2012 IEEE 6th International Conference ‘Intelligent Systems’

Battery Aging Detection Based on Sequential Clustering and Similarity Analysis

Gancho Vachkov
Graduate School of Science and
Engineering, Yamaguchi University,
753-8512 Yamaguchi, Japan
vachkov@yamaguchi-u.ac.jp

Stefan Byttner
Intelligent Systems Laboratory
Halmstad University
301 18 Halmstad, Sweden
Stefan.Byttner@hh.se

Magnus Svensson
Volvo Group Trucks Technology
Götaverksgatan 10
405 08 Göteborg, Sweden
Magnus.Svensson@volvo.com

Abstract— **The battery cells are an important part of electric and hybrid vehicles and their deterioration due to aging directly affects the life cycle and performance of the whole battery system. Therefore an early aging detection of the battery cell is an important task and its correct solution could significantly improve the whole vehicle performance. This paper presents a computational strategy for battery aging detection, based on available data chunks from real operation of the vehicle. The first step is to aggregate (reduce) the original large amount of data by much smaller number of cluster centers. This is done by a newly proposed sequential clustering algorithm that arranges the clusters in decreasing order of their volumes. The next step is the proposed fuzzy inference procedure for weighed approximation of the cluster centers that creates comparable one dimensional fuzzy model for each available data set. Finally, the detection of the aged battery is treated as a similarity analysis problem, in which the pair distances between all battery cells are estimated by analyzing the predicted values from the respective fuzzy models. All these three steps of the computational procedure are explained in the paper and applied to real experimental data for battery aging detection. The results are positive and suggestions for further improvements are made in the conclusions.**

Keywords – battery aging detection; data aggregation; sequential clustering; fuzzy inference; weighted approximation; similarity analysis

I. Introduction

In hybrid electric vehicles (HEVs) and electric vehicles (EVs), it is essential to design the battery system well since it is a major part of the total cost of a vehicle. Due to its high cost it is important to maximize the usage of the battery during its lifetime and thus to achieve an efficient and reliable management of the battery system.

A battery pack in a HEV or EV typically consists of several battery cells. The voltage of each cell should in general be measured carefully because this is a strong indicator of the state of charge of the battery [1]. It is sufficient that even a single cell deviation from the normal operation condition would affect the entire battery pack performance by limiting its output.

Deviations in cell behavior generally may occur due to two reasons [2];

- Changes in internal impedance and/or
- Cell capacity reduction due to aging.

There are then two main approaches for balancing a battery system [2]:

- *Passive*; cells with too high voltage are removed by resistors connected in parallel with the cells
- *Active*; moves energy from high voltage cells to low voltage cells

The active approach has generally less losses but is more costly to implement, and both require detection of a deviating cell behavior.

Battery systems that are used for HEVs are subject to especially high requirements since they are exposed to rapid charge and discharge cycles (such as regenerative braking, a common method to convert kinetic energy to electric energy that is further used to charge the battery). Additionally, batteries in vehicles can be exposed to high temperatures which can speed up the aging process.

Aging of batteries is unavoidable during the period of their usage, and typically this leads to gradually decreased power capacity of the battery. For real vehicle applications it is of special importance to detect an “old” battery cell, since it will limit the driving range of the vehicle. Battery manufacturers typically provide data on the aging of a battery based on standard laboratory tests, but these are made for specific (identical) charge and discharge cycles that may not resemble the real usage in the vehicle.

II. Problem Statement and the Proposed Method

It becomes clear from the above preliminaries that the problem of *battery aging detection* belongs to the large group of diagnostics and anomaly detection problems. More precisely, it is the problem of finding the “old” battery cell within the whole battery pack, by analyzing the characteristics (behavior) of all the battery cells, based on the available data and comparing them to each other. It is preferable for this detection to be done in *real time,* by using the measured *data stream* from the real operation of the electric vehicle. As mentioned in the Introduction, in this paper we use data from standard laboratory tests that resemble to a large extent the real operation of the electric vehicle.

