State Machine

An HFSM [2,5] permits the execution of an HGS (such as that shown in figure 1) and contains two stacks, one for states (*FSM_stack*) and the other for modules (*M_stack*). The stacks are managed by a combinational circuit (CC) that is responsible for new module invocations and state transitions in any active module that is designated by outputs of the *M_stack*. Since each particular module has a unique code, the same HFSM states (state codes) can be repeated in different modules. Any non-hierarchical transition is performed through a change of a code only on the top register of the *FSM_stack*. Any hierarchical call alters the states of both stacks in such a way that the *M_stack* will store the code for the new (called) module and the *FSM_stack* will be set to the initial state (normally to a_0=0…0, containing all zeros) of the module. Any hierarchical return just activates a pop operation without any change in the stacks. As a result, a transition to the state following the state where the terminated module was called will be performed. The stack pointer *stack_ptr* is common to both stacks. If the *End* node is reached when *stack_ptr*=0, the algorithm terminates execution.

4 Software Modelling and Hardware Implementation of Hierarchical Algorithms

All blocks of the HFSM considered in [2,5] have been described as classes in C++. The software that has been developed allows different hierarchical algorithms (HGS) to be tested, which can be read from files and executed in an HFSM. The latter is considered to be a set of objects instantiated from the appropriate classes (namely the stack class common to both stacks, and the combinational circuit - CC). CC is configured to run the supplied algorithm, which is read from the file. The algorithm can be executed in a virtual space allowing components of execution units to be modelled much like the technique proposed in [7]. A virtual execution unit is presented on a monitor screen in the form of interconnected visual objects, such as the execution part of a transfer line. A specific class in the C++ programme supports each visual object.

This technique has a very important advantage. All the required algorithms can be tested in software and then converted to the proper hardware implementation. If the required execution unit cannot be found in the library, it can easily be created and attached to the system under investigation through pre-designed interfaces.

The functionality of the HFSM can be described in a hardware description language (HDL) much like it was done in [8], where all the required details can be found. The distinctive feature of the HDL code (compared with traditional FSMs) is the provision of state transitions at the following two levels:

- At the first level the code on the top register of the *M_stack* enables the HFSM to recognise the active module.
- At the second level the code on the top register of the *FSM_stack* and the values of external input variables (from the set X) enable the HFSM to execute the proper state transition, i.e. to form the code of the next state and, in the case of a hierarchical transition, the code of the next module.

Methods [7] make it possible to communicate between circuits implemented in hardware (in FPGA, in particular) and between virtual objects of robots and embedded systems simulated in software. This technique provides for a systematic approach to the development of hierarchically specified control systems in complex environments.

5 Parallel Logical Control

The hardware implementation of parallel algorithms is a complicated problem. In the general case it is very difficult and resource consuming to implement hierarchy and parallelism within the same control system. We would like to underline that we are talking about real parallelism and we will not be discussing methods for the implementation of pseudo-parallel algorithms that are widely used in software for embedded systems. This section is divided into three subsections showing: 1) how to describe parallel logical control algorithms; 2) how to implement parallelism without hierarchy using a modified technique [9]; 3) how to implement hierarchy and parallelism within the same control system.

5.1 Description of Parallel Logical Control Algorithms

Parallel control algorithms can be described with the aid of the PRALU language proposed in [10]. However due to some unclear constructions in [10] that might lead to errors, the language was slightly modified, which permitted the gap between PRALU and graphical specifications (such as HGS [2,9] and GS [6]) to be bridged. Consequently not only textual specifications, but also graphical specifications can easily be used for describing parallel logical control algorithms.

Let us consider the sets $Y=\{y_1,...,y_N\}$ and $X=\{x_1,...,x_L\}$, introduced in section 2, and associated with the actuators and sensors of the respective execution unit. To make the relationship between the elements of Y/X and the actuators/sensors of the physical systems clearer, we will use abbreviations formed from the first letters of the relevant names in normal font for actuators, and in *italic* font for sensors. For example, the abbreviation for a “left sensor” will be written down as *ls* and the abbreviation for an actuator “move left” will be written down as ml. Let us introduce the following syntactic constructions for our specification, which differ from [10]:

- -(*ls*=0) or -(*ls*=1) – wait for *ls*=0 or *ls*=1 and only after that proceed to the operation written on the right-hand side of the relevant expression such as -(*ls*=0) or -(*ls*=1);
- (ml⇒0) / (ml⇒1) – assign ml the value 0 / 1;
- Specifying just an abbreviation (such as ml) for an actuator without any value (such as ⇒1) means that the actuator receives an active value just during the respective transition and passive value after the transition. Later on this feature will be demonstrated on examples;
- Square brackets are used to combine expressions, for example “-(*stop*=0) […] (*stop*=1) → 3” means: when *stop*=0 execute sequence in square brackets and when *stop*=1 execute transition to label 3.
- All the other constructions and rules are similar to [10].

Each specification has to be provided in such a way that ambiguity is avoided. For example, for any line like “-(*ls*=0) → (ml⇒1) → …”, which says “assign ml⇒1 when *ls*=0", we also have to know what to do when *ls*≠0, which is unclear in expressions in [10]. If in the expression above the value -(*ls*=1) is not explicitly specified, it denotes: "no operation until *ls*=0 and ml⇒1 when *ls*=0".

5.2 Implementation of Parallel Algorithms Based on One-hot Encoding Technique

It is known [9] that ordinary (non hierarchical) GS [6] can be directly mapped onto FPGA-based circuits. The primarily idea of [9] is to set a correspondence between operational (rectangular) nodes of GS and individual flip-flops in FPGA (this is a pure GS-based one-hot state encoding technique). As a result, the number of flip-flops will be essentially increased compared with binary encoding. In practice this is not a problem because the cost of a flip-flop is negligible in electronic circuits (such as an FPGA or an ASIC). The method [9] can also be used for parallel control algorithms, which will be demonstrated through an example. Figure 2 depicts two interacting objects of a self-controlled transport section: a robot on the left hand side; and a container on the right-hand side that can be transported from left to right and vice versa. The following operations (provided through the relevant actuators) have to be executed:

- take (t) – the robot takes something (on the left-hand side);
- deliver (d) – the robot delivers something to the container;
- move right (mr/MR) – the robot/the container has to be moved right;
- move left (ml/ML) – the robot/the container has to be moved left.

The following sensors will be used:

- *completed* (*c*) – delivery is complete when (*c*=1) and it is not complete when (*c*=0);
- *left sensor* (*ls*) / *LEFT SENSOR* (*LS*) – left sensor for the left / right transport section;
- *right sensor* (*rs*) / *RIGHT SENSOR* (*RS*) – right sensor for the left / right transport section;
- *FULL* (*F*) – the container is full when (*F*=1) and the container is not full when (*F*=0).

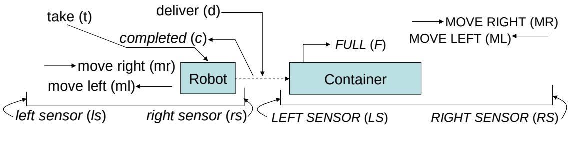


Figure 2: Self-controlled transport section.

Let the container on the right-hand side be unloaded somehow (we will not worry about this). All the other operations will be included in the specification of the algorithm. In accordance with the description above, the robot is controlled by two actuators mr and ml: if mr⇒1, the robot moves to the right; when ml⇒1, the robot moves to the left; (mr⇒1)&(ml⇒1)≡0. The track, where the robot is moving, is bounded by the sensors *ls* and *rs* in such a way that if *ls*=1/*rs*=1, the robot is at the left/right edge. The container has exactly the same behaviour, except capital letters are used instead of lower-case letters in the description. The transfer is controlled by two actuators d and t in such a way that if t⇒1 the robot takes something at the left-hand side of the transfer line and if d⇒1, the robot delivers something to the container. Sensor *c* signals (*c*=1) that the delivery has completed. Assuming that several iterations are needed for the robot to load the container, the following algorithm described in modified PRALU language is suggested (all the details of PRALU language can be found in [10]).

```
Start Parallel Algorithm: →1.2                          (1)
1:  -(stop=0) → [-(ls=0) → (ml⇒1)  → 1;
                   -(ls=1) → (ml ⇒0) t → -(rs=0) →
                                        (mr⇒1) → ;
                                          -(rs=1) →
          (mr⇒0) -(F=0; LS=1) → d → -(c=1) → 1;]
    -(stop=1)  → 3
2:  -(stop=0) → [-(F=0;LS=0) → (ML⇒1) → 2;
                 -(LS=1) → (ML⇒0) → -(F=1; RS=0)
                                    → (MR⇒1) → 2;
                                       -(RS=1) →
                                     (MR⇒0) → 2;]
    -(stop=1) → 4
3,4 → end
```

Here the symbol "-" is interpreted as "wait for"; the symbol "→" means "execute"; when *stop*=1 the execution of the algorithm is ended (however any currently executing operation has to be completed before the termination of the algorithm). If there is just a semicolon on the right-hand side of the operator "→", the previous test of sensors (logical conditions) has to be repeated. For example, after the rightmost operator "→" in the expression "-(*rs*=0) → (mr⇒1)→;", the logical condition *rs* has to be tested again in such a way that while *rs*=0, mr⇒1. Chains with labels 1 and 2 are executed in parallel and they describe the control of the robot and the container, correspondingly. Note that such control is considered just for explanatory purposes and is greatly simplified. It is assumed that at the beginning the robot and the container are at arbitrary positions.

The rules (derived from [9]) for constructing a parallel control device (implementing the described algorithm) are the following:

- Use the method [9] without any change for ordinary chains like -(*F*=0;*LS*=0)→(ML⇒1)→2; -(*LS*=1)→

$(ML \Rightarrow 0) \rightarrow (F=1; RS=0) \rightarrow (MR \Rightarrow 1) \rightarrow 2;$ - $(RS=1) \rightarrow (MR \Rightarrow 0) \rightarrow 2;$

- Activate more than one flip-flop for parallel branches like *Start Parallel Algorithm: →1.2* (we have to activate in this particular transition two flip-flops for the chains 1 and 2 in parallel);
- Use the AND function in the case of chains merging such as 3,4 → end. This permits the two parallel chains 3 and 4 to be terminated.

Since we do not have other types of transitions, the method [9] can be applied directly. Output functions can be implemented trivially (similarly to any finite state machine based on one-hot state encoding – see [9] for details).

Note that any description such as that considered above can be formally converted to an equivalent description in HGS, which was shown in [11]. For example, formulae (1) can be converted to the HGS depicted in figure 3a (~~RE~~ designates that re-entrance to any module Z_1,Z_2 is prohibited). Figure 3b demonstrates how the HGS can be extended to provide a closer relationship to (1). Figure 3c shows how the module Z_1 (the chain 1) can be described in HGS. The other module Z_2 (chain 2) can be similarly and easily described. In an analogous way, an HGS specification can be converted to PRALU. Thus, the two specification methods considered are mutually convertible.

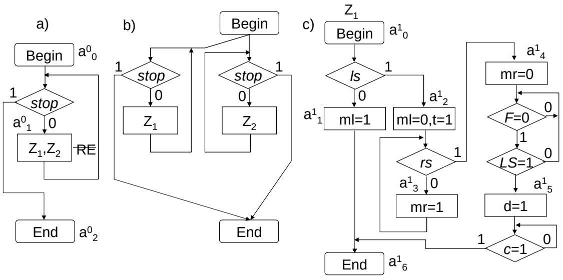


Figure 3: (a) HGS for the example in figure 2, (b) interpretation of parallel chains, (c) HGS for the module Z_1.

Figure 4 demonstrates how to implement the HGS depicted in figures 3a and 3b in hardware. Labels a^i_j in figure 3 correspond to FSM states (see [2,9] for details), where i is an index of module Z_i and j is a number of state in the module Z_i. In one-hot state encoding for each state a flip-flop is allocated (see figure 4).

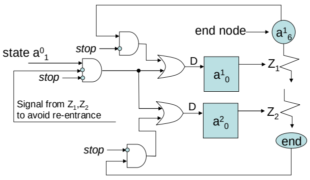


Figure 4: Hardware implementation of parallel branching and merging operations.

Let us assume that initially all the flip-flops (of D type) are reset to 0 (i.e. all their outputs shown in figure 4 are set to 0). To start the algorithm we have to activate the state a^0_0 (see figures 3 and 4). This can be done by setting the flip-flop a^0_0 in the state "1". Subsequently, if *stop*=0 a transition to the state a^0_1 is executed and then two flip-flops a^1_0 and a^2_0 are set to 1 in parallel and they force parallel execution of the modules Z_1 and Z_2. After any transition to the "end" node, the condition (*stop*=0)&("end node is active") is tested. If *stop*=1, the first state in the relevant module (either Z_1 or Z_2) is not activated and all the flip-flops used for this module will be set to 0. When both Z_1 and Z_2 are terminated a transition to the state a^0_2 is executed. Thus, the modules Z_1 and Z_2 cannot be executed until the flip-flop a^0_0 is set to "1" again. Thus, to start execution again it is necessary to reset a^0_2 and to set a^0_0. It should also be mentioned that the requirement considered above, namely "any currently executing operation has to be completed before termination of the algorithm", is satisfied. Implementation of any sequential algorithm, such as that shown in figure 3c, can be provided using the method [9] without any change.

Note that it is not necessary to allocate states such as a^1_0 and a^2_0. Indeed, for each entry point to the modules Z_1, Z_2 we can use states immediately following the node "begin", such as a^1_1 and a^1_2 in figure 3c. However, the relevant logical conditions (such as *ls* in figure 3c) have to be properly tested.

5.3 Hierarchical Parallel Logical Control Systems

It is known [10] that PRALU descriptions are synthesisable. The relevant hardware circuits can be built on the basis of a parallel FSM with partial states in such a way that the FSM can be in several of these states at the same time. This is achieved through a special ternary encoding allowing state codes to be assigned such that they are intersected for parallel algorithmic branches (chains) making it possible to recognise two or more different states at the same time. For example, the code 0-1-0 enables us to recognise the state 011-0 for one chain and the state 0-100 for another chain. The details of this technique are rather complicated and we will omit them due to limitations in the length of this paper. This method is entirely described in [10]. We would like to underline that using the method [10], parallel algorithms can be implemented based on ternary state encoding allowing the size of state registers to be decreased significantly compared with one-hot state encoding (although the actual size is, as a rule, much larger than $\lceil \log_2 R \rceil$, where R is the number of states). Decreasing the size of state codes is very important for stack-based implementations. Since the PRALU language can be formally converted to HGS and the latter can be implemented in hardware using an HFSM model, we can synthesise circuits for parallel hierarchical algorithms (although some constraints that are presented in [12] have to be taken into account). In other words using the HFSM model we can also provide a stack-based invocation of sub-algorithms (modules/chains) allowing us to take advantage of the method [2]. Since the mechanisms that support hierarchy are based on pre-established operations with a stack memory, and are separated from the logic synthesis of primary combinational circuits, we can also apply the methods for the synthesis and verification of parallel algorithms that are considered in [10,11].

6 Experiments

The primary goal of the experiments was to prove on arbitrarily selected working examples that the methods presented in the paper are correct. The control systems were implemented in FPGAs of Spartan-IIE family of Xilinx. We have used the stand-alone board TE-XC2Se [13]. The results of experiments have shown that the proposed methods and tools make possible to construct control systems implementing hierarchical or parallel algorithms and can be used for practical applications. We found that although hierarchical and parallel algorithms can be implemented within the same system, using hierarchical parallel algorithms is very resource consuming and many constraints have to be taken into account. In many cases, autonomous HFSMs working in parallel are better. Hierarchical circuits based on the proposed templates are very fast and do not require many resources. Parallel implementations based on one-hot encoding technique are also quite efficient in terms of resources and execution time.

7 Conclusion

The paper describes two types of specifications that can be efficiently used to describe the functionality of parallel and hierarchical control systems. Such systems can be seen as embedded controllers and they are needed for many practical applications in robotics. The specifications considered (namely HGS and PRALU) are mutually convertible and permit the methods and tools available for them to be combined. For example, an HGS is a formally synthesisable specification based on the model of a hierarchical finite state machine and PRALU is supported by numerous methods for the synthesis of the combinational part for a parallel FSM and for formal verification. Finally, the paper shows how to design hierarchical, parallel and hierarchical parallel systems from specifications presented in the languages considered and summarises the advantages and disadvantages of each particular implementation. The tools that have been developed for software/hardware modelling of control systems interacting with execution units are also described.

8 Acknowledgment

The authors would like to acknowledge Ivor Horton for his very useful comments and suggestions.

9 References

[1] G. Sen Gupta, C.H. Messom and S. Demidenko, “State transition based (STB) role assignment and behaviour programming in collaborative robotics”, *Proceedings 2nd International Conference on Autonomous Robots and Agents*, Palmerston North, New Zealand, pp 385-390 (2004).

[2] V. Sklyarov, “Hierarchical finite-state machines and their use for digital control”, *IEEE Trans. on VLSI Systems*, 7(2), pp 222-228 (1999).

[3] Y. Meng, “A dynamic self-reconfigurable mobile robot navigation system”, *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, USA, pp 1541-1546 (2005).

[4] B.W. Kernighan and D.M. Ritchie, *The C Programming Language*, Prentice Hall (1988).

[5] V. Sklyarov, “FPGA-based implementation of recursive algorithms”, *Microprocessors and Microsystems, Special Issue on FPGAs: Applications and Designs*, 28(5-6), pp 197-211 (2004).

[6] S. Baranov, *Logic Synthesis for Control Automata*, Kluwer Academic Publishers (1994).

[7] V. Sklyarov, “Hardware/software modeling of FPGA-based systems”, *Parallel Algorithms and Applications*, 17(1), pp 19-39 (2002).

[8] V. Sklyarov, “Models, methods and tools for synthesis and FPGA-based implementation of advanced control systems”, *Proceedings 2nd International Conference on Mechatronics*, Kuala Lumpur, pp 1122-1129 (2005).

[9] V. Sklyarov, A. Ferrari, “Design and implementation of control circuits based on dynamically reconfigurable FPGA”, *Proceedings IEEE International Conference on Electronics, Circuits and Systems*, Lisbon, Portugal, pp 527-530 (1998).

[10] A. Zakrevskij, *Parallel Algorithms of Logical Control*, Academy of Science, Minsk (1999).

[11] A. Zakrevskij and V. Sklyarov, “The specification and design of parallel logical control devices”, *Proceedings PDPTA 2000*, Las Vegas, pp 1635-1641 (2000).

[12] V. Sklyarov, “Graphical description and hardware implementation of parallel control algorithms”, *Proceedings PDPTA'99*, Las Vegas, pp 1390-1396 (1999).

[13] Trenz Electronic, “Spartan-IIE development platform”, www.trenz-electronic.de, visited on 7/10/2006.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Development of a Companion Robot “SELF” with which Humans are not Bored

Yoshikazu Mori[1] and Naoyuki Kubota[2]
[1]Department of Intelligent Systems Engineering,
Ibaraki University, Hitachi, Ibaraki, Japan
[2]Faculty of System Design,
Tokyo Metropolitan University, Tokyo, Japan
mori-zen@mx.ibaraki.ac.jp, kubota@comp.metro-u.ac.jp

Abstract

In this paper, we discuss how to generate behaviours of a companion robot “SELF” with which humans are not bored. In the simulation, a red ball and a blue ball are displayed. A human moves the red ball with a mouse and another human or a PC moves the blue ball. By using this model, first, we classify the action patterns of the human into four groups under natural interhuman interaction. Next we examine the impressions that the actions of a robot give to a human using SD method. And then we analyse how the impressions of a human change with the differences of the information transfer efficiency. Lastly we design a behaviour generator for SELF by using a statistical model, and examine the impression of the behaviours of SELF.

Keywords: entertainment robotics, interactive robot, companion, boredom

1 Introduction

Robot entertainment and healing are new fields for the practical application of autonomous robots. For autonomous robots used for surgery or dangerous operations, the accuracy and reliability are critically important. Therefore, the technological hurdles are higher. In contrast, more imperfections are forgivable in the entertainment and healing robots. However, due to lack of progressive features that will keep users entertained and involved, conventional robot pets are destined to lose popularity [1]. To give people a feeling of attachment to robot pets, Shibata has developed robot pets appealing enough to make users want to touch them [2]. Fujita, the creator of the pet robot AIBO (SONY Corp.), regards a wider variety of actions and a larger random selection of actions as the essential element to stimulate human interest [3]. However, those are still insufficient and various key elements remain missing.

In this paper, as a first step in developing an appealing robot pet, we multilaterally and quantitatively analysed boredom through experiments [4]. We consider it possible to provide a mechanical system with ongoing appeal if interactions between the user and the mechanical system are analysed in greater depth. We hope to find clues to a behaviour algorithm that will enable a robot to coexist with people. The key point for not boring a human easily is the interactive ability of the robot. We would like to produce a robot that has a thirst for knowledge, and it autonomously learns various things through interactions with its surroundings including humans. In order to realise the autonomous actions, we use a statistical model [5]. We named our robot “SELF” to represent the individualistic nature of the word self.

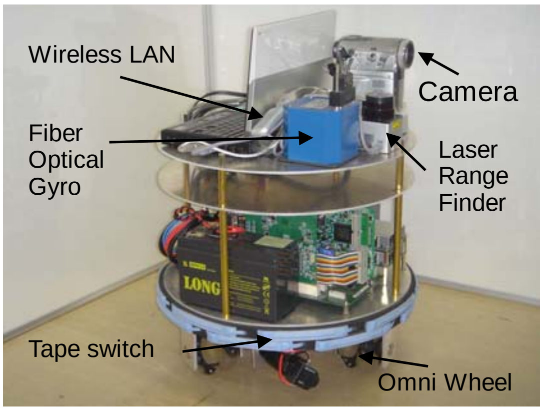


Figure 1: SELF.

The hardware of SELF is shown in figure 1. We used ART-Linux as an operating system for reasons of stability. PC was libretto U100 (CPU 1.10GHz, RAM 1GB). SELF is able to execute the same action in the simulation because SELF can move omni-directionally by using Omni-wheels, As the sensor system, a video camera, a laser range finder, sixteen touch sensor tapes, an optical gyroscope, and an encoder for every wheel were used. The interface board was a Ritech Interface Board (IF-0145-1, ZUCO Co., Ltd.). The command was transmitted via wireless LAN.

Our target is to inspire SELF.

2 Interhuman Interaction

We analysed interhuman interaction for producing the behaviours of SELF. The simulation model consisted of two PCs connected through a network. It was used to analyse human actions by converting interhuman

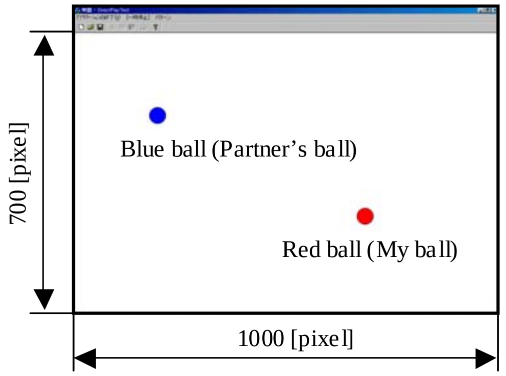


Figure 2: Simulation screen.

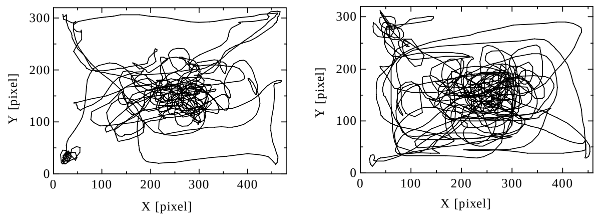


Figure 3: Each path of two balls.

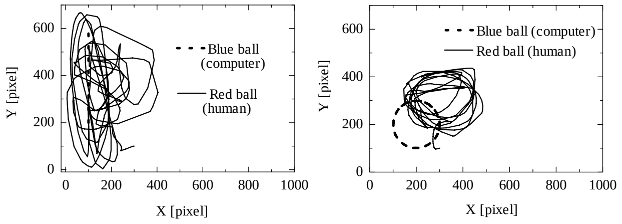


Figure 4: Experiment on entrainment.

Table 1: Patterns of the motions and actions.

1	linear motion	6	slightly quick following
2	circular motion	7	considerably quick following
3	random motion	8	slightly slow fleeing
4	considerably slow following	9	slightly quick fleeing
5	slightly slow following	10	considerably quick fleeing

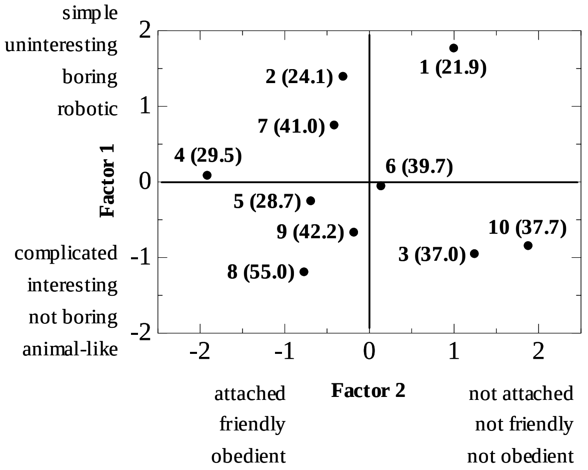


Figure 5: Experimental result of SD technique.

interaction into numerical values. Each PC displayed a red ball and a blue ball (see figure 2). The size of the screen was 700 pixels high and 1000 pixels long. The diameter of the ball was 40 pixels. A subject was allowed to move the red ball on one PC freely with a mouse. The action of the red ball was displayed as that of the blue ball on the screen of the other PC through the network.

In a series of experiments conducted with seven pairs of male college students in their twenties, each was informed in advance that the action of the red ball on one screen was displayed as that of the blue ball on the other PC. The subjects were then allowed to continue interaction between the red and blue balls as much as was desirable. Figure 3 shows paths of the two balls by one pair of subjects in interactive communication. The path of one ball resembles that of the other ball, as is the case for the other six pairs of subjects. People thus tend to conduct actions influenced by the actions of the others. Such unconscious actions as above are known as "entrainment," which is, for example, typified by interaction seen between mothers and babies [6]. In order to confirm the entrainment phenomenon clearly, the response of each subject was examined when the blue ball was moved linearly or circularly by a computer. Figure 4 shows each path of the two balls. These clearly show that people's actions were influenced by the actions of others.

Through the experiments, it was clear that both test subjects generated common behaviours such as "fleeing action," "following action," "no interferential action" and "no action" in unknown timing while causing an entrainment phenomenon.

3 Impression of Robot Action

In this section, we analyse how a human is impressed by a robot. A programme for the blue ball in simulation gave ten types of action patterns as shown in table 1. The impression of each action was examined by SD method. The number of the adjective pairs was nine and they were as follows: simple / complicated, interesting / uninteresting, boring / not boring, animal-like / robotic, attached / not attached, friendly / unfriendly, fearful / not fearful, easy / strict, and obedient / disobedient.

Each subject was required to interact with the blue ball on the screen until he got bored with it, and then the subject was requested to respond to the impression of the action on a 7-point scale. The action patterns were repeated from 1 to 10. The subjects participating in the experiment were 30 males and females from in their teens to thirties. Figure 5 is the experimental result of SD method and shows the actions of the robot and the impression given to the human based on the actions. Figures in parentheses beside actions

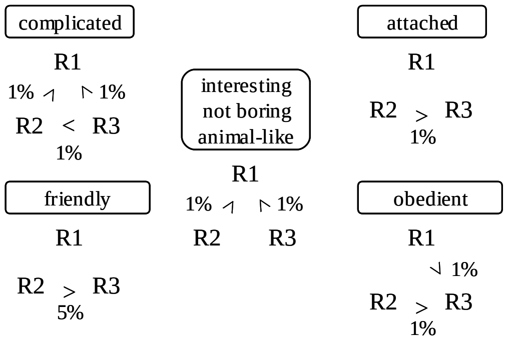


Figure 6: Impression for each reaction mode.

represent the average interaction time. Results of the experiment are as follows:

(1) If the human cannot interact with a robot, the impression of the robot is not preferable.

(2) Following and fleeing actions at moderate speed create the agreeable impression that the robot is interesting, animal-like, and attached. In this case, the interaction time is long.

(3) Actions to be conducted in the same manner as those in (2) but at an extremely high speed create the disagreeable impression that the robot is simple, uninteresting and boring.

4 Effect of Information Transfer Efficiency

Nakata et al. has shown the method of calculating the information transfer efficiency in the interaction between two persons using dyadic interaction [7]. They discuss the relationship between the information transfer efficiency and the human impression using a touch-type robot of which the appearance is a teddy bear. The evaluation items are “intention,” “interest, ” and “prettiness.” In this study, we examined the similar relationship from the viewpoint of the boredom for the same nine evaluation items as those in section 3. Information transfer efficiency, consisting of distinguishability and controllability, shows the influence on the actions of a transmitter and a receiver. Distinguishability shows the ratio that the difference in the actions of the one is reflected in the actions of the other. Controllability shows the ratio that the one can control the choice of actions of the other.

Based on the result of section 2, the reaction actions of the robot were classified into four types: “following action,” “fleeing action,” “no interferential action,” and “no action.” For the experiments in this section, the following and fleeing actions that were the least boring in section 3 are adopted. Here the no interferential action is the random motion. Three reaction modes are used for the experiments as follows: (R1) non-random 1-to-1 reaction, (R2) semirandom reaction, and (R3) full-random reaction. The information transfer efficiencies are 1, 0.5, and 0 respectively. In R1, if a subject (a red ball) approaches a robot (a blue ball), then the robot flees. If the subject flees from the robot, then the robot approaches. If the distance of the each other is over a fixed value, then the robot moves at random. If the subject is still, then the robot is also still. Therefore the subject can easily guess the reaction pattern of the robot. In R2, the reaction probability is a half. In R3 all the reactions happen at the equal probability.

Each of the ten male college students in their twenties participating in this experiment as a subject was required to interact with all reaction modes of the blue ball. They ranged from (R1) to (R3) against each of the evaluation items to examine the impression. “Simple,” for example, corresponds to the items to be selected such as “least complicated,” “secondly complicated,” and “most complicated.” Mann-Whitney U test was used for the test of statistics. Figure 6 shows the result of the test and the impression for each reaction mode. (R2) was more complicated than (R1) and the level of significance was 1 %. The results of the experiment are summarised as follows:

(1) When a robot is too obedient, people feel bored and receive a disagreeable impression of it (R1).

(2) When people cannot retain a good understanding with the robot, they receive a complicated, interesting and not boring impression. However, in this case, they do not feel the robot to be animal-like or intimate. So effect of this action is short terminable (R3).

(3) When people cannot retain a sufficiently good understanding of the robot and there is a certain amount of room for random discretion, they feel the robot to be animal-like (R2).

5 Behaviour Generator for SELF

In this section, we discuss the behaviour generator for SELF. The simulation model consists of one PC and the blue ball is operated by the PC. The situation screen is the same as that of the previous section.

We propose a behaviour generator for SELF by using statistical model that is the simplest one of dynamic Bayesian networks (DBNs). “Bayesian networks are directed acyclic graphs in which the nodes represent variables, the arcs signify the existence of direct causal influences between the linked variables, and the strengths of these influences are expressed by forward conditional probabilities [8].” Dynamic Bayesian networks are expanded in order to handle time series data [9].

Figure 7 shows the flowchart for producing behaviours of SELF that consists of four phases: (i) Observation phase, (ii) Learning phase, (iii) Reasoning phase and (iv) Introspection phase referring to Inamura’s paper [10]-[11].

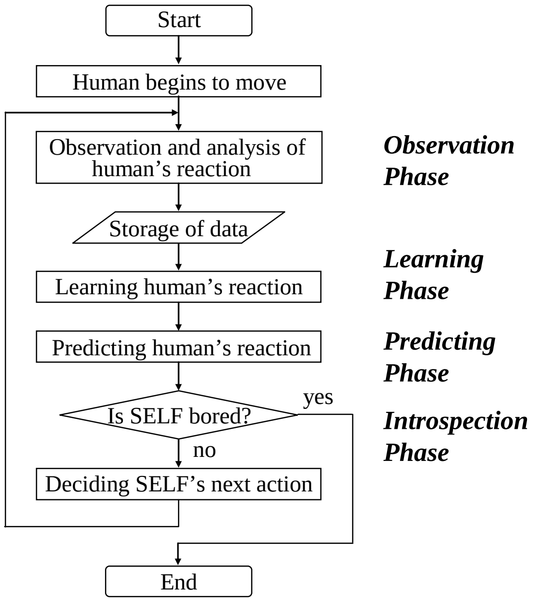


Figure 7: Flowchart.

5.1 Observation Phase

In the observation phase, SELF observes the distance and angle between it and its partner: a human. The behaviours of the partner are classified into four groups: "fleeing action," "following action," "no interferential action" and "no action," as well as in section 3.

5.2 Learning Phase

A matrix called a *conditional probabilities table* (CPT) is formed based on the network topology. It expresses the relationship between each node. In this study, the CPT is as follows:

$$CPT = \begin{pmatrix} P(p1 \mid s1) & \cdots & P(p4 \mid s1) \\ \vdots & \ddots & \vdots \\ P(p1 \mid s4) & \cdots & P(p4 \mid s4) \end{pmatrix} \qquad (1)$$

The CPT is composed of two nodes: *S*= {*s1, s2, s3, s4*} is a node that represents the actions of SELF, and *P*= {*p1, p2, p3, p4* } is a node that represents the reactions of its partner. The suffix of *s* and *p* corresponds to the behaviours respectively.

The conditional probability $P(pj \mid si)$ (*i, j* = 1,2,3,4), which represents that the partner's reaction *pj* was observed after SELF chose the action *si*, is calculated as follows:

$$P(pj \mid si) = P(P = pj \mid S = si) = \frac{P(pj, si)}{P(si)} \qquad (2)$$

and each probability is filled in the CPT correspondingly.

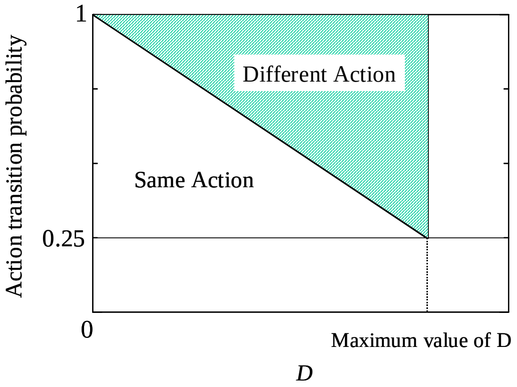


Figure 8: Relationship between difference of information quantity *D* and action transition probability.

5.3 Reasoning Phase

SELF infers what reaction its partner takes after it chose some action. In effect, if SELF chooses an action si, SELF can infer the next partner's reaction by referring to the behaviour that has the highest probability in *i* line of the CPT.

5.4 Introspection Phase

In general, humans learn the reaction patterns of some unknown object through the process of reasoning and confirmation. At first, humans feel interest if the reaction of an object is the same as what he anticipated, and he repeats the same action toward the object. After some trials, however, he feels bored and wants to experience other reactions from the object. Consequently, he will test other actions. We assume that humans feel animal-like impressions to the object if the above reaction can be realised, and such an interaction algorithm is connected to the prevention of boredom of humans. In order that SELF may produce these behaviours autonomously, we use the following variable *D* that is the difference of information quantity:

$$D = (-\log_2 Pa(pj)) - (-\log_2 Pe(pj)) \qquad (3)$$

where *Pa(pj)* is an actual probability that the partner took the reaction *pj. Pe(pj)* is an estimated probability by SELF. If those two probabilities are equal, *D* is equal to zero. In the beginning, SELF keeps the same action if *D* is small. In the course of time, however, SELF tends to change its action because of its programmed "boredom." Figure 8 shows the relationship between the value of *D* and the action transition probability.

6 Impression of the Behaviour of SELF

In this section, we analyse how humans are impressed by the behaviours of SELF. As a comparison, the arbitrary action was produced as well as SELF, but it

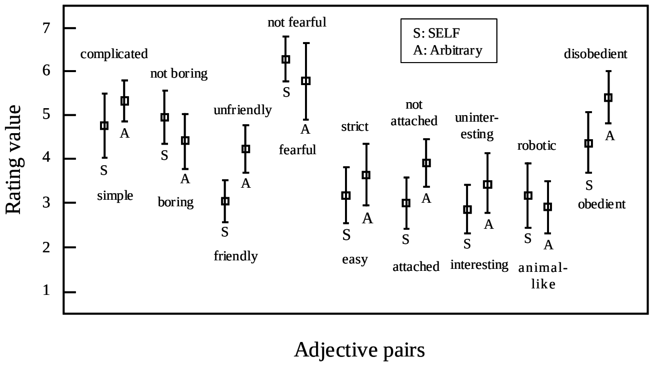


Figure 9: Impression of the behaviours of SELF.

