A New Algorithm for a Fast Testing and Sorting System Applied to Battery Clustering

Qing Wang[a)1)], Xianzhe Cheng[a)b)] and Jie Wang[a)c)]

[a)]School of Engineering and Computing Sciences, Durham University, DH1 3LE, Durham, United Kingdom

[b)]College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, Hunan 410073, China

[c)]Postgraduate Institution, China Academy of Space Technology, Beijing, 100086, China

***Abstract*—Battery clustering is to sort out homogeneous battery cells to form a battery pack with high uniformity, which is of great importance to prolong the cycle life of the lithium-ion battery. The traditional method for battery clustering is to compare the charge and discharge characteristic curves of the battery cells. This paper proposes a new algorithm, the squeeze algorithm, for a fast testing and sorting system to save the test time and improve efficiency. The algorithm is based on a database of complete charge and discharge characteristic curves. With a battery cell to be clustered, it will go through a short time test and get a partial curve. And then the partial curve will be compared with the database curves to get clustered. Experiments have been conducted with 100 battery cells to be clustered with 1111 database batteries. The clustering results show high accuracy, which indicates that the proposed algorithm is feasible for battery clustering.**

***Index Terms*--Battery Refurbishment, Capacitance-voltage characteristic Curves, Clustering algorithms, Euclidean distance**

I. Introduction

Since the first commercialization of the lithium-ion battery (LIB) in 1991, the LIB market has grown dramatically in the last two decades and now worldwide sales for LIBs have surpassed $13,000 million [1]. Compared with other type of batteries (e.g. lead-acid, nickel-metal hydride and etc.), LIBs have outstanding performance with higher operating voltages, higher energy densities and lower self-discharge rates, thus being widely used in cameras, mobiles, laptop computers and other portable IT devices [1]-[3]. Especially in the electrical vehicle field, LIBs play an important role to act as the power source and energy storage center [4].

Generally, a LIB is made up of several Li-ion cells (with 18650 most commonly used). In an electrical vehicle battery pack, the number of cells reaches up to dozens or even hundreds. Those cells in a battery pack are connected in a series-parallel structure. A series structure would increase the voltage of the whole battery pack while a parallel structure increases the capacity, which enables the battery pack to adapt to various applications [5]. However, problems lie in this structure along with battery cycling use. There exist diversities on dynamic property among cells in the same battery pack. As a consequence, a battery pack would stop discharging when the weakest cell in it gets exhausted, while the capacities in other cells are not fully utilized. The case is similar in charging process, with the weakest cell not fully charged. What is worse, over-charging or over-discharging may take place, resulting in great harm to the battery cells and decreasing the cycle life of the whole battery. The discrepancies among cells bring huge impacts for the usage of battery packs [6].

To deal with this problem, there are two main solutions. One is to apply a battery management system (BMS). The BMS would monitor the charge condition of each cell and manage to balance the capacities of those cells, optimizing the performance of the whole battery pack. Many researchers attempted to promote efficiency of the BMS [7]-[9]. The second method is to sort out homogenous battery cells to form a battery pack with high uniformity. This solution would prevent the discrepancies among cells to make the best of the assembled battery pack.

In the research on battery cell uniformity, it is widely adopted to compare feature vectors of different cells and classify them into groups according to the distances among these feature vectors [6], [10]-[14]. Raspa et al. chose open circuit voltage (OCV), total capacity and parameters from equivalent circuit model (ECM) as the feature vector and worked out the distance among these vectors by using self-organizing maps neural networks to get the clustering results [6]. Shen et al. employed different parameters including "aging voltage, capacity, resistance, 1C discharging time and thickness" [10] to cluster battery cells into groups. According to [11] and [12], battery charge and discharge characteristic curve was a better choice for feature vector as it reflected the dynamic property of the battery cell. Zheng and Wang