The complexity of the problem for battery aging detection is demonstrated on existing real experimental data from two battery cells, as shown in Fig. 1, 2 and 3. One of the battery cells is “Good” and the other is “Old”. The differences between the time characteristics *Ampere – Time* and *Volt – Time* of the

978-1-4673-2278-2/12/$31.00 ©2012 IEEE

Good and Old battery, shown in Fig. 1 and Fig. 2 respectively, are hardly noticeable. However the difference between the static behaviors (*Ampere – Volt)* of these two batteries, depicted in Fig. 3, is easier to be noticed visually, and needs to be discovered in an automatic computational way.

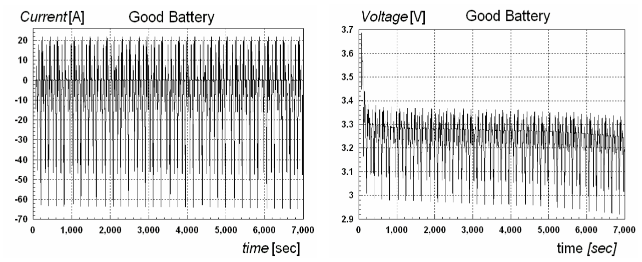

Figure 1. Experimental data taken from a “good” battery cell.

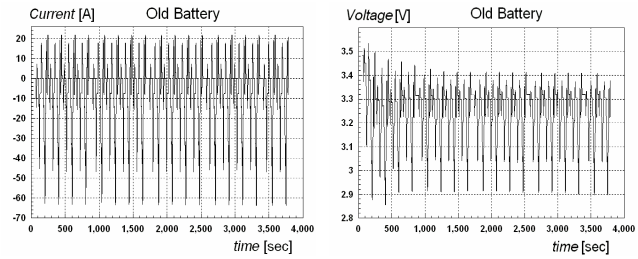

Figure 2. Experimental data taken from an “old” battery cell.

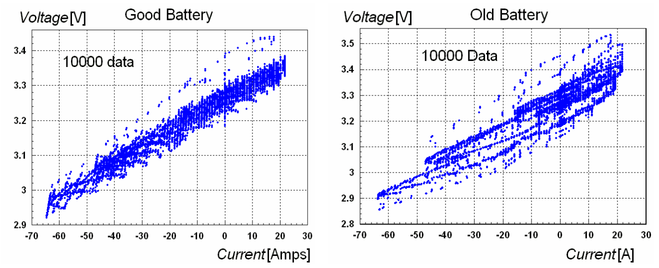

Figure 3. Static (*Volt* - *Current*) behaviour of the Good and Old battery cells.

The proposed method for battery aging detection in this paper works in not pure *real time* mode, but rather in a *semi-real time* mode by using fixed portions of data (data chunks). It consists of the following main computational steps:

- *Aggregation* of the original experimental data by applying sequential clustering algorithm. This step aims to reduce the original large number of data into a smaller number of “representative” cluster centers, thus reducing the computation time in the next steps;

- *Fuzzy Inference* algorithm for weighted approximation of the cluster centers. This step aims to create a one dimensional fuzzy model that represents in a plausible way the relationship *Voltage- Curren*t for each battery cell separately;

- *Similarity analysis* of all pairs of battery cells, based on the prediction from their fuzzy inference models for extracting the most dissimilar (old) cell.

The computational details of the above three steps are explained in the sequel of the paper, followed by results for an old battery detection, based on real experimental data.

III. Aggregation of the Raw Experimental Data by Sequential Clustering

The original (raw) experimental data are not ideally suited for direct usage by the algorithm for similarity analysis, because of several reasons. One of them is the large amount of original data, which significantly slows the computational procedure. Another reason is that the experimental data from the battery cells often include many *ambiguous data*, in a sense that there could be multiple measured outputs (voltage) under the same load condition (the same current). Such cases can be easily noticed in the experimental data from Fig. 3, and they lead to further computation problems (ill defined matrix) if some analytical computation methods involving matrix operations are used afterwards.

Therefore an appropriate computation method is needed to make a kind of *aggregation* of the original large data set of *M* data into a smaller set of *N* representative points, called *information granules* or *cluster centers.*

Such aggregation can be performed by different unsupervised competitive learning algorithms, such as clustering algorithms [3,4], Neural-Gas and its modifications [5,6] etc. All these algorithms aim to find the most appropriate locations of the predefined number of *N* clusters in the *K*-dimensional data space, so that the resulting group of clusters resembles as much as possible the density distribution of the original set of *M* data in the *K*-dimensional space.