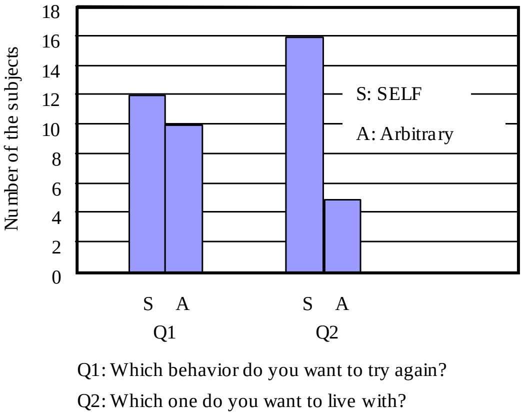


Figure 10: Results of the questionnaires.

changed its action at random every 1.0 s. The adjective pairs were the same as those in section 3.

Each subject was required to interact with the blue ball on the screen until he got bored with it, and then he was requested to respond to the impression of the behaviour on a 7-point scale. The subjects participating in the experiment were 22 males and females from in their twenties to fifties.

Figure 9 is the experimental result of the Mann-Whitney U test. SELF was more “friendly” at the significant level 1%, and was more “attached” and “obedient” at the significant level 5%. SELF produced preferable impression such as “interesting” and “not boring” although there is no difference within the significant level 5%. Arbitrary action is more “animal-like” than the behaviour of SELF at 0.3 point on average. The questionnaires said that the former and the latter looked like a cat and a dog respectively.

Figure 10 shows the results of the questionnaires. The first question, Q1, is “Which behaviour do you want to try again?” and the second question, Q2, is “Which one do you want to live with.” As for the Q1 and Q2, the behaviour of SELF was selected 12 to 10 and 16 to 5, respectively. There is a difference at the significant level 1% under the Mann-Whitney U test. According to the results of figures 9 and 10, the behaviour of SELF is preferable for the long-term boredom.

7 Conclusions and Future Work

In this study, we quantitatively analysed the interaction between a human and a robot, and clarified the following points:

(1) People naturally receive the impression that following and fleeing actions at an appropriate speed are interesting and are not boring.

(2) People receive the impression that a robot related to interaction is animal-like and that it somehow presents the substance of life, provided that the information transfer efficiency is 0.5.

We proposed a behaviour generator of a companion robot “SELF.” Compared with an arbitrary action, the behaviour of SELF was friendly, attached and obedient, and its impressions were agreeable.

In future work, we plan to apply the behaviour algorithm to the real robot.

8 Acknowledgements

This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B), 2006, 18700208.

9 References

[1] T. Sato, T. Nakata and T. Mori, “Pet robot performance direction engineering”, *Proc. 17th Annual Conference of the Robotics Society of Japan*, pp 1111-1114 (1999).

[2] T. Shibata, “Mental commit robot for healing human mind”, *Journal of the Robotics Society of Japan*, 17(7), pp 943-946 (1999).

[3] M. Fujita, “Emotional expressions of a pet-type robot”, *Journal of the Robotics Society of Japan*, 17(7), pp 947-951 (1999).

[4] Y. Mori, K. Ota and T. Nakamura, “Robot motion algorithm based on interaction with human”, *Journal of Robotics and Mechatronics*, 14(5), pp 462-470 (2002).

[5] Y. Mori, M. Ueno and N. Kubota, “Development of a companion robot ‘SELF’”, *Proc. 3rd International Symposium on Autonomous Minirobots for Research and Edutainment*, pp 351-359 (2005).

[6] H. Shimizu, T. Kume, Y. Miwa and Y. Miyake, *Ba and Co-creation*, NTT Publishing (2000).

[7] T. Nakata, T. Ko, T. Mori and T. Sato, “Informational analysis on impression of human robot interaction”, *Journal of the Robotics Society of Japan*, 19(5), pp 667-675 (2001).

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*: *Networks of Plausible Inference*, Morgan Kaufmann (1988).

[9] S. Kim, S. Imoto and S. Miyano, “Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data”, *BioSystems*, 75, pp 57-65 (2004).

[10] T. Inamura, M. Inaba and H. Inoue H, “PEXIS: probabilistic experience representation based adaptive interaction system for personal robots”, *Journal of the Institute of Electronics, Information and Communication Engineers*, J84-D-I(6), pp 867-877 (2001).

[11] T. Inamura, M. Inaba and Inoue H, “Incremental acquisition of behavior decision model based on interaction between human and robots”, *Journal of the Robotics Society of Japan*, 19(8), pp 983-990 (2001).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Spatial-temporal Expression Based on Spiking Neurons for Prediction of a Partner Robot

Kenichiro Nishida[1], Naoyuki Kubota[2,3]
[1]Department of Mechanical Engineering,
Tokyo Metropolitan University, Japan
[2]Department of System Design, Tokyo Metropolitan University
[3]SORST, Japan Science and Technology Agency
nisida-kenitirou@ed.tmu.ac.jp, kubota@comp.metro-u.ac.jp

Abstract

This paper discusses prediction-based perceptual system for communication between a partner robot and a human. The prediction is very important to extract the perceptual information for the natural communication with a human in the future. Perceptual information must be represented as spatial-temporal information. The proposed method is composed of four layers: the input layer, clustering layer, prediction layer, and perceptual module selection layer. In the input layer, the perceptual information is inputted to the spiking neurons and spiking neurons output spatial-temporal perceptual information for clustering of human behaviour patterns. The spatial-temporal perceptual information outputted by the input layer is very important for appropriate clustering. Furthermore, we show experimental results of the communication between a partner robot and a human based on our proposed method.

Keywords: partner robot, prediction, spiking neurons, human-robot interaction

1 Introduction

Recently, there has been many researches of human-robot interaction for social communication of human-friendly robots [1-4]. These robots require various capabilities for the social interaction based on verbal communication and non-verbal communication by gestures and others. Furthermore, these robots also require a perceptual system to communicate with a human. In the classical approach of environmental perception in artificial intelligence, a robot tries to extract environmental attitude and to build the environmental model [5]. When the robot interacts with a human, the robot selects the perceptual information from the built environmental model for the natural communication with the human. However, it is very difficult for the robot to extract the perceptual information to be shared with the human beforehand. Furthermore, it is computationally expensive to build an environmental model including unnecessary perceptual information for human interaction. Therefore, the robot should have the predictive capability [6] to extract the perceptual information to be shared with the human.

Most of the previous researches on the prediction have focused on the prediction for given patterns. If the patterns to be predicted are given, the robot only has to memorise and recall the patterns. However, the robot does not know what pattern has to predict in the communication with a human, because the patterns included necessary perceptual information in the communication, especially the human behaviour patterns, can change variously by the other and the situation. Consequently, the robot should extract the patterns included perceptual information to be paid attention and predict the next pattern. Therefore, we propose a prediction-based perceptual system. We apply spiking neurons for the prediction because spiking neurons can deal with the spatio-temporal context.

This paper is organised as follows. Section 2 proposes a prediction-based perceptual system. Section 3 shows experimental results of a partner robot based on the proposed method and discusses the effectiveness of our proposed method.

2 Prediction-based Perceptual System for a Partner Robot

2.1 A Partner Robot: Hubot

We developed a human-like partner robot, Hubot, in order to realise the social communication with a human (figure 1). This robot is composed of a mobile base, a body including two CPUs, two arms with grippers. The robot has various sensors such as Pan-tilt CCD camera, two line sensors (infrared sensors), microphone, ultrasonic sensors, touch sensors in order to perceive its environment. Furthermore, many encoders are equipped with the robot. One CPU is used for motion controlling, and the other CPU is used for image processing. In previous researches, we proposed a human detection method using a series of images from the CCD camera (figure 2), an interactive trajectory planning method for a hand-to-hand behaviour based on human evaluations, utterance and voice recognition for natural communication based on relevance theory, multiple value functions, map building, multi-objective behaviour coordination for reproducing acquired behaviours. These methods were proposed for the situation sharing between the robot and a human.

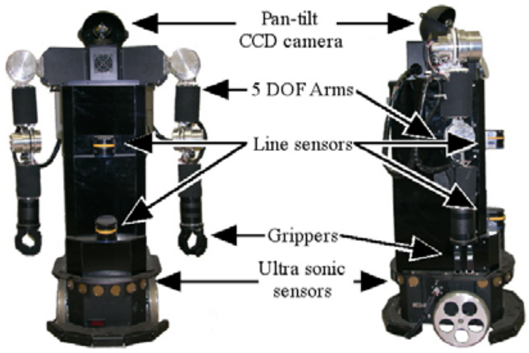


Figure 1: A partner robot: Hubot.

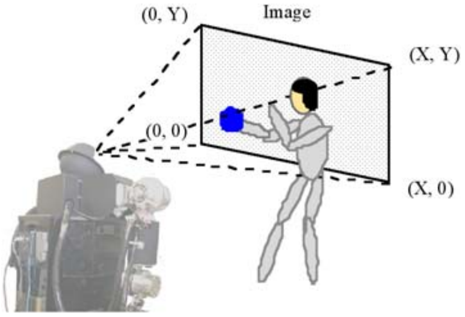


Figure 2: Visual perception of a partner robot.

2.2 Prediction-based Perceptual System

The prediction-based perceptual system for prediction of human behaviour patterns is composed of four layers: the I-layer, C-layer, P-layer, and S-layer. These layers indicate the input layer, clustering layer, prediction layer, and perceptual module selection layer, respectively (figure 3). The I-layer is composed of spiking neurons used for recognising a specific state. The S-layer is also composed of spiking neurons used for selecting the perceptual modules and control sampling intervals of the perceptual modules. Spiking neurons for the I-layer and for the S-layer are spiking neuron for perception (SN-P) and spiking neuron for selection (SN-S), respectively. The inputs to a SN-P are the outputs from a neuron of the S-layer corresponding to a perceptual module. Therefore, the time series of outputs from the SN-Ps construct the spatio-temporal pattern of the perception. Since the change of the firing patterns indicates the dynamics of perception, the robot can select perceptual modules to be used in the next perception by learning the changing patterns as a role of prediction. Therefore, the C-layer as the next layer performs unsupervised learning based on the outputs of the SN-Ps. As a result of unsupervised learning, each neuron at the C-layer acquires the boundary among the sets of perceptual information. Here a clustered perceptual state is called a perceptual mode. A perceptual mode is defined as a reference vector composed of the outputs of the SN-Ps. Each perceptual mode uses different perceptual modules to extract its perceptual information. The transition of perceptual modes indicates the dynamics of human interaction. Therefore, the robot should select perceptual modules in the next perception. Next, the robot learns the transition among perceptual modes to select perceptual modules for the next perception. The P-layer calculates the mode transition probability among perceptual modes. According to the mode transition probabilities, the S-layer selects the perceptual modules for extracting perceptual information required in the next perception.

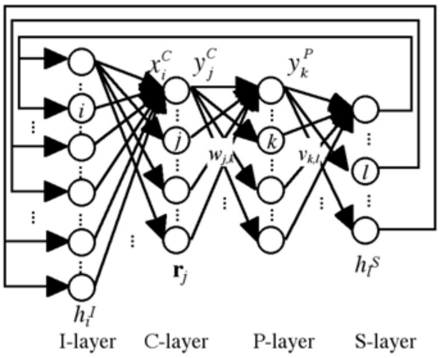


Figure 3: A prediction-based perceptual system composed of the I-layer, the C-layer, the P-layer and the S-layer.

2.3 Input Layer based on Spiking Neurons

Various types of artificial neural networks have been proposed to realise clustering, classification, nonlinear mapping, and control [7-11]. Basically, artificial neural networks are classified into pulse-coded neural networks and rate-coded neural networks from the viewpoint of abstraction level. A pulse-coded neural network approximates the dynamics introduced the ignition phenomenon of a neuron, and the propagation mechanism of the pulse between neurons. Hodgkin-Huxley model is one of the classic neuronal spiking models with four differential equations. An integrate-and-fire model with a first-order linear differential equation is known as a neuron model of a higher abstraction level. A spike response model is slightly more general than the integrate-and-fire model, because the spike response model can choose kernels arbitrarily. On the other hand, rate-coded neural networks neglect the pulse structure, and therefore are considered as neuronal models of the higher level of abstraction. A neuron model of McCulloch-Pitts and Perceptron are well known as famous rate-coded models [7].

One important feature of pulse-coded neural networks is the capability of temporal coding [10]. In fact, various types of spiking neural networks have been applied for memorising spatial and temporal context. Therefore, we apply spiking neurons for representing the time series of perceptual information. The inputs to a SN-P are the outputs from a SN-S corresponding to a perceptual module. In this paper, the robot uses four perceptual modules for differential extraction, human detection, object detection, and hand motion recognition. Each perceptual module outputs several perceptual information, e.g., certainty whether a human is detected, human face direction in human detection module. In this paper, we use a simple spike response model to reduce computational cost. First of all, the internal state of the *i*th SN-P $h_i^I(t)$ is calculated as follows;

$$h_i^I(t) = h_i^{syn}(t) + h_i^{ref}(t) \tag{1}$$

Here, $h_i^{syn}(t)$ is calculated by the following equation,

$$h_i^{syn}(t) = \gamma^{syn} \cdot h_i^I(t-1) + q_i^I(t) \tag{2}$$

where $h_i^{ref}(t)$ indicates the refractoriness of the SN-P; $q_i^I(t)$ is the input to the *i*th SN-P from the SN-S corresponding to the perceptual module; *N* is the number of neurons; γ^{syn} is a discount rate ($0< \gamma^{syn} <1.0$). For example, $q_i^I(t)$ is calculated based on certainty whether a human is detected in SN-P corresponding to human detection. When the SN-P is fired, *R* is subtracted from $h_i^{ref}(t)$ in the following,

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & if \;\; p_i^I(t-1) = 1 \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & otherwise \end{cases} \tag{3}$$

where γ^{ref} is a discount rate satisfying $0< \gamma^{ref}<1.0$; $p_i^I(t)$ is the output of the *i*th SN-P at the discrete time *t*. When the internal potential of *i*th SN-P is larger than the predefined threshold, a spike is outputted as follows;

$$p_i^I(t) = \begin{cases} 1 & if \;\; h_i^I(t) \geq \theta_i^I \\ 0 & otherwise \end{cases} \tag{4}$$

where θ_i^I is a threshold for firing. Here the spike outputs are transmitted to the next layer of the C-layer. The value received from the SN-P is gradually reduced to represent the temporal sequence of spikes.

2.4 Clustering Layer Based on Unsupervised Learning

The spatio-temporal context in the perception can be represented by the simultaneous fires and sequential fires of SN-Ps. Therefore, the C-layer performs unsupervised learning like self-organising map (SOM) [11]. Cluster analysis is used for grouping or segmenting observations into subsets or clusters based on similarity. SOM, growing neural gas, K-means algorithm, and Gaussian mixture model are often applied as clustering algorithms [11, 12]. SOM can be used as incremental learning, while K-means algorithm and Gaussian mixture model use observed all data in the learning phase (batch learning). SOM is often applied for extracting a relationship among observed data, since SOM can learn the hidden topological structure from the data. Furthermore, a growing neural gas can update the number of nodes according to a data set.

The sequence of the fired SN-Ps is used as an input for clustering by the unsupervised learning in order to extract the perception pattern of a human interaction. The inputs to the C-layer $\mathbf{X}^C=(x_1^C,x_2^C, \ldots , x_N^C)$ are the temporally-discounted spikes from SN-Ps of the I-layer. *N* is the number of inputs (the number of SN-Ps at the I-layer). The input to the C-layer from ith SN-P x_i^C is calculated as follows;

$$x_i^C(t) = 1 - \exp\left(-\left(\eta^C \cdot p_i^I(t) + u_i(t)\right)\right) \tag{5}$$

where η^C is a coefficient. When t_m is defined as time step which the *m*th spike counted backward from current time step is outputted, spikes are added to temporally-discounted term $u_i(t)$ as follows;

$$u_i(t) = \kappa^{C1} \cdot u_i(t-1) + \sum_{m=1}^{s} \frac{m}{t_m} \cdot p_i^I(t - t_m) \tag{6}$$

where κ^{C1} is discount rate. The time sequence to be considered is *T*.

To express the temporally-discounted spikes, following equation can also be used;

$$x_i^C(t) = \sum_{n=0}^{T} (\kappa^{C2})^n p_i^I(t-n) \tag{7}$$

where κ^{C2} is the discount rate for temporal transmission of spike outputs; *t* is the current time step; *T* is the time sequence to be considered. We compare these (6) and (7) in section 3.

When the *j*th reference vector is represented by $\mathbf{r}_j$, the Euclidian distance between the input vector and reference vector is defined as

$$d_j = \left\| \mathbf{x}^C - \mathbf{r}_j \right\| \tag{8}$$

where $\mathbf{r}_j=(r_{j,1},r_{j,2}, \ldots , r_{j,\mathrm{N}})$, the number of neurons (reference vectors) at the C-layer is *M*. Here two-dimensional structure is used for representing the neighbouring relationship among neurons of the C-layer. Next, the s^Cth neuron minimising the distance dj is selected by

$$s^C = \arg\min_j \left\{ \left\| \mathbf{x}^C - \mathbf{r}_j \right\| \right\} \tag{9}$$

Furthermore, the reference vector of the *j*th neuron is trained by

$$\mathbf{r}_j \leftarrow \mathbf{r}_j + \xi^C \cdot \zeta^C_{s^C,j} \cdot \left(\mathbf{x}^C - \mathbf{r}_j\right) \tag{10}$$

where ξ^C is a learning rate ($0< \xi^C <1.0$); $\zeta^C_{s^C,j}$ is a neighbourhood function ($0< \zeta^C_{s^C,j}<1.0$). In the unsupervised learning algorithm, the parameters of ξ^C and $\zeta^C_{s^C,j}$ are gradually reduced toward 0 according to the learning state.

2.5 Prediction and Perceptual Module Selection

The robot learns the transition among perceptual modes. The output from the *j*th neuron of the C-layer is used as the input to the neuron of the P-layer and calculated by,

$$y_j^C(t) = \exp(-d_j) \tag{11}$$

The output of the *k*th neuron of the P-layer is

$$y_k^P(t) = \sum_{j=1}^{M} w_{j,k} y_j^C(t) \tag{12}$$

where $w_{j,k}$ is the connection weight between the *j*th neuron of the C-layer and *k*th neuron of the P-layer. This value indicates the mode transition parameters.

When the perceptual mode transits to the other mode, the prediction difference d^P is calculated based on the gaussian membership function to evaluate the learning state. The prediction difference d^P is calculated as follows,

$$d^P = 1 - \exp\left(-\frac{\sum_{k=1}^{M} w_{s^C(t-1),k} \left(\mathbf{r}_{s^C(t)}(t) - \mathbf{r}_k(t-1) \right)^2}{2\sigma^2} \right) \tag{13}$$

where σ is the standard deviation and $w_{s^C(t\text{-}1),k}$ is the connection weight between the $s^C(t\text{-}1)$th neuron of the C-layer and *k*th neuron of the P-layer. The standard deviation σ determines the shape of the curve. We determine the value of the standard deviation σ by preliminary experiments.

The internal state of the *l*th SN-S $h^S_l(t)$ is calculated as follows;

$$h_l^S(t) = \gamma^S \cdot h_l^S(t-1) + \xi^{S1} \cdot \sum_{k=1}^{M} v_{k,l} \cdot y_k^P(t) + q^S \tag{14}$$

where γ^S is discount rate, ξ^{S1} is a learning rate; q^S is the input to a SN-S ($q^S = 1 - \xi^{S1}$). If h_j^S is higher than the threshold θ^S, its corresponding perceptual module is selected. In the beginning of learning, all of the perceptual modules is easily selected to extract various perceptual information because learning rate ξ^{S1} is small. The learning rate ξ^{S1} is increased if the prediction difference d^P is lower than the threshold θ^P in the P-layer

$$\xi^{S1} \leftarrow \xi^{S1} / \alpha^S \tag{15}$$

Otherwise, the learning rate is decreased,

$$\xi^{S1} \leftarrow \beta^{S1} \cdot \xi^S \tag{16}$$

α^{S1} and β^{S1} are the update rate for the learning rate.

2.6 Learning for Prediction and Perceptual Module Selection

The learning of connection weights between the C-layer and P-layer is performed by the Hebbian rule. Since the prediction is performed by the changing pattern of clustering state of perceptual modes, the weight parameter between the *j*th neuron of the C-layer and *k*th neuron of the P-layer is updated according to the temporal input values at *t* and *t*-1

$$w_{j,k} \leftarrow \xi^P \cdot y_j^C(t-1) y_k^C(t) + \left(1 - \xi^P\right) w_{j,k} \tag{17}$$

where ξ^P is a learning rate. The connection weights are normalised as follows,

$$w'_{j,k} = \frac{w_{j,k}}{\sum_{n=1}^{M} w_{j,n}} \tag{18}$$

Next, the $w'_{j,k}$ is substituted into $w_{j,k}$. The learning rate ξ^P is updated according to the learning state of the C-layer because the boundary among clusters is under learning if the clustering is not efficiently performed. In this way, the learning of the P-layer is performed according to the accuracy of the C-layer.

Next, we explain how to decide the connection weights between the P-layer and S-layer. The connection weights $v_{k,l}$ are easily decided according to the reference vectors as follows,

$$v_{k,l} \leftarrow \xi^{S2} \cdot \frac{\sum_{n \subseteq PM_l} r_{k^C,n}}{S_l} + (1 - \xi^{S2}) \cdot v_{k,l} \tag{19}$$

where PM_l and S_l are the set and the number of features extracted in the *l*th perceptual module, respectively.; ξ^{S2} is a learning rate ($0 < \xi^{S2} < 1.0$). The learning rate is updated according to the learning state of the P-layer.

3 Experimental Results

This section shows experimental results of the proposed perceptual system for the partner robot, Hubot. Figure 4 shows interaction between the robot and a human. We conducted several experiments of prediction in communication by using proposed method [4]. These experiments suggest that expression of perceptual information by the I-layer is very important, because learning of the C-layer is base of prediction. Therefore, we conducted simple experiment to improve the behaviour of the I-layer.

In these experiments, the robot looked at the human behaviour of writing some sentences with a red marker on a white board (figure 5). First, the human comes to the front of the robot (figures 5a, 5b), next he writes on the white board (figure 5c), finally he leaves from front of the robot (figures 5d, 5e).

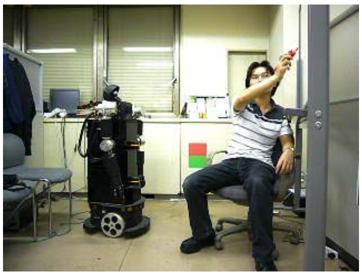


Figure 4: Interaction between the robot and a human.

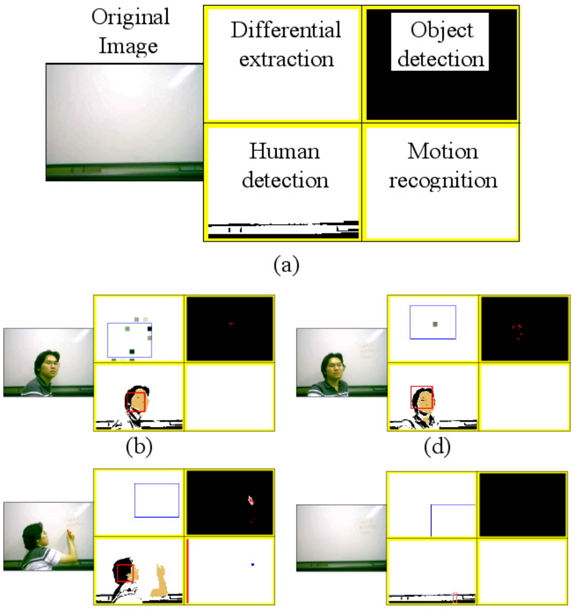


Figure 5: Snapshots of image processing conducted by the robot.

The robot looked at the human behaviour and processed images taken by CCD camera. The size of an image (*X*, *Y*) is (160, 120). Figure 5 shows several results of image processing. In these experiments, the robot uses perceptual modules of the differential extraction, human detection, object detection, and hand motion recognition modules. Each perceptual module output several perceptual information. For example, the human detection module outputs the perceptual information of whether there is a human or not, and human face direction (right or left). The object detection module outputs the perceptual information of object colour (red, green, or blue) and shape (circle, triangle, and square). We focused on the typical perceptual information of human detection, and red object detection in this experiments.

Figure 6 shows changes of the input $q_6^I(t)$, the internal state $h_6^I(t)$, and the spikes $p_6^I(t)$ of the 6th SN-P for red object detection. Each alphabet in figure 6 corresponds to the same alphabet in figure 5. We use steady-state genetic algorithm for object detection [4], and the input to the 6th SN-P is calculated based on fitness value. When the fitness value is high, the input to the SN-P is high and several spikes are outputted. But the result of the red object detection is unstable, because the red marker used in this experiment is small (figure 5c). However, the 6th SN-P integrates the input and outputs spikes as shown in figure 6-*2 and the robot can detect the red object well.

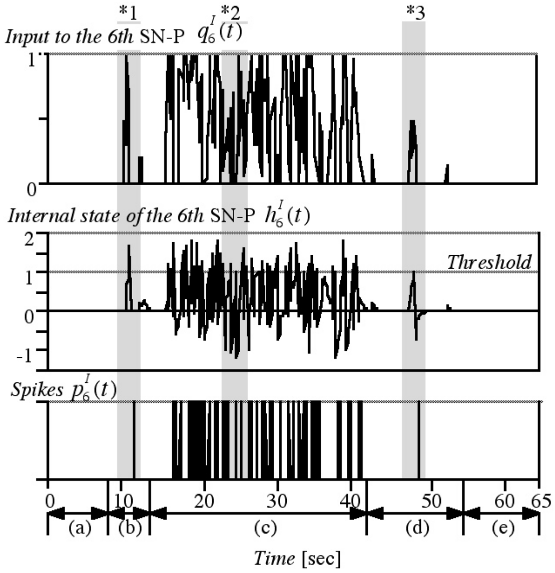


Figure 6: Change of input, internal state, and spikes of the 6th SN-P for red object detection.

Next, we conducted a experiment of comparison of (5) and (7). The human behaviours are taken as a movie for the comparison. Figure 7 shows change of the internal state of the 1st SN-P h_1^I(t) for human detection, and the input to the C-layer from the 1st SN-P $x_1^C(t)$. Figure 8 shows change of the internal state of the 6th SN-P h_6^I(t), and the input to the C-layer from the 6th SN-P $x_6^C(t)$. In figures 7 and 8, (1) and (2) show the result when (5) and (7) are used, respectively. In this experiment, the robot can detect human better than the red object. When unexpected noises are outputted as shown in figure 6-*1 and *3, (7) cannot treat these spikes as noises and discounts them as shown in figure 8-*1 and *2. On the other hand, (5) can treat unexpected spikes as noises.

The robot requires spatial-temporal perceptual information to classify and predict human behaviours. In this experiment, we show the affection of different expression of temporally-discounted spikes by using two methods. (5) is the most appropriate equation for temporal perceptual information. Several temporal perceptual information outputted from the I-layer is gathered and express spatial-temporal perceptual information.

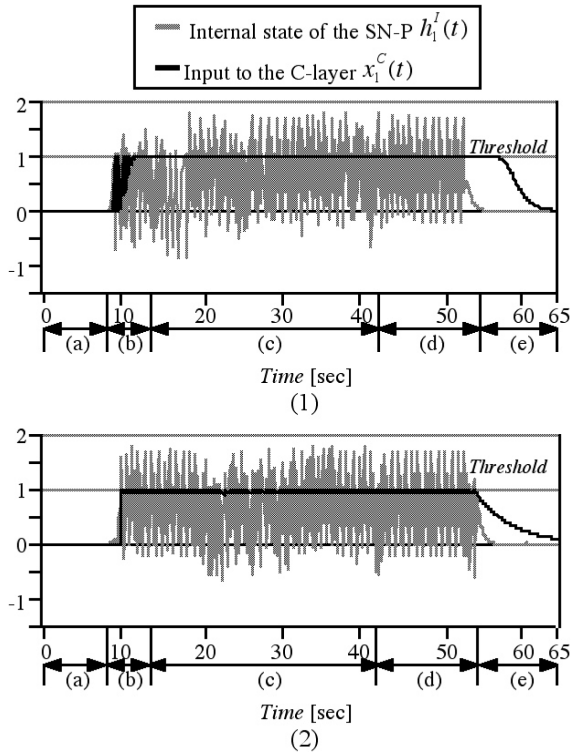


Figure 7: Change of internal state of 1st SN-P for human detection and the input to the C-layer from 1st SN-P.

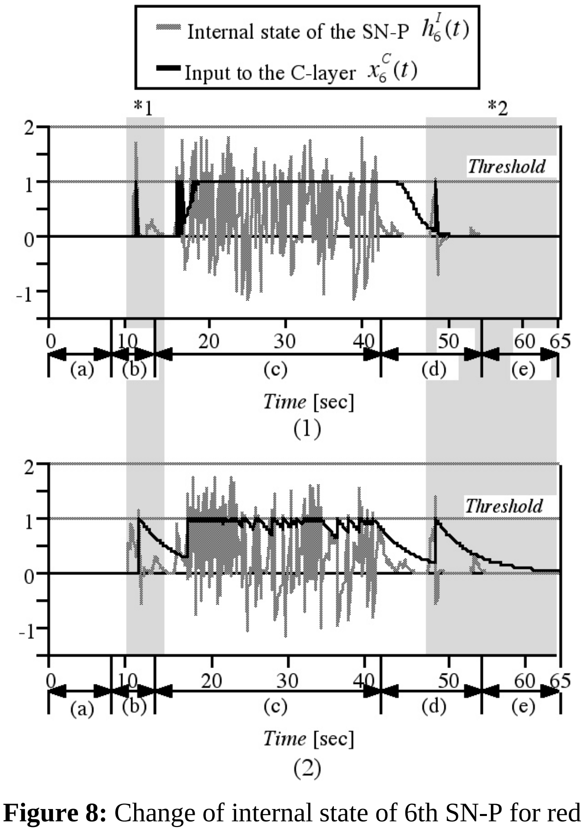


Figure 8: Change of internal state of 6th SN-P for red object detection and the input to the C-layer from 6th SN-P.

4 Conclusion

In this paper, we proposed a prediction-based perceptual system for a partner robot. In the proposed method, we use spiking neurons for extracting spatial-temporal perceptual information from time series of images for the clustering of human behaviour. The experimental results show that the proposed method can express the appropriate spatial-temporal perceptual information.

However, the proposed method has many variables to be tuning. Therefore, we intend to propose an interactive learning method between the robot and human as a future work, because the robot can not adjust the variables by itself without human reaction.

5 References

[1] S. Calinon and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot", *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai, pp 2769-2774 (2004).

[2] L. Steels, "Evolving grounded communication for robots", *Trends in Cognitive Science*, 7(7), pp 308-312 (2003).

[3] N. Kubota, "Computational intelligence for structured Learning of a partner robot based on imitation", *Information Sciences*, 171, pp 403-429 (2005).

[4] N. Kubota and K. Nishida, "The role of spiking neurons for visual perception of a partner robot", *Proceedings of 2006 IEEE World Congress on Computational Intelligence (WCCI 2006)*, Vancouver, pp 530-537 (2006).

[5] R. Pfeifer and C. Scheier, *Understanding Intelligence*, MIT Press., Cambridge, MA (1999).

[6] M.W. Eysenck, *Psychology: An Integrated Approach*, Longman, Essex (1998).

[7] J.A. Anderson and E. Rosenfeld, *Neurocomputing*, MIT Press, Cambridge, MA (1988).

[8] G.A. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network", *Computer*, 21, pp 77-88 (1988).

[9] W.T. Miller, R.S. Sutton and P.J. Werbos, *Neural Networks for Control*, MIT Press, Cambridge, MA (1990).

[10] W. Gerstner, "Spiking neurons", in W. Maass and C.M. Bishop (eds), *Pulsed Neural Networks*, MIT Press, Cambridge, MA, pp 3-53 (1999).

[11] T. Kohonen, *Self-Organizing Maps* (3rd edition), Springer-Verlag, Berlin/Heidelberg (2001).

[12] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, New York (2001).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Human Tracking and Gesture Recognition of a Partner Robot Based on Computational Intelligence

Naoyuki Kubota[1, 4], Yu Tomioka[2], Kenichiro Nishida[3]
[1]Department of System Design, [2]Department of Precision Engineering,
[3]Department of Mechanical Engineering,
Tokyo Metropolitan University, Tokyo, Japan
[4]SORST, Japan Science and Technology Agency, Japan
kubota@comp.metro-u.ac.jp, tomioka-yu@c.metro-u.ac.jp, nisida-kenitirou@ed.tmu.ac.jp

Abstract

This paper discusses how to extract human motion patterns. Typically, human motion patterns are classified into body motions and gesture motions. A body motion is considered as a trajectory of the human in an environment, while a gesture motion is considered as a movement of a physical point. Therefore, the combination of body motions and gesture motion is useful to analyse a human behaviour. These motion patterns are extracted by human tracking and gesture recognition, respectively. The proposed method for human tracking is composed of differential extraction and particle velocity estimation. The proposed method for gesture recognition is composed of a steady-state genetic algorithm for human hand detection, a spiking neural network for spatio-temporal pattern extraction of a human hand motion, and a self-organising map for gesture clustering. In this paper, we discuss the effectiveness of the proposed method through several experimental results.

Keywords: partner robot, human tracking, gesture recognition, computational intelligence

1 Introduction

Recently, various types of image processing methods for human motion analysis have been developed as computational capabilities increase [1-4]. Typically, human motion patterns are classified into body motions and gesture motions. A body motion is considered as a trajectory of the human in an environment, while a gesture motion is considered as a movement of a physical point. Therefore, the combination of body motions and gesture motion is useful to analyse a human behaviour. These motion patterns are extracted by human tracking and gesture recognition, respectively.

Various types of gesture recognition methods have been proposed so far [5-7]. The research stream can be classified into hand shape recognition and hand motion recognition. Since a hand is a complex object, it is very difficult to detect a hand from an image. Therefore, many researchers have used a simple background, coloured gloves, and others. In general, since skin colour information is used for detecting a hand, we must distinguish the hand from its similar colour of objects. Therefore, a hierarchical detection method of skin colour region extraction, hand extraction, and hand shape recognition has been used in order to reduce computational cost. Hand shape recognition is performed by using hand contour or 3D hand model. On the other hand, hidden Markov Model has been used for extracting gesture sequence as one of hand motion recognition. If knowledge database of gesture sequence patterns is available in hand motion recognition, the number of template candidates can be reduced in gesture recognition. We should use both of hand shape recognition and hand motion recognition simultaneously.

We also proposed an alternative method of hand extraction and gesture recognition based on computational intelligence [13-15]. We apply a steady-state genetic algorithm [16] for human hand detection. The dynamics of the human hand motion itself includes the meaning to be communicated. We apply a spiking neural network [19] for extracting a human hand motion and a self-organising map [21] for clustering motion patterns. The performance of gesture recognition depends mainly on the extraction capability of the spatio-temporal human hand motion pattern. Therefore, we discuss the effectiveness of the temporal coding in the spiking neural network. Finally, we show experimental results of gesture recognition for a partner robot, and discuss the effectiveness of the proposed method.

2 Human Motion Analysis

2.1 A Partner Robot

We developed a partner robot; MOBiMac as shown in figure 1. Two CPUs are used for PC and robotic behaviours. The robot has two servo motors, six ultrasonic sensors, and Pan-tilt CCD camera. Therefore, the robot can take various actions such as collision avoiding, human approaching, and line tracing. The behaviour modes used for this robot are human detection, human communication, behaviour learning, behavioural interaction. The communication with a human is performed by the utterance as the result of voice recognition and human motion recognition. The behaviour learning includes the reinforcement learning through interaction with the environment, and imitative learning through interaction with the human. The behavioural interaction includes the soccer and games with a human. In the following, we focus on image processing for gesture recognition.

The robot takes an image from the CCD camera, and extracts a human. If the robot detects the human, the robot extracts the motion of the human hand. According to the human hand motion, the robot decides the action outputs. Furthermore, the robot expresses the internal or perceptual state by utterance. A behaviour of the robot can be represented using fuzzy rules based on simplified fuzzy inference [17,18]. The logical structure written by fuzzy rules is easy for humans to understand and to design. Outputs of the robot are motor output levels. Fuzzy controller is used for collision avoidance and target tracing behaviours. The inputs to the fuzzy controller for collision avoidance and target tracing are the measured distance to the obstacle by ultrasonic sensors, and the relative direction and to a target point, respectively. Basically, a target point is generated by using the humans and objects on the image.

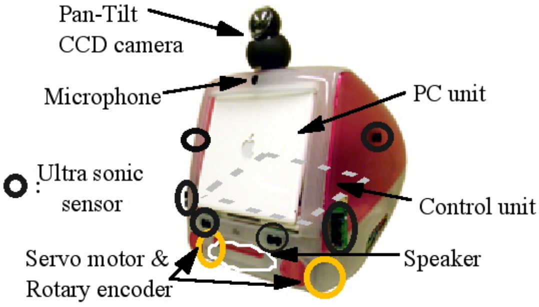


Figure 1: A partner robot; MOBiMac.