The most often used is the *Fuzzy C-means* clustering [3,4], which belongs to the group of S*imultaneous Clustering* algorithms. Here the number *N* of clusters should be set before the calculations. However it is rarely known in advance, so this leads to obtaining some *implausible* clustering solutions, containing smaller or larger number of clusters than necessary.

There is another group of clustering algorithms that use the idea of *Sequential Clustering.* Here the number of clusters is not predetermined, but *grows* gradually (one after another) in a sequence, according to a given criterion, until appropriate stopping conditions are satisfied.

The advantages of the sequential clustering algorithms are twofold. First, the *redundant computations* for unnecessary large number of clusters are avoided. Second, the clusters are usually extracted in an *ordered sequence*, starting from the most significant cluster (the cluster with the largest volume) to the least significant (the smallest volume) cluster. Such representation of the clustering results is helpful for the next computation step – similarity analysis of the available data.

One of the most famous and original sequential clustering algorithm is the *Mountain Clustering* [7] with some of its versions and modifications [8,9]. The general concept here is to use of the so called *mountain* (or *potential*) function, in order to estimate the current areas of highest density in the data space. The potential function decreases gradually in a sequence with each new extracted cluster. This algorithm is easy to implement, but has some problems with the proper selection of the parameters, especially the *width* of the each new subtracted mountain function.

In this paper we use our previously proposed sequential clustering algorithm from [10], with some modifications. This algorithm has been experimentally proven to be a robust clustering with automatic stopping criterion. Its computational details are summarized, as follows:

The assumption is that we have a large number of *M* data in the *K*-dimensional space: $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{iK}], i = 1,2,...,M$. The objective is to extract the centers (prototypes) $\mathbf{C}_i = [c_{i1}, c_{i2}, ..., c_{iK}]$, $i = 1,2,...,n$ of the clusters in a sequence such that their cluster volumes: $V_S, s = 1,2,...,n$ are (most likely) arranged in a *decreasing order:* $V_1 \geq V_2 \geq ... \geq V_n$.

The *cluster volume* V_S can be defined in different ways, but in general this is a kind of measure of the *density* or *size* of the cluster in the *K*-dimensional data space.

The clustering algorithms belong to the group of the *unsupervised learning* algorithms, but in our proposed sequential clustering algorithm we repeatedly solve an *optimization* problem, i.e. maximizing the cluster volume V_S. Therefore we convert the unsupervised clustering problem into a *supervised learning* algorithm.

First, the so called *Cover Function* H_i is defined in the form of a *Gaussian* function with a movable center location $\mathbf{c}$ during the iterations and a fixed (predefined) *K*-dimensional *width* $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, ..., \sigma_K]$ as follows:

$$H_i = \exp\left(-\sum_{j=1}^{K}\frac{(c_{ij} - x_{ij})^2}{2\sigma_j^2}\right), i = 1,2,...,M \qquad (1)$$

The *Cover Function* calculates the *proximity* between the data point $\mathbf{x}_i$ and its current location $\mathbf{c}$. Here $H_i \to 0$ means a *Low Proximity* (*large* distance between the location $\mathbf{c}$ and the data point $\mathbf{x}_i$), and $H_i \to 1$ means *High Proximity* (*short* distance between $\mathbf{c}$ and $\mathbf{x}_i$).

The *volume* *V* of the current cluster is defined in a cumulative way, by adding the weighted proximities of all data $\mathbf{x}_i$ to the current $\mathbf{c}$ location of the cover function, namely:

$$V = \sum_{i=1}^{M} P_i H_i = \sum_{i=1}^{M} P_i \exp\left(-\sum_{j=1}^{K}\frac{(c_{ij} - x_{ij})^2}{2\sigma_j^2}\right) \qquad (2)$$

The weight parameter $P_i \in [0,1]$, $i = 1,2,...,M$ is called *Capacity* of the respective data point $\mathbf{x}_i$. At the beginning of the computation process all data have a *full power* (full capacity): $P_i = 1.0$, $i = 1,2,...,M$. Once a cluster *s* is extracted from the data set, then the capacity of all data points is *reduced* by the following recursive calculation:

$$P_i = P_i - P_i H_i = P_i(1 - H_i), i = 1,2,...,M \qquad (3)$$

Now the problem of extracting the current cluster s, $s = 1,2,...$ is transformed into solving one *optimization* problem of *maximizing* the volume *V* of the cluster, computed by (2) at each step of the sequential clustering procedure.

The proposed sequential clustering algorithm was called in [10] the *moving cup* algorithm, because of the physical analogy with a cup that moves in the search space and tries to gather as many data points as possible at the end of the movement (the end of the optimization).

As for the optimization algorithm, we used here again the popular Particle Swarm Optimization (PSO) algorithm with linearly decreasing *Inertia Weight*. The details of the algorithm are available in [11] and are omitted here.

In our specially modified version of the PSO we have included constraints, as in [10], in order to take into account the fixed size of the search space. Another modification implemented here is for automatic termination of the iterations, according to the following rule: if the criterion *V* is *stabilized* within a small predetermined threshold, the algorithm terminates automatically, thus saving computation time.

At each step s, $s = 1,2,...$ of the sequential clustering, a new cluster is being extracted, when the PSO algorithm terminates. Then the *Average Capacity* AC of all data points, as well as the *total volume* TV of all *s* currently extracted clusters can be calculated, as follows:

$$AC_s = \sum_{i=1}^{M} P_i^s \Big/ M \quad \text{and} \quad TV_s = \sum_{i=1}^{s} V_i \qquad (4)$$

Here P_i^s is the *remaining capacity* of the *i*-th data after the end of the *s*-th step, calculated by (3).

The following trends exist during the progress of the sequential clustering: $AC_s \to 0$ and $TV_s \to M$ with $s \to \infty$. They allow us to establish a meaningful *stopping criterion* of the sequential clustering, by predefining the amount of the *information loss* ε (e.g. $\varepsilon = 0.02$). The sequential clustering will stop after *s* steps (clusters), when the following inequality becomes valid:

$$TV_s \leq M(1 - \varepsilon) \qquad (5)$$

In all further computations we have used the above stopping criterion with $\varepsilon = 0.02$.

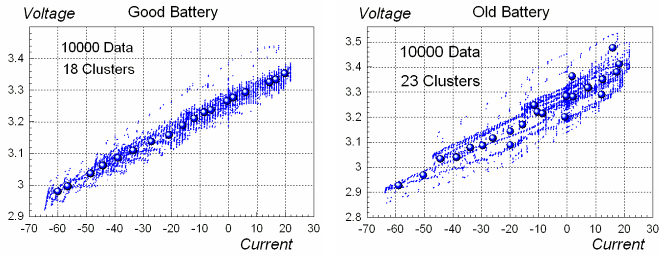

Figure 4. Example of aggregation of two experimental data sets .

The average capacity *AC* of all data decreases monotonously with increasing the number *s* of clusters, as it is shown in Fig. 5 for the example of the data sets from Fig. 4.

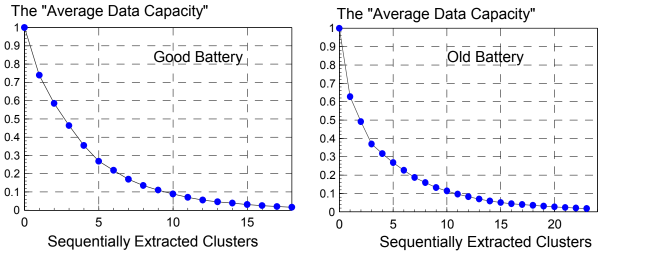

Figure 5. The average capacity of all data are monotonously decreasing with increasing the number of steps (number of clusters) *s*.

At the end of this sequential clustering algorithm, the original large data set is aggregated (reduced) into a smaller number of s clusters with their volumes V_i, $i=1,2,...,s$ and *Cluster Coordinates* saved in vector: $\mathbf{x}_{oi}$, $i=1,2,...,s$.