2.2 Image Processing for Human Tracking

The image processing for human tracking is composed of differential extraction and particle velocity estimation. An image is taken from the CCD camera equipped on the ceiling. The robot must detect a human face and objects from complex background speedily for natural communication. Since the image processing takes much time and computational cost, the robot detects a moving object for the fast human search. First, the robot selects pixels by the differential extraction, and *K*-means algorithm is used for the clustering of pixels to reduce the search area on the image. The *K*-means algorithm is one of most popular iterative descent clustering methods. The inputs to *K*-means algorithm are the central position of templates candidates; $\mathbf{v}_j$ (=($p_{j,1}$, $p_{j,2}$)), j=1,2, ... , n). When the reference vector of the ith cluster is represented by $\mathbf{r}_i$, the Euclidian distance between the jth input vector and the ith reference vector is defined as:

$$d_{j,i} = \left\| \mathbf{v}_j - \mathbf{r}_i \right\| \tag{1}$$

where $\mathbf{r}_i$=($r_{i,1}$,$r_{i,2}$) and the number of reference vectors (output units) is l. Next, the reference vector minimising the distance $d_{j,i}$ is selected by

$$c_j = \underset{i}{\operatorname{argmin}} \left\{ \left\| \mathbf{v}_j - \mathbf{r}_i \right\| \right\} \tag{2}$$

where c_j is the cluster number of the jth input. After selecting the nearest reference vector to each input, the ith reference vector is updated by the average of the inputs belonging to the ith cluster. This updating process is continued until all reference vectors are not changed at the clustering process. These points corresponding to reference vectors are used as the searching points in the next image for the human tracking. The average velocity of these points is used for predicting the next point on the next image. If the prediction is approximately correct, the points are maintained to the next image. Otherwise, the points are removed from the set of search points. The maximal number of search points is N_{max}. The candidate points by the differential extraction and k-means are added to the set of search points.

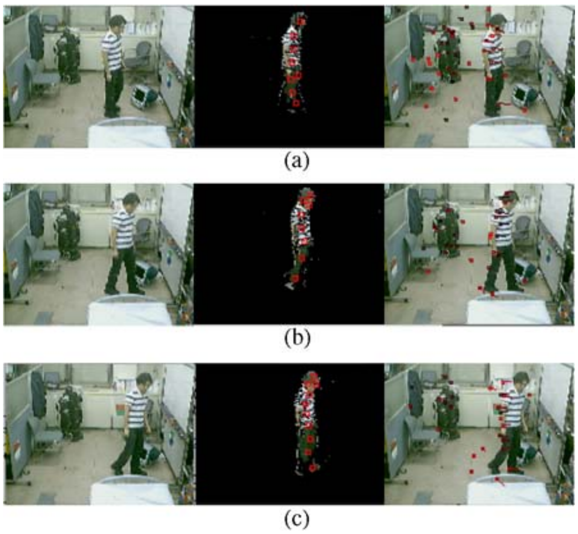


Figure 2: An experimental result of human tracking.

Figure 2 shows an experimental result of human tracking the upper left, the lower left, and the lower right show the original image, differential extraction, and the particle velocity estimation, respectively. Each particle is depicted as a small square, and the velocity of each particle is depicted as a vector from the particle. In the figures, there are many particles without the vector, but these are automatically removed from the set of search points within next two images. This result shows that the proposed method can perform the human tracking with predicting the change of the search points.

2.3 Image Processing for Gesture Recognition

The image processing for gesture recognition is composed of human hand detection, spatio-temporal pattern extraction of human hand motion, and gesture clustering (figure 3). A human has a blue ball for performing a gesture displayed to a human in order to simplify the problem. Since the image processing takes much time and computational cost, the full size of image processing to every image is not reasonable. Therefore, we use the attention area to narrow the search area for detecting a blue ball. First, the robot calculates the centre of gravity of the pixels different from the previous image as the differential extraction. The size of image used in the differential extraction is 20×15. The attention range is formed according to the centre of gravity.

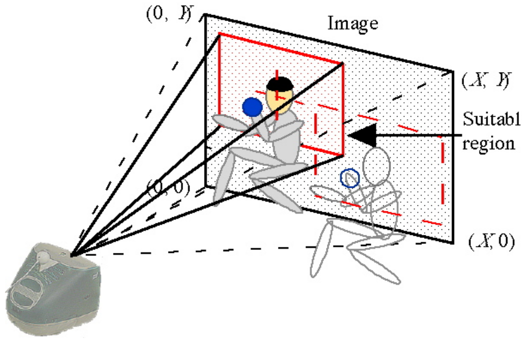


Figure 3: Image processing for gesture recognition.

To detect a blue ball, we focus on colour-based object and shape recognition with SSGA-O based on template matching. The SSGA simulates the continuous model of the generation, which eliminates and generates a few individuals in a generation (iteration). The shape of a candidate template is generated by the SSGA-O. We used an octagonal template of the angle fixed at 45º. In order to improve the shape recognition capability, we propose an octagonal template of variable angles**.** Figure 4 shows a candidate template used for detecting a target where the *j*th point $g_{i,j}^O$ of the *i*th template is represented by ($g_{i,1}^O$,+$g_{i,2j}^O$cos($g_{i,2j+1}^O$), $g_{i,2}^O$+$g_{i,2j}^O$sin($g_{i,2j+1}^O$)), *i*=1, 2, ... , *n*, *j*=1, ... , 2×*m*+2; O_i (=($g_{i,1}^O$, $g_{i,2}^O$)) is the centre of a candidate template on the image; *n* and *m* are the number of candidate templates and the searching points used in a template, respectively. Therefore, a candidate template is composed of numerical parameters of ($g_{i,1}^O$, $g_{i,2}^O$,... , $g_{i,2m+2}^O$). The number of individuals is *G*. In SSGA-O, only a few existing solutions are replaced by new candidate solutions generated by genetic operators in each generation. In this paper, the worst candidate solution is eliminated ("Delete least fitness" selection), and is replaced with the candidate solution generated by the crossover and the mutation. We use elitist crossover and adaptive mutation. The elitist crossover randomly selects one individual and generates an individual by combining genetic information from the randomly selected individual and the best individual. Next, the adaptive mutation is performed to the generated individual. Fitness value is calculated as follows.

$$f_i^O = C_{\text{Target}}^{\text{O}} + \eta_1^O \cdot C_{\text{Other}}^{\text{O}} \tag{3}$$

where η_1^O is a coefficient for penalty; C_{Target}^O and C_{Other}^O indicate the numbers of pixels of the colours corresponding to a target and other colours included in the template, respectively. The target colour is selected according to the pixel colour occupied mostly in the template candidate. Therefore, the largest area of a single colour is extracted on the reduced colour space of the image. One iteration is composed of selection, crossover, and mutation. The iteration of SSGA-O is repeated until the termination condition is satisfied. In general, the maximum of evaluations or generations is used as a termination condition.

The robot extracts the motion of the human hand from images using SSGA-O where the maximal number of images is *T*. The sequence of the hand position is represented by **G**(*t*)=(G_x(*t*), G_y(*t*)) where *t*=1, 2, ... , *T*.

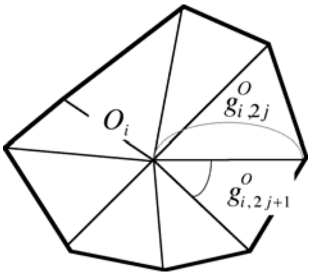


Figure 4: A template used for object recognition in SSGA-O.

The sequence of human hand positions is used for recognising spatio-temporal pattern as a human gesture by spiking neural network (SNN). SNN is often called a pulsed neural network, and is one of the artificial NNs imitating the dynamics introduced the ignition phenomenon of a cell, and the propagation mechanism of the pulse between cells [21]. In this paper, we use a spike response model to reduce the computational cost. First of all, the internal state *hi*(*t*) is calculated as follows;

$$h_i(t) = \tanh\left(h_i^{syn}(t) + h_i^{ext}(t) + h_i^{ref}(t)\right) \tag{4}$$

Here hyperbolic tangent is used to avoid burst firing, $h_i^{ext}(t)$ is the input to the *i*th neuron from the external environment and $h_i^{syn}(t)$ including the output pulses from other neurons. Furthermore, $h_i^{ref}(t)$ indicates the refractoriness factor of the neuron. When the internal potential of the *i*th neuron is larger than the predefined threshold, a pulse $p_j(t)$ is outputted as follows;

$$p_i(t) = \begin{cases} 1 & if\ h_i^{ref}(t) \geq q_i \\ 0 & otherwise \end{cases} \tag{5}$$

where q_i is a threshold for firing. Here spiking neurons are arranged on a planar grid (figure 5) and the number of neurons N=25. By using the value of a human hand position, the input to the ith neuron is calculated by a Gaussian membership function as follows;

$$h_i^{ext}(t) = \exp\left(-\frac{\left\|\mathbf{c}_i - \mathbf{G}(t)\right\|^2}{\sigma^2}\right) \tag{6}$$

where $\mathbf{c}_i$=($c_{x,i}$, $c_{y,i}$) is the position of the ith spiking neuron; s is a standard deviation. The sequence of pulse outputs $p_i(t)$ is obtained by using the human hand positions $\mathbf{G}(t)$. The weight parameters are trained based on the temporal Hebbian learning algorithm as follows,

$$w_{j,i} \leftarrow \tanh\left(\gamma^{wgt} \cdot w_{j,i} + \xi^{wgt} \cdot p_i(t) \cdot p_j(t-1)\right) \tag{7}$$

where γ^{wht} is a discount rate and ξ^{wgt} is a learning rate. Because the adjacent neurons along the trajectory of the human hand position are easily fired as a result of the temporal Hebbian learning, the SNN can memorise the temporally firing patterns of various gestures.

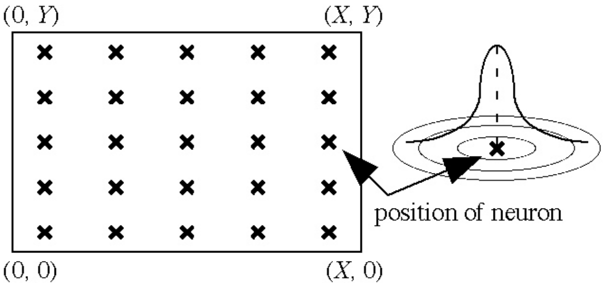


Figure 5: Spiking neurons arranged on the image.

By using the value of a human hand position, SNN generates time series of spike outputs. Because the weight of the adjacent neurons along the trajectory of the human hand position is trained by the internal state, the path value can memorise patterns of various gestures. Next, the path values concerning the weights of spiking neurons are as an input for clustering by a self-organising map (SOM) [21] in order to extract a spatial pattern of a human gesture.

3 Experimental Results

This section shows several experimental results of gesture recognition using the partner robot MOBiMac. The size (X,Y) of an image is (160, 120). The numbers of individuals and iterations in SSGA-O are 150 and 100. The number of reference vectors in SOM is 10. We use the gesture patterns listed in figure 6.

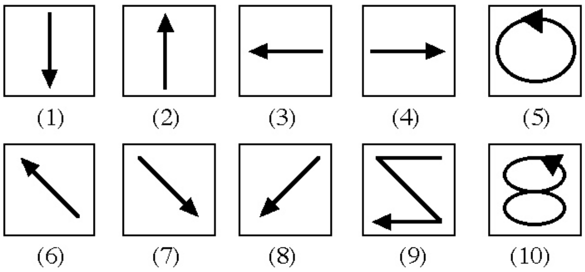
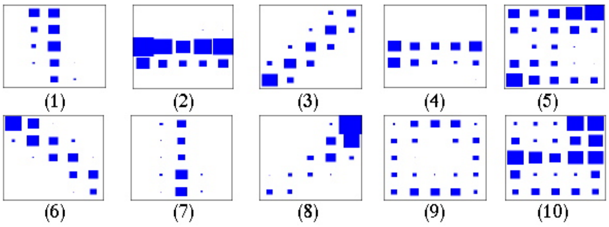


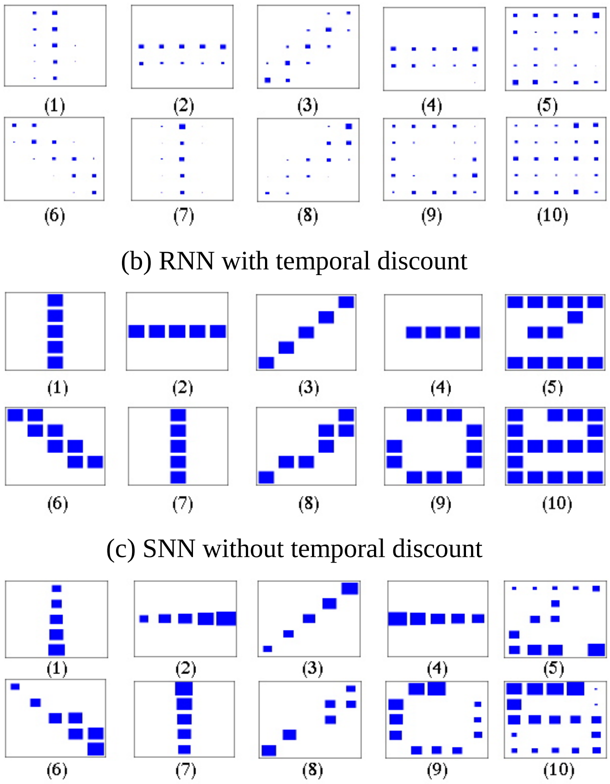
Figure 6: Gesture patterns used in experiments.

Next, we show comparison results of performance between spiking neurons and non-spiking neurons. Here we use the internal state instead of spike outputs to represent the non-spiking neuron. Therefore, the behaviour of the NN without spike outputs is similar to a recurrent (RNN). Furthermore, we discuss the effectiveness of temporal discount for spike outputs used in SOM.

Figure 7 shows the comparison results of (a) RNN without temporal discount, (b) RNN with temporal discount, (c) SNN without temporal discount, (d) SNN with temporal discount. Temporally discounted spike outputs (blue square) are transmitted to nodes of SOM. In order to use the same condition for the comparison, we use the recorded movie. Because the internal state is gradually decreasing, the behaviour of RNN without temporal discount is similar to SNN with temporal discount (figures 7a and 7d). However, the internal state of adjacent neurons in figure 7a is also increasing, the gesture pattern to be recognised is easily confused with similar gesture patterns (e.g. (5) and (10) in figure 7a). The value of the internal state in RNN with temporal discount easily becomes small. In this way, RNN based on the internal state does not need the temporally discounting factor, but it seems difficult to extract temporal and spatial features simultaneously. It is clear that that spike outputs of SNN without temporal discount extracts the spatial pattern but the temporal pattern is not extracted (figure 7c), while SNN well extracts both of spatial and temporal patterns simultaneously (figure 7d). Actually, the recognition rate of SOM by using case (a), (b), (c), and (d) in the learning stage are 0.7, 0.6, 0.7, and 0.9, respectively.



(a) RNN without temporal discount



(b) RNN with temporal discount

(c) SNN without temporal discount

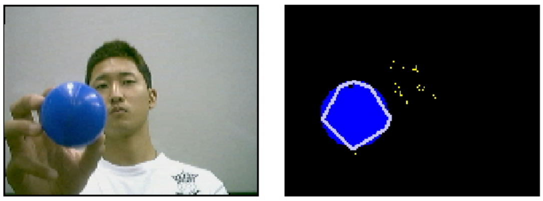
Figure 7: The comparison results of gesture recognition.

Next, we show the comparison result of SOM by using different learning rate (table 1). In this result, the learning rate 0.2 is enough, because the number of the teaching data is relatively small in the experiment. In this way, these preliminary experimental results support that the proposed method is suitable to gesture recognition.

Table 1: Comparison of the learning results of SOM.

Learning Rate	Teaching Data	Testing Data
0.4	0.7	0.65
0.3	0.85	0.7
0.2	0.9	0.8
0.1	0.8	0.75

Next, we show an experimental result of the interaction with a person. Here the person uses a blue ball for the interaction with the robot. Figure 8 shows the detection result of the blue ball from an original image. This result shows the blue ball is detected from the original image well. Figures 9 and 10 show a result of gesture recognition different from the results in figure 6. At the initial stage of the learning, SNN was unable to extract the blue ball (figure 10a), but SNN gradually detects the sequence of the blue ball motion (figure 10b).



(a) An original image (b) Its extraction result

Figure 8: Blue ball detection by SSGA.

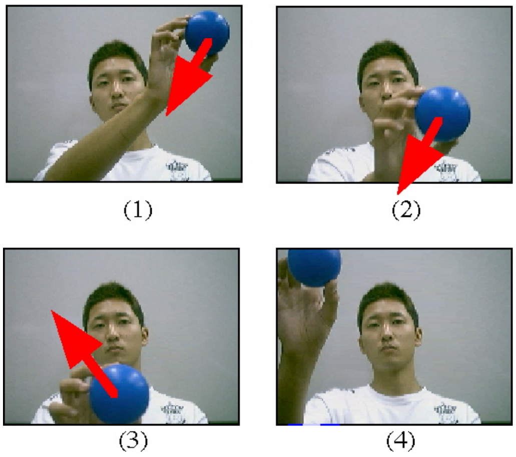
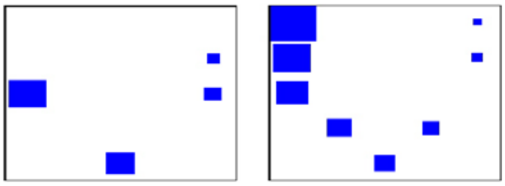


Figure 9: A typical pattern of a V-figure motion.



(a) (b)

Figure 10: The change of learning by SNN, (a) at the initial stage, (b) at the final stage

4 Summary

This paper discusses the effectiveness of the temporal coding in a spiking neural network for gesture recognition. We conducted several experiments and showed the comparison result among the spiking neural network and its related methods. The experimental results show the effectiveness of the proposed method. However, we should discuss the performance of the self-organising map for gesture clustering in detail. The number of nodes in the self-organising map should be changeable according to the learning state of the spiking neural network, because the spike output patterns of the spiking neural network generates the dynamics of the input to the self-organising map. If the learning of the spiking neural network is performed well, the spike output patterns will be clearly ordered. In this situation, the self-organising map can perform the clustering very well. In this way, we can incorporate the interdependency among the learning modules, and we proposed the concept of structured learning.

As one of future works, we discuss the structured learning for gesture recognition in detail. Furthermore we will develop the method for extracting the meanings of gestures through interaction with a human from the viewpoint of communication.

5 References

[1] D. Marr, *Vision*, W.H. Freeman, San Francisco (1982).

[2] K. Fukushima, "Neural network model for selective attention in visual pattern recognition and associative recall", *Applied Optics*, 26, pp 4985-4992 (1987).

[3] P. Dayan, S. Kakade and P.R. Montague, "Learning and selective attention", *Nature Neuroscience*, 3, pp 1218-1223 (2000).

[4] H.H. Bulthoff, S.W. Lee, T.A. Poggio and C. Wallraven, *Biologically Motivated Computer Vision*, Springer-Verlag (2002).

[5] Y. Hamada, N. Shimada and Y. Shirai, "Hand shape estimation using image transition network", *Proceedings of Workshop on Human Motion (Humo 2000)*, Texas, pp 161-166 (2000).

[6] E. Ong and R. Bowden, "A boosted classifier tree for hand shape detection", *Proceedings 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004)*, pp 889-894 (2004).

[7] V. Athitsos and S. Sclaroff, "An appearance-based framework for 3D hand shape classification and camera viewpoint estimation", *Proceedings 5th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2002)*, p 45 (2002).

[8] R.P.N. Rao and A.N. Meltzoff, "Imitation leaning in infants and robots: towards probabilistic computational models", *Proceedings Artificial Intelligence and Simulation of Behaviors (AISB 2003)*, Aberystwyth (2003).

[9] S. Calinon and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai, pp 2769-2774 (2004).

[10] A. Maurer, and A. Billard, "Extended Hopfield network for sequence learning: application to gesture recognition", *Proceedings of International Conference on Artificial Neural Networks (ICANN'05)*, Warsaw, pp 493-498 (2005).

[11] S. Schaal, A. Ijspeert and A. Billard, "Computational approaches to motor learning by imitation", *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, 358, pp 537-547 (2003).

[12] K. Morikawa, S. Agarwal, C. Elkan and G. Cottrell, "A taxonomy of computational and social learning", *Proceedings of Workshop on Developmental Embodied Cognition (DECO 2001)*, pp 46-50 (2001).

[13] N. Kubota, "Computational intelligence for structured learning of a partner robot based on imitation", *Information Sciences*, 171, pp 403-429 (2005).

[14] N. Kubota and M. Abe, "Interactive learning for a partner robot based on cyclic gestures", *Proceedings 5th International Conference on Simulated Evolution and Learning (SEAL'04)*, Busan, SWA-1 (2004).

[15] N. Kubota and K. Nishida, "Fuzzy computing for communication of a partner robot based on imitation", *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, pp 4391-4396 (2005).

[16] G. Syswerda, "A study of reproduction in generational and steady-state genetic algorithms", in *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA (1991).

[17] J.S.R. Jang, C.T. Sun and E. Mizutani, *Neuro-fuzzy and Soft Computing*, Prentice-Hall, New Jersey (1997).

[18] T. Fukuda and N. Kubota, "An intelligent robotic system based on a fuzzy approach", *Proceedings of IEEE*, 87(9), pp 1448-1470 (1999).

[19] W. Gerstner, "Spiking neurons", in W. Maass and C.M. Bishop (eds), *Pulsed Neural Networks*, MIT Press, Cambridge, MA, pp 3-53 (1999).

[20] J.A. Anderson and E. Rosenfeld, *Neurocomputing*, MIT Press, Cambridge, MA (1988).

[21] T. Kohonen, *Self-Organization and Associative Memory*, Springer, New York (1984).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Particle Swarm Procedure for the Capacitated Open Pit Mining Problem

Jacques A. Ferland,[1] Jorge Amaya,[2] Melody Suzy Djuimo[1]
[1]Département d'informatique et de recherche opérationnelle,
Université de Montréal, Montréal, Canada
[2]Centro de Modelamiento Matemático,
Universidad de Chile, Santiago, Chile
ferland@iro.umontreal.ca, djuimoys@iro.umontreal.ca, jamaya@dim.uchile.cl

Abstract

In the capacitated open pit mining problem, we consider the sequential extraction of blocks in order to maximise the total discounted profit under an extraction capacity during each period of the horizon. We propose a formulation closely related to the resource-constrained project scheduling problem (RCPSP) where the genotype representation of the solution is based on a priority value encoding. We use a GRASP procedure to generate an initial population (swarm) evolving according to a particle swarm procedure to search the feasible domain of the representations. Numerical results are introduced to analyse the impact of the different parameters of the procedures.

Keywords: Open pit mining, GRASP, particle swarm, evolutionary process, RCPSP, priority encoding

1 Introduction

Consider the problem where a mining industry is analysing the profit of extracting the ore contained in some site. In fact, the analysis includes two stages where two different problems have to be solved. The first problem is to determine the maximal open pit corresponding to the maximal gain that the mining industry can get from extraction at the site. This problem can be formulated as identifying the maximal closure of an associated oriented graph. Picard proposes a very efficient procedure to solve this problem in [1].

The second problem, denoted block extraction problem, is to determine the extraction order leading to the maximum profit accounting for the discount factor and particular constraints related to the extracting operation. In general, the mining site is partitioned into blocks characterised by several numbers. One of these is the net value of the block estimated from prospect ion data. This net value is equal to the profit associated with the block corresponding to the difference between the ore content value and the cost of extracting the block. Hence this value can be negative if the ore content value is smaller than the cost of extracting it. Furthermore, the physical nature of the problem may require extracting blocks having negative net values in order to have access to valuable blocks.

In their pioneering work, Lerchs and Grossman [2] deal with this problem with an approach generating a sequence of nested pits. Then several other approaches have been proposed: heuristics [3,4], Lagrangian relaxation [3], parametric methods [5],[6],[7],[8], dynamic programming [9], mixed integer programming [3],[5],[10] and metaheuristics or artificial intelligence methods [11],[12]. In this paper, we rely on an analogy with the resource-constrained project scheduling problem (RCPSP) [13] to propose a particle swarm procedure [14] to solve a specific variant of the problem referred to as the capacitated open pit mining problem. Recall that a particle swarm procedure is an evolutionary population-based process. The swarm (population) is evolving through the feasible domain searching for an optimal solution to the optimisation problem. In the concluding remarks, we indicate how to extend it to other variants of this problem.

The paper is organised as follows. In section 2, we formulate the two problems and we summarise Picard's procedure to solve the maximal open pit problem. Our solution approach relying on an analogy with the RCPSP to deal with the scheduling block problem is summarised in section 3, and section 4 includes our particle swarm procedure. In section 5, we introduce numerical results to analyse the impact of the different parameters of the procedures. The numerical results allow verification that the quality of the solutions generated with the particle swarm procedure is better than those obtained with greedy genotype priority vectors.

2 Problem Formulations

In this section we first introduce a mathematical model for the maximal open pit problem and a method to solve it. Then the scheduling problem specifying the extraction order of the blocks is formulated. The solution approach for this problem is given in the next section.

2.1 Maximal Open Pit

To formulate the maximal open pit model used to determine the maximal gain expected from the extraction, we denote by N the set of blocks in the mining site and by b_i the net value of extracting block $i \in N$. A binary variable x_i is associated with each block i:

$$x_i = \begin{cases} 1 & \text{if block } i \text{ is extracted} \\ 0 & \text{otherwise.} \end{cases}$$

The maximal open pit corresponds to the solution x maximising the objective function $\sum_{i\in N} b_i x_i$.

The only physical constraints considered are related to the maximal pit slope leading to identify a set of predecessors B_i including the blocks of the preceding layer above i that must be removed before block i can be extracted. The problem can be formulated using an oriented graph (open pit graph) $G=(V, A)$ where $V = \{ i \in N : \text{node } i \text{ corresponds to block } i \}$ and the set of arcs $A = \{ (i,j) : j \in B_i, i \in N \}$ is used to indicate the precedence order of the block extraction. Hence the mathematical model associated with the maximal open pit problem can be summarised as follows

$$\begin{array}{llll} \text{(MOP)} & \text{Max} & \sum_{i\in N} b_i x_i & \\ & \text{Subject to} & x_j - x_i \ge 0 \quad (i,j) \in A & (1) \\ & & x_i = 0 \text{ or } 1 \quad i \in N. & (2) \end{array}$$

Now recall that the closure of an oriented graph [1] is a subset of nodes such that all successors of any of its node also belong to the set. It is easy to see that the problem (MOP) is equivalent to determine the maximal closure of the open pit graph $G=(V, A)$.

In [1], Picard shows that the problem of finding the maximal closure of $G=(V, A)$ is equivalent to determine a minimum cut $(S, \overline{S})$ of the associated Picard graph $\overline{G} = (\overline{V}, \overline{A})$, specified as follows. The set on nodes $\overline{V} = V \cup \{s, t\}$ where s and t correspond to a super source and and a super sink, respectively. The set of arcs $\overline{A} = A \cup A^+ \cup A^-$, where $A^+ = \{(s,i) : i \in V, b_i > 0\}$, $A^- = \{(i,t) : i \in V, b_i \le 0\}$. Furthermore, the capacity d_{ij} of arc $(i,j) \in \overline{A}$ is as follows:

$$d_{ij} = \begin{cases} +\infty & \text{if } (i,j) \in A \\ b_j & \text{if } i = s \text{ and } (s,j) \in A^+ \\ -b_i & \text{if } j = t \text{ and } (i,t) \in A^-. \end{cases}$$

Picard [1] shows that a maximal closure of G is the set (S – {s}). It follows that the Ford and Fulkerson labelling procedure [15] can be applied to determine the maximal flow in the Picard's graph associated with G, and then the maximal open pit is equal to N^* = (S – {s}) where $(S, \overline{S})$ is the minimal cut associated with the maximal flow.

2.2 Scheduling Block Extraction

The maximal open pit indicates the set of blocks to extract in order to maximise the total profit, but it does not give any indication of the order for extracting the blocs under operational constraints, nor does it account for the discount factor during the extraction horizon. This problem is more complex and hence more difficult to solve. In this paper, we consider the capacitated open pit mining problem where only one additional operational constraint related to the maximal quantity that can be extracted during each period of the horizon. In the concluding remarks we indicate how to extend the solution procedure to include other operational constraints.

Denote by p_i the weight of block $i \in N$, by C_t the maximal weight that can be extracted during period t, $1 \le t \le T$, and $\frac{1}{1+\alpha}$ the discount rate per period. A binary variable x_i^t is associated with each block i for each period t:

$$x_i^t = \begin{cases} 1 & \text{if block } i \text{ is extracted during period } t \\ 0 & \text{otherwise.} \end{cases}$$

Referring to the notation introduced in section 2.1, the scheduling block extraction problem can be formulated as follows:

$$\begin{array}{lll} \text{(SBE)} \quad \text{Max} & \sum_{t=1}^{T} \sum_{i\in N} \frac{b_i}{(1+\alpha)^{t-1}} x_i^t & (3) \\ \text{Subject to} & \sum_{t=1}^{T} x_i^t \le 1 \quad i \in N & (4) \\ & \sum_{l=1}^{t} x_j^l - x_i^t \ge 0 \quad (i,j) \in A, t = 1, \cdots, T & (5) \\ & \sum_{i\in N} p_i x_i^t \le C_t \quad t = 1, \cdots, T & (6) \\ & x_i^t = 0 \text{ or } 1 \quad i \in N, t = 1, \cdots, T. & (7) \end{array}$$

The objective function (3) accounts for the discount factor in evaluating the net values of the blocks when they are extracted. Constraints (4) guarantee that each block i is extracted at most once during the horizon. The extraction precedence is enforced by constraints (5) and the constraints (6) are related to the extraction capacity during each period of the horizon.

It can be shown that the set of blocks extracted in an optimal solution of (SBE) is a subset of the maximal open pit. Hence, in the model (SBE), we can replace N by $N^* = (S - \{s\})$ determined in section 2.1.

3 Solution Approach

To find a schedule for the block extraction problem, we consider the similarity between (SBE) and the well known resource constraint project scheduling problem (RCPS) [13]. The open pit mine extraction corresponds to the project, and the extraction of each block to an activity of the project. The precedence relationship between the activities (blocks) is derived from the open pit graph G = (V, A): for all $i \in N^*$

$P_i = \{\text{block } i \text{ predecessors}\} = \{ j \in N^* : (i,j) \in A \}$. The capacity constraints correspond to constraints (6) in problem (SBE). Hence, for any period t, let $E_t = \{ i \in N^* : \text{block } i \text{ is extracted in period } t \}$, and the extraction capacity constraints can be written as:

$$\sum_{i \in E_t} p_i \le C_t \quad t = 1, \cdots, T .$$

The reward associated with activity (block) i depends on the extraction period t: $\dfrac{b_i}{(1+\alpha)^{t-1}}$.

The genotype representation $PR = \lfloor pr_1, \cdots, pr_{|N|} \rfloor$ that we use is similar to the priority value encoding introduced by Hartman in [13] to deal with the RCPSP. The i^{th} element $pr_i \in [0,1]$ corresponds to the priority of scheduling block i to be extracted. Hence the priority of extracting block i increases with the value of pr_i, and these values are such that $\sum_{i=1}^{|N|} pr_i = 1$.

Starting from the genotype vector PR, a feasible phenotype solution x of (SBE) is generated using the following serial decoding scheme where the blocks are scheduled sequentially to be extracted. To initiate the first extraction period t = 1, we first remove the block among those having no predecessor (i.e., in the top layer) having the largest priority. During any period t, at any stage of the decoding scheme, the next block to be removed is one of those with the highest priority among those whose predecessors have already been extracted, such that the capacity C_t is not exceeded by its extraction. If no such block exists, then a new extraction period (t + 1) is initiated. We denote by $v(PR)$ the value of the feasible solution x of (SBE) associated with the genotype vector PR. It is worth noticing that this decoding scheme is closely related to the Serial SGS procedure used to generate phenotype schedule from genotype representation for the RCPSP.

In order to associate a priority to a block, we need to consider not only its net value b_i, but also its impact on the extraction of other blocks in future periods. One such measure proposed by Tolwinski and Underwood in [9] is the block lookahead value $\bar{b}_i$. This value is determined by referring to the spanning cone SC_i of block i in the optimal open pit N^* specified as follows:

$$SC_i = \{ j \in N^* : \text{there exists a path from } j \text{ to } i \text{ in } G \}.$$

Hence, the value $\bar{b}_i = \sum_{j \in SC_i} b_j$.

Referring to the lookahead values, several different genotype priority vectors can be generated using the following GRASP procedure [16] to determine sequentially the block priorities. At the k^{th} ($k = 1, \cdots, N^*$) iteration, a block i_k is selected as follows:

- determine the subset of the β% best (in terms of their $\bar{b}_i$ values) blocks not yet selected
- randomly select one block i_k in this subset.

Then, $pr_{i_k} = \dfrac{(|N^*| - k + 1)}{1 + 2 + \cdots + |N^*|} = \dfrac{2(|N^*| - k + 1)}{|N^*|(|N^*| + 1)}$.

Note that the percentage β% is a parameter of the procedure.

Several feasible solutions of (SBE) can be generated by decoding different genotype vectors generated with the GRASP procedure. Furthermore, other processes can be used to generate other genotype vectors. In the next section, we introduce an evolutionary process evolving in the set of genotype vectors in order to converge to an improved feasible phenotype solution of (SBE).

4 Particle Swarm Procedure

In this section, we introduce a particle swarm procedure [17] evolving through the set of genotype vectors in order to search the feasible domain of the problem (SBE). Consider a population (swarm) P of M genotype vectors (particles): $P = \{PR^1, \cdots, PR^M\}$. The M initial genotype vectors are generated using the GRASP procedure described in the preceding section. To initialise the first iteration of the particle swarm procedure, the genotype vector PR^k corresponds both to the current vector and the best achieved vector $\overline{PR}^k$ of the individual k (i.e., at the start of the first iteration, $\overline{PR}^k = PR^k$). During the iterations of the procedure, the individuals (genotype vectors) of the population are evolving and we denote by PRb the best overall genotype vector achieved so far. Hence, at the end of each iteration, $\overline{PR}^k, k = 1, \cdots, M$, and PRb are updated as follows:

$$\overline{PR}^k := \text{ArgMax}\left\{ v(PR^k), v(\overline{PR}^k) \right\} \tag{8}$$

$$PRb := \underset{1 \le k \le M}{\text{ArgMax}} \left\{ v(\overline{PR}^k) \right\}. \tag{9}$$

Now we describe a typical iteration of the particle swarm procedure. Each current vector PR^k evolves individually to a new current genotype vector according to a probabilistic process accounting for its current value PR^k, for its best achieved value $\overline{PR}^k$, and for the best overall achieved vector PRb. More specifically, with each component i of each vector k, we associate a *velocity* [17] factor vc_i^k evolving at each iteration of the procedure. Its value is initialised at 0 (i.e., vc_i^k = 0) when the procedure starts. At each iteration, it evolves as follows

$$vc_i^k := wvc_i^k + c_1 r_{1i}^k (\overline{pr}_i^k - pr_i^k) + c_2 r_{2i}^k (prb_i - pr_i^k), \qquad (10)$$

and we define

$$ppr_i^k := vc_i^k + pr_i^k \ . \qquad (11)$$

The probabilistic nature of the procedure follows from the fact that $r_{1i}^k \in [0, 1]$ and $r_{2i}^k \in [0, 1]$ are different uniform random numbers selected at each iteration. The impact of the terms is scaled by so-called acceleration coefficients c_1 and c_2. The inertia weight w was introduced in [18] to improve the convergence rate.

Using the vector $PPR^k = \left[ppr_1^k, \cdots, ppr_{|N^*|}^k \right]$, we determine the new current genotype PR^k. First, we have to verify whether all components of PPR^k are non negative. Hence, determine

$$ppr^k = \operatorname*{Min}_{1 \le i \le |N^*|} \{ppr_i^k\},$$

and if $ppr^k < 0$, then replace

$$ppr_i^k := ppr_i^k - ppr^k.$$

Thus the new current genotype vector PR^k is specified as follows:

$$pr_i^k := \frac{ppr_i^k}{\sum_{j=1}^{|N^*|} ppr_j^k}$$

where $pr_i^k \ge 0, i = 1, \cdots, |N^*|$, and $\sum_{i=1}^{|N^*|} pr_i^k = 1$.

5 Numerical Results

Numerical results are now introduced to analyse the impact of the parameter values on the efficiency of the solution approach. The different procedures are implemented in Java 1.5 and the tests are executed on an AMD Athlon(tm) 64 Processor 3200, 2.2 GHz with 2.00 Go Ram.

We are using 20 different problems randomly generated over a two dimensional grid with 20 layers and 60 blocks wide. For each problem, 10 clusters including the blocks containing ore are randomly generated (note that these clusters can overlap). The value b_i of the blocks in the clusters is selected randomly in the set {6, 8, 12, 16}. The rest of the blocks outside the clusters contain no ore and have a negative value b_i equal to -4. The optimal pit of each problem is determined using the approach described in section 2.1. The number of blocks $|N^*|$ in the optimal pit of each problem is indicated in table 1.

Table 1: Optimal pit size.

Problem	$\|N^*\|$	**Problem**	$\|N^*\|$
1	64	11	291
2	78	12	307
3	88	13	320
4	90	14	353
5	95	15	382
6	113	16	385
7	156	17	428
8	158	18	434
9	205	19	495
10	230	20	508

The weight p_i of each block i is equal to 1, and the maximal weight $C_t = 3$ for each period.

We use the first 10 problems having smaller maximal open pit to analyse the impact of the different parameters. Each problem is solved using 12 different sets of parameter values as summarised in table 2. Note that *iter* denotes the number of iterations of the particle swarm procedure.

For each set of parameter values l, each problem ρ is solved 5 times to determine:

$va_{l\rho}$ the average of the best values $v(PRb)$ achieved

$vb_{l\rho}$ the best values $v(PRb)$ achieved

$\%_{l\rho}$ the average % improvement

$$\%_{l\rho} = \frac{va_{l\rho} \text{ - } v(PRb) \text{ at first iter.}}{v(PRb) \text{ at first iter.}} \times 100$$

$it_{l\rho}$ the last iteration where an improvement of PRb occurs.

Table 2: Parameter values.

Set	***β***	***M***	***Iter***	***w***	***c*₁**	***c*₂**
1	10	50	20	0.7	1.4	1.4
2	50	50	20	0.7	1.4	1.4
3	100	50	20	0.7	1.4	1.4
4	10	25	20	0.7	1.4	1.4
5	50	25	20	0.7	1.4	1.4
6	100	25	20	0.7	1.4	1.4
7	50	50	20	0.1	1.4	1.4
8	50	50	20	0.7	0.7	1.4
9	50	50	20	0.7	1.4	0.7
10	50	25	20	0.1	1.4	1.4
11	50	25	20	0.7	0.7	1.4
12	50	25	20	0.7	1.4	0.7

Then. for each set of parameter l, we compute the average values va_l, vb_l, $\%_l$ and it_l over the 10 problems. These values are summarised in table 3 and are used as criteria for evaluating the impact of the parameter values.

Table 3: Average values of criteria.

Set	it_l	$\%_l$	va_l	vb_l
1	4.06	0.351	240.25	242.71
2	5.52	3.580	233.82	239.69
3	5.06	4.600	217.44	226.84
4	6.14	0.566	239.35	241.64
5	5.10	2.795	231.22	237.72
6	4.88	5.566	213.02	220.80
7	3.04	2.721	232.91	238.55
8	3.26	2.663	233.52	237.98
9	5.58	2.031	235.27	240.92
10	3.68	2.016	231.63	236.44
11	3.06	2.488	230.08	235.50
12	4.80	2.872	232.00	236.32

Impact of the *β%* in the GRASP procedure

Comparing rows 1, 2 and 3 of table 3, we observe that the values of va_l and vb_l decrease while the value of $\%_l$ increases as the value of β increases. The same observations apply for rows 4, 5 and 6. Referring to the definition of the GRASP procedure in section 3, it is easy to see that in general, the priority of blocks with larger lookahead values increases as β decreases. Hence, we can expect the individuals in the initial population to have better profit values inducing that we can reach better solutions, but that the percentage of improvement $\%_l$ is smaller.

Impact of the size *M* of the population

The value va_l in row 1 is larger than in row 4. The same is true if we compare rows 2 and 5, and rows 3 and 6. This indicates that the values of the solutions generated are better when the size of the population is larger. This makes sense since we generate a larger number of different solutions.

Impact of the values of the particle swarm parameters

Comparing the results in rows 2, 7, 8 and 9 and those in rows 3, 10, 11 and 12, there is no clear impact of modifying the values of the parameters w, c_1 and c_2.

Note that the values $w = 0.7$, $c_1 = 1.4$, and $c_2 = 1.4$ were selected following the authors in [19] who showed that setting the values of the parameters close to $w = 0.7298$ and $c_1 = c_2 = 1.49618$ gives acceptable results.

Note also that the number of iterations it_l used by the particle swarm procedure to reach the best solution is smaller than the number *iter* of iterations completed.

One could legitimately ask how the solutions obtained with an evolutionary process like particle swarm compare with that obtained by decoding a greedy genotype priority vector generated with a variant of the GRASP procedure where, at each iteration, the block i with the largest value $\bar{b}_i$ among those left is selected. To complete this analysis we use the last 10 problems (numbered from 11 to 20) with a larger maximal open pit to obtain the results in table 4. Using the parameter values in set 1, each problem ρ is solved 5 times to determine:

$va_{1\rho}$ the average of the best values $v(PRb)$ achieved for parameter values in set 1

$vb_{1\rho}$ the best value $v(PRb)$ achieved for parameter values in set 1

$vw_{1\rho}$ the worst value $v(PRb)$ achieved for parameter values in set 1

v_{greedy} the value obtained by decoding a greedy genotype priority vector.

This choice of the parameter values is justified by the results in table 3, where the best value for va_l is obtained with the parameter values in set 1.

The results in table 4 indicate that even the worst value $vw_{1\rho}$ is better than v_{greedy} for all problems. Furthermore, the percentage of improvement of $va_{1\rho}$ over v_{greedy} ranges from 2.32% to 52.24%. We can conclude that using particle swarm leads to a gain in the solution quality.

Table 4: Improved solutions with particle swarm.

Problem	$va_{1\rho}$	$vb_{1\rho}$	$vw_{1\rho}$	v_{greedy}
11	166.07	170.08	163.09	156.56
12	545.34	547.32	544.43	532.66
13	144.93	155.49	137.59	116.84
14	254.14	287.45	232.67	173.72
15	72.36	84.92	70.84	43.24
16	289.85	314.99	277.77	240.86
17	400.36	408.72	384.06	356.74
18	450.11	459.71	442.40	425.59
19	384.27	389.60	380.19	343.11
20	72.01	81.25	62.86	34.39

Referring to table 4, we can also observe that for each problem, the interval $[vw_{1\rho}, vb_{1\rho}]$ including the values $v(PRb)$ is quite small. Indeed, the smallest ratio $\Delta_\rho = \dfrac{vw_{1\rho}}{vb_{1\rho}}$ is equal to 0.774 for problem $\rho = 20$. This indicates that the particle swarm procedure is stable.

6 Conclusion

In this paper we consider the capacitated open pit mining problem where we have to determine the sequential extraction of blocks maximising the total discounted profit under an extraction capacity during each period of the horizon. We propose a formulation closely related to the resource-constrained project scheduling problem (RCPSP) where the genotype representation of the solution is based on a priority value encoding. We use a GRASP procedure to generate an initial population (swarm) evolving according to a particle swarm procedure to search the feasible domain of the representations. The numerical results indicate that the better the individuals in the initial population, the better the solution generated, but the smaller the benefit of using an evolutionary process like particle swarm. Also, the quality of the solution increases with the size of the population. The numerical results in our experimentation do not seem to be very sensitive to the parameter values of the particle swarm procedure. Finally, the numerical results allow verification that the quality of the solutions generated with the particle swarm procedure is better than those obtained with greedy genotype priority vectors.

Additional experimentation should be undertaken on problems of larger size closer to real world applications. Furthermore, it is easy to see that the decoding procedure of the genotype priority vectors can be adapted to account for additional constraints found in other variants of the scheduling block extraction problem.

7 References

[1] J.C. Picard, "Maximal closure of a graph and applications to combinatorial problems", *Management Science*, 22, pp 1268-1272 (1976).

[2] H. Lerchs and I.F. Grossman, "Optimum design for open pit mines", *CIM Bulletin*, 58, pp 47-54 (1965).

[3] L. Cacetta, P. Kelsey and L.M. Giannini, "Open pit mine production scheduling", *Computer Applications in the Mineral Industries International Symposium* (*3rd Regional APCOM*), Austral. Inst. Min. Metall. Publication Series, 5, pp 65-72 (1998).

[4] M. Gershon, "Heuristic approaches for mine planning and production scheduling", *International Journal of Mining and Geological Engineering*, 5, pp 1-13 (1987).

[5] K. Dagdelen and T.B. Johnson, "Optimum open pit mine production scheduling by lagrangian parameterization", *Proceedings 19th APCOM Symposium of the Society of Mining Engineers* (*AIME*), Littleton, CO, pp 127-142 (1986).

[6] D.M. François-Bongarçon and D. Guibal, "Parametization of optimal designs of an open pit beginning a new phse of research", *Transactions SME, AIME*, 274, pp 1801-1805 (1984).

[7] G. Matheron, *Le Paramétrage des Contours Optimaux*, Technical Report no. 403, Centre de Géostatistiques, Fontainebleau, France (1975).

[8] J. Whittle, *Four-X User Manual*, Whittle Programming, Melbourne (1998).

[9] B. Tolwinski and R. Underwood, "A scheduling algorithm for open pit mines", *IMA Journal of Mathematics Applied in Business and Industry*, 7, pp 247-270 (1996).

[10] M. Gershon, "Mine scheduling optimization with mixed integer programming", *Mining Engineering*, 35, pp 351-354 (1983).

[11] B. Denby and D. Schofield, "The use of genetic algorithms in underground mine scheduling", *Proceedings 25th APCOM Symposium of the Society of Mining Engineers* (*AIME*), Brisbane, pp 389-394 (1995).

[12] B. Denby, D. Schofield and S. Bradford, "Neural network applications in mining engineering", *Department of Mineral Resources Engineering Magazine,* University of Nottingham, pp 13-23 (1991).

[13] S. Hartmann, "A competitive genetic algorithm for the resource-constrained project scheduling", *Naval Research Logistics*, 456, pp 733-750 (1998).

[14] J. Kennedy and R.C. Eberhart, "Particle swarm optimization", *Proceedings IEEE International Conference on Neural Networks IV*, Perth, Australia, pp 1942-1948 (1995).

[15] L.R. Ford and D.R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, New Jersey (1962).

[16] T. Feo and M.G.C. Resende, "Greedy randomized adaptive search procedure", *Journal of Global Optimization*, 2, pp 1-27 (1995).

[17] U. Paquet and A.P. Engelbrecht, "A new particle swarm optimiser for linearly constrained optimization", *Proceedings 2003 Congress on Evolutionary Computation*, Canberra, pp 227-233 (2003).

[18] Y. Shi and R.C. Eberhart, "A modified particle swarm optimizer", *Proceedings IEEE Congress on Evolutionary Computation,* Piscataway, New Jersey, pp 69-73 (1998).

[19] R.C. Eberhart and Y. Shi, "Comparing inertia weights and construction factors in particle swarm optimization" *Proceedings of the Congress on Evolutionary Computation*, La Jolla, CA, pp 84-88 (2000).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Gain-scheduled Congestion Control for TCP/AQM Networks using State Predictive Observers

Takehito Azuma[1], Masashi Uchida[2] and Yasunori Kawai[3]
[1]Utsunomiya University, Utsunomiya, Japan
[2]HOKUSHO Co. Ltd., Shino-machi, Kanazawa, Japan
[3]Ishikawa National College of Technology, Kahoku-gun, Ishikawa, Japan
tazuma@cc.utsunomiya-u.ac.jp

Abstract

In this paper, a design problem of congestion controllers is discussed for TCP/AQM (transmission control protocol/active queue management) networks. The proposed method consists of two control techniques. First one is based on a gain-scheduling technique considering nonlinearities of TCP/AQM networks. But the designed congestion controller is a state feedback controller and it is impossible to embed the congestion controller in real computer networks. To avoid this problem, second one is based on a design technique of state predictive observer for linear time-delay systems. Thus it can be possible to embed the congestion controller in real networks by combining these methods. Firstly dynamical models of TCP/AQM are described as linear systems with self-scheduling parameters, which also depend on information delay. Here it is distinguishing to focus on constraints on the maximum queue length and TCP window-size, which are the network resources in TCP/AQM networks. And a design method of memoryless state feedback controllers is shown for linear system with a self-scheduling parameter and an information delay. But the designed gain-scheduled congestion controller is a state feedback controller and it is impossible to apply this controller to computer networks directly. Thus observers are also designed and the observer-based congestion controllers are derived.

Keywords: systems and control, computer networks, congestion controllers, observer-based control

1 Introduction

Computer networks or wireless networks have become main communication tools. Especially the high reliable exchange of data using the Internet has been important for its explosive growth and utilisation. The transmission control protocol (TCP) is well known as this exchange. Under TCP, a window flow-control mechanism is used to set its transmission rate (figure 1).

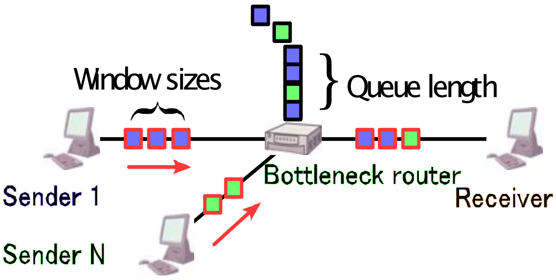


Figure 1: Congestion in computer networks.

In this mechanism, TCP increases the window size during successful data transmission. Conversely TCP cuts the window size in half whenever a data does not reach the receiver. Such data losses called “packet losses” can affect network performance. One of causes of this is that TCP has no information of network mechanisms contributing to packet loss [1][2][3]. Considering this, recently TCP/AQM (transmission control protocol /active queue management) networks having queue information of the congested node are considered in some papers [4][5][6][7]. Other papers considering congestion control are about scalable control [8] [9] [10].

Some kinds of AQM schemes are proposed, e.g. random early detection (RED) [11][12], virtual queue [13], random early marking (REM) [14], adaptive virtual queue (AVQ) [15], proportional integral controller [16], gain-scheduled controller [20][21] and robust state predictive controller [26]. Based on the control theory, it seems possible to design that congestion controllers (AQM schemes) achieve better performances than those AQM schemes do. AQM design problems are important and become useful in future researches because AQM is embedded in the router having much information about circumstances of current networks.

In this paper, a design method of congestion controllers is proposed for TCP/AQM networks. The proposed method consists of two congestion control approaches. First approach is the gain-scheduled control [20][21] based on gain scheduling techniques [17][18]. This controller is a state feedback controller but can achieve good performances and consider nonlinearities of TCP/AQM networks. Second one is based on the robust state predictive controller [26]. This approach is based on linearised systems of TCP/AQM networks but it is possible to design observers for TCP/AQM networks. In [26], the efficacy of the designed observers is shown. Thus it seems possible to design good congestion controllers by combinations of these two approaches. In the numerical example, the effectiveness of the proposed method is evaluated by using *ns-2* (Network Simulator ver.2) by comparing the result of [26] and the proposed approach.

2 Gain-scheduled Congestion Control

In this section, a design method of gain-scheduled congestion controllers [20][21] is summarised.

Firstly dynamical models of TCP/AQM networks considered in this paper are shown. The dynamical models of TCP/AQM networks consist of models of the TCP window sizes, queue lengths and AQM mechanisms. Models are described as nonlinear equations but the dynamical models of TCP/AQM networks can be derived as linear systems with self-scheduling parameters and time delays which we call as information delay.

2.1 Dynamical Models of TCP/AQM Networks

Dynamics of TCP networks:

Dynamical models of TCP were investigated by [22] [23]. The dynamical models of TCP networks are given as the following nonlinear differential equations.

$$\dot{w}_s(t) = \frac{1}{h_r(t)} - \frac{w_s(t)}{2}\frac{w_s(t-h_r(t))}{h_r(t-h_r(t))}p(t-h_r(t)) \tag{1}$$

$$\dot{q}(t) = N(t)\frac{w_s(t)}{h_r(t)} - C \tag{2}$$

where w_s is the average TCP window size, *q* is the average queue length, *C* is the queue capacity, *N* is the number of TCP sessions and *p* is probability of packets dropped at AQM. h_r is round-trip time $RTT = \frac{q}{C} + T_p (T_p\text{:}$ propagation delay).

This parameter h_r is information delay in TCP networks. We also assume that w_s and *q* are constrained as follows,

$$q \in [0, q_{\max}], \tag{3}$$

$$w_s \in [0, w_{s\max}]. \tag{4}$$

In the equation (1), the parameter *p* is very important. In case of *p=0* , (1) is equal to the first term. This dynamics denotes that TCP window size increases linearly. On the other hand, in case of *p=1*, (1) is equal to the second term. These facts mean that the TCP window size decreases largely. Thus it is important to change the TCP window size by setting *p* adequately to avoid congestions in networks but traditional AQM designs are impossible to change the TCP window size by setting *p* adequately because parameter tunings of AQM schemes, which are explained in the next subsection, are manual procedures.

AQM scheme:

Active queue management is a core process where packets are dropped depending on the queue length. Dropped packets in this way amounts to reducing TCP source rate as the queue length grows. The objective of AQM is to manage the buffer size as a mean value (the average queue length).

In equations (1) (2), the average queue length can be adjusted by the TCP window size. Thus it is proper to avoid congestions that we make the TCP window size small if the average queue length is large. It is also proper to achieve effective data transmission that we make the TCP window size large if the average queue length is small. In this paper, our objective is to design AQM scheme which has the above properties.

2.2 Representation as Linear Systems with Self-scheduling Parameters and Information Delays

Here we derive dynamical models of TCP/AQM networks as linear systems with self-scheduling parameters and information delays. To simplify the discussion, we assume that the number of TCP sessions and round-trip time (information delay) are time invariant, e.g. $N(t) = N, h_r(t) = h_r$.

If $w_s >> 1$ [16], the dynamical models of TCP/AQM networks (1) (2) can be described as follows,

$$\dot{w}_s(t) = \frac{1}{h_r} - \frac{w_s(t)^2}{2h_r}p(t-h_r), \tag{5}$$

$$\dot{q}(t) = N\frac{w_s(t)}{h_r} - C. \tag{6}$$

Introducing the equilibrium points (w_{so}, q_o, p_o) which are given by

$$p_o = \frac{2}{{w_{so}}^2}, w_{so} = \frac{h_r C}{N}, \tag{7}$$

equations (5) and (6) become

$$\delta\dot{w}_s(t) = -p_o\frac{\delta w_s(t) + 2w_{so}}{2h_r}\delta w_s(t) - \frac{(\delta w_s(t) + 2w_{so})^2}{2h_r}\delta p(t-h_r), \tag{8}$$

$$\delta\dot{q}(t) = N\frac{\delta w_s(t)}{h_r}, \tag{9}$$

where the following conditions are satisfied,

$$w_s(t) = w_{so} + \delta w_s(t),$$

$$q(t) = q_o + \delta q(t),$$

$$p(t) = p_o + \delta p(t).$$

Now the state variable and the input variable are introduced as follows,

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \delta w_s(t) \\ \delta q(t) \end{bmatrix}, u(t) = \delta q(t).$$

Then TCP/AQM networks (5) and (6) (that is (1) and (2)) can be described as the following linear system with a self-scheduling parameter and an information delay.

$$\dot{x} = A(\theta_x(x))x + B(\theta_x(x))u(t - h_r) \tag{10}$$

$$\theta_x(x) = x_1(t) = \delta w_s(t) \tag{11}$$

where

$$A(\theta_x(x)) = A_0 + A_1\theta_x(x),$$

$$A_0 = \begin{bmatrix} -p_o\frac{w_{so}}{h_r} & 0 \\ \frac{N}{h_r} & 0 \end{bmatrix}, A_1 = \begin{bmatrix} -p_o\frac{1}{2h_r} & 0 \\ 0 & 0 \end{bmatrix}, \tag{12}$$

$$B(\theta_x(x)) = B_0 + B_1\theta_x(x) + B_2\theta_x(x)^2,$$

$$B_0 = \begin{bmatrix} -\frac{1}{2h_r} \\ 0 \end{bmatrix}, B_1 = \begin{bmatrix} -\frac{w_{so}}{h_r} \\ 0 \end{bmatrix}, B_2 = \begin{bmatrix} -\frac{w_{so}{}^2}{2h_r} \\ 0 \end{bmatrix}, \tag{13}$$

and $\theta_x(x)$ is called as a self-scheduling parameter. From the above discussion, TCP/AQM networks are described as linear system with a self-scheduling parameter and an information delay (10)- (13) having constraints (3) and (4).

2.3 A Statefeedback Gain-scheduled Controller Design

Now consider the following linear time-delay system with a self-scheduling parameter (10), (11) where $x \in R^n$ is the state, $u \in R^m$ is the input and h_r is a constant time delay. $\theta_x(x)$ is a self-scheduling parameter and is equal to x_1 which denotes the first element of the state *x*. In this paper, we consider this simple case. Of course it is possible to extend more general case, but to simplify the discussion we consider only a simple case. Additionally we consider the constraint of the state variable *x* as follows,

$$x \in X_e = \{x | x' Q_x{}^{-1} x \le 1\}. \tag{14}$$

The matrix Q_x is a given matrix and denotes a constraint of the state variable *x*. Since the linear time-delay system with a self-scheduling parameter (10) is nonlinear, the constraint of the state variable *x* (14) is usual and meaningful. Thus the global problem of the linear time-delay system with a self-scheduling parameter is not discussed in this paper and only a semi-global problem is considered.

Now considering the state constraint (14), the linear time-delay system with a self-scheduling parameter can be treated as a linear parameter varying (LPV) system with a time delay as follows,

$$\dot{x} = A(\theta(t))x + B(\theta(t))u(t - h_r), \tag{15}$$

$$\theta(t) = [\theta_{\min}, \theta_{\max}]. \tag{16}$$

where $\theta(t)$ is the scheduling parameter of this LPV system with a time delay and $\theta_{\min}$, $\theta_{\max}$ are given from the constraint (14). We call this LPV system with a time delay as a linear time-delay system with a scheduling parameter.

Note that the system (10)-(13) includes the linear time-delay systems with a scheduling parameter (15) with (16). Here we define the design problem of stabilising controllers for the linear time-delay system with a self-scheduling parameter (11).

Problem:

Design the memoryless state feedback controller

$$u(t) = Kx(t) \tag{17}$$

which stabilises the linear time-delay system with a self-scheduling parameter (10) with (11) and assures that the state of the closed loop system satisfies the constraint (14).

The following theorem can solve the problem defined in the previous subsection [20].

Theorem 1: Let $\theta_{\min}$ and $\theta_{\max}$ be given parameters. If there exist matrices $X > 0$, $P_1 > 0$, $P_2 > 0$ and Y such that the following two LMIs are feasible for all $\theta \in \Theta = [\theta_{\min}, \theta_{\max}]$,

$$\begin{bmatrix} M_c & XA'(\theta) & Y'B'(\theta) & B(\theta)Y \\ A(\theta)X & -P_1 & 0 & 0 \\ B(\theta)Y & 0 & -P_2 & 0 \\ Y'B'(\theta) & 0 & 0 & -X \end{bmatrix} < 0, \tag{18}$$

$$X > P_1 + P_2, \tag{19}$$

where

$$M_c = h_c^{-1}\{XA'(\theta) + A(\theta)X + Y'B'(\theta) + B(\theta)Y\}$$

and moreover defining K, A_c and B_c as

$$K = YX^{-1},$$

$$A_c(\theta) = A(\theta) + B(\theta)K,$$

$$B_c(\theta) = B(\theta)K,$$

be using matrices of (18) and (19), if there exists $\alpha > 0$ which satisfies

$$\begin{bmatrix} A_c Q_x + Q_x A'_c(\theta) + \alpha Q_x & B_c(\theta) \\ B_c'(\theta) & -2\alpha Q_x \end{bmatrix} < 0, \qquad (20)$$

for all $\theta \in \Theta$.

Then the controller

$$u(t) = Kx(t), K = YX^{-1} \qquad (21)$$

is a stabilising controller for the system (10)-(13) and the state of the closed loop system satisfies the constraint (14).

Here a simple explanation of Theorem 1 is given. The first LMIs (18) and (19) assure that the closed-loop system (10)-(13) is asymptotically stable without considering the constraint (14). The next LMI (20) assures that the state of the closed-loop system satisfies the constraint (14).

Remark 1: Parameters $\theta_{\min}$ and $\theta_{\max}$ are design parameters and should be chosen adequately because conservativeness of the result from Theorem 1 is decided by these parameters. Usually $\theta_{\min}$ and $\theta_{\max}$ are obtained by considering the constraint (14) in case of a given matrix Q_x because the self-scheduling parameter $\theta_x(x)$ is equal to $x_1(t)$. The given matrix Q_x is an estimated matrix from a reachable set of the controlled system (10)-(13). The derived conditions are dependent on the parameter θ and it is needed to reduce finite dimensional conditions because the parameter dependent conditions are infinite and difficult to solve. But it is possible to reduce finite dimensional conditions using the technique [24][25].

3 Gain-scheduled Congestion Control using State Predictive Observers

State predictive control is applied to a design of congestion controllers for TCP/AQM networks [26] and it was shown that good performances are achieved in comparison with RED [11]. In this paper, gain-scheduled congestion controllers with state predictive observers are proposed. This approach is combination of [20][21] and [26].

3.1 Linearised Models of TCP/AQM Networks

To design state predictive observers, linearised models of nonlinear models introduced in the previous section are derived. Linearised models are given as the following linear system with an input time-delay

$$P(s) = \frac{\frac{C^2}{2N}}{(s + \frac{2N}{R_0^2 C})(s + \frac{1}{R_0})}, \qquad (22)$$

where $R_0 = q_0 / C + T_p$.

Now the state variable and the input variable are introduced as follows,

$$x(t) = \begin{bmatrix} \delta w_s(t) \\ \delta q(t) \end{bmatrix}, y(t) = \delta q(t)\ , \ u(t) = \delta p(t).$$

The state equation of the linear systems with an input time-delay is given as follows,

$$\dot{x}(t) = Ax + Bu(t - R_0), \qquad (23)$$

$$y(t) = Cx(t), \qquad (24)$$

$$A = \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}, B = \begin{bmatrix} -\frac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

3.2 State Predictive Observers

It is impossible to obtain the state of the above time-delay system directly because the controller is embedded in the bottleneck router and the TCP window size is not available on-line. The state *x(t)* is estimated by using the following observer.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t - R_0) - L_p(y(t) - C\hat{x}(t)), \qquad (25)$$

where L_p is an observer gain which is designed as $A + L_p C$ becomes stable. Considering an estimated error of the state $e(t) = x(t) - \hat{x}(t)$, the error system is given as follows,

$$\begin{aligned} \dot{e}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\ &= A(x(t) - \hat{x}(t)) + L_p C(x(t) - \hat{x}(t)) \\ &= (A + L_p C)(x(t) - \hat{x}(t)) \\ &= (A + CL_p)e(t) \end{aligned} \qquad (26)$$

and it is possible to design observers if *(C,A)* is observable.

Now it is assumed that *(C,A)* is observable and the observer gain L_p is designed as a synthesis problems of a Kalman filter in case of considering that the small change of the input time delay is a white noise. The general description of linear systems with white noises is given as follows,

$$\dot{x}(t) = Ax(t) + B_1 u(t) + B_2 \xi(t),$$
$$y(t) = Cx(t) + D_1 u(t) + D_2 \xi(t) + \eta(t),$$

where $B_1 = B$, $B_2 = \left[w_{s0} * 2 \quad q_0 / 2\right]^T$, $D_1 = 0$, $D_2 = 0$, $E\left[\xi(t)\xi(t)^T\right] = 1$ and $E\left[\eta(t)\eta(t)^T\right] = 1$ B_2 denotes the small change of the input time delay. Finally the observer gain L_p is easily designed as a synthesis problem of a Kalman filter for this system by using MATLAB.

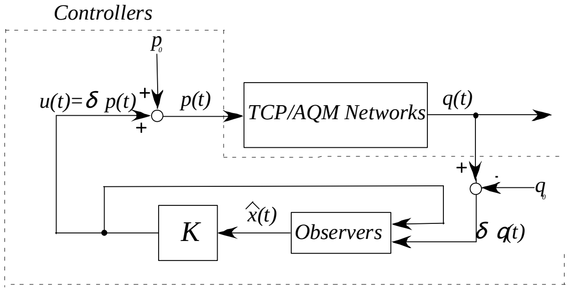


Figure 2: The closed loop system.

Remark 2: The closed loop system designed in the section 3 and 4 is shown in figure 2. The block diagram in the dotted line denotes the designed controller which is embedded as AQM. Observers are designed by using the method in this section and the feedback gain *K* is designed in the previous section.

4 A Numerical Example

The network topology is in figure 1 and parameters are given in table 1.

Table 1: Parameters of the network topology.

Parameter	Value
A number of TCP sessions *N*	8
Information delay h_r	0.56 [s]
Propagation delay T_p	14 [ms]
Link capacity *C*	73.3 [packets/s]
Maximum window size $w_{s\max}$	8 [packets]
Maximum queue size $w_{s\max}$	80 [packets]
w_{s0}	5.13 [packets]
q_0	40 [packets]
p_0	0.076

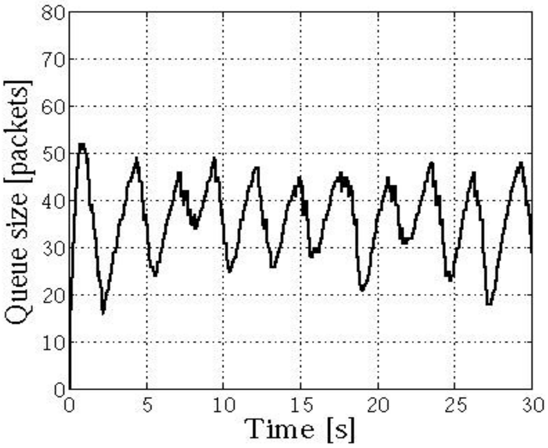


Figure 3: The queue size (proposed method).

Simulation is done by using *ns-2*. Figure 3 shows the result in case of the gain-scheduled congestion controller using the state predictive observer which is proposed in this paper. Figure 4 shows the result in [26].

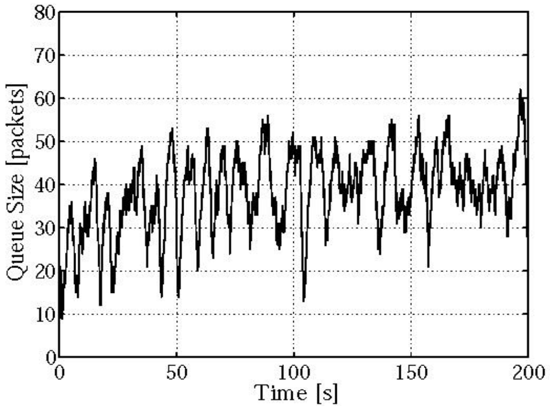


Figure 4: The queue size ([26]).

Note that time axis is different from figure 3 and figure 4. Thus it is obvious that the time response of the proposed approach is very fast. From these results, the proposed approach can achieve better performances than the approach in [26].

5 References

[1] S. Low, F. Paganini and J. Doyle, "Internet congestion control", *IEEE Control Systems Magazine*, pp 28-43 (2002).

[2] C. Jin, D. Wei, S. Low, G. Buhrmaster, J. Bunn, D. Choe, R. Cottrell, J. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot and S. Singh, "Fast TCP: from theory to experiments", *IEEE Network*, pp 4-11 (2005).

[3] S. Low D. Wei, C. Jin and S. Hegde, "Fast TCP: motivation, architecture, algorithms, performance", *IEEE/ACM Trans. on Networking*, (2007)(forthcoming).

[4] C. Hollot, V. Misra, D. Towsley and W. Gong, "On designing improved controllers for AQM

routers supporting TCP flows", *Proceedings of IEEE/INFOCOM* (2001).

[5] C. Hollot, V. Misra, D. Towsley and W. Gong, "A control theoretic analysis of red", *Proceedings of IEEE/INFOCOM* (2001).

[6] S. Low, F. Paganini, J. Wang, S. Adlakha and J. Doyle, "Dynamics of TCP/AQM and a scalable control", *Proceedings of IEEE INFOCOM* (2002).

[7] S. Low, F. Paganini, Z. Wang and J. Doyle, "A new TCP/AQM for stable operation in fast networks", *Proceedings of IEEE INFOCOM* (2003).

[8] F. Paganini, J. Doyle and S. Low, "Scalable laws for stable network congestion control", *Proceedings of Conference on Decision and Control* (2001).

[9] G. Vinnicombe, "On the stability of networks operating TCP-link congestion control", *Proceedings of IFAC World Congress* (2002).

[10] I. Lestas and G. Vinnicombe, "Scalable robust stability for nonsymmetric heterogeneous networks", *Automatica* (2006).

[11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", *IEEE/ACM Trans. on Networking*, pp 397-413 (1993).

[12] W. Feng, D. Kandlur, D. Saha and K. Shin, "A self-configuring red gateway", *Proceedings of IEEE/INFOCOM* (1999).

[13] R. Gibbson and F. Kelly, "Distributed connection acceptance control for a connectionless network", *Proceedings of the 16th International Telegraffic Congress* (1999).

[14] S. Low, S. Athuraliya, V. Li and Q. Yin, "REM: active queue management", *IEEE Network* (2001).

[15] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue", *Proceedings of ACM SIGCOMM* (2001).

[16] C. Hollot, V. Misra, D. Towsley and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows", *IEEE Trans. on Automatic Control*, pp 945-959 (2002).

[17] R. Watanabe, K. Uchida, M. Fujita and E. Shimemura, "L2 gain and H-infinity control of linear systems with scheduling parameter", *Proceedings 33rd IEEE Conference on Decision and Contro*l, pp 1412-1414 (1994).

[18] W. Rugh and J. Shamma, "Research on gain scheduling", *Automatica*, pp 1401-1425 (2000).

[19] T. Azuma, R. Watanabe and K. Uchida, "Constrained state feedback h-infinity control of nonlinear systems", *Proceedings 3rd Asian Control Conferenc*e, pp 476-479 (2000).

[20] T. Azuma, H. Naito and M. Fujita, "Stabilization of linear time-delay systems with self-scheduling parameters and the application to a stabilizing congestion control synthesis for TCP/AQM networks," *Proceedings IFAC Workshop on Time Delay Systems* (2004).

[21] T. Azuma, H. Naito and M. Fujita, "Experimental verification of stabilizing congestion controllers using the network testbed", *Proceedings 2005 American Control Conference*, pp 1841-1846 (2005).

[22] V. Misra, W. Gong and R. Srikant, "Robust rate control for ABR sources", *Proceedings of ACM/SIGCOMM* (2000).

[23] C. Hollot and Y. Chait, "Nonlinear stability analysis for a class of TCP/AQM networks", *Proceedings of Conference on Decision and Control* (2001).

[24] T. Azuma, R. Watanabe and K. Uchida, "An approach to solving parameter dependent LMI conditions based on finite number of LMI conditions", *Proceedings of American Control Conference*, pp 510-514 (1997).

[25] T. Azuma, R. Watanabe and K. Uchida, "A new LMI approach to analysis of linear systems depending on scheduling parameter in polynomial forms", *Automatisierungstechnik*, pp 119-204 (2000).

[26] T. Azuma and H. Hirano, "A design of congestion controllers for computer networks using robust state predictive control", *Proceedings of Computational Intelligence, Robotics and Autonomous Systems (CRIAS2005)* (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Optimal Design of Manipulator Parameters Using Evolutionary Optimisation Techniques

B.K. Rout, N.N. Sharma and R.K. Mittal
Mechanical Engineering Group
Birla Institute of Technology and Science, Pilani, Rajasthan, India
rout@bits-pilani.ac.in, nns@bits-pilani.ac.in, rkm@bits-pilani.ac.in

Abstract

This paper discusses an approach to optimise the parameters of a manipulator for optimal energy usage given a task specification. To obtain the optimum parameters two evolutionary optimisation techniques have been used. The objective function minimises energy usage while performing a straight-line motion subject to different physical constraints. The application has been illustrated by taking a SCARA manipulator. It is observed that DE converges faster as compared to GA and requires significantly less number of function evaluations. Better results are also achieved by reaching smaller objective functions.

Keywords: differential evolution optimisation technique, genetic algorithm, SCARA, energy usage

1 Introduction

Mostly energy usages of robots are dependent on kinematic and dynamic parameters of robots and the type of task. In this direction, very few attempts have been made. The design of robotic manipulators begins with the dimensioning of the various links to meet task specifications. To design a robot to satisfy the desired task requirement is a complex activity for the reason that the relationship between the robot’s actuators and end-effector varies with position and orientation. There are many efficient search methods available, but most of them are incompatible with robot design problems, because objective function is highly non-linear and coupled. The disadvantages are the requirement of objective function in terms of decision variables, the requirement of initial guessing solution to decide which point to explore next and premature convergence of optimisation process at local optimum. In order to overcome above shortcomings evolutionary based technique is employed in place of conventional optimisation methods. The method is illustrated by optimum design of SCARA manipulator based on task specification. The proposed methodology is a viable alternative to the costly prototype testing, where mathematical model is investigated which describes energy usage in terms of the set of design parameters.