IV. Fuzzy Inference for Weighted approximation of the Cluster Centers

The next step of the proposed method for battery aging detection is to *approximate* the aggregation results from all data in an uniform (equal) way, so that a fair comparison between the performances of all battery cells could be done. Here the problem is that the aggregation of the data from each battery cell is accomplished by different final number of clusters (granules), varying from *14* to *27* for the real available data. Additionally, each cluster has its own “weight” (volume) which means that it has different “importance” during the comparison between the battery cells.

The proposed idea in this paper is to make the so called *weighted* one dimensional model: *Current – Voltage* (*C-V*) that approximates in a plausible way the relationship between these two measured parameters for each cell. Then, a direct comparison between all the models can be applied to discover the differences (or similarities) between the battery cells.

An illustration of this idea is given in the following Fig. 6 for two data sets, belonging to one *Good* and one *Old* battery cells. The ball-type curve symbols represent the location of the respective clusters at the end of the clustering algorithm. The curve between them is the result from the final *weighted quasi-linear* approximation of the one-dimensional model.

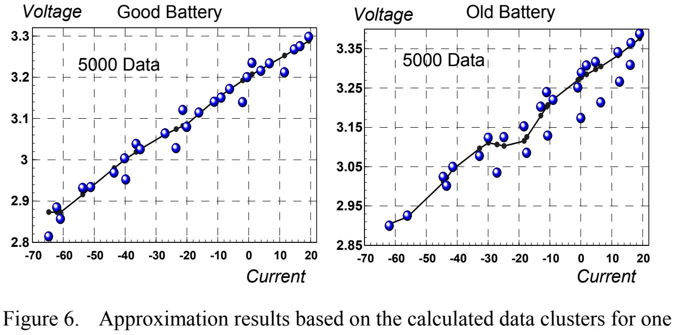

Figure 6. Approximation results based on the calculated data clusters for one Good and one Old battery cell. The curves between the cluster centers are the weighted approximations based on the fuzzy inference of the clusters.

The whole approximation procedure is accomplished into the following two steps:

- *First*, we identify a one-dimensional *Takagi-Sugeno* (TS) fuzzy model by using as input data the coordinates of all cluster centers and their respective volumes (relative weights). Here we assume fixed number and positions of *nine* triangular membership functions for the input *Current*, as shown in Fig. 7.

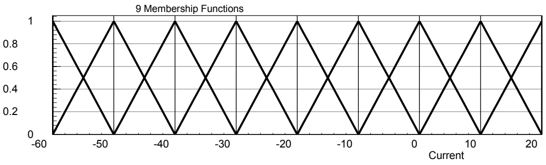

Figure 7. Nine uniformly distributed Triangular Membership Functions.

Here the identification problem is to determine the values of the consequents (*singletons*) of all *9* fuzzy rules such that to minimize the following objective function *Q:*

$$Q=\sqrt{\sum_{i=1}^{S} W_i (V_{oi}-V_i)^2 \Big/ S}, \tag{6}$$

wher S is the number of all extracted clusters; W_i is the weight (volume) of the *i*-th cluster, calculated by (2); V_{oi} is the cluster center (voltage), assumed as experimental value and V_i is the modeled result (voltage) calculated by the fuzzy model.

The minimization of the objective function (6) is done here in a numerical way, by applying (again) the PSO algorithm with constraints, almost identical to the algorithm, used in the previous Section III.

- *Second*, the identified TS fuzzy model is used to *predict* the values for the voltage *V* at each cluster location and also around them, in whole experimental *range* of the Current.

V. Similarity Analysis for Battery Aging Detection

The behavior of all battery cells should be compared in order to discover the differences between them and isolate (detect) the old (aged) battery. This is essentially a *Similarity Analysis* procedure, in which we try to detect the battery, whose behavior is *most deviated* from the behaviors of all other batteries in the list. In Fig. 8 the approximated *C-V* characteristics of six battery cells is shown.