1.1 Background of Research

Optimum robot design has been attempted by researchers on kinematic and dynamic specifications [1-2]. Zhuang et al. [3] had applied GA to select the optimal robot measurement configurations, which is an important element in robot calibration experiments. Aspragathos and Foussias [4] had used GA for searching the best location of a path in the robot workspace considering the velocity performance and later Paredis [5] used a distributed agent based GA to create fault tolerant serial chain manipulators from a small inventory of links and modules. Oral and Idler [6] had optimised weight of high-speed robot for a particular motion using sequential quadratic programming. Agarwal and Veeraklaew [7] attempted to design robots using collocation techniques so that their parameters can be adjusted to optimal value for a given sequence of motion. Coello [8] developed new technique that combines GA and the weighted min-max multi-objective optimisation method for robot design. Yang and Chen [9] applied the minimised degree of freedom concept in a modular robotic system to maximise the load carrying capacity and reduce the power requirements using an evolutionary algorithm. Shiakolas et al. [10] optimised the required torque of robots using three evolutionary techniques for a defined motion subject to kinematic, dynamic and structural condition constraints. Garg and Kumar [11] proposed a path planning method that searched the manipulator configuration for an optimum path based on performance index value using GA and simulated annealing techniques.

GA and their variants are part of the evolutionary techniques and have been extensively used for modular robot design, inverse and forward kinematics, calibration and optimum motion and path planning. DE is a recently developed evolutionary technique [12] and has been successfully applied to many other areas [13]. The evolutionary optimisation techniques have been illustrated using SCARA robot for optimum energy while performing a straight-line motion.

2 Optimisation Problem

The optimisation problem dealt here is to find optimum manipulator kinematic and dynamic parameters to optimise energy usage while performing the task. The task specification comprises, the kinematic characteristics i.e. the required position and orientation of the end effector. Constraints are imposed on the minimum and maximum (range) values of the link parameters (link length and mass), and reachability of the manipulator.

2.1 Objective Function

Present study focuses on optimising the energy usage to perform the task, therefore objective function become summation of energy used at every joint during straight-line motion of the manipulator. The objective function become

$$\text{Minimise } z = \sum_{p=1}^{point} \sum_{j=1}^{joints} \tau_{pj} q_{pj} \qquad (1)$$

where τ_{pj} and q_{pj} are the torque and joint displacement of j^{th} joint at p^{th} point respectively.

2.2 Constraints

The constraints for this optimisation problem include the link parameter ranges and physical constraints such as limits on the joint variables as given below.

$$L_l \le L_a \le L_u \qquad (2)$$

$$m_l \le m_a \le m_u \qquad (3)$$

where L_l, m_l and L_u, m_u being the lower bounds and the upper bounds of the length and mass of link *a* respectively. The constraint on the joint limits or range of motion of the manipulator actuators are imposed due to physical constraints. The joint constraints are defined as

$$\theta_{j,\min} \le \theta_j \le \theta_{j,\max} \qquad (4)$$

where θ_j is the joint value for joint *j*.

3 Steps Followed for Optimal Design

The first step here is to define the problem, define the design variables, assign ranges of values to all the parameters, and define the constraint vectors. The problem definition consists of defining the kinematic structure of the manipulator to be analysed using Denavit–Hartenberg (D-H) parameters, the desired initial and final position in Cartesian space and desired time for the motion.

To perform a task, the torque vector required to be applied at joints need to be determined for a given task. The torque vector is determined using process of inverse dynamics. The information regarding joint displacement, velocity and acceleration vectors of a specified trajectory along with the geometric parameters i.e. link lengths, link masses, link inertias, of the manipulator are used for the computation of torque vector.

The defined values are used in the analysis routines to obtain values for the design parameters. The design parameters are checked for constraint violation and if constraints are not violated then design parameters are used in evaluating the objective function. These evaluations are used in the optimisation routine where new values for the design variables are generated. One function evaluation is completed when one set of design variables is analysed. This process is repeated until certain criteria are met. The analysis routines related to manipulator motion analysis are inverse kinematics, joint space trajectory generation, and inverse dynamics.

3.1 Inverse Kinematics

Inverse kinematics evaluates the required joint variables (*i*) and (*f*) based on the desired Cartesian coordinates of the initial positions. To develop a solution to the inverse kinematics problem the desired tool configuration are specified as input data which is given by {*p*, *R*}, where *p* represents the tool position relative to the base and *R* represents the tool orientation relative to the base. Specifying tool-tip position with a translational vector *p* is a natural and convenient technique. The compact representation of tool configuration available in [15] is used for analysis.

3.2 Trajectory Planning

The goal of trajectory planning is to describe the requisite motion of the manipulator as a time sequence of joint/link/end-effector locations and derivatives of locations, which are generated by specified time law. Joint space trajectory generates the position $\theta(t)$, velocity $\dot{\theta}(t)$ and acceleration $\ddot{\theta}(t)$ based on the desired motion.

3.3 Inverse Dynamics

Dynamics evaluates the individual joint torques based on the geometric characteristics of the links, and position, velocity and acceleration profiles. The evaluation of the dynamic equations includes friction losses which is proportional to the magnitude of the velocity. The generalised dynamic model of the *n*-DOF manipulator in a compact form is represented by [14]

$$\tau_i = \sum_{j=1}^{n} M_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_j\dot{q}_k + G_i + \mu\dot{q}_i \qquad (5)$$

where M_{ij} is the symmetric inertia matrix, h_{ijk} the centrifugal and coriolis force vector, G_i the gravity force vector, μ is the viscous friction coefficient at the joints.

3.4 Straight-line Motion

To plan a trajectory in joint space which generates a straight-line trajectory in tool-configuration space, the inverse kinematics equations is used. Assuming that w^0 and w^1 denote the initial and final point, respectively, in tool-configuration space. If the movement is to be carried out in T seconds, then a general straight-line trajectory for the tool is represented as follows.

$$w(t) = \lfloor 1 - s(t) w^0 + s(t) w^1 \rfloor \quad 0 \le t \le T \tag{6}$$

where $s(t)$ is a differentiable speed distribution function mapping [0, T] with $s(0) = 1$ and $s(T) = 1$.

It is not sufficient to simply follow a straight-line path in joint space between the end points, because in general, this will not produce a straight-line path in tool configuration space. However, if the distance between the adjacent points in joint space is sufficiently small, the straight line segment in joint space will approximate a straight-line path segment in tool-configuration space. Thus, a straight line path is approximated by visiting an approximate sequence of closely spaced knot points in joint space. A simple, yet effective technique for selecting the knot points called the bounded deviations method has been proposed by Taylor (1979). The algorithm available for approximating straight-line motion [15] with a point-to-point robot is used.

4 Optimisation Approaches

Two evolutionary techniques, genetic algorithm (GA) and differential evolution (DE) are discussed briefly.

4.1 Genetic Algorithm

Over the last decade, GA has been used extensively as optimisation tools in various areas such as sciences, commerce, and engineering. The primary reasons for the success are broad applicability, ease of use, and global perspective. GA is an adaptive search and optimisation algorithm based on the mechanics of natural selection and natural genetics. It is robust in complex search space and is versatile in its application. The genetic algorithm operates on the principle of “survival of the fittest” where weak individuals die before reproducing and stronger individuals survive and bear many offspring who often inherit qualities that are in most cases superior to their parents. The working principle of GA is described in [10].

4.2 Differential Evolution

Differential Evolution Algorithms (DEs) are search and optimisation tools, which work differently, compared to classical search and optimisation methods. Because of their broad applicability, and ease of use DEs have been increasingly applied to various search and optimisation problems in the recent past. Unlike simple GA which uses binary coding for representing design parameters, DE uses real coding of floating point numbers. Storn [13] gave the working principle of DE. The key parameters of control are NP- the population size, CR- the crossover constant, F- the weight. The detailed DE algorithm used in the present study is given below.

The genetic operation of mutation in the DE uses the vector differentiation method (adding the weighted difference between two population vectors to the third vector) to generate a new vector. DE is a parallel search method that operates on D-dimensional parameter vectors, where D is the number of design variables. The number of vectors is equal to user defined population size. The initial vector population is chosen randomly. The DE process starts from selecting a target vector. Then, it randomly selects two other vectors and generates a difference vector, which is multiplied with a user defined weighting factor F to obtain ‘weighted difference vector’. The weighted difference vector and randomly chosen mutation vector create a noisy vector, which is subjected to crossover process with the target vector in order to generate the trial vector. The trial vector is then compared with the original target vector. The vector having less cost function is allowed into the new population. A termination criterion is then checked. If the termination criterion is not satisfied, the population of solutions is modified by three main operators and a new population is created.

5 Manipulator Parameter Design Optimisation

The manipulator models, kinematics, inverse kinematics and dynamic analysis for the manipulator considered are based on the D-H parameters. In the presented example, the parameters are expressed in the SI units and the angles are in degrees. The problem consists of finding the design parameter combination that has minimum energy usage. To illustrate the application SCARA manipulator [15] is considered. The following relationship has been formulated for the SCARA manipulator to specify the corresponding fitness function. The joint variables q_i are constrained to lie in the joint space work envelope where, $\beta \ge 0$ is the physical constraint imposed on joint2, and h and H are the joint constraints on the prismatic joint 3.

$$\begin{bmatrix} -\pi \\ -\pi+\beta \\ h \\ -\pi \end{bmatrix} \le \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} \le \begin{bmatrix} \pi \\ -\pi-\beta \\ H \\ \pi \end{bmatrix} \tag{7}$$

Therefore, the locus of points p reachable by the tool tip satisfies the following inequalities:

$$a_1^2 + a_2^2 - 2a_1a_2\cos\beta \le p_1^2 + p_2^2 \le (a_1 + a_2)^2 \qquad (8)$$

$$d_1 - d_4 - H \le p_3 \le d_1 - d_4 - h \qquad (9)$$

where p_1, p_2, p_3 are the *x, y* and *z* co-ordinates of the tool tip and a_1, a_2, d_1 and d_4 are the link kinematic parameters.

$$a_{il} \le a_i \le a_{iu} \qquad (10)$$

$$d_{il} \le d_i \le d_{iu} \qquad (11)$$

$$m_{il} \le m_i \le m_{iu} \qquad (12)$$

where $a_{il}, d_{il}\, m_{il}$ and $a_{iu}, d_{iu}\, m_{iu}$ are the lower bounds and the upper bounds of the length and mass of link *i* respectively.

5.1 Kinematic and Dynamic Models of SCARA Manipulator

The link-coordinate diagram built according to D-H algorithm [15] and the kinematic parameters are listed in table 1.

Table 1: Kinematic parameters of SCARA robot.

Axis	θ	d	a	α
1	q_1	d_1	a_1	π
2	q_2	0	a_2	0
3	0	q_3	0	0
4	q_4	d_4	0	0

The tool position with respect to base is given by

$$T_{base}^{tool} = \begin{bmatrix} C_{1-2-4} & S_{1-2-4} & 0 & a_1C_1 + a_2C_{1-2} \\ S_{1-2-4} & -C_{1-2-4} & 0 & a_1S_1 + a_2S_{1-2} \\ 0 & 0 & -1 & d_1 - q_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

where the notation C_{1-2-4} denotes $Cos(q_1 - q_2 - q_4)$ and similarly, S_{1-2-4} denotes $Sin\,(q_1 - q_2 - q_4)$. The vector representation of dynamic model SCARA is given by

$$\tau_1 = M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + M_{14}\ddot{q}_4 + (H_1V)^* + \mu\dot{q}_1 \qquad (14)$$

$$\tau_2 = M_{21}\ddot{q}_1 + M_{22}\ddot{q}_2 + M_{24}\ddot{q}_4 + (H_2V)^* + \mu\dot{q}_2 \qquad (15)$$

$$\tau_3 = M_{33}\ddot{q}_3 + \mu\dot{q}_3 \qquad (16)$$

$$\tau_4 = M_{41}\ddot{q}_1 + M_{42}\ddot{q}_2 + M_{44}\ddot{q}_4 + \mu\dot{q}_4 \qquad (17)$$

The joint trajectories are planned using straight line motion as discussed in the section 3.4. The task specifications and constraints are given in tables 2-3. It is assumed that tasks are completed in 2sec. The constraint values for the design variables, the link lengths, and mass parameters are assumed to be same for both the optimisation approaches. Taking the control parameter values available in Table 4 and the values in tables 1-2 both the optimisation processes are implemented one after another.

Control parameters for the optimisation process are selected by trial and error method for which number of function evaluation is lowest. The results of the optimisation process utilising the two evolutionary techniques are presented in table 6. In this table, the values of the design variables, the objective function value, and number of function evaluations for each technique are presented.

Table 2: Task specifications and number of points generated using straight line algorithm.

Initial configuration	Final configuration	Error ε	No. of points generated
$W_0 = \begin{bmatrix} 800 \\ 0 \\ 677 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$W_1 = \begin{bmatrix} 203.4 \\ 662.7 \\ 557 \\ 0 \\ 0 \\ 1.6487 \end{bmatrix}$	1×10^{-3}	9
		1×10^{-4}	33
		1×10^{-5}	93
$W_0 = \begin{bmatrix} 800 \\ 0 \\ 677 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$W_1 = \begin{bmatrix} 500 \\ 325 \\ 278.5 \\ 0 \\ 0 \\ 1.3243 \end{bmatrix}$	1×10^{-3}	9
		1×10^{-4}	33
		1×10^{-5}	93

Table 3: Constraints for optimisation algorithm.

	Link parameter	Joint ranges
Link length parameter (mm)	$420 \le a_1 \le 430$ $370 \le a_2 \le 380$ $872 \le d_1 \le 882$ $195 \le d_2 \le 205$	$-\pi + 15 \le q_2 \le \pi - 15$ $15 \le d_3 \le 180$
Link mass parameter (kg)	$8 \le m_1 \le 22$ $13 \le m_2 \le 17$ $8 \le m_3 \le 12$ $3 \le m_4 \le 7$	

Table 4: Control parameters of GA and DE.

Control parameters	DE	GA
Population size	40/	40
Number of generations	50/	50
Crossover probability	0.5/	0.8
Weighting factor	0.8	–
Mutation probability	–	0.175

It is observed that the GA and DE results are closer to each other. However, the DE approach consistently obtains smaller function values within small number of function evaluations. The number of function evaluations is an indication of the computing effort required in reaching the optimum function value for the same number of generations. The theoretical number of function evaluations is given by

$$\text{NOF} = (\text{Population size} \times \text{Number of Generations}) \qquad (18)$$

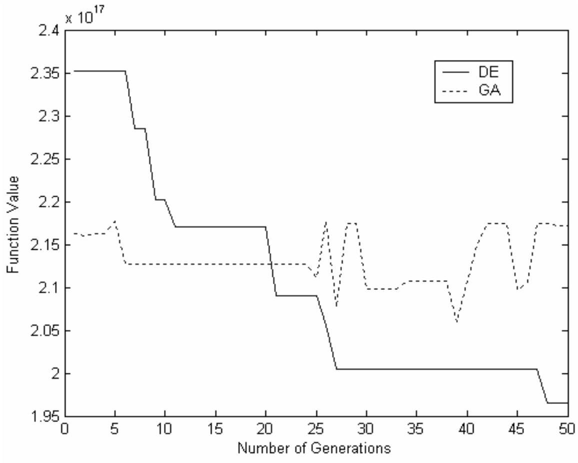


Figure 1: Function history for task 1 with ε =0.001.

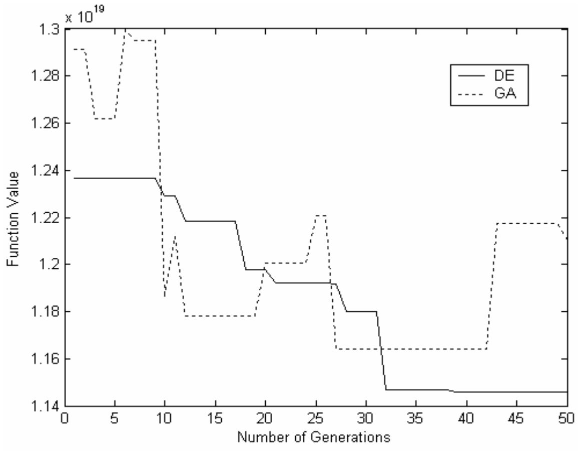


Figure 2: Function history for task 1 with ε =0.0001.

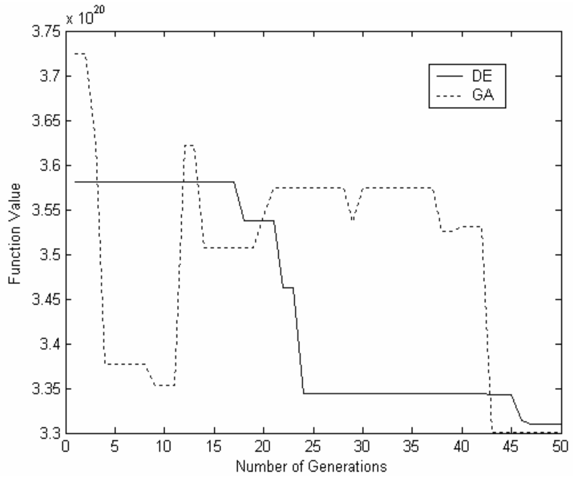


Figure 3: Function history for task 1, ε =0.00001.

However, in the case of DE after the trail vector is computed each member of the trail vector is subjected to bound check and only those members who pass the bound check will have cost function evaluation step. The objective function history for each case is shown in figures 1 to 6 respectively. It is observed that the DE converges faster in a monotonic fashion than GA after 10 generations.

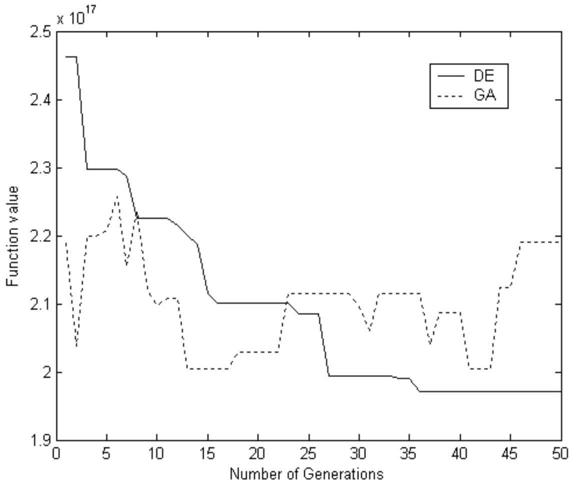


Figure 4: Function history for task 2 with ε =0.001.

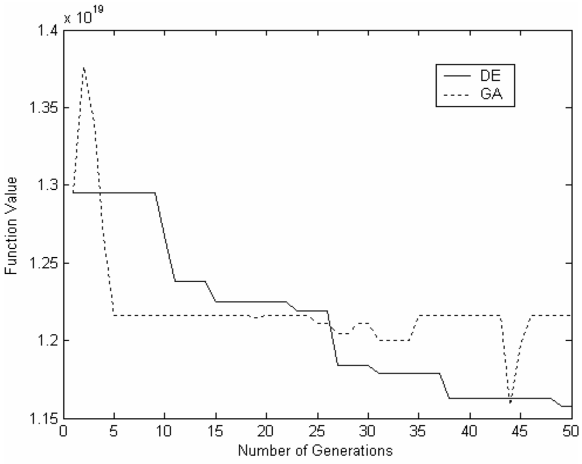


Figure 5: Function history for task 2 with ε =0.0001.

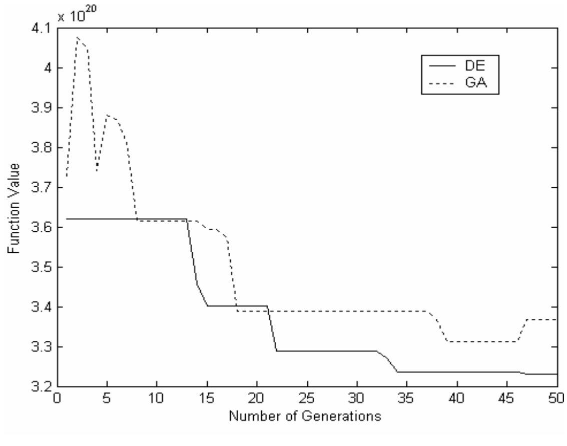


Figure 6: Function history for task 2, ε =0.00001.

6 Conclusions

Here two evolutionary optimisation approaches GA and DE, are used and compared for the design of SCARA manipulator based on task specifications and constraints. The design process considered the kinematic and dynamic characteristics of the links of the manipulator.

Table 6: Summary of DE and GA results for tasks 1 and 2.

	$\mathcal{E}$ **(mm)**	**a$_1$ (mm)**	**a$_2$ (mm)**	**d$_1$ (mm)**	**d$_4$ (mm**	**m$_1$ (kg)**	**m$_2$ (kg)**	**m$_3$ (kg)**	**m$_4$ (kg)**	**Energy (J)**	**NOF**
DE task1	1×10^{-3}	420.817	374.430	872.774	204.362	18.199	13.057	8.012	3.120	1.965×10^{17}	876
	1×10^{-4}	422.804	371.598	876.614	202.145	19.043	13.071	8.075	3.049	1.146×10^{19}	832
	1×10^{-5}	426.883	377.098	872.533	202.256	18.140	13.189	8.074	3.196	3.309×10^{20}	858
DE task2	1×10^{-3}	424.213	374.636	878.214	201.784	18.424	13.192	8.015	3.028	1.971×10^{17}	958
	1×10^{-4}	422.158	374.285	880.253	197.748	18.085	13.257	8.097	3.096	1.157×10^{19}	852
	1×10^{-5}	428.195	379.865	875.034	202.794	18.422	13.008	8.034	3.006	3.231×10^{20}	909
GA task1	1×10^{-3}	425.215	377.490	879.607	203.509	21.482	13.282	8.345	3.203	2.171×10^{17}	2040
	1×10^{-4}	422.235	376.117	877.411	204.647	18.250	13.015	8.031	3.784	1.210×10^{19}	2040
	1×10^{-5}	428.901	376.313	878.705	198.372	18.690	13.031	8.156	3.062	3.574×10^{20}	2040
GA task2	1×10^{-3}	425.960	378.156	876.274	196.764	20.933	13.188	8.486	3.250	2.190×10^{17}	2040
	1×10^{-4}	422.000	377.490	875.098	200.803	19.396	13.376	8.313	3.141	1.216×10^{19}	2040
	1×10^{-5}	422.588	380.000	878.784	195.117	18.925	13.501	8.141	3.094	3.367×10^{20}	2040

The objective has been to minimise energy usage to perform the defined motion subject to constraints on link parameters length, and mass. The results indicate that the DE reaches a steady-state objective function quickly requiring only a small number of generations and function evaluations as compared to GA. The fast performance of DE indicates that it can be a viable optimisation approach for real-time optimisation. The results show that the evolutionary optimisation techniques are reliable, despite the fact that it is relatively slow. However, it is not a serious drawback, since the selection of optimal design is an off-line procedure.

7 References

[1] Z. Shiller and S. Sundar "Design of multi-degree-of-freedom mechanisms for optimal dynamic performance", *ASME Journal of Mechanical Design*, 115, pp199-206 (1993).

[2] J.J. Paredis and P.K. Khosla "Kinematic design of serial link manipulators from task specifications", *International Journal of Robotics Research*, 12(3), pp 274–287 (1993).

[3] H. Zhuang and W. Haung "Optimal selection of measurement configurations for robot calibration using simulated annealing", *Proc. of IEEE, Int. Conf. Robotics and Automation*, San Diego, pp 393-398 (1994).

[4] N.A. Aspragathos "Optimal location of path following tasks in the workspace of a manipulator using genetic algorithms", *Proc. Recent Advances in Robot Kinematics* (*ARK'96*), Portroz, Slovenia, pp 179-188 (1996).

[5] J.J. Paredis and P.K. Khosla "Agent-based design of fault tolerant manipulators for satellite docking". *Proc. IEEE International Conference on Robotics and Automation*, 4, pp 3473-3480 (1997).

[6] S. Oral and S.K. Ider "Optimum design of high-speed flexible robotic arms with dynamic behaviour constraints", *Computer and Structures*, 65, pp 255-259 (1997).

[7] S.K. Agarwal and T. Veeraklew **"**Designing robot for optimal performance during repetitive motion", *IEEE Trans. of Robotics and Automation*, 14, pp 771-777 (1998).

[8] C.A. Coello Coello, A.D. Christiansen, and A.H. Aguirre, "Using a new GA-based multi-objective optimization technique for the design of robot arms", *Robotica*, 16, pp 401-414 (1998).

[9] G. Yang and I.M. Chen, "Task based optimization of modular robot configurations: minimized degree-of-freedom approach", *Mechanism and Machine Theory*, 35, pp 517-540 (2000).

[10] P.S. Shiakolas, D. Koladiya and J. Kebrle, "Optimum robot design based on task specifications using evolutionary techniques: kinematic, dynamic and structural constraints", *Inverse Problems in Engineering*, 10(1), pp 359-375 (2002).

[11] D.P. Garg and M. Kumar "Optimization techniques applied to multiple manipulators for path planning and torque minimization", *Engineering Applications of Artificial Intelligence*, 15, pp 241-252 (2002).

[12] R. Storn, *Differential Evolution Design of an IIR-Filter with Requirements for Magnitude and Group Delay*, International Computer Science Institute (1995).

[13] R. Storn "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, 11(4), pp 341-359 (1997).

[14] R.K. Mittal and I.J. Nagrath, *Robotics and Control*, Tata McGraw-Hill, New Delhi, 2003.

[15] R.J. Schilling, *Fundamentals of Robotics: Analysis and Control*, Prentice Hall of India, New Delhi, 1996.

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Automatic Gain-Scheduled Control Using Stability Radius

Rini Akmeliawati [1] and Iven M. Y. Mareels [2]
[1] School of Engineering, Monash University Malaysia
[2] Department of Electrical and Electronic Engineering, University of Melbourne
rini.akmeliawati@eng.monash.edu.my, i.mareels@unimelb.edu.au

Abstract

In this paper we present a systematic and automatic gain-scheduled control using stability radius. The technique provides automatic selection of sets of operating points for a gain-scheduled control. The technique is illustrated on a longitudinal flight control problem.

Keywords: gain-scheduled control, automation, flight control, complex stability radius

1 Introduction

Gain-scheduled control is one of most common control techniques used in practice. This control technique is based on linear control theory but results in a nonlinear control. In general, gain-scheduled control is obtained as follows: 1. Selecting a set of operating points; 2. Designing (linear) controller for each operating point selected; 3. Scheduling or interpolating the resulting family of the controllers to obtain a global controller.

This technique is initially known as an ad-hoc technique, whereby design validation and verification relies much on extensive simulations. Not until the last decade that analytical and mathematical justifications are available to demonstrate the stability and performance robustness of GS control [1], [2].

Many studies can be found on various types of gain-scheduled control, such as switching control by control Lyapunov function [3], linear-fractional transformation (LFT)[5], linear-parameter varying (LPV) control [6] and [7]. Most of those are summarised in [8] and [9]. All those techniques are based on assumption that the set of operating points are known *a priori*. The issues of how many operating points to use and how to obtain the *appropriate* set of operating point are overlooked and have never been addressed. Our proposed technique will address these issues. In practice, this first step of GS technique is still done based on heuristics and extensive simulations. Prior knowledge of the system to be controlled becomes necessary to select the operating points. Thus, GS control results in a time-consuming and expensive technique.

In [10] an automation procedure for a missile autopilot using the structured singular value μ is proposed. The results indicate conservatism due to the inadequacy of the μ-analysis in analysing mixed uncertainties (parametric and dynamic) of the missile dynamics [10].

In this paper we propose a systematic way of selecting the operating points for GS control system such that the stability and performance requirement can be achieved. With the proposed automatic gain-scheduled control, we construct the grid points at the scheduling-variable space based on the complex stability radius. Thus, for any trajectories, the closed-loop system will guarantee the stability and performance robustness. This technique is an extension to the work that was done in [11]. The complete process from selection of operating points which guarantees stability and performance robustness to design and verification is addressed. The technique is illustrated on a flight control problem of a civil aircraft model.

Our proposed automatic GS technique is similar to the stability-based switching controller developed in [3]. The difference is that our proposed technique explores the idea of complex stability radius whereas in [3] the technique uses equilibria-dependent Lyapunov functions. The advantage of using the complex stability radius is that it can handle a wider class of time variations [13]. This is not the case with the equilibria-dependent Lyapunov functions. Furthermore, in our proposed technique, there is no need to compute the Lyapunov function for each equilibrium or operating points. Our technique also addresses the problem encountered in [10] whereby μ-analysis cannot handle mixed parametric and dynamic uncertainties as mentioned earlier.

This paper is organised as follows. In section 2 we will briefly introduce mathematical notations and definitions. The definition of complex stability radius is presented in section 3. Section 4 presents the automatic GS algorithm. Section 5 presents an example of the technique in flight control problem. Finally we summarise this paper in section 6.

2 Mathematical Preliminaries

Here we introduce some basic mathematical terminology and definitions that are used in the rest of this paper.

- $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$. $\mathbb{R}$: the set of real numbers. $\mathbb{C}$: complex plane.
- Let S be a set. For $S \subset \mathbb{R}^n$, ∂S , and S^o (or $int(S)$) are denoted as the *boundary* and the *interior* of S, respectively. $\overline{S}$ is the complement of S.
- Given a family of sets $S_{j\,j\in J}$, $S \subset X$ the *intersection* of this family is defined as a set $\cap_{j\in J} S_j = \{x \in S_j, \forall j \in J\}$. The *union* of this family is defined as a set $\cup_{j\in J} S_j = \{x \in S_j, for\ some\ j \in J\}$.
- Let $T : D \to W$. The set $Im(T) := \{T(x) \in W; x \in D\}$ defines the *image* of T.
- $\mathbb{C}_- = \{(x,y) \in \mathbb{R}^2; x < 0\}$; $\mathbb{C}_- = \{z \in \mathbb{C} | Real(z) < 0\}$.
- $\mathbb{C}_+ = \mathbb{C} \backslash \mathbb{C}_-$, is the closed right half plane.
- $L^\infty(I; \mathbb{F}^{n\times m})$ = the set of functions $f : I \to \mathbb{F}^{n\times m}$ that are essentially bounded on the interval $I \subset \mathbb{R}$.
- $\|f\|_{L^\infty(I;\mathbb{F}^{n\times m})} = \sup_{t\in I} \|f(t)\|$.

Consider a nonlinear system of the form

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_o, \quad t \in \mathbb{R}, \tag{1}$$

where $x(t) \in \mathscr{D} \subseteq \mathbb{R}^n$ is the system state vector, $\mathscr{D}$ is an open set, $u(t) \in \mathscr{U} \subseteq \mathbb{R}^m$ is the control input, $\mathscr{U}$ is the set of all admissible controls such that $u(.)$ is a measurable function with $0 \in \mathscr{U}$, and $f : \mathscr{D} \times \mathscr{U} \to \mathbb{R}^n$ is continuously differentiable on $\mathscr{D} \times \mathscr{U}$.

Definition 1 *$x^* \in \mathscr{D}$ is an* equilibrium point *of (1) if there exists $u^* \in \mathscr{U}$ such that $f(x^*, u^*) = 0$.*

Definition 2 *The trajectory $x(t) \in \mathscr{D}$ denotes the solution of (1) with $x(0) = x_0$ evaluated at time t.*

3 Complex Stability Radius

Here we summarise basic results on the stability radius, which have been initially studied by [15]. Stability radius is a measure of stability bound of a system. Consider a nominal system of the form

$$\dot{x} = Ax \tag{2}$$

The nominal system (2) is subject to affine parameter perturbations of the form $A \mapsto A + \mathscr{E}\Delta\mathscr{H}$, where $\mathscr{E} \in \mathbb{F}^{n\times l}, \mathscr{H} \in \mathbb{F}^{k\times n}$ are the scale matrices that define the structure of the perturbation and $\Delta(.) \in L^\infty(\mathbb{R}_{\geq 0}; \mathbb{F}^{l\times k})$ is unknown. The resulting perturbed system is

$$\dot{x} = (A + \mathscr{E}\Delta\mathscr{H})x, \tag{3}$$

$A, \mathscr{E}, \Delta$ and $\mathscr{H}$ may be interpreted as time-varying matrices for complex stability radius, though we eliminate the dependency of t in the representation mainly to simplify the notation. A may also be interpreted as A_{cl}, a closed-loop system matrix. In this case the perturbation Δ contains plant and input signal perturbations.

Thus, the nominal system (with input), with $A \in \mathbb{F}^{n\times n}, B \in \mathbb{F}^{n\times m}$ and $x \in \mathbb{F}^n$ is in the form:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t). \tag{4}$$

$(A(t), B(t))$ is assumed to be stabilisable and with $u(t) = -K(t)x$(t), $K(t) \in \mathbb{F}^{m\times n}$ be a stabilising controller. Hence, (4) can be rewritten as

$$\dot{x}(t) = A_{cl}(t)x(t), \tag{5}$$

where $A_{cl}(t) = A(t) - B(t)K(t)$ and $\sigma(A_{cl}(t)) \subset \mathbb{C}_-$ and the perturbed closed-loop system

$$\dot{x}(t) = (A_{cl}(t) + \mathscr{E}(t)\Delta(t)\mathscr{H}(t))x(t), \tag{6}$$

where $\|\Delta(.)\|_{L^\infty(0,\infty)} := \sup_{t\geq 0} \|\Delta(t)\|$.

Consider a ”frozen system” which is defined as, for fixed $\tau \geq 0$,

$$\dot{x}(t) = (A_{cl}(\tau) + \mathscr{E}(\tau)\Delta(t)\mathscr{H}(\tau))x(t). \tag{7}$$

Based on Theorem 4.4 in [13], suppose that $(A_{cl}(\tau), \mathscr{H}(\tau))$ is observable for every $\tau \geq 0$ and $\|\dot{A}_{cl}(\tau)\| + \|\dot{\mathscr{E}}(\tau)\| + \|\dot{\mathscr{H}}(\tau)\| < \delta$, $\delta > 0$, for almost all $\tau \geq 0$, then there exists $\rho > 0$, such that

$$\rho \leq r_{c,dyn}(A_{cl}; \mathscr{E}(.), \mathscr{H}(.)).$$

Definition 3 *[15] The stability radius of a ”frozen” A_{cl} is defined by*

$$\begin{aligned} r_F(A_{cl}; \mathscr{E}, \mathscr{H}, \mathbb{C}_+) \quad = \quad & \inf\{\|\triangle\|; \triangle \in \mathbb{F}^{l\times k}, \\ & \sigma(A_{cl} + \mathscr{E}\triangle\mathscr{H}) \cap \mathbb{C}_+ \neq \emptyset\}. \end{aligned}$$

The complex stability radius $r_c(A; \mathscr{E}, \mathscr{H})$ of the system (2) with respect to the perturbation structure $(\mathscr{E}, \mathscr{H})$ is the spectral norm $||\Delta||$ of a smallest destabilising perturbation matrix $\Delta \in \mathbb{C}^{l\times k}$. Complex stability radius measures the robustness of stable linear systems under complex (parameter) perturbations. Obviously, complex stability radius is more conservative than its real counterpart as the later is a measure of robustness under real perturbations only.

Given the controlled perturbed system in the form (6) and let $G(s) = \mathscr{H}(sI - A_{cl})^{-1}\mathscr{E}$ be the ”transfer matrix” associated with $(A_{cl}, \mathscr{E}, \mathscr{H})$, then the complex stability radius is defined by the following theorem.

Theorem 1 *[14, 15] The complex stability radius, r_c:*

$$r_c(A_{cl};\mathscr{E},\mathscr{H},\mathbb{C}_+) = [\max_{s\in\partial\mathbb{C}_+} \|G(s)\|]^{-1}, \tag{8}$$

where $\partial\mathbb{C}_+ = \bar{\mathbb{C}}_- \cap \mathbb{C}_+$, is the boundary of $\mathbb{C}_+$.

Our proposed automation exploits the complex stability radius to determine which operating points should be selected for the controller design points. The result of this automation is a family of operating points with associated controllers such that the stability and performance robustness of the overall design is guaranteed.

4 Automation of Gain-Scheduled Control

We describe our gain-scheduled control as follows. Let $\Omega \subset \mathbb{R}^q$, a compact set, denotes the set of GS parameters so that for every $\theta \in \Omega$ we have an equilibrium (x_θ, u_θ) of (1), i.e. , $f(x_\theta, u_\theta) = 0$. More precisely, we assume that

$$\begin{aligned} s \times c : \Omega &\rightarrow \{(\bar{x},\bar{u}) \in \mathbb{R}^n \times \mathbb{R}^m | f(\bar{x},\bar{u}) = 0\}, \\ \theta &\mapsto (s(\theta), c(\theta)) = (x_\theta, u_\theta). \end{aligned} \tag{9}$$

Let θ_0, $\theta_1 \in int(\Omega)$. Let $\theta_R(t)$ be a piecewise smooth, time-parameterised path contained in $int(\Omega)$ connecting $\theta_0 = \theta_R(0)$ and $\theta_1 = \theta_R(T)$, or
$\theta_R(.) : [0,T] \rightarrow \Omega, T > 0$. See figure 1.

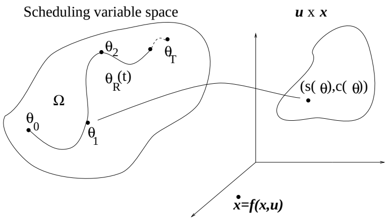


Figure 1: Mapping from scheduling-variable space to $(x \times u)$ space.

The system (1) is linearised about an equilibrium (operating point) (x_θ, u_θ) of (1) for each scheduling parameter $\theta \in \Omega$. Consider now the control law for the system (1)

$$u(t) = u_{\theta(t)} - K_\theta(x - x_{\theta(t)}), \tag{10}$$

where $K_{\theta_R} \in \mathbb{R}^{m\times n}$ is a feedback gain that strictly stabilises the linearised plant at the equilibrium $(x_{\theta(t)}, u_{\theta(t)})$. Hence, for each equilibrium (x_θ, u_θ) we use Taylor's Theorem to derive

$$f(x,u) = A_\theta[x - x_\theta] + B_\theta[u - u_\theta] + R(x_\theta, u_\theta, x, u),$$

where $A_\theta \in \mathbb{R}^{n\times n}$, $B_\theta \in \mathbb{R}^{n\times m}$, and $R : \mathbb{R}^{2n+2m} \rightarrow \mathbb{R}^n$ is a continuous function satisfying

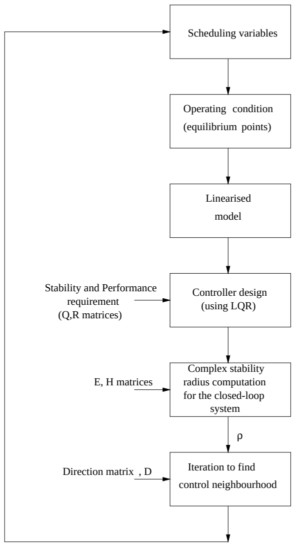


Figure 2: The GS automation flowchart [11].

$$\begin{aligned} \|R(x_\theta, u_\theta, x, u)\| &\leq r_\theta(x,u) \left\| \begin{matrix} x - x_\theta \\ u - u_\theta \end{matrix} \right\|, \\ \forall(x_\theta, u_\theta, x, u) &\in \mathbb{R}^{2n+2m}, \end{aligned}$$

and $r_\theta(.,.) : \mathbb{R}^{n+m} \rightarrow \mathbb{R}_{\geq 0}$ is also continuous and $r_\theta(x,u) \rightarrow 0$ as $(x,u) \rightarrow 0$.

Based on the linear approximation of (1), i.e.

$$\frac{d}{dt}[x(t) - x_\theta] = A_\theta[x(t) - x_\theta] + B_\theta[u(t) - u_\theta], \tag{11}$$

a state feedback matrix $K_\theta \in \mathbb{R}^{m\times n}$ is designed so that (10) applied to (11) yields an exponentially stable closed loop system

$$\frac{d}{dt}[x(t) - x_\theta] = [A_\theta - B_\theta K_\theta][x(t) - x_\theta].$$

4.1 Automation

The structure of the automation process is shown in figure 2. The following definition describes the automation process.

Definition 4 *[11] With reference to system (1), let $\theta \in \mathbb{R}^q$ be a scheduling variable, $\mathscr{P}$ be a partition of the operating envelope $\Omega \subset \mathbb{R}^q$ in scheduling-variable space and $\mathscr{U}_i$ is convex.*

$$\begin{aligned} \mathscr{P} = \{\mathscr{U}_i, i = 1\cdots N\}; \mathscr{U}_i^\circ = \mathscr{U}_i \backslash \partial\mathscr{U}_i, \\ \mathscr{U}_i^\circ \cap \mathscr{U}_j^\circ = \emptyset, i \neq j, \bigcup_i \mathscr{U}_i = \Omega. \end{aligned}$$

Let θ_i be the centre of $\mathscr{U}_i$, with associated controllers (K_i, u_i) and equilibrium x_i in the operating envelope, $u_i = c(\theta_i), x_i = s(\theta_i)$ (see equation (9) and (10)). The partition $\mathscr{P}$ is acceptable provided the induced GS controller is stabilising $\forall\theta(t)$ such that $\theta(t) \in \Omega$ and $\dot{\theta}(t)$ is sufficiently small $\forall t$, the system

$$\begin{aligned} \dot{x}(t) &= f(x(t), u_{in(\theta(t))} + K_{in(\theta(t))}\hat{x}(t)), && (12) \\ \hat{x}(t) &= x(t) - x_{in(\theta(t))}, && (13) \\ in(\theta(t)) &= j \texttt{ if } \theta(t) \in \mathscr{U}_j, && (14) \end{aligned}$$

is stable and achieves that $x(t)$ remains inside the operating envelope. Further, $\|x(t) - s(\theta(t))\| \ll 1$.

The partition $\mathscr{P}$ is computed by computing the centre θ_i. Starting from a given θ_0 we compute the neighbouring centre in an iterative manner, verifying that the new centre is such that the changes are within the complex stability radius.

The automatic GS control of system (1) with the control law (10) proceeds as follows.

1. The first operating point and the corresponding scheduling variable(s) selection. (This operating point is generally the nominal point or a point where the nonlinearities of the system are significant.)

2. Controller design. Feedback controller is designed for the linearised system to satisfy the performance and stability requirement.

3. Determine $\mathscr{E}$, $\mathscr{H}$ matrices of the system's perturbation and compute the complex stability radius, ρ_i.

4. Construction of stability region, i.e. a ball around θ_i contained in stability radius, more precisely (see figure 3):

   (a) $\mathscr{E}\Delta_\theta\mathscr{H} = (A_i - B_iK_i) - (A_{s(\theta)} - B_{s(\theta)}K_i)$,

   (b) $r(\theta) = \|\Delta_\theta\|$,

   (c) $\Theta_i = \{\theta : r(\theta) \leq \rho_i\}$,

   (d) $B_{\theta_i}(\alpha) = \{\|\theta - \theta_i\| \leq \alpha\}$ (for a given norm) with polygonal level curve.

   (e) Find maximal $\bar{\alpha}$ such that $B_{\theta_i}(\bar{\alpha}) \subset \Theta_i$.

5. Choose θ_{i+1} on the boundary of $B_{\theta_i}(\bar{\alpha})$ and repeat the process.

6. Optimise the θ_i lattice to obtain minimal overlap, and a full partition.

5 Example on A Civil Aircraft Model

In this section we present our proposed automated gain-scheduling technique that was used to design an autopilot system for a research civil aircraft model, RCAM developed by GARTEUR [16]. We only consider the longitudinal dynamics of the aircraft.

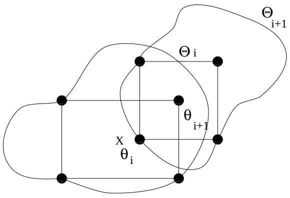


Figure 3: Defining $B_{\theta_i}(\alpha)$ [11].

Table 1: Performance criteria.

Responses	t_r	t_s	RMSE
Altitude	$< 12\ s$	$< 45\ s$	$9.15m$
FPA	$< 5\ s$	$< 20\ s$	-
Airspeed	$< 12\ s$	$< 45\ s$	$< 1\%$
Pitch angle	$< 12\ s$	$< 45\ s$	$\pm 0.5\frac{\pi}{180}\ rad$

The automation is aimed to construct an acceptable partition in scheduling-variable space such that the gain-scheduled control guarantees stability and performance robustness.

5.1 Controller Design

The longitudinal model of RCAM in the body-axes is defined as

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), && (15) \\ y(t) &= g(x(t), u(t)), && (16) \end{aligned}$$

where $x \in \mathbb{D}^4 = [\ u_b \quad w_b \quad q \quad \theta\ ]^T$, the forward and downward velocity in the body-axes, the pitchrate [rad/s], and the pitch angle [rad], respectively; $u \in \mathbb{D}^2 = [\ \delta_e \quad \delta_{th_1} \quad \delta_{th_2}\ ]^T$, the elevator and the left and right throttle deflection angle, respectively. $\mathbb{D} \subset \mathbb{R}$ specifies the region within the flight envelope of interest. $y \in \mathbb{D}^6 = [\ q \quad n_z \quad V \quad \theta \quad \alpha_b \quad \gamma\ ]^T$, and $\alpha_b = \arctan\frac{w_b}{u_b}$, $V = \sqrt{u_b{}^2 + w_b{}^2}$, $\gamma = \theta - \alpha_b$, $n_z = \dot{u}_b/g$, are the angle of attack, the airspeed, the flight path angle (FPA) and the vertical load, respectively. The aircraft mass and the centre of gravitation (COG) position are assumed constant. Detailed descriptions of the aircraft can be found in [16].

The stability criteria follows the Flying Qualities requirement [17], whereas the performance criteria are summarised in table 1. t_r, t_s are the rise time and the settling time, respectively. RMSE is the root mean square error of the response.

Based on figure 2, the automation follows the following steps.

STEP 1: Choose scheduling variables. We select V and γ. Intuitively, V and γ are slowly time-varying and capture the dynamic nonlinearities, so they are prima facie candidates.

STEP 2: The first operating point is selected as

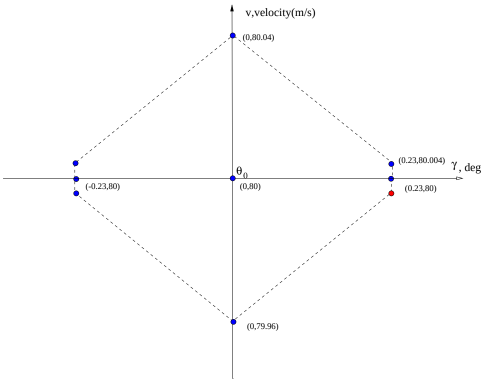


Figure 4: 'Beehive' mapping of the stability region of point (0,80) in the scheduling-variable space.

$(\gamma_0, V_0) = (0, 80)$. This point is considered to be our nominal operating point.

STEP 3: Obtain the linear model. In this step, we linearise the aircraft model, defined by (15) around the operating point selected in STEP 2.

STEP 4: Controller design. In this step, we design a feedback controller for the system defined in STEP 3, $u = -Kx$. For our nominal system, we use an LQR method to design full-state feedback controllers. The weighting matrices are $Q = diag(0.01, 0.001, 0.001, 0.01)$, and $R = diag(10, 10, 10)$. The weighting matrices are chosen such that the stability and performance criteria can be achieved. The choice of Q is mainly aimed to improve the phugoid mode, which is characterised by u_b. The choice of R is to prevent unreasonably large control inputs.

STEP 5: Compute the (complex) stability radius using (8). In this step, we first determine the structure of the perturbation matrices. The computed complex stability radius for the nominal operating point turns out to be 0.0374.

STEP 6: Iterate to find a neighbourhood, i.e. the next operating points to be selected. The neighbourhood of the point (0,80) in the form of a 'beehive' is shown in figure 4. This corresponds to eight directions, which are described by the direction matrix, D (See figure 2). The iterative procedure above is guaranteed to stop reasonably in finite time as $(A - BK)$ is Hurwitz.

Figure 5 shows the resulting (γ, V) flight envelope grid. Our algorithm generates a partition on the basis of stability radius for the linearised system at each centre. A formal verification requires a Lyapunov argument, which can be completed along the lines of Theorem 1 in [11] on the realisation that the complex stability radius accounts for all possible variations. Our result is conservative. This is due to the fact that the complex stability radius allows for a large class of time variations in the system, larger than the system robustness actually requires.

Notice also that the grid structure is quite regular. This regularity is not guaranteed *a priori* as the grid is the projection via a highly nonlinear mapping of a ball in parameter space into the scheduling-variable space.

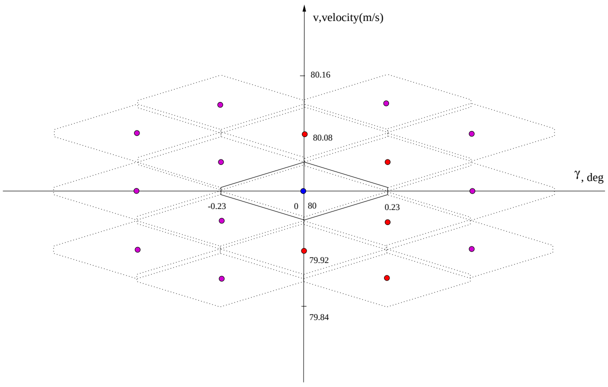


Figure 5: The resulting grid.

The type of controller used affects the magnitude of the stability radius. As computed earlier, the stability radius of the closed-loop system (with the LQR controller) is 0.0374. The (closed-loop) stability radius of the same open-loop system with a feedback controller designed using a pole-placement technique is 0.0168. Both closed-loop systems have the same (closed-loop) eigenvalues. Thus, with the designed LQR controller, the grid is slightly bigger than that with a pole-placement controller.

5.2 Simulation Results

Using the resulting (γ, V) flight envelope grid (figure 5), we simulate the closed-loop systems to achieve trajectory tracking. Seventeen LQR controllers and trim controllers (based on 17 points of the grid) are computed using a MATLAB$^{\text{TM}}$ programme. These controllers are then interpolated using spline interpolation. The flight condition is normal, i.e. no wind disturbances. The desired trajectory is given. The initial operating point is $[u_{bo}, w_{bo}, q_0, \theta_0] = [80, 0.5, 0, 0.03]$. The output responses are shown in figure 6. The control actions are shown in figure 7.

We observe that there is small tracking error in the state responses during climbing. However, the error is still within the design specifications. The maximum deviations in steady-state of the airspeed and the pitch responses are less than 1% and 0.5^o, respectively. The airspeed, the pitch and the flight-path angle responses satisfy the settling time and the rise time requirements.

6 Conclusion and Future Work

The main contribution of the paper is on addressing the issue of operating point selection in a systematic and automatic way with guaranteed stability robustness. The automation allows the GS control design starting from the selection of operating points. The automation is based on the complex stability radius, which can handle a large class of perturbations or uncertainties,

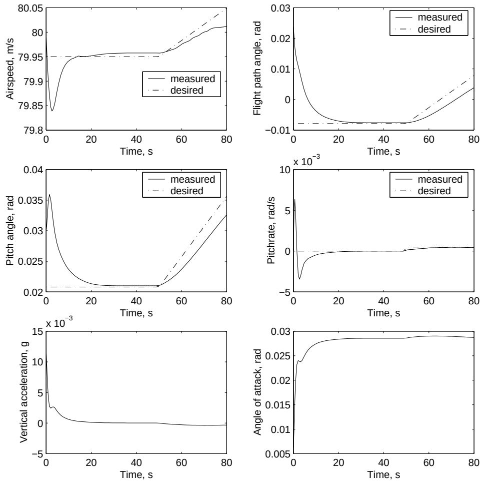


Figure 6: Output responses.

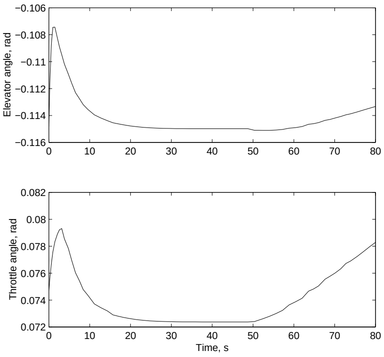


Figure 7: Elevator and throttle activities.

including nonlinear, dynamics and time-varying. Thus, the stability robustness of the closed-loop system is guaranteed regardless of the types of trajectories (in the scheduling-variable space) that the closed-loop system is going through.

Our automatic gain-scheduling technique is successfully illustrated on a flight control problem. However, the result shows a degree of conservatism of the complex stability radius. This is as expected as complex stability radius can deal with a wider class of perturbation, which may not be necessary for our aircraft model in the example.

Future study is required to overcome the conservatism in the result. One approach is to select a different linear control technique for each linearised model to achieve maximum stability radius. Incorporating performance robustness measure could improve the presented result though this normally introduces more conservatism. Nonetheless, no matter how many operating points are used in the end, the automation proposed in this paper can reduce the amount of time and cost in the design process of a gain-scheduled control.

7 References

[1] J. Shamma and M. Athans, ‘Analysis of gain scheduled control for nonlinear plants’, *IEEE Trans. on Automatic Control*, 35(8), pp 898-907 (1990).

[2] W. Rugh, ‘Analytical framework for gain-scheduling’, *IEEE Control Systems Magazine*, pp 79-84 (1991).

[3] A. Leonessa, W.M. Haddad and V. Chellaboina, ‘Nonlinear system stabilization via hierarchical switching control’, *IEEE Trans. on Automatic Control*, 46(1), pp 17-28 (2001).

[4] A. Packard, ‘Gain scheduling via linear fractional transformations’, *Systems & Control Letters*, 22, pp 79-92 (1994).

[5] F. Wu and K. Dong, ‘Robust and gain-scheduled H_2 synthesis for LFT parameter-dependent systems’, *Proc. of the 2005 American Control Conference*, 4, pp 2851-2856 (2005).

[6] C. Scherer, ‘LPV control and full block multipliers’, *Automatica*, 37(3), p 361-375 (2001).

[7] A.R. Mehrabian and J. Roshanian, ‘Design of gain-scheduled autopilot for a highly-agile missile’, *Proc. of 1st Intl. Symp. on Systems and Control in Aerospace and Astronautics*, pp 144-149 (2006).

[8] W.J. Rugh and J.S. Shamma, ‘Survey paper: research on gain scheduling’, *Automatica*, 36, pp 1401-1425 (2000).

[9] D.J. Leith and W.E. Leithead, ‘Survey of gain-scheduling analysis and design’, http://www.hamilton.ie /pubs_Leithead.htm, visited on 01/09/2006.

[10] K. Wise and R. Eberhardt, ‘Automated gain schedules for missile autopilots using robustness theory’, *Proc. of IEEE Conf. of Control on Applications*, pp 243-250 (1992).

[11] R. Akmeliawati and I. Mareels, ‘Stability analysis of gain-scheduled control and automation for flight control systems’, *Proc. of the 14th IFAC World Congress,* Beijing, pp 289-294 (1999).

[12] W. Rugh, *Linear System Theory*, Prentice-Hall, New Jersey (1996).

[13] I. Mareels and A. Ilchmann, ‘On stability radii of slowly time-varying systems’, *Advances in Mathematical Systems Theory: A Volume in Honor of Diederich Hinrichsen*, Birkhauser, Boston (2001).

[14] B. Kelb, D. Hinrichsen and A. Linnemann, ‘An algorithm for the computation of the structured complex stability radius’, *Automatica*, 25(5), pp 771-775 (1989).

[15] D. Hinrichsen, A.J. Pritchard, ‘Stability radius for structured perturbations and the algebraic Riccati equation’, *Systems & Control Letters*, 8, pp 105-113 (1986).

[16] S. Bennani, J. Magni, J. Terlouw, *Robust Flight Control: A Design Challenge,* Springer-Verlag, London (1997).

[17] *USAF MIL-1797-A: Military Standard, Flying Qualities of Piloted Vehicles*, USAF, Washington (1990).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Research on Reduction of Eye Fatigue in Stereovision Using Move-parallel Method

Yuichi Koyama, Hongte Kim, Manami Mizukami and Junichi Takeno
Robot and Science Laboratory,
Meiji University, Kanagawa, Japan
ce66717@isc.meiji.ac.jp, takeno@cs.meiji.ac.jp

Abstract

With conventional stereovision shooting methods using ordinary cameras arrayed in parallel, the convergence distance to sense the distance to the object being viewed does not agree with the focal distance to obtain sharp retinal images. This results in a visual burden that is unique to stereovision. The authors have been studying a mechanism to adjust the convergence of the stereo cameras and a technique to reduce the visual burden caused by attempting to match convergence and focal distance. Visual burden was successfully reduced using cross points but it was found that space perception was distorted in the peripheral vision. The authors have developed a new stereo image shooting method to correct distortion in space perception and reduce the visual burden as much as possible. The present paper reports on this new shooting method.

Keywords: Stereovision, visual burden, camera, remote, teleportation

1 Introduction

Humans desire three main qualities for visual media:

- to be able to see distant views
- to be able to fix the view in front of the observer
- to be able to represent imagined scenery

This desire is inherent to humans and has driven the development and spread of the media we see today. Consumers increasingly see stereovision, such as on mobile phones and game machines with 3D liquid crystal displays, as a result of the development of advanced technologies.

Stereovision is natural to humans and perspective and sense of distance are important sensual information essential in daily life. The present study aims at verifying and establishing the practical use of stereovision and the development of a remote stereovision system (RSVS).

1.1 Factors of Stereovision

Important factors of stereovision are classified into three categories in the present study: 1) Internal stimulation, 2) Single images, and 3) Binocular vision. Of these, the effect of depth perception by disparity is relatively important. Reproduction of internal stimulation (1) is difficult. Remote stereovision will be realised by presenting images based on single images (2) and binocular vision (3). Particularly important is the scheme of modelling and transmitting binocular vision to the observer, or how to view different images with the right and left eyes independently.

1.2 Reproduction of Stereovision

Various stereovision systems are available today. By type, they include viewer, glass and naked eye systems. Time-sharing shutters are adopted in the RSVS. The right and left images are alternately displayed at high speed and the LCD shutters of the goggles switch synchronously. The right and left eyes view only the right and left images, respectively, to achieve independent right and left vision. The image quality is not deteriorated, and this is one of the features of the system developed by the authors.

2 Outline of RSVS

Figure 1 shows the overall functional structure of the system. Each component will be described, followed by the plan for realisation.

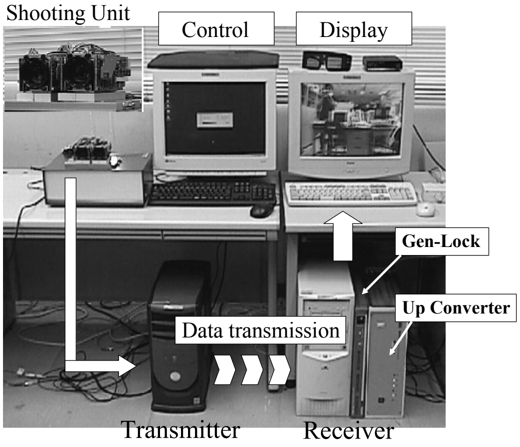


Figure 1: Remote stereovision system.

Shooting Unit: To imitate the human vision mechanism for acquiring factors of stereovision by disparity, two cameras are arrayed, each corresponding to the right and left eyes. Standard camera functions such as zoom and auto-focus are incorporated to adjust the image quality.

Transmitter: Stereo images are transmitted for viewing stereovision remotely. Established networking technologies and the Internet infrastructure are used to send the stereo images. MPEG-2 image compression was selected because wide ranges of compatible commercial products are available on the market.

Display: For an observer to view stereovision remotely, the received stereo images must be displayed to present independent right and left views. To develop the display, stereo image display technology and hardware were selected and developed based on the development concept.

The RSVS Versions 1 and 2 have been prototyped in the present study. RSVS Version 2, which is currently used, is designed to achieve remote stereovision with high image quality. The right and left images are compressed to MPEG-2 and remotely transmitted after synchronising. The received MPEG images are decoded to video signals and converted to double-speed (60 to 120 Hz) images by an up-scan converter for alternate display. Images for each side of the stereovision are alternately displayed on a 720 x 480 pixel screen for very sharp images and the resultant clear stereovision. The transmission rate for the binocular images is about 13 Mbps. The transmission delay is about 1 sec, which poses a problem for real-time observation. The authors will improve the RSVS for real-time viewing while maintaining high image quality.

3 Visual Burden

Visual burden in stereovision systems comes from the sense of fatigue caused by wearing gadgets such as glasses and the mismatch with the visual system. While recent display units feature ever-higher resolution and hardware is lighter due to advanced technology, mismatch with the visual system has not been discussed much by researchers. Stereo image shooting cameras are generally arrayed and fixed parallel to each other. This means that the information on disparity in the shooting space is fixed. This is similar to looking at an object within the visual field while watching a certain distant point in ordinary vision, and it is very unnatural. The authors believe that this unnatural condition causes a burden to the brain that is manifested as a visual burden, and offer adequate binocular images to reduce visual burden in the present paper.

3.1 Cross Method and Parallel Method

Two methods are used to shoot stereo images: the cross method and the parallel method. The features of these methods are described below.

Parallel Method: The right and left cameras shoot images with their optical axes kept parallel to each other. The obtained images are free from geometrical distortion. But the stereovision viewer must diverge their eyes. The parallel method is effective for shooting distant views.

Cross Method: The optical axes of the right and left cameras are crossed. This is the most popular stereo image shooting method. The cross point of the optical axes of the cameras is reproduced on the stereo display plane. Stereo effects are adjustable by selecting the object on which the cross point is set before shooting.

3.2 Visual Burden due to Different Convergence Distance and Focal Distance

When stereo images taken using the parallel and cross methods are shown on a display, the difference of convergence and accommodation becomes apparent. In natural vision, convergence and accommodation rest on a single object. Assume the cross point is set on a different point than the object (in front of or in back of the object), then the convergence of the viewer of the stereo images moves to the stereo image while accommodation stays in the same place on the display. Convergence moves stereoscopically to the back and front of the display while accommodation basically moves plenary on the display. This poses a problem in that convergence of the visual system and accommodation provides different information about distance to the viewer of the stereo images, leading to a mismatch of stereo effects and visual system (figure 2).

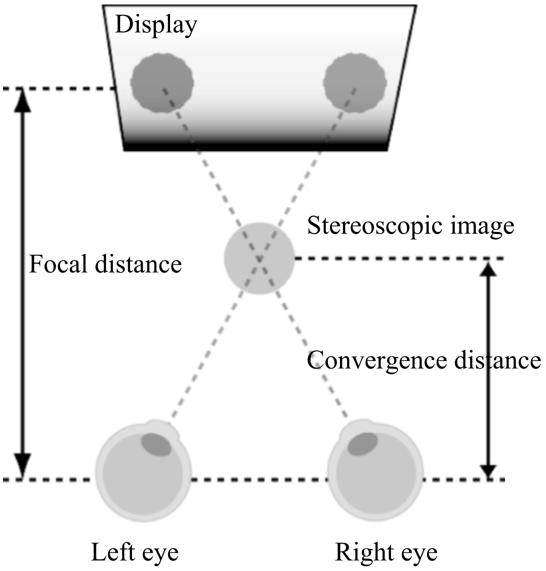


Figure 2: Disagreement of distance information.

The authors believe that the different information about distance given by convergence and accommodation or an unnatural visual environment makes the viewer uneasy and causes visual burden at least partly. To alleviate the visual burden, convergence is adjusted by changing the optical axes of RSVS cameras to provide a visual environment in which the distance of convergence and accommodation agree with each other relative to the object being viewed.

3.3 Shooting with Convergence and Accommodation

The cross point in stereovision shooting is the intersection of optical axes of the right and left cameras. The object with the cross point set is imaged at the centre of the display. In reality, there is more than one object; there are normally several objects being shot. When the observer's eyes move from the object with the set cross point to another without the cross point, the difference between the convergence and accommodation makes the observer feel fatigued. With a twin-lens camera system, there is a locus on which the difference of pixels between two images taken by the cameras is zero next to the cross point. According to geometric observations, the locus is a circle passing the centres of the camera lenses and the object of vision (or three points in all). This circle is called a horopter [2][3]. Using the concept of the horopter, the object point referred to in this paper is defined as the point where the disparity between the right and left cameras is minimal, excluding the cross point (intersection of optical axes of both cameras). Using this definition, we propose a stereovision shooting method with reduced visual burden.

3.4 Amount of Camera Motion for Object Point and Cross Point

The amount of camera motion to catch the object of vision changed using object points can be smaller than when using the cross point alone. Reduction in the amount of motion of stereo cameras may reduce the adverse effect of the changing background on the observer's eyes. The difference of the amount of camera motion between shooting a changing object of vision using the cross point and shooting it using object points is described below.

In figures 3 and 4, assume the object of vision is aligned to P and P' and then to Q (cross point) and Q' (object point). Points A, B and P is the same as points A', B' and P', respectively. Circles A, B and Q are identical to Circles A, B, and Q'. Circles A, B and P are identical to Circles A', B', and P'.

In figure 3, the following equation holds:

$$\theta_\gamma = \theta_\alpha + \theta_b = \theta_a + \theta_\beta \qquad (1)$$

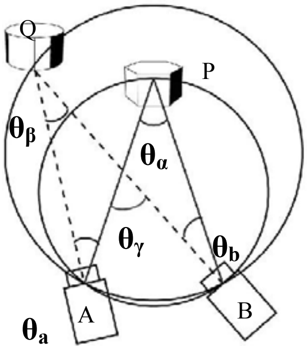


Figure 3: Shooting at cross points.

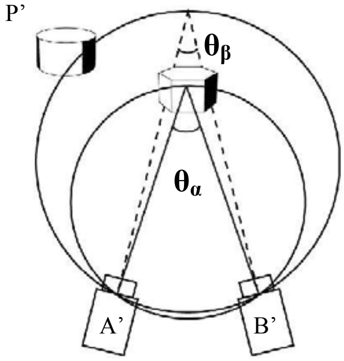


Figure 4: Shooting at object points.

The amount of camera motion from P to Q using the cross point is given by the following equation using (1):

$$\theta_a + \theta_b = \theta_\alpha - \theta_\beta + 2\theta_b \qquad (2)$$

The amount of motion (moving the object of vision from P' to Q') when using object points as shown in figure 4 is θ_α- θ_β, or is smaller by $2\theta_b$ from (2).

3.5 Vertical Disparity

Human eyes are set approximately 6 cm apart horizontally and almost zero vertically. This means that generally no disparity occurs vertically. The allowable range is therefore small and some reports state that stereovision is difficult to obtain with as small a gap as approximately 0.5° [1].

With the RSVS system with horizontally arranged cameras, vertical disparity could exceed the allowable range when the image is enlarged by zooming due to installation error.

4 Remote Control System

Referring to visual burden discussed in the preceding section, the authors have developed a system in which the viewer can remotely adjust the vergence and vertical disparity of the stereo cameras to reduce vertical disparity and adequately set the convergence distance and focal distance. This system has with three harmonic drive motors to adjust the direction of the cameras. With this new camera adjusting mechanism, the motors are installed from below the right and left cameras to enable them to move horizontally. The smallest adjustable distance between the cameras is 6.8 cm. The right camera is adjustable vertically to avoid the problem of vertical motion of the cameras when zooming. The cameras can be moved vertically by turning the aluminium disc after shifting its centre. It is also possible to follow and shoot a moving object.

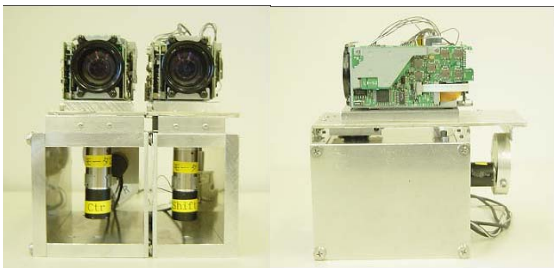


Figure 5: Camera adjustment mechanism.

5 A New Shooting Method

The visual burden caused by using the cross method for shooting stereo images has been reduced by employing the remote control system introduced in the preceding section. The other problem of the cross method is distortion in space perception, which is also a factor of visual burden. The move-parallel method is a new stereo vision shooting method designed to eliminate distortion in space perception.

5.1 Principle of the Move-Parallel Method

The principle of the move-parallel method is described below. In figure 6, when shooting objects A and B which are equidistant from the cameras, the optical axes of the cameras pass the centres of the cameras O1 and O2. Images a1 and b1 of object A are on the line connecting object A and the camera centres. Similarly, images a2 and b2 of object B are on the line connecting object B and the camera centres. Since objects A and B are equidistant from the cameras and the right and left cameras are arrayed in parallel, ΔABO1 and Δa1b1O1 are similar. In like manner, ΔABO2 and Δa2b2O2 are similar. The segment l on the straight line connecting two points of objects A and B is perpendicular to the optical axes of the right and left cameras. Assume segment m connects the centres of the two camera lenses and segment n connects the CCD planes of the cameras, then these three segments are parallel to each other, or l // m // n. Since ΔABO1~Δa1b1O1~Δa2b2O2, the ratio of all corresponding sides is identical. Hence, AB:a1b1 = AB:a2b2. Since the scale factors are identical, the lengths of segments a1b1 and a2b2 are equal. The objects A and B have no disparity and the viewer sees the objects as being equally distant.

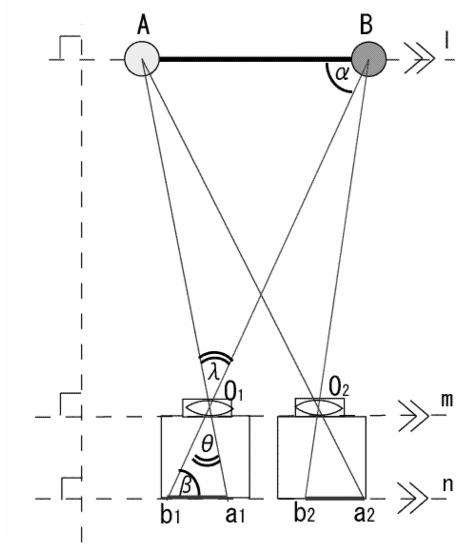


Figure 6: Principle of move-parallel method.

5.2 Merits of the Move-Parallel Method

Merits of the move-parallel method are described below referring to figures 7 and 8.

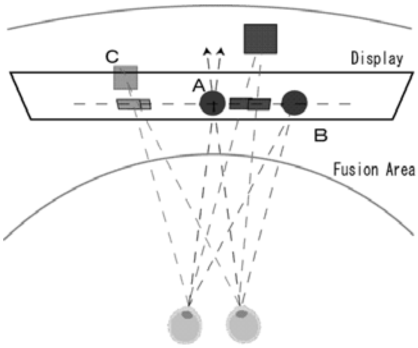


Figure 7: Shooting using the cross method.

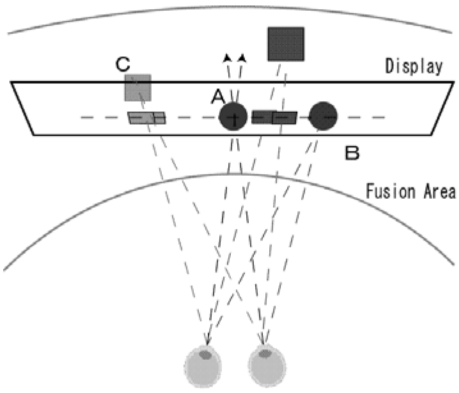


Figure 8: Shooting using the move-parallel method.

Since the cross method uses the cross point, distortion of space perception due to the horopter, as described in the preceding section, is unavoidable. Object B on the hooter is closer to the viewer than object A on the cross point. From the positional relationship of coupling, object B sees that the object A and the viewer are equidistant from the viewpoint of disparity. For C, in contrast, distortion is generated because disparity plays a role on the screen. When using the move-parallel method for shooting, the cross point is not used and the locus of the same disparity can be drawn as a straight line starting at the fixation point. The actual positional relationship and the positional relationship of coupling are equal to obtain the correct sense of distance.

5.3 Animation Shooting Software

Currently available RSVS Versions 1 and 2 are adequate to achieve remote stereovision. The present paper proposes the move-parallel method because the purpose of the present study is the development of stereo images with a low visual burden. The authors have developed a software programme to record and replay the experiments conducted using the parallel, cross and move-parallel methods to verify the effectiveness of the move-parallel method in reducing the visual burden more than other methods through experiments with animation. The new software programme allows the user to smoothly shoot stereo images and conduct the required experiments.

6 Experiments Shooting with Stereo Cameras

Animation experiments were conducted on the move-parallel method introduced in the preceding section. Near and medium-distance perspectives were prepared for the experiments. The cross method and move-parallel method were compared for each distance perspective. By comparing the two methods, the authors have verified the usefulness of the move-parallel shooting method for the respective distance perspectives and distortion in space perception.

6.1 Outline of the Experiments

The distance between the cameras and the object was nearly identical for the two shooting methods.

The motion of the object was also nearly the same. The near and medium-distance areas were all white or black to prevent diversion of the viewer's attention from the object being viewed.

Images shot using the move-parallel method have a dark spot on the right-hand side of each image after translation. The dark spot was added intentionally to the images using the cross method to hide the difference between the two types of images.

The subjects in the experiments were not told which method was used for the respective images.

Fifty students in their early 20s participated in the experiment. After the experiment, they were asked to evaluate the ease of viewing the images, stereoscopic effects and the level of fatigue in five ranks. They briefly reported what they thought about the images and selected the best image taking all aspects into consideration from their own points of view.

6.2 Experiment 1: Verification of Near-Distance Shooting

Purpose: The purpose of the experiment was to verify which caused visual burden: the cross method (with distortion in the near-distance area where disparity tends to occur easily) and move-parallel method (no distortion).

Outline of the Experiment:

- Disparity is set on the object
- The object is placed 30 cm from the cameras
- The object does not move but turns on a turn-table

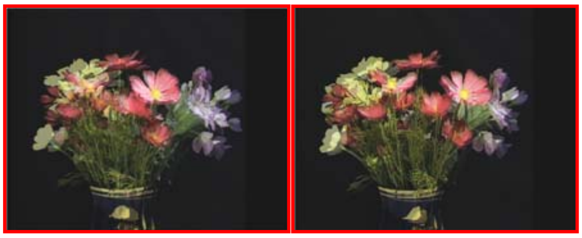


Figure 9: Taking a picture of near-distance (left: cross method; right: move-parallel method).