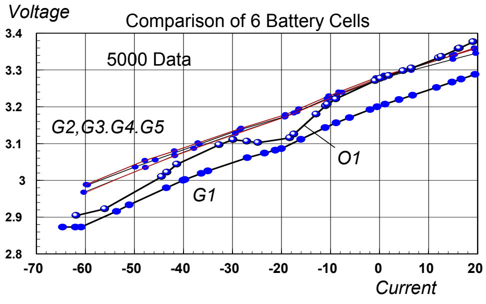

Figure 8. Behaviour characteristics of six Battery Cells, represented by their fuzzy inference approximations from Section IV.

One of the batteries is labeled as old (aged) *O1* and the others *G1.G2,...,G5* are considered as good batteries. The difference between their behaviors (characteristics) can be noticed visually, but we still need a numerical (automatic) way to estimate and sort the differences.

Calculating the *Dissimilarity Degree* between all pairs of battery cells is essential part of the similarity analysis of all available batteries. This is obviously a combinatorial procedure that has to be performed in order to make the final decision about detection of the old battery.

We propose here a simple way to compute the *dissimilarity degree* between a given pair of battery cells {B$_i$,B$_j$} in (7), by taking the *pair distance* (PD) between the approximated values for the voltage, for a given number *R* of *check points*, namely:

$$PD(B_i,B_j)=\sum_{r=1}^{R}\left|V_r^i - V_r^j\right| \Big/ R \qquad (7)$$

The check points are usually uniformly distributed in the *overlapped* input space of the *Current* C, for all pairs of batteries.

The next step is to calculate the *mean distances* between each battery cell and all other batteries in the complete list of *L* batteries, namely:

$$MD_k = \sum_{\substack{i=1\\ i\neq k}}^{L} PD(B_k,B_i) \Big/ (L-1),\ k=1,2,\ldots,L \qquad (8)$$

The final step is to *detect (isolate)* the most distant battery, i.e. the battery, whose characteristics differs at most from all other characteristics. The "aged" battery, denoted by B$_o$ will have the *largest mean distance* to all other batteries in the list, i.e.

$$B_o = \max_{1\le k\le L}\{SD_k\} \qquad (9)$$

Another way to calculate the dissimilarity degree between a given pair of batteries is to use the *weighted mean distance* WMD, by applying a Gaussian function that gives bigger weights to the smaller distances. This gives actually a greater importance to the batteries that are closer to the current one.

Experimental results for detection of the old battery cell are given in the next section by using both calculations: MD and WMD.

VI. Experimental Results from Battery Aging Detection

The data used for battery aging detection in this paper are measured using both new batteries and an aged battery cell in a real (non-uniform) drive cycle (EUCAR cycle), realistic to a vehicle application. The aged battery had been used with this drive cycle for about *4* months, with about *18* hours of usage per day.

A. Organization of the Experimental Data

From the whole available amount of experimental data we have extracted three relatively long data chunks (each of them with *5000* data) representing the behavior of five "good" (new) battery cells, named as G1, G2, G3, G4 and G5, and one "old" (aged) battery cell named as O1. The three successive data chunks are further named as D1, D2 and D3 and contain the individual experimental data for each battery cell.

In order to test the robustness of the proposed method for a correct detection of the aged battery O1, we have run the method for all *3* data chunks D1, D2 and D3 separately and also for longer sequences of data, obtained by merging of these data chunks. Thus we obtained 3 longer data sets, named as: D12 and D23 (with *10,000* data each) and data set D123 (containing the full amount of *15,000* data).

The next Fig. 9 depicts the data sets D3 containing *5000* data each for all battery cells: G1, G2, G3, G4, G5 and O1.