Table 1: Results of questionnaire on near-distance comparison.

Near-distance	Move-parallel method	Cross method
Stereoscopic effect	3.85	3.65
Fatigue	3.25	2.5
Ease of viewing	3.95	3.35
Overall evaluation	38 (people)	12 (people)

Table 1 shows the results of the experiment. For the near-distance shooting, three times more subjects selected the move-parallel method over the cross method. Both the cross and move-parallel method provide stereoscopic effects but the move-parallel method was preferred to the cross method considering fatigue and ease of use as far as near-distance observation was concerned. Although this is because no distortion due to space perception was generated, it was known from this experiment that absence of distortion provides people with ease of viewing and less fatigue in terms of the visual burden. This indicates that the move-parallel method is fully useful in near-distance shooting of animation.

6.3 Experiment 2: Verification and Change in Disparity in Medium-Distance Shooting

Purpose: The purpose of this experiment was to verify the behaviour of visual burden caused by an increase or decrease of disparity as the object being viewed moved back and forth with disparity set to the medium distance.

Outline of the Experiment:

- The object was placed 1 m from the cameras
- The object was moved towards and away from the cameras in a range of 30 to 130 cm
- Disparity was maintained at the initial position (1 m) of the subject
- Four blocks were placed to assist in relative distance perspective. The blocks were placed in positions that did not obstruct the travel of the object

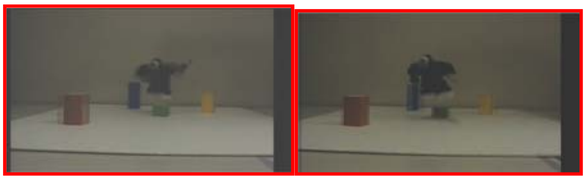


Figure 10: Taking a picture of medium-distance (left: cross method; right: move-parallel method).

Table 2: Results of questionnaire on medium-distance comparison.

Middle-distance	Move-parallel method	Cross method
Stereoscopic effect	3.65	3.85
Fatigue	2.75	3.15
Ease of viewing	3.2	3.35
Overall evaluation	23 (people)	27 (people)

Table 2 shows the results of the experiment. In medium-distance shooting, the cross method was favoured more than the move-parallel method by a small margin. Subjects who were not accustomed to stereovision must have found fusion more difficult than in near-distance shooting because of the increasing magnitude of disparity as the object approached. For medium-distance shooting, the object is located at a somewhat distant place and the angle of convergence is relatively small. For these reasons, the difference caused by distortion was not sensed much, and only a relatively small difference in fatigue and ease of viewing was noted between the two methods. The object moved only back and forth, and lateral motion (in which distortion is easily sensed) was not involved. This is another reason why not much difference was felt between the two methods. The authors believe that these are the reasons why the differences were fairly insignificant.

7 Conclusions

Results of our experiments show that the move-parallel method was generally more highly evaluated for shooting near-distant images. We also found that the reactions to both methods were not very different for shooting images at medium-distance. The cross method has been considered a good shooting method because the focal distance and convergence distance are identical and the visual burden on the observer can be reduced. Our experiments have proved that the move-parallel method is as good as the cross-point method also for shooting animation at medium-distance. For these reasons, the move-parallel method can reduce the visual burden and is a better method for obtaining the correct distance perspective. Our experiments have also shown that ease of viewing and fatigue in near-distance stereovision were improved because distortion was eliminated. Based on the above findings, the move-parallel method is useful for near-distance stereovision and also for shooting stereovision images of animation as it received an evaluation similar to the cross method for medium-distance stereovision.

8 References

[1] H. von. Helmholtz, *Treatise on Physiological Optics*, trans. J.P.C. Southall, Optical Society of America, New York (1925).

[2] A. Grieco, G. Nolteni, B. Piccoli and E. Occhipinti, "The vertical horopter and the angle of view", *Work With Display Units 94*, pp 131-136 (1995).

[3] M. Ohkura, "Measurement of horopter and alleys in auditory space", *Proceedings 2002 International Conference on Auditory Display*, Kyoto, Japan, pp 333-341, (2002).

[4] H. Harashima, *Science of 3-Dimensional Pictures and Human Beings*, Ohmsha Publishing, Tokyo (in Japanese) (2000).

[5] *Letter Symbols for Quantities*, ANSI Standard Y10.5 (1968).

[6] S. Tachi, "Fusing virtual reality, robotics and networks together", *Proceedings 14th International Symposium on Measurement and Control in Robotics: The Changing World*, Houston, TX (2004).

[7] H. Kim, K. Nishikawa and J. Takeno, "Research on convergence control and reduction of eye fatigue", *Proceedings International Conference on Communication and Control Technologies (CCCT2004), The International Institute of Informatics and Systemics (IIIS),* Austin, TX, 2, pp 401-406, (2004).

[8] M. Mizukami, H. Kim and J. Takeno, "Acquisition of accurate distance information in stereovision using move-parallel method for teleoperation", *Proceedings IEEE International Conference on Mechatronics and Automation*, Niagara Falls, Canada, pp 1197-1202 (2005).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Extraction of Association and Phrases Information from the Internet and Creation of a Knowledge Database

Yukihiro Kato, Yukinobu Imai, Takashi Kanamori, Katsuya Furuike and Junichi Takeno
Robot and Science Laboratory, Department of Computer Science,
Meiji University, Kawasaki, Japan
ce66709@isc.meiji.ac.jp, ce44705@isc.meiji.ac.jp, ce55704@isc.meiji.ac.jp, ee27122@isc.meiji.ac.jp,
juntakeno@aol.com

Abstract

It is important for robots to have stream of consciousness and be capable of expressing feelings in order to communicate with humans. The present paper proposes an association and phrases network that serves as the information source for the robot to feel and have stream of consciousness. A method to create a knowledge database using text data collected from the internet is discussed. This knowledge database handles both the association of words and kansei (comprehensive mental images), allowing the robot to make human-like associations. Experiments were carried out to verify the appropriateness of the associations and mental images in the knowledge database. Both associations and mental images were found to be close to those of humans. The database is an effective memory model capable of supporting associations and mental images for incorporation into robots.

Keywords: association and phrases network, association word, phrases, kansei, internet

1 Introduction

In this information-oriented society, computers have been evolving to have higher-speed processing, greater accuracy and increased memory capacity. Therefore, many innovative robot technologies have been developed in recent years and are now used in daily life. Robots must be able to communicate clearly with humans to co-exist with them in the future. By doing this, robots can more easily communicate and collaborate with humans in many situations, such as in care or life-saving efforts. For a robot to be capable of communicating with humans, it must be able to recognise, judge and take action, as well as express feelings. However, it is difficult to realise this with existing artificial intelligence systems. Therefore, human stream of consciousness must be reproduced on a computer. Human stream of consciousness uses memory as its information source and is inspired by associations and inside or outside stimulation. As the first step in reproducing stream of consciousness, we try to create a database which provides a dependable association memory system, without considering stimulation. For this purpose, we propose the *association and phrases network*. This network consists of words associated with each other and mental-image-expressing factors. We term these words *association words*, the mental images *kansei* and the mental-image-expressing factors *phrases*. This is a comprehensive memory model where association of words is networked and then connected to feelings. This study positions itself as a memory system forming the basis for creating the capability to communicate with a robot in this way. On the basis of this study, we want to build a system which will perform a stream of consciousness and communicate with a language by feelings.

A key feature of our present study is the use of text data collected from the internet to create the database. Another feature is the inclusion of both knowledge and mental images in the database and their being treated together. Therefore, feelings that could be expressed by a robot are generated from knowledge and mental images. Very few studies take such an approach. Still another important feature is the individual handling of multiple concepts inherent in each word.

Fairly similar research is being performed at Doshisha University's Graduate School of Engineering in a project called the Commonsense Consciousness Judgment System. The theme of the project is not exactly consciousness per se, but for a given input, a database is searched for a similar concept to generate an image, which makes this a kind of expert system [1]. The primary objective of that research is to have a robot possess a degree of “common sense” and it is not particularly concerned with developing communication capability and constructing consciousness. The KwMap website, a search site on the internet, collects related words using the principle of co-occurrence. The site's method and results for collecting related words (nearly synonymous with what we term association words) are very similar to our approach as described in this paper. However, the study on the KwMap website is not relevant to feelings. Co-occurrence is discussed in section 3.

In the present paper, we discussed a model and method for creating the association and phrases network. We also describe a model of stream of consciousness which can be caused by this memory system in near future.

2 Modelling of Knowledge and Consciousness

This section presents the definitions and characteristics of the association and phrases network, which is modelled after the knowledge-governing capabilities of humans. We also discuss the consciousness network, representing consciousness.

2.1 The Association and Phrases Network

The association and phrases network comprises certain keywords and the words associated with them (association words), connected by directed edges to form a network. Each edge has a certain strength of connection or ease of association (association value). The association and phrases network also includes kansei-expressing words (phrases). The association words and phrases constitute the nodes in the network. Associations differ from person to person and the network changes by experience and learning, but in the association and phrases network created in this study, the associations and mental images are general and average, but are hard to form under a peculiar case in a property of the creating method. Figure 1 shows an example of an association and phrases network. The round nodes are the various association words and the square nodes are the phrases. These are bound together in directed edges, and a node pointed at can be associated with a node at the root of directed edge. In this example, "pet" and "wolf" can be associated with "dog," and "wolf" has "fear" as a mental image. Mental-image-expressing words such as "fear" and "dear" are *phrases* [2][3][4].

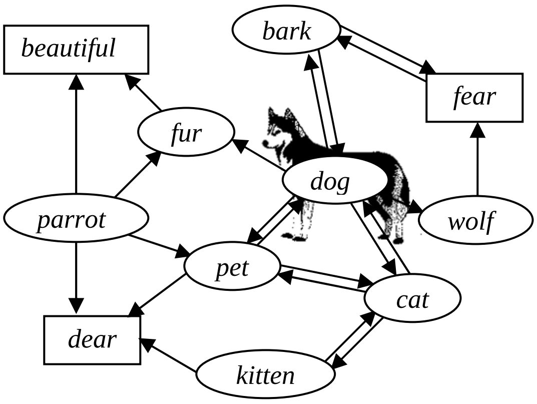


Figure 1: An example of an association and phrases network.

2.2 Consciousness Network

In order to develop robots that think like humans, it is necessary to model the human stream of consciousness. Stream of consciousness, which refers to continuously changing consciousness, involves some related factors, e.g. external and internal bodily stimuli such as voice, visual image and hunger, which are the internal energy of consciousness itself that has hitherto built up the stream of consciousness.

Stream of consciousness can be modelled as follows: upon perceiving an internal or external stimulation, the core of consciousness and related concepts are extracted from the association and phrases network as a partial graph. This is the consciousness network. The core of the consciousness network moves to other concepts as they may occur due to desire or association, on receiving new stimulation or the internal energy of consciousness itself. This causes the consciousness network to change. The continuation of this series of events is called the stream of consciousness. For example, one thinks of "dog" upon viewing an image of dogs. The next moment, one may think of "pet" as the association word for "dog." This means that consciousness has moved from the node "dog" to the latent node "pet" existent in the subconscious on the association and phrases network. The process by which the subject of consciousness changes with time is called stream of consciousness [2][3].

3 Knowledge Database

We have created the association and phrases network as a knowledge database. This section describes association words, association value, phrases and kansei value and our method of creating a database using the internet. Meaning classification of nodes, a recent development by the authors, is also outlined.

3.1 Association Word and Association Value

The association and phrases network consists of words associated with certain keywords that are connected by edges as described above. Words associated with other words are called association words. For example, in figure 1, association words of "dog" are "pet," "cat," "wolf," "bark" and "fur." The quantified strength of the connection between association words is called the association value.

3.2 Phrases and Kansei Value

A key feature of this study is that kansei, the basis for feeling, is analysed together with word groups. Kansei is the mental image of a given word. One may think of "joy" for the word "peace", or "fear" for "war." We call the adjectives expressing such mental images *phrases*. Phrases are used to quantify and extract images. Typical examples include "good", "beautiful", "fear" and "dangerous." These phrases are designated freely and the database user specifies which adjective is relevant to which kansei [5]. The kansei value indicates a word's level of kansei on a 0 to 1 range.

3.3 Creation of Knowledge Database Using the Internet

This section describes the creation of a knowledge database and the extraction of association words, association value and kansei value. We used the internet because it is the world's largest data repository and contains an incomparable amount of information. Website content is renewed almost daily by users throughout the world and encompasses a wide range of subject matter from the latest news to various other kinds of information. It is intended to create the database which installs various human associations and kansei gathered using information from the internet.

3.3.1 Flow of Database Creation

The text data used to construct the knowledge database in this study was extracted from the internet. The actual work procedure involved browsing several websites and picking up text data from their pages. Any available links to external sites were also visited to pick up their text data. This was repeated until a sufficient amount of text data was collected. The collected data was analysed to find the number of appearances of key nodes and of other nodes appearing in the same sentences as the key nodes. These nodes were then added to the database as association words.

3.3.2 Extraction of Association Words and Association Value

To find association words for a given word, the relevance between the given word and other words must be known. In this study, association words are extracted based on the principle of co-occurrence. Co-occurrence means that a group of certain words appear in the same sentence or context simultaneously. The more relevant any two words are, the closer they appear to each other and the more frequently they appear. Words with co-occurrence are considered the most appropriate association words. Therefore, co-occurrence of words is used for extracting association words, but semantic analysis, morphological analysis and text-mining are not used in this way. Co-occurrence was confirmed on the text data obtained from the internet.

Association value is the probability of any two or more words appearing in the same sentence simultaneously by reason of co-occurrence. Specifically, the total number of simultaneous appearances of individual words and the number of simultaneous appearances of a group of certain words are stored in the database. This information is used to extract association words and association value. The association value $p(a_i)$ of an association word a_i of the core word x is defined by the equation below:

$$p(a_i) = \frac{c(a_i)}{c(x)} \tag{1}$$

where $c(x)$ = appearances of x and $c(a_i)$ = appearances of x and a_i simultaneously.

3.3.3 Extraction of Kansei Value

Like the association value, the kansei value is determined by the frequency of the appearance of phrases in the same sentence as the core word. The kansei value is calculated in two steps. First, by assuming $c(x)$ is the total number of appearances of x and $c(k)$ the number of phrases appearing simultaneously with x, we obtain the equation below. We define this as the pre-kansei value $R_p(x)$ for descriptive purposes.

$$R_p(x) = \frac{c(k)}{c(x)} \tag{2}$$

The pre-kansei value is greater than or equal to zero, but can exceed 1 if many phrases exist. For convenience, the maximum for the kansei value is set to unity and the kansei value is normalised in the 0 to 1 range. After obtaining the pre-kansei values for all nodes, the highest value is defined as $R_{\max}$. Divide the pre-kansei value $R_p(x)$ of n a given node x by $R_{\max}$, and the result is the kansei value $R(x)$:

$$R(x) = \frac{R_p(x)}{R_{\max}}. \tag{3}$$

3.4 Meaning Classification

Meaning classification is a procedure for handling multisense words (similar words with multiple meanings) as individual nodes of a different meaning in the knowledge database. This allows the creation of a database with a one-image-one-node structure. For example, "book" means both a work of literature and the act of making a reservation; "virus" means a disease-causing organism as well as a harmful and contagious computer programme. The single word "virus" gives quite different images and meanings as it is used in biochemistry and computing contexts, although the root meaning is the same. Meaning classification of a node means grouping of the association words of the node by meaning. The original node is split by the derived meaning groups.

A hierarchical clustering method is used for meaning classification. Words associated from a given node are called *primary association words*; words associated from the primary association words are called the *secondary association words* of a given node. Primary association words with secondary association words of similar content are considered to have similar meaning. Primary association words with similar meaning are put in groups with the same meaning in the order of their level of similarity. These meaning groups are used to classify the nodes by

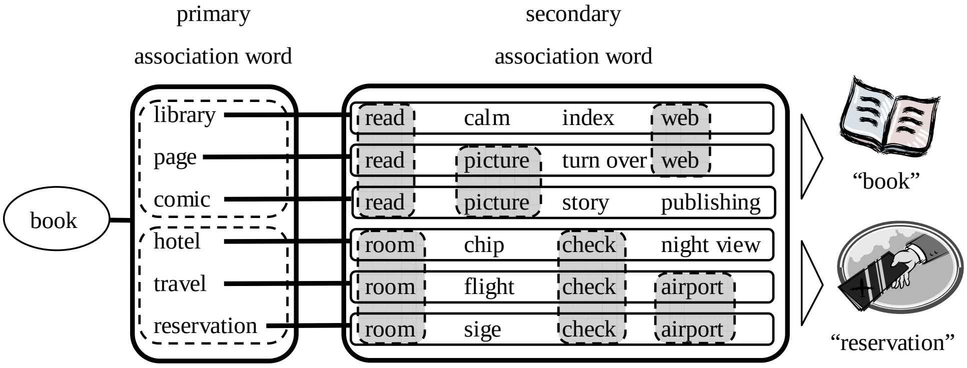


Figure2: An example of meaning classification.

meaning. An example of meaning classifications is shown in figure 2. Comparing each set of secondary association words associated from each of the primary association words of "book" gives us the level of similarity. This is a criterion of meaning classification. "Library", "page" and "comic" belong to the same meaning group. "Hotel", "travel" and "reservation" belong to another meaning group. The original node "book" is split into each meaning.

In addition, clustering methods are hierarchical clustering and non-hierarchical clustering as a major k-means method. Non-hierarchical clustering methods are not adopted in this experiment, because the k-means method needs to clarify the number of clusters and it may be difficult to generate proper meaning groups.

4 Experiments

As explained above, although these techniques are very simple, but we found them effective. For our knowledge database, three experiments were carried out to confirm the efficiency of our technique:

1. matching with dictionaries to find out how many nodes existed in the database;
2. questionnaires to learn how close the association and kansei of the database was to human interpretations; and
3. verification of the appropriateness of our technique by performing experiments on meaning classification.

4.1 Comparison of Database with Dictionaries

We browsed the internet for about two weeks to extract text data from 1,280,000 websites and successfully maintained about 500,000 nodes in the database. We compared the database with a dictionary to survey how much vocabulary the database had. For this comparison we used Eijiro, a dictionary of English that includes 300,000 words. The results were as follows.

Our database contains 1,867 (97.9%) of the 1,906 basic English words found in Eijiro. This indicates that the database is effective for practical use, except in special cases, because 90% of common English sentences are composed of basic English words.

We further compared our database with an English dictionary containing more than 310,000 words. About 150,000 (48.57%) words matched. One reason the matching rate is not very high is because many words appearing in the dictionary are not actually used on the internet. Another possible reason is that sufficient time was not taken to extract the text data. Viewed from another perspective, the matching of 150,000 words means that 350,000 words or concepts not covered by the dictionary were collected, including important words or concepts. Additionally, it is possible that new words and expressions not yet appearing in the dictionary were able to be installed in our database. This is an important characteristic of our study. By taking more time in future, we will extract nodes which are not in the database but are in the dictionary, to create a more reliable database.

4.2 Comparison of Association and Kansei between Our Database and Humans

Two questionnaires were used to learn how close the associations and kansei in our database were to human interpretations. The questionnaire was given to 85 people (all Japanese).

4.2.1 Questionnaire on Association Words

The first questionnaire was used to check the reliability of the association words in the database. Twelve general words in extensive use in a variety of fields were selected from the nodes in the database. The association words for the selected words were presented to the questionnaire participants who were asked to evaluate the association words by selecting one of three reply options: "natural", "unnatural" and "not sure".

Table 1: Results of questionnaire on association words and five association words of selected node.

	card	loan	bike	computer	food	taxi	bomb	rain	moon	meat	pork	sashimi	average ratio
Natural	33	75	33	80	39	30	46	60	29	68	66	84	0.630
Not sure	33	8	33	5	31	30	30	18	40	14	15	0	0.251
Unnatural	19	2	19	0	15	25	9	7	16	3	4	1	0.117

Node	**Association words**
loan	home, payday, mortgage, credit, cash
computer	system, software, reservation, internet, information
rain	forest, snow, heavy, showers, water

Node	**Association words**
card	business, extreme, online, phone, poker
bike	ride, car, exercise, shop, pocket
taxi	airport, service, transportation, hotel, limousine

Table 2: Results of questionnaire on kansei value. Kansei value of “smile” and “virus”.

		happiness	disgust	fear	sadness	anger
smile	questionnaires	**6.929**	1.318	1.647	1.871	1.012
	Database	**7.389**	4.896	3.063	4.864	2.140
bomb	questionnaires	1.116	5.188	**6.471**	5.459	5.376
	Database	2.718	7.439	**8.192**	7.838	4.382
virus	questionnaires	1.176	5.271	**6.118**	3.553	3.435
	Database	2.130	**5.852**	5.299	1.497	2.760

Table 3: Partial results of meaning classification.

party	Meaning group 1	（wedding, favours, gift, bridal, baby, Halloween, theme, shower）
	Meaning group 2	（democratic, democrats, election, republicans, leader, political, law）
book	Meaning group 1	（books, write, wrote, written, author, title, library, reading, published）
	Meaning group 2	（hotel, guide, travel, reservation, vacation, store, buy, club, room）
run	Meaning group 1	（kick, failed, rush, game, pass, running, golf, gambling）
	Meaning group 2	（business, company, office, site, system, money, president）
log	Meaning group 1	（message, web, access, blog, data, review, book, read, page）
	Meaning group 2	（house, home, mountain, cabin, cedar, property, construction）

The questionnaire results (upper table) and some association words of the nodes in the database (lower table) are shown in table 1.

As the average ratios in table 1 show, “natural” was selected by about 60% of the respondents, in contrast to “unnatural”, selected by about 10%. This indicates that the overall association was generally natural. Breaking down the findings further to individual words, two groups of words exist. The first group of words were rated “natural” by nearly all respondents (e.g. “loan”, “computer”, “rain”). The words in the other group (e.g. “card”, “bike”, “taxi”) received almost even numbers of the three possible responses. In the first group, the association is definitely “natural”. In the latter group, the disparities were due to the fact that people make different associations. The “unnatural” response was not concentrated on any single word. This indicates that overall the database is effective and reliable for practical use so far as the association words are concerned.

4.2.2 Questionnaire on Kansei Value

As with the association words, the kansei value was examined for its reliability using a questionnaire. We defined the kansei value for five words: “joy”, “abhorrence”, “terror”, “sorrow” and “anger” [5]. Under the same conditions as used for the association words, ten nodes were selected (e.g. “liar”, “bomb”, “smile”, “hospital”, “virus”, “fair”, “kidnap”, “dead”, “family”, “curse”) and participants were asked to rate the level of kansei in each of the nodes by assigning a point value between 0 and 10. The average of the questionnaire results is compared with the kansei value of the database multiplied by 10 (to use the same scale).

As the results in table 2 show, the kansei value that obtained the highest mark in the survey matched the database in nine out of 10 nodes. Humans have multiple feelings simultaneously only in complicated circumstances. Humans usually have only a single feeling, i.e. the kansei of the maximum value. The fact that the kansei values matched fairly between the database and the questionnaire responses is quite important and meaningful. The word “virus” was the only exception, but this difference is very small and the five kansei values were close to one another. In this way it became clear that kansei of a database is near to a human sense. In this light, the kansei values in the database are certainly an effective means to help a robot express human-like feelings.

4.3 Meaning Classification of Multisense Words

The meaning classification procedure was applied to twelve multisense words to verify the appropriateness of the meaning classification method we had devised. Table 3 shows part of the content of the generated meaning groups for three of the words.

There are many association words of which these are just a few. According to the results, our meaning groups have a certain tendency of similar reason meaning, indicating that our approach to meaning classification is successful. However, there are discrepancies that degrade the overall results. The derived meaning groups do not necessarily match the meanings of the individual words.

The results of our experiments indicate that meaning classification using hierarchical clustering is practical to some extent. It is necessary to continue our research to improve accuracy.

5 Future Tasks and Prospects

The results of our experiments indicate that our knowledge database is effective to a certain extent with regard to the number of nodes, association words and kansei in the database. It is particularly noteworthy that these factors were found to be natural or similar to those of humans according to our purpose. This verifies the appropriateness of the association and phrases network. Based on this, the next step is building a consciousness network which forms stream of consciousness while considering stimulation.

The text data used in this study was extracted from the internet. Its effectiveness was demonstrated to some extent by experiments, but no objective comparisons have been made. We will create another association and phrases network using text data extracted from information sources other than the internet for the sake of comparison. The procedure for selecting and extracting text data for comparison will be studied in the future.

There are two types of associations. One is semantic association, while the other uses vocabulary such as spelling and vocal sound. This study aimed ot create a network based on semantic associations. We will also study the development and practical use of a network based on vocabulary. Phonetic signs and spelling will be analysed and vocabulary-based association values will be defined based on similarities.

The meaning classification technique will be elaborated to make it more accurate. Practical application of the meaning classification technique to the nodes of the database will be studied. We conducted the meaning classification procedure but did not split the original node based on the results of the classification. We will study the calculation of association values of nodes that are split by meaning classification. Specifically, a method for using a vocabulary network or other networks will be studied.

6 Summary

Our study proposes the *association and phrases network*, a memory model that handles associations and kansei simultaneously to create a stream of consciousness. Our database with the embedded association and phrases network (memory model) has the flexibility to respond to current usage because it uses the internet. The database has achieved human-like associations and kansei. That we could achieve a result by using such a simple method shows effectiveness and originality. Using this method we are now able to generate a stream of consciousness from the consciousness network. We think that robots will be able to possess the capability to judge and to express feelings in the near future. This may eventually lead to the development of a robot capable of interactive communication with humans.

It is important not only that a robot have the ability to make human-like judgment, behaviour and interactive communication, but also that it have a human mind. This study will lead to robot development and improved robot-human-interactions.

7 References

[1] A. Kometani, H. Watabe and T. Kataoka, “Constructing a commonsense consciousness judgment system”, *Proceedings 17th Conference of Japanese Society for Artificial Intelligence*, Niigata, Japan, pp 3C1-07 (2003).

[2] T. Kaneko and J. Takeno, “Information scientific research of funniness and sadness”, *IEEE Int. Workshop on Robot and Human Interactive Communication*, Bordeaux, pp 450-455 (2001).

[3] T. Kanamori, Y. Imai and J. Takeno, “Extraction of 430,000 association words and phrases from internet web sites”, *International Conference on Machine Automation*, Toronto, pp 1929-1934 (2005).

[4] Y. Imai, M. Tamada and J. Takeno, “Method for extracting associated words, association value and kansei value from the internet”, *Proceedings 8th World Multiconference on Systemics, Cybernetics, and Systemics (IIIS)*, Orlando, FL, pp 160-165 (2004).

[5] A. Ogiso, S. Kurokawa, M. Yamanaka, Y. Imai and J. Takeno, “Expression of emotion in robots using flow of artificial consciousness”, *IEEE Int. CIRA2005*, Helsinki, pp 421-426 (2005).

[6] N. Goldblum, *The Brain-Shaped Mind*, Cambridge University Press, Cambridge (2001).

[7] M. Takashi, “How to make a conscious robot (fundamental idea based on passive consciousness model)”, *Journal of the Robotics Society of Japan* 23(1), pp 51-62 (2006).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Design Tools for Rapid Prototyping of Embedded Controllers

Manuel Almeida, Bruno Pimentel, Valery Sklyarov and Iouliia Skliarova
Department of Electronics, Telecommunications and Informatics, IEETA
University of Aveiro, Aveiro, Portugal
manuel.almeida@ieeta.pt, pimentel@ieeta.pt, skl@det.ua.pt, iouliia@det.ua.pt

Abstract

Electronic devices used in the scope of robotics and embedded systems have to be adapted to numerous external events and many of them might be unknown in advance. This application-specific particularity requires environment-specific adaptation and frequent changes in the pre-defined behaviour. In general, all feasible functionality cannot be incorporated in the device during the design phase, i.e. some eventual modifications are postponed until physical tests in real working conditions. Even after executing physical tests, some unexpected events (requiring the appropriate device reaction) might appear. Thus, either sophisticated adaptable electronics has to be implemented or the device has to be able to communicate with another more intelligent host computer, which would assist to cope with emerging problems. The paper suggests tools that provide support for dealing with the considered situations. This is achieved through the following: 1) FPGA-based prototyping core board (reconfigurable platform) establishing both wired and wireless interactions with host computers; 2) Design templates and libraries for interacting with standard peripheral equipment and widely used components for different types of control and computations; 3) Software providing support for interactions with the core board; 4) Intellectual property cores for solving a number of optimisation problems common to many engineering applications.

Keywords: FPGA, prototyping, VHDL templates, design library, reconfigurable systems, remote configuration

1 Introduction

Modern Field Programmable Gate Arrays (FPGA) are composed of programmable logic cells, memories, arithmetical devices, processors, circuits for advanced synchronisation, etc. The majority of the FPGA components can change their functionality and interconnections between the components can arbitrarily be established through reprogramming the relevant chip. This opportunity opens practically unlimited capabilities of FPGA-based systems for rapid prototyping, which is a demanded technique for robots and embedded systems.

Let us summarise the most important basic features of FPGA-based systems:

- The implemented circuit can be optimised for a particular application. This permits to reduce the number of the necessary clock cycles, to execute (as many as required) operations in parallel, to choose the most appropriate device architecture, etc.;
- Although a clock frequency of FPGAs is much lower than a clock frequency of ASICs we can benefit from reconfiguration and practically unlimited prototyping facilities, which, in particular, allow different competitive and alternative implementations and algorithms to be examined and compared;
- Since FPGA-based systems might be configured not only statically but also dynamically, we can construct virtual systems than might require more resources than the resources available on an FPGA chip. Indeed, since dynamic reconfiguration makes it possible to change the functionality of FPGA during run time, we can partition a complex system into subsystems functioning sequentially. As soon as one subsystem has completed the required sub-task, hardware for the subsequent subsystem can be provided through reconfiguration of the same FPGA;
- The design lead time for FPGA-based systems is much shorter than for ASICs.

Numerous advantages of FPGAs make reconfigurable platforms an ideal target for modern embedded systems that combine high computation demands with dynamic task sets [1]. A number of FPGA-based prototyping boards have been manufactured and they enable the designers to verify alternative and competitive engineering solutions. Using such boards simplifies significantly the design of new FPGA-based applications and allows the development lead time to be shortened. Very often we can take full advantage range of hardware capabilities of prototyping boards if the relevant design tools are available, namely design templates, design libraries and intellectual property (IP) cores.

Note that a large number of available FPGA-based prototyping systems makes it difficult to find the best choice for a particular application and, as a rule, it is necessary to find a compromise between the required hardware/software resources and the price. Taking into account the fact that the majority of prototyping boards/systems include many typical components (such as memories, liquid crystal displays - LCDs, standard interfaces, etc.) it is very difficult to find a board optimally targeted to the particular application, i.e. such a board that contains only those elements that are required for a particular design problem and no other components, which just increase the cost and occupy the space. The paper suggests a technique permitting to overcome this problem, which has been achieved through the design of an extendable set of hardware/software tools easily retargeted to different engineering application areas. Any particular problem can be solved using just the selected subset from the considered set, which includes only the needed hardware/software components and excludes all the other available components. In case if the desired components are not available they can be constructed easily.

In general, the suggested tools possess the following distinctive features:

- The core FPGA can be configured using wired (USB) and, in future, wireless (Bluetooth) interfaces, which makes the prototyping system ideal for remote applications, such as that are needed for robotics and embedded systems;
- The developed software/hardware components provide support for both dynamic onboard reconfiguration and remote wireless reconfiguration and/or interaction;
- The design process is supported by various supplied tools, such as hardware description language (VHDL, in particular) templates, design libraries and IP cores. Some of them are targeted to the remote control and the reconfigurability.

The remainder of this paper is organised in six sections. Section 2 describes FPGA-based embedded controllers. Section 3 considers the developed FPGA-based prototyping system. Section 4 shows how remote interactions with the board can be established. Sections 5, 6 present the developed software tools, hardware and language templates, IP cores and design libraries. The conclusion is in section 7.

2 FPGA-based Embedded Controllers

The main objective of the proposed tools is illustrated in figures 1 and 2. We would like to divide the design process of an embedded controller into two stages. The first stage is verification and debugging of the developed circuit at different levels of abstraction, namely:

- Simulation in computer using general purpose and application-specific software;
- Hardware/software co-simulation [2] in such a way that the developed FPGA-based controller interacts with virtual sensors and actuators displayed on PC monitor screen (see figure 1) and their activity (much like the activity of analogous sensors and actuators in physical systems) is supported by the relevant software models.
- Using hardware libraries that enable the designers to communicate with typical peripheral devices (see the right-hand part of figure 1). This is very helpful for debugging purposes.

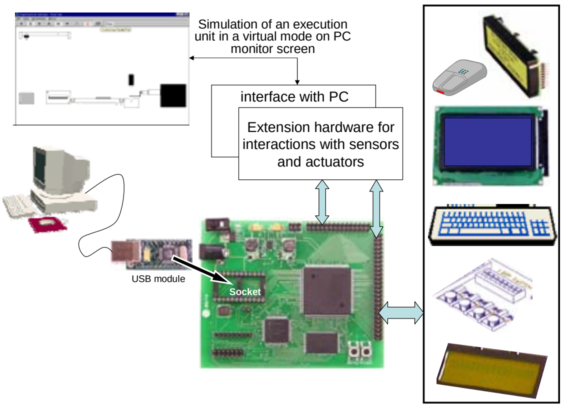

Simulation of an execution unit in a virtual mode on PC monitor screen
interface with PC
Extension hardware for interactions with sensors and actuators
USB module
Socket

Figure 1: Using wired interfaces for testing and debugging purposes.

After the controller has been tested it can be connected to the proper physical system (see figure 2). At this stage we can use the same prototyping board replacing the USB interface block with a Bluetooth wireless interface block, which can be inserted in the same socket. After that we still have support for debugging through remote interactions. This permits many useful functions to be implemented, such as:

- Reading and verifying the controller's states;
- Remote reconfiguration;
- Intellectual assistance from the host computer, which possesses more powerful hardware/ software resources;
- Hardware support for virtual capabilities allowing the controller to be constructed on an FPGA that does not have sufficient hardware resources to accommodate all the required functionality, etc.

In order to implement the considered technique many different software/hardware components have been developed, namely:

- Drivers for USB/Bluetooth interfaces;
- Software supporting functions, illustrated in figures 1 and 2;

- Design templates permitting dynamically reconfigurable circuits to be constructed;
- Design libraries to support interfaces with typical peripheral devices (see figure 1);
- IP cores for solving numerous optimisation problems formulated over binary and ternary matrices;
- Hardware/software tools for data compression and decompression.

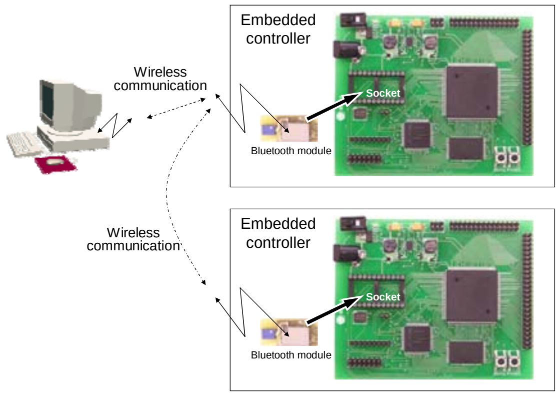


Figure 2: Using wireless interfaces.

3 FPGA-based Prototyping System

The basic architecture of the developed prototyping system (see figure 3) is organised in such a way that it permits to provide the following features:

- Powering and programming the board from PC through USB port. If necessary an external power source can also be used;
- Keeping *bitstreams* for the FPGA in a flash memory, which permits to use the board as an autonomous device without any connection to PC and only external powering has to be provided;
- Keeping more than one *bitstream* in the flash memory for dynamic reconfiguration of FPGA. The capacity of the selected flash memory permits to store up to 8 *bitstreams*. This is very practical not only for run-time reconfiguration but also for verification of different types of alternative and competitive implementations;
- User-friendly software interface for programming the board and data exchange with PC;
- Data exchange with any other device supporting standard USB port;
- Extension connectors for interacting with application-specific externally connected devices.

In general, this architecture presents further improvements over the previously designed prototyping system [3].

The flash memory is divided into three logical sections, as shown in figure 4. The first section contains a *bitstream* that has to be pre-loaded to FPGA in order to allow the following set of operations: 1) transferring an application-specific *bitstream* to the second section; 2) erasing flash memory sectors; 3) transferring data from a host device to the third section of the flash memory and vice versa. This technique has already been used in Trenz prototyping boards [4]. The second logical section is used to store an application-specific *bitstream* for subsequent quick loading into the FPGA (using the "project" pushbutton available on the board). The third memory section enables the designer to store additional *bitstreams* for configuring the FPGA or any arbitrary data such as bitmaps for a VGA monitor.

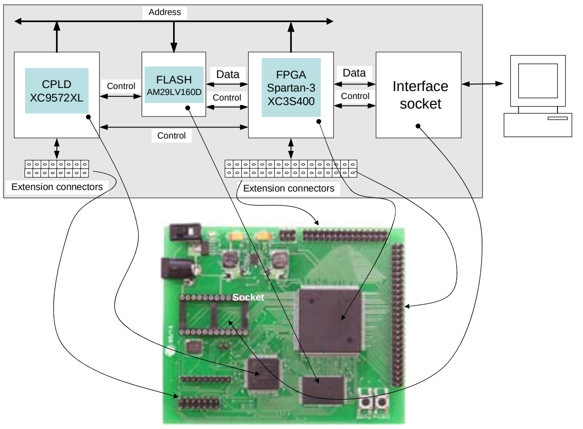


Figure 3: Basic architecture of the developed prototyping system.

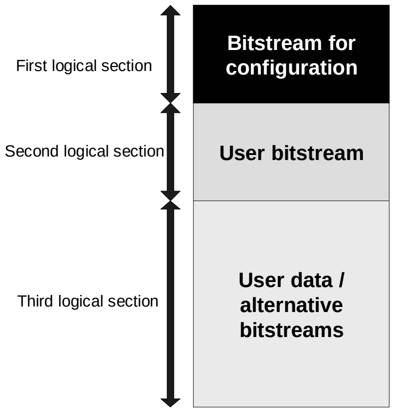


Figure 4: Dividing the flash memory into logical sections.

To download a *bitstream* to the first logical section of the flash memory a JTAG connector [5] is employed. JTAG mode has to be used just once during the board manufacturing. After that, the developed software permits to store new *bitstreams* if required.

The CPLD (see figure 3) is needed for controlling the flash memory and pushbuttons assembled on the board because during configuration the FPGA cannot execute these functions. The CPLD generates also an initial reset signal for FPGA circuits as soon as a new configuration is completed.

The board contains a powerful FPGA of Xilinx Spartan-3 family, namely XC3S400 [5], based on 90nm technology, with 400000 system gates, 56Kb of distributed RAM, 288Kb of block RAM, 16 multipliers and 264 inputs/outputs. For shortening the reconfiguration time, a parallel mode has been chosen.

Extension connectors permit to attach any application-specific external hardware, which enables the designer to optimise resources, to improve performance and to extend the functionality (see figure 1). General-purpose extensions make possible to construct embedded systems interacting with the desired peripheral devices.

The USB controller (version 2.0) provides data exchange with PC for downloading FPGA *bitstreams* and interactions between the FPGA and external hardware.

4 Remote Functions

By replacing the USB controller with a Bluetooth module, the FPGA can be configured remotely. The *bitstream* stored in the first section of the flash memory, which is used for configuration purposes, automatically identifies the module attached to the socket and changes its behaviour accordingly. In contrast to parallel mode provided for USB interface, the Bluetooth module functions in a serial mode, (8 bit data, 115200 baud-rate, no parity bit and one stop bit).

The developed software tools [3] (see section 5 for details) have been modified in order to provide support for the new functionality based on serial interface for the constructed Bluetooth module. From the end-user point of view this functionality is exactly the same as for the USB module and the difference is just in an opportunity of a remote interaction with the board instead of a wired interaction.

Note that an external power source is required if Bluetooth module is used. A small battery-based power source can be supplied to provide the required portability.

5 Software Tools

A software program called PBM (Prototyping Board Manager) has been developed and it provides a convenient user-friendly interface (partially demonstrated in figure 5) and debugging tools.

The most important function of PBM is managing a user *bitstream* in the second section for quick loading into the FPGA (by pressing the “project” button). This technique is the most appropriate to integrate design workflows for single-*bitstream* projects.

PBM also features a terminal window for run-time data exchange between the user and the prototyping system, thus constituting an integrated input/output peripheral, which is ideal for project monitoring and testing.

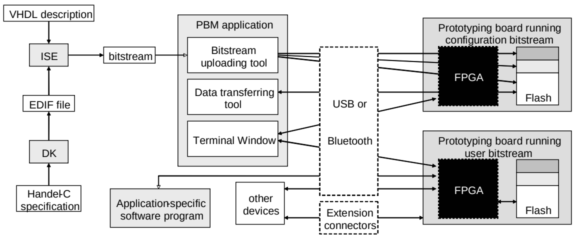


Figure 5: An example of interactions between software and hardware.

A more advanced function allows to send multiple *bitstreams* and to store them in the third section of the flash memory (see figure 4). This function is appropriate for three different purposes:

1. Autonomous experiments with different single-*bitstream* projects without connection to a host computer. In particular, this mode can be used to compare and validate alternative/competitive implementations. The third section of the flash memory is logically subdivided in 6 pre-defined sectors for storing *bitstreams*. Selecting the desired sector is achieved with the aid of a simple additional switch attached through extension connectors, which indicates the proper sector for the CPLD. The same function can be executed remotely through wireless interface.
2. FPGA dynamic reconfiguration, using techniques such as those described in [6,7]. These techniques permit to implement circuits that require more resources than the resources available in the FPGA through run-time reconfiguration.
3. Programming FPGAs installed on additional extension boards. In this case, the core FPGA is considered to be a controller (manager) for a run-time reconfigurable system, which is composed of multiple FPGAs.

The software application includes a user manual in English and in Portuguese languages (also available online [8]) which gives detailed information on how to take full advantage of all the available functionality.

In order to be able to work with PBM, the user must first press the board’s “configuration” button allowing to load the *bitstream* from the first section of the flash memory into the FPGA. This *bitstream* configures the FPGA to implement a control circuit in accordance with the desired protocol (either USB or Bluetooth), which is automatically recognised by PBM. Each function available for the user generates a sequence of basic operations supported by this protocol, such as:

erase a sector, read from pre-specified range of addresses, write a sequence of bytes, etc.

Table 1 presents the average time for performing some of the tasks listed above using the USB protocol. Note that a) each sector has 64 KB, b) writing operations time includes the time for erasing the corresponding sectors and c) writing a *bitstream* involves 4 sectors. Table 1 demonstrates that the developed PBM is faster than the tools [4].

Table 1: Average time for executing PBM functions.

Function	**Average time (s)**
Erase a sector	0.7
Read a sector	0.4
Write a sector	1.5
Write a *bitstream*	5.5

In order to allow PBM to be used with future prototyping hardware, some design guidelines have been established, which guarantee future compatibility. The set of guidelines covers 3 scopes: board architecture, configuration *bitstream* and board's specification file. The first relates to the board construction and hardware properties; the second, to the protocol employed to communicate with the application; the last one applies to the file containing information about the board allowing PBM to manage *bitstreams* and data transfers correctly.

PBM allows not only to download user bitstreams but also to update the configuration (system) bitstream stored in the first section of the flash memory and providing interface between the PBM and the prototyping system. Reloading the system bitstream is required very rarely just in the case if either flash memory contents has been damaged or a new version of PBM is launched.

From figure 5 we can see that the developed software can collaborate with commercial CAD systems in such a way that PBM supplies all kinds of low-level functionality, device drivers, interface and debugging facilities and CAD systems make it possible FPGA based circuits to be designed.

For example, user projects can completely be managed in Xilinx ISE [5] or in any similar environment, which finally generates a *bitstream* that is ready for downloading to FPGA. System-level specification tools (such as Celoxica DK design suite [9]) can also be used. For instance, in figure 5 the design suite of Celoxica translates a Handel-C project description to an electronic design interchange format (EDIF) file, which is further converted in ISE to a *bitstream* for the FPGA.

It is also possible to develop application-specific software which will communicate directly with the board (see figure 5), i.e. without using PBM. This possibility is useful to build solutions that require collaboration between software and hardware components, such as co-processing systems and portable devices with computer-based maintenance.

6 Templates, Design Libraries and IP Cores

Basically, templates, design libraries and IP cores can be taken from the previously developed tools for other prototyping boards namely TE-XC2Se [4], RC100, RC200 [9] and ADM-XPL PCI [10]. Thus, this section reviews the previous authors' results and shows how they can be adapted to the new prototyping platform.

The tools proposed in [11] include reusable specifications of hardware components (modules) that have been developed for two types of CAD environments; Xilinx ISE [5] and Celoxica DK [9]. The components can be employed to implement both application-specific blocks for optimisation purposes and a number of standard interfaces that are very useful for interaction and data exchange with devices attached to the FPGA, such as LCD and touch panels, bus controllers, etc. (see figure 1). The designed modules can be easily integrated into any application-specific digital system and used for visualising the results, fast data transfer, debugging of internal sub-circuits, etc. They were constructed in such a way that their functionality can be either fixed or modifiable (both statically and dynamically). The latter capability was provided with the aid of re-loadable RAM-based blocks. To illustrate the capabilities of the tools suggested, four design examples were discussed in [11].

There are three following types of reusable components that have been developed (based on the previous results) for the new reconfigurable platform:

1. Hardware and VHDL templates. A hardware template is a circuit with pre-defined connections between all primarily components. The functionality of the circuit can be customised through reconfiguration of the alterable circuit blocks, which are RAM blocks. An example of such circuit is given in [12]. VHDL templates are considered to be skeletal code fragments (described in a hardware description language, namely in VHDL), which can easily be customised for particular functionality. They are used much like language templates supplied with Xilinx ISE [5]. An example of VHDL template for a hierarchical finite state machine is given in [13].
2. Design libraries, which are considered to be fragments of parameterisable VHDL code, which can be inserted to any project requiring the respective facilities (for example, requiring USB or RS232 serial communication). VHDL libraries of such type have been developed for many standard interfaces and peripheral devices, such

as RS232, VGA, mouse, keyboard, static RAM, flash RAM, etc. Parametrisation has been provided through VHDL *generic* and *generate* statements.

3. IP cores have been implemented for matrix based optimisation (1) and compression/decompression (2) algorithms, which are often required for robotics and embedded applications. The first group (1) of algorithms permits to solve various problems over binary and ternary matrices, which include matrix covering [14], Boolean satisfiability [15], graph colouring, etc. These tasks are very important for many engineering applications [16,17]). The second group (2) of algorithms makes possible the volume of transmitted data between a host computer and the prototyping board to be reduced. For such purposes methods described in [18] have been employed.

7 Conclusion

The paper suggests tools enabling the designers of electronic devices to simplify many synthesis and verification problems. They include: 1) the core prototyping system (CPS); 2) CPS-oriented software; 3) design templates, design libraries and IP cores for solving optimisation problems over binary/ternary matrices and for data compression/decompression.

8 References

[1] C. Steiger, H. Walder and M. Platzner, “Operating systems for reconfigurable embedded platforms: online scheduling of real-time tasks”, *IEEE Trans. on Computers*, 53(11), pp 1393-1407 (2004).

[2] V. Sklyarov, “Hardware/software modeling of FPGA-based systems”, *Parallel Algorithms and Applications*, ISSN 1063-7192, 17(1), pp 19-39 (2002).

[3] M. Almeida, V. Sklyarov, I. Skliarova and B. Pimentel, “Design tools for reconfigurable embedded systems”, *Proceedings 2nd International Conference on Embedded Software and Systems,* Xian, China, pp 254-261 (2005).

[4] Trenz Spartan-IIE Development Platform, http://www.trenz-electronic.de, visited on 10/7/2006.

[5] Xilinx, “Products and services”, http://www.xilinx.com/, visited on 10/7/2006.

[6] N. Shirazi, W. Luk and P.Y.K. Cheung, “Run-time management of dynamically reconfigurable designs”, *Proceedings 8th International Workshop on Programmable Logic and Applications* (*FPL'98),* Tallinn, Estonia, Springer, pp 59-68 (1998).

[7] V. Sklyarov, A.A. da Rocha and A.B. Ferrari, “Synthesis of reconfigurable control devices based on object-oriented specifications”, in J.C. López, R. Hermida and W. Geisselhardt (eds), *Advanced Techniques for Embedded Systems Design and Test*, Kluwer Academic Publishers, pp 151-177 (1998).

[8] PBM User’s Manual, http://www.ieeta.pt/~iouliia, visited on 10/7/2006.

[9] Celoxica, “Home Page”, http://www.celoxica.com, visited on 10/7/2006.

[10] Alpha Data, “Home Page”, http://www.alpha-data.com, visited on 10/7/2006.

[11] V. Sklyarov, I. Skliarova, P. Almeida and M. Almeida, “Design tools and reusable libraries for FPGA-based digital circuits”, *Proceedings EUROMICRO Symposium on Digital Systems Design (DSD’2003),* Belem, Turkey, IEEE Computer Society, pp 255-263 (2003).

[12] V. Sklyarov, I. Skliarova, A. Oliveira and A. Ferrari, “A dynamically reconfigurable accelerator for operations over Boolean and ternary vectors”, *Proceedings EUROMICRO Symposium on Digital Systems Design (DSD’2003),* Belem, Turkey, IEEE Computer Society, pp 222-229 (2003).

[13] V. Sklyarov, “FPGA-based implementation of recursive algorithms”, *Microprocessors and Microsystems*, 28(5-6), pp 197-211 (2004).

[14] I. Skliarova and A.B. Ferrari, “The design and implementation of a reconfigurable processor for problems of combinatorial computation”, *Journal of Systems Architecture*, 49(4-6), pp 211-226 (2003)

[15] I. Skliarova and A.B. Ferrari, “A software/ reconfigurable hardware SAT solver”, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(4), pp 408-419 (2004).

[16] I. Skliarova and A.B. Ferrari, “Reconfigurable hardware SAT solvers: a survey of systems”, *IEEE Transactions on Computers*, 53(11), pp 1449-1461 (2004).

[17] R. Feldman, C. Haubelt, B. Monien and J. Teich, “Fault tolerance analysis of distributed reconfigurable systems using SAT-based techniques”, *Proceeding FPL’2003,* Lisbon, Portugal, pp 478-487 (2003).

[18] V. Sklyarov, I. Skliarova, B. Pimentel and J. Arrais, “Hardware/software implementation of FPGA-targeted matrix-oriented SAT solvers”, *Proceedings 14th International. Conference on Field-Programmable Logic and Applications (FPL’2004),* Antwerp, pp 922-926 (2004).

3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Stimulation Point of FES for Undulation of Fish

Isao Segou, Keiji Suzuki
Institute of System Information Sciences
Future-University Hakodate, Hokkaido, Japan
g2105016@fun.ac.jp, suzukj@fun.ac.jp

Abstract

Is it feasible to control the action of a shoal of fish by controlling fish and utilise native fish ability or capability that applies to underwater observation? This work proposes CyberFish based on FES (functional electrical stimulation). At the beginning, it is thought that it is necessary to generate the swimming action of fish as the element of the achievement of the control. The fish moves by swimming. In this work, we consider that to generate the swim is a critical factor in the action control of the fish. So, to generate undulation is one of some swimming actions of fish. The aim of this work generates undulation by using electric stimulation used with FES on the muscles of fish. Electric stimulation is generated by microcomputer, and an electrical stimulation pattern is controlled, too. In this experiment, we set some stimulation points and stimulation patterns and stimulated the fish's muscle. As a result of the experiment, we succeeded in confirming the undulation of fish.

Keywords: undulation, FES, stimulation point

1 Introduction

Is it feasible to control the action of the shoal of fish by controlling fish? Is the fish's high exercise ability and capability underwater able to be observed and investigated? In the CyberFish plan, the aim is to control the action of fish by electrical stimulation. At the beginning, it is thought that it is necessary to generate the swimming action of fish as the element of the achievement of the control. This work submits that control action of fish by electrical stimulation based on FES (functional electrical stimulation). FES is electrical stimulation that assists or controls biofunction, and it is used to rehabilitate, to treat of paralytic. In this work, we aim at controlling the action of the fish by generating the fish's swim motion based on the FES system. Next, adopt undulation that is the general swimming motion of fish and aim at reproducing it by adding electrical stimulation based on the FES to the fish's muscle. In this paper, we discuss the feasibility of a cyber fish by the method based on FES and Effective stimulation point.

2 Generate Swimming Motion

2.1 Functional Electrical Stimulation (FES)

FES is electrical stimulation that assists or controls biofunction, and it is used to rehabilitate, to treat of paralytic. The electrical stimulation is a short duration pulse current and is used on the nerve and muscular systems. In the method of modulating the stimulation pulse to control the muscle amount of contraction, Figure 1 shows the three kinds of methods of AM, PWM, and FM. In the clinical application of FES, the FM method is not used so much. AM and PWM methods are often used. In this paper, it stimulates muscle by using the PWM method. Next, the installation configuration of electrodes can be divided roughly into three types, the surface electrode, the percutaneous electrode and the implanted electrode. Figure 2 shows three installation configurations. It is thought that it is difficult to install the surface electrode in fish's surface when it stimulates the fish underwater. So, this work stimulates muscle by using the percutaneous electrode.

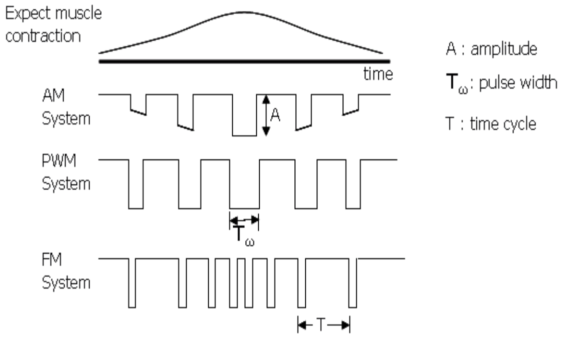


Figure 1: Three kinds of modulation methods of stimulation current or voltage.

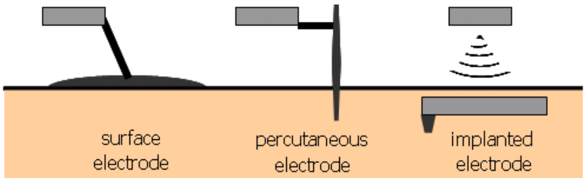


Figure 2: Three kinds of setting methods.

2.2 Fish Swimming Motion

There are various swimming motions of fish: oscillation, undulation, anguilliform, carangiform, thunniform, ostraciiform and so on. The general swimming motion of the fish is undulation by lateral muscle. When the fish swims ahead by using undulation, the wave of the fish's flexion advances from the beak side to the tail side. The fish's arthromere anatomy is a prerequisite for the wavelike movement of the lateral muscle. In the lateral muscle, arthromere anatomy appears as myotome. The muscle fibre becomes like the spiral, and all muscle fibre flex to almost the same length. It is near an efficient isometric contraction. It reflects the arrangement of neurons in the spinal cord and in the medulla. There are spinal cord segments that are units proportional to one myotome. There are spinal nerves from either side of each spinal cord segment.

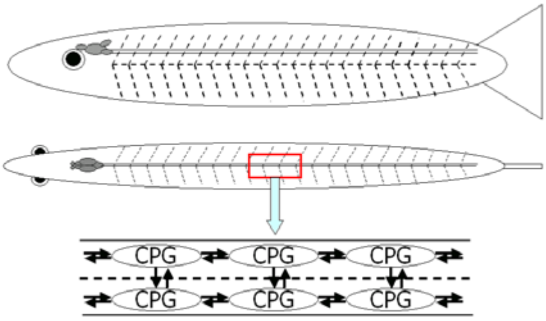


Figure 3: Pattern diagram showing spinal cord segment [2].

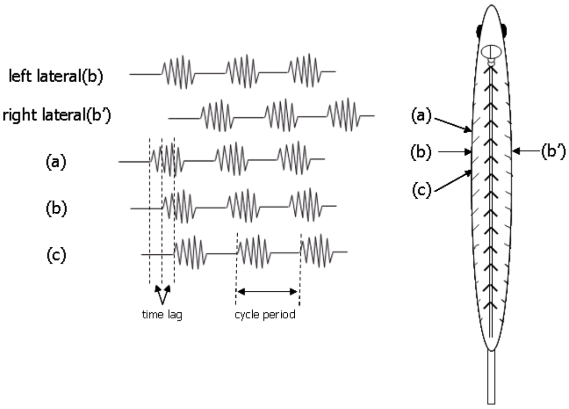


Figure 4: Activity of spinal motor neuron [2].

Figure 3 is a pattern diagram showing the spinal cord segment. In the spinal cord there is a pair of neural circuitry (central pattern generator or CPG) corresponding myotome in each segment. And, right, left, front, and back CPG influence mutually. When a myoelectric signal leading from the fish that is swimming, the burst is recorded alternately from in opposition to right and left myotome. The burst including the little time difference is recorded from the same side front and back myotome. Figure 4 shows the activity of CPG obtained from the electromyogram. A right and left muscle at the same level alternately acts by the interaction of right and left CPG and the muscle in the back and forth acts from the travelling direction sequentially. In this work, we try to generate undulation that assumes a straight forward swimming by stimulation pattern like CPG on lateral muscle.

2.3 Stimulation Experiment

2.3.1 Setting

In this experiment, rainbow trout that swim by undulation were used.

Figure 5: Rainbow trout.

In the stimulation experiment, we put fish to sleep with an anaesthetic. Stimulation waveform and patterns are generated and controlled by the microcomputer. Stimulation waves and patterns are generated by the microcomputer.

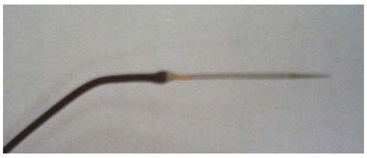


Figure 6: Needle electrode.

Figure 6 shows the needle used as a stimulation electrode. Figure 7 shows the appearance of the experiment.

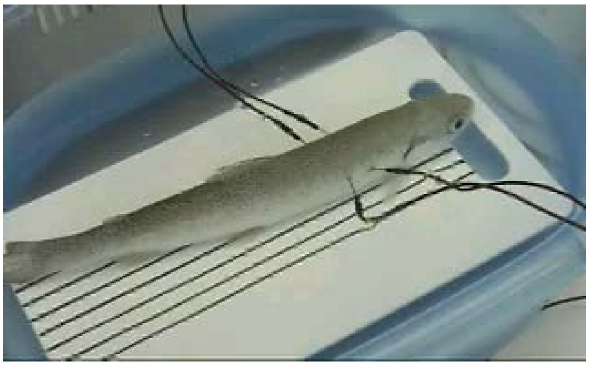


Figure 7: Appearance of experiment.

Figures 8 to 13 show various setting conditions, stimulation positions and patterns.

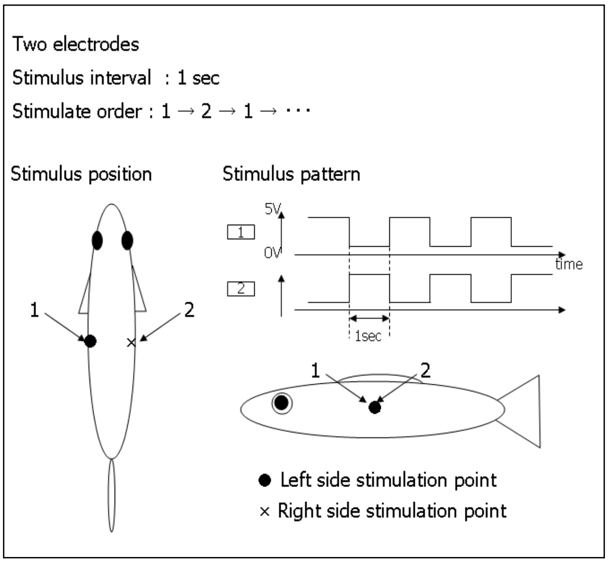

Two electrodes
Stimulus interval : 1 sec
Stimulate order : 1 → 2 → 1 → · · ·
Stimulus position
Stimulus pattern
1
2
5V
0V
time
1sec
● Left side stimulation point
× Right side stimulation point

Figure 8: Setting 1.

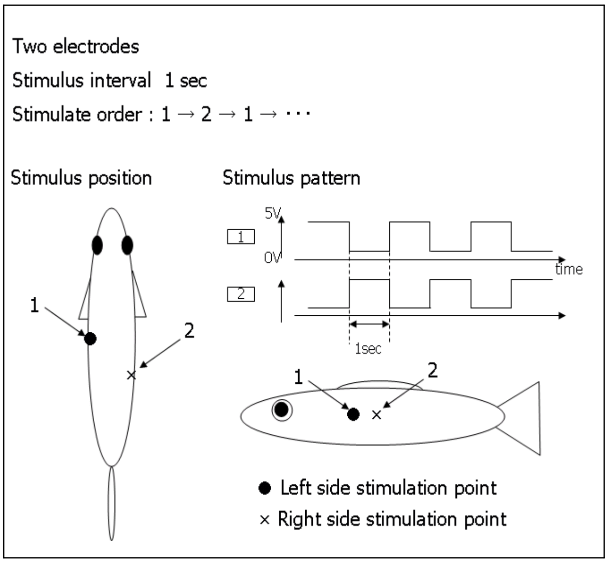

Two electrodes
Stimulus interval 1 sec
Stimulate order : 1 → 2 → 1 → · · ·
Stimulus position
Stimulus pattern
1
2
5V
0V
time
1sec
● Left side stimulation point
× Right side stimulation point

Figure 9: Setting 2.

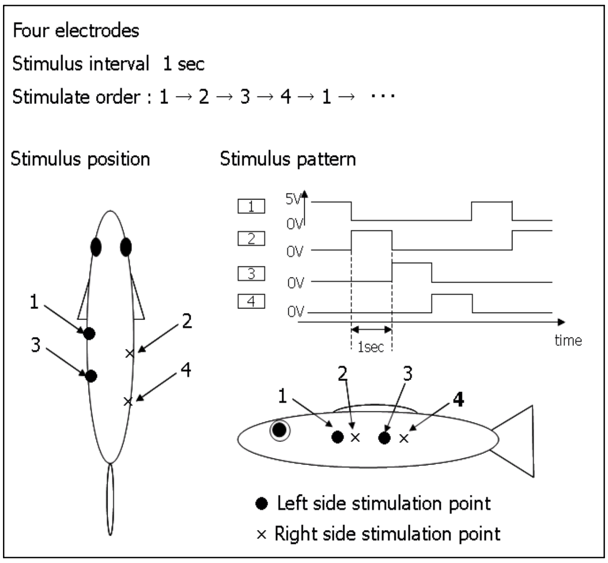

Four electrodes
Stimulus interval 1 sec
Stimulate order : 1 → 2 → 3 → 4 → 1 → · · ·
Stimulus position
Stimulus pattern
1
2
3
4
5V
0V
time
1sec
● Left side stimulation point
× Right side stimulation point

Figure 10: Setting 3.

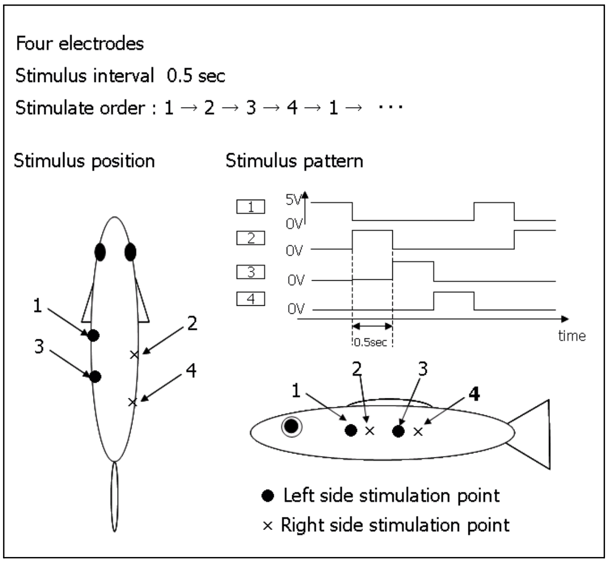

Four electrodes
Stimulus interval 0.5 sec
Stimulate order : 1 → 2 → 3 → 4 → 1 → · · ·
Stimulus position
Stimulus pattern
1
2
3
4
5V
0V
time
0.5sec
● Left side stimulation point
× Right side stimulation point

Figure 11: Setting 4.

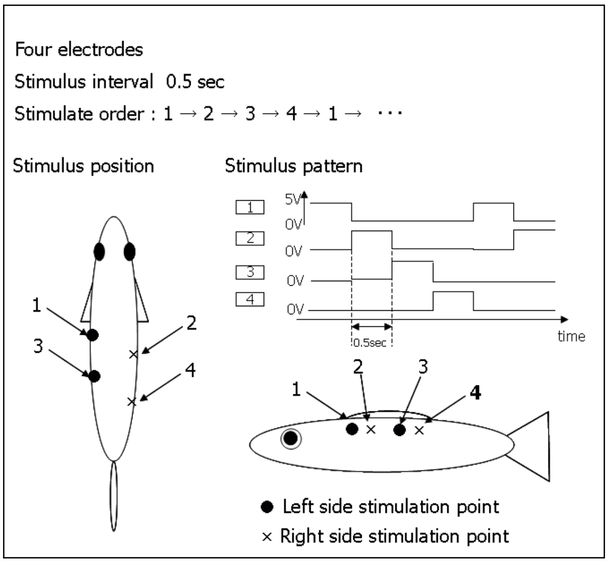

Four electrodes
Stimulus interval 0.5 sec
Stimulate order : 1 → 2 → 3 → 4 → 1 → · · ·
Stimulus position
Stimulus pattern
1
2
3
4
5V
0V
time
0.5sec
● Left side stimulation point
× Right side stimulation point

Figure 12: Setting 5.

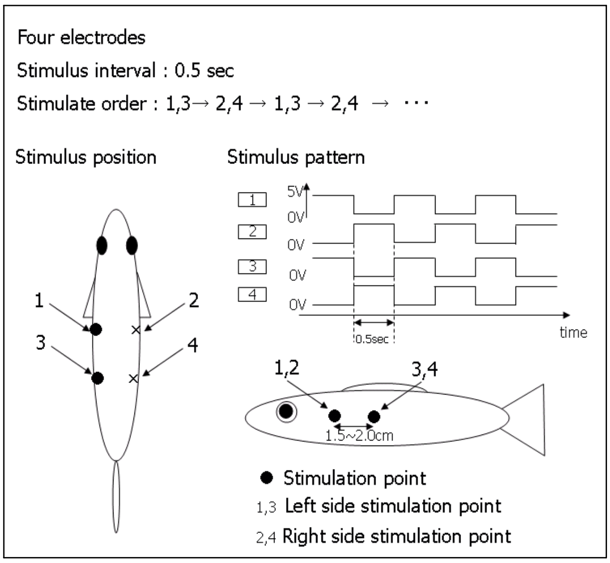
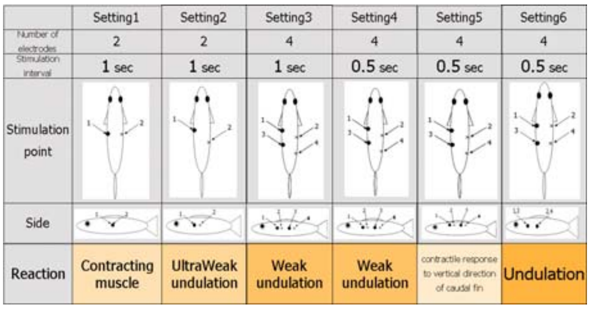

Four electrodes
Stimulus interval : 0.5 sec
Stimulate order : 1,3→ 2,4 → 1,3 → 2,4 → · · ·
Stimulus position
Stimulus pattern
1
2
3
4
5V
0V
time
0.5sec
1,2
3,4
1.5~2.0cm
● Stimulation point
1,3 Left side stimulation point
2,4 Right side stimulation point

Figure 13: Setting 6.

2.3.2 Results

In the condition in figure 8, one stimulation electrode was set up on the opposite side of the body, and was stimulated. As a result, the undulation by the lateral muscle was not able to be generated though contractile response of the lateral muscle was able to be confirmed. In the condition in figure 9, dislocate the stimulation electrode a little and set up. As a result of that condition, micro wave like motion was able to be confirmed by the contractile response. In the condition in figure 10, electrode increases to four. As a result, the contractile response of the lateral muscle was able to be shown, and to confirm the motion which the lateral muscle waves. It was confirmed the wavy motion propagate to caudal fin, but it was not a powerful wavy motion that can generate the impellent. In the condition in figure 11, the stimulation interval changed 0.5 seconds from 1 second. As a result, a similar result to pattern d in figure 11 was able to be confirmed. Pattern d as shown in figure 11 was near a usual swim motion of the rainbow trout. In the condition in figure 12, changed stimulation position to upper of lateral line from near lateral line. In this case, contractile response of the muscle was able to be confirmed, but the caudal fin shows upward flexion. In the condition in figure 13, the distance of the electrode on the same side is adjusted about 2cm, and stimulates the right side muscle and the left side muscle alternately. The purpose of setting 6 is a confirmation of the effect of widening the stimulation of a muscular range. In this case, undulation of the muscle like figure 15 was able to be confirmed.



	Setting1	Setting2	Setting3	Setting4	Setting5	Setting6
Number of electrodes	2	2	4	4	4	4
Stimulation interval	1 sec	1 sec	1 sec	0.5 sec	0.5 sec	0.5 sec
Stimulation point						
Side						
Reaction	Contracting muscle	UltraWeak undulation	Weak undulation	Weak undulation	contractile response to vertical direction of caudal fin	Undulation

Figure 14: Experiment results.

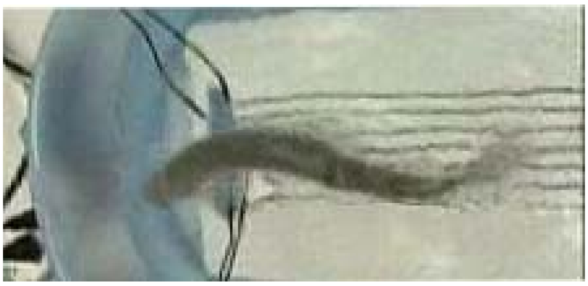


Figure 15: Generated undulation.

In addition, we experimented by using the method of stimulating setting 6 that was able to confirm the undulation. Three electrode installation positions, a, b, and c were set. The stimulation pattern of setting 6 is given at each position and the reaction is observed. The setting of the stimulation position is in figure 16. The result of the reaction is shown in figure 17. When stimulating at the position of A and B, an almost similar reaction was able to be observed. In position c, undulation with a little small amplitude was observed compared with A and B.

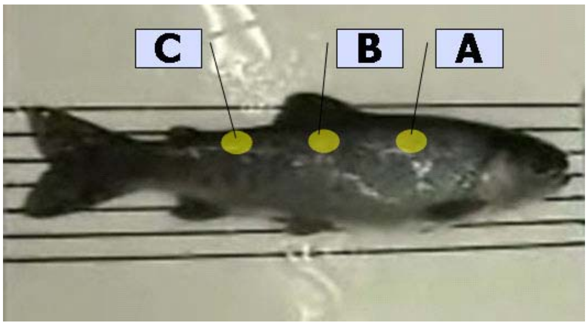


Figure 16: Three stimulation points.

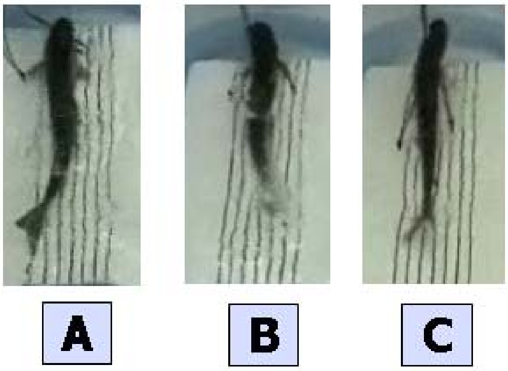


Figure 17: Reactive result at each stimulation position.

3 Discussion

From the results of the experiment, it can be said electric stimulation that is done to wavelike is effective to generate figure 15: it generated undulation, the wavy motion of the lateral muscle. But, we thought that it is still difficult for the fish to swim by the undulation that was able to be confirmed in figures 8 to 11. In the point stimulation like the needle electrode used this time, a muscular range of shrinkage is narrow, and it is felt to be insufficient in generating the wavy motion that can be advanced. When one considers that the fish's muscle fibre shows a reaction near the isometric contraction, widening the range of stimulation can stimulate more muscles, and the generation of a powerful undulation is expected. Then, we experimented with setting 6. In this result, we succeeded in generating steady undulation. By experiment that changes stimulation position by stimulation pattern of setting 6 and observes reaction, it is thought that the stimulated position should stimulate to the area between A and B in figure 16. When you expand the distance of the electrode too much on the same side by the same setting, the contracting muscle is individually shown, and the generation of the wavy motion is disturbed. In this experiment, we confirmed that undulation was able to generate and generable stimulation points. The speed adjustment of the undulation and the generation of the whirling motion would be possible by increasing the number of the stimulation electrodes and improving the stimulation pattern.

4 Conclusion

In this work, we generated swimming action that is an element necessary to control the action of the fish. It experimented by using the microcomputer by various stimulation methods, and the reaction of the fish's muscle was confirmed. In this experiment, we succeeded in generating undulation of the lateral muscle. We confirmed a generable stimulation point for undulation of fish. And, the feasibility of the action control of the fish was discussed.

5 References

[1] N. Hoshimiya, *BioEngineering*, Shokodo, Tokyo (in Japanese) (1990).

[2] K. Uemastu, Y. Oka and H. Ito, *Neuroscience in Fish*, Koseisha Koseikaku, Tokyo (in Japanese) (2002).

Author Index