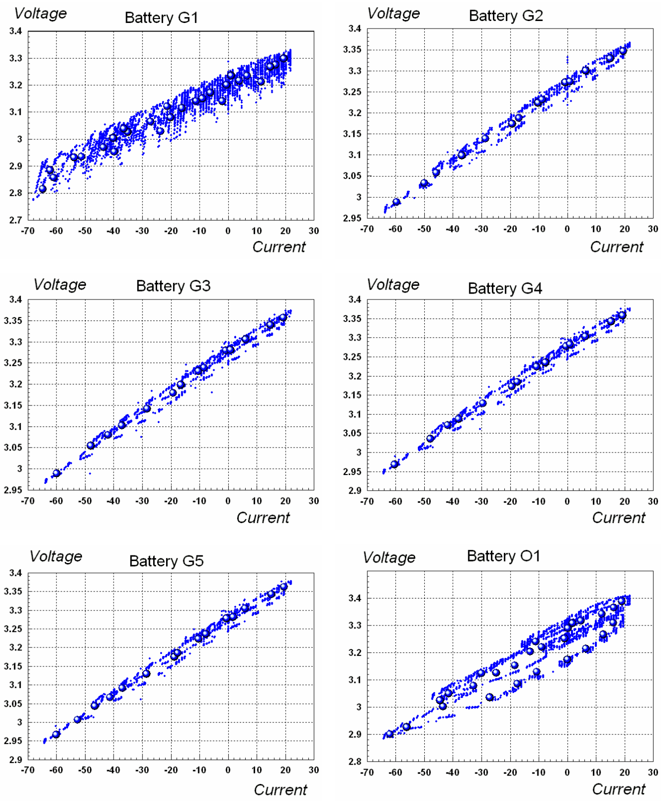

Figure 9. Data Sets D3 containing 5000 data for all battery cells .

B. Results from the Aged Battery Detection

The following are some of the experimental results for aging battery detection. The have been produced by assuming *R = 80* check points located uniformly within the interval [-60, +20] of the Current C.

Table I shows the pair distances PD, calculated by (7) for all pairs of batteries by using the experimental Data set D3.

Table II presents the calculated mean distances MD by (8) and also the weighted mean distances WMD, in the case of using Data Sets D3. The detected *old battery* is shown in bold for both cases: MD and WMD.

Tables III, IV, V and VI show the detection results in the case of using data sets D1, D12, D23 and D123 respectively.

In all experimental results, the assumed spread parameter of the Gaussian function was: $\sigma = 0.025$

TABLE I. THE PAIR DISTANCES PD OF ALL BATTERY CELLS FOR DATA SET D3.

Battery	*G1*	*G2*	*G3*	*G4*	*G5*	*O1*
G1	0.00	0.0125	0.0131	0.0147	0.0128	0.0152
G2	0.0125	0.00	0.0052	0.0097	0.0071	0.0158
G3	0.0131	0.0052	0.00	0.0052	0.0062	0.0150
G4	0.0147	0.0097	0.0052	0.00	0.0066	0.0154
G5	0.0128	0.0074	0.0062	0.0066	0.00	0.0163
O1	0.0152	0.0158	0.0150	0.0154	0.0163	0.00

TABLE II. DETECTION RESULTS FROM DATA SET D3

Battery	*G1*	*G2*	*G3*	*G4*	*G5*	*O1*
MD	0.0137	0.0101	0.0089	0.0103	0.0099	**0.0155**
WMD	0.0029	0.0028	0.0027	0.0028	0.0029	**0.0032**

TABLE III. DETECTION RESULTS FROM DATA SET D1

Battery	*G1*	*G2*	*G3*	*G4*	*G5*	*O1*
MD	0.0110	0.0091	0.0092	0.0097	0.0094	**0.0114**
WMD	0.0022	0.0022	0.0019	**0.0025**	0.0021	0.0024

TABLE IV. DETECTION RESULTS FROM DATA SET D12

Battery	*G1*	*G2*	*G3*	*G4*	*G5*	*O1*
MD	0.0103	0.0083	0.0084	0.0089	0.0080	**0.0112**
WMD	**0.0023**	0.0021	0.0021	0.0022	0.0021	0.0022

TABLE V. DETECTION RESULTS FROM DATA SET D23

Battery	*G1*	*G2*	*G3*	*G4*	*G5*	*O1*
MD	0.0124	0.0085	0.0082	0.0098	0.0087	**0.0126**
WMD	0.0022	0.0024	0.0023	0.0025	0.0025	**0.0026**

TABLE VI. DETECTION RESULTS FROM DATA SET D123

Battery	*G1*	*G2*	*G3*	*G4*	*G5*	*O1*
MD	0.0117	0.0083	0.0087	0.0096	0.0087	**0.0118**
WMD	0.0023	0.0021	0.0020	**0.0026**	0.0024	0.0025

It is seen from the above tables that the detections by using the mean distances MD in (8) show correct results, i.e. the old battery **O1** was discovered properly. As for the case of the weighted mean distances WMD, for three data sets (D1, D12 and D123) the detection showed different (not the correct) results. However, the "true" old battery O1 is listed in these 3 cases as a "second guess" (the second maximal value in the Table). Obviously the proper selection of the spread parameter σ here plays an important role for the correct detection and is usually determined heuristically.

A look at Fig. 8 and Fig. 9 can explain such deviation in the results, namely that the battery cell G1 has a quite different behavior compared to that one of the other good cells: G2, G3, G4 and G5. Therefore it can be easily mistaken as an old battery. This implies that the proposed method, even if not currently a perfect one is robust enough to detect the battery cells that behave "very differently" from the normal good cells.

VII. CONCLUSIONS

The proposed computational procedure for battery aging detection in this paper consists of three main steps: data aggregation by sequential clustering; fuzzy inference for weighted approximation of the cluster centers and similarity analysis by using the pair distances method.

The initial information for detection of the aged battery is in the form of relatively large portions of real operation data (data chunks) from the electric vehicle. The result from the detection is to find a certain battery cell, from the list of all cells, whose performance is most different from the performances of all the other batteries.

The proposed method basically does not use problem dependent parameters in the final detection stage, with exception for calculation of the weighted mean distance WMD. However, it still uses some tuning parameters in the PSO algorithms for sequential clustering and for fuzzy model identification. It is experimentally proven that they have very little influence to the final detection results.

There are some ways to improve the detection results, such as creating another type of model that describes more precisely each data set before the comparison. Another direction for future development of the proposed method is to modify it for handling not only data chunks, but also data streams in real time. Thus it would become an evolving method being able to detect the gradual changes in the performance (aging) of each battery cell separately.

REFERENCES

[1] B. A. Johnson, R. E. White. Characterization of Commercially Available Lithium-Ion Batteries", Journal of Power Sources, Vol. 70, pp. 48-54, 1998.

[2] J. Hellgren, L. Feng, B. Andersson, R. Blanc. An Experimental System for Battery Management Algorithm Development. The 25th World Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exhibition, Shenzen, China, Nov 5-9 (2010).

[3] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York: Plenum Press, 1981.

[4] A. Jain, "Data Clustering: 50 years beyond K-means", *Pattern Recognition Letters*, vol. 31, pp. 651-666, 2010.

[5] T. Martinetz, S. Berkovich and K. Schulten, "Neural-Gas Network for Vector Quantization and Its Application to Time-Series Prediction", *IEEE Trans. Neural Networks*, Vol. 4, No. 4, pp. 558-569. 1993.

[6] L. Xu, A. Krzyzak and A. Oja, "Rival Penalized Competitive Learning for Clustering Analysis, RBF Net and Curve Detection", *IEEE Trans. Neural Networks*, Vol. 4, No. 4, pp. 636-649, 1993.

[7] R. Yager and D. Filev, "Approximate Clustering via the Mountain Method", *IEEE Trans. on Systems, Man and Cybernetics*, vol. 24, No. 8, pp.1279-1284, 1994.

[8] A. Clark and D. Filev, "Clustering Techniques for Rule Extraction from Unstructured Text Fragments", CD-ROM Proc. of the NAFIPS'05 Conference, Ann Arbor, USA, July 2005.

[9] P. Angelov, "An Approach for Fuzzy Rule-based Adaptation using On-line Clustering", *International Journal of Approximate reasoning*", vol. 35, no. 3, pp. 275-289, 2004.

[10] G. Vachkov, "Temporal and Spatial Evolving Knowledge Base System with sequential Clustering", Proc. of the 2010 World Congress WCCI 2010 (FUZZ-IEEE 2010), Barcelona, Spain, pp. 3010-3017. July, 2010.

[11] R. Poli, J. Kennedy and T. Blackwell, Particle Swarm Intelligence. An Overview", *Swarm Intelligence*, vol. 1, pp. 33-57, 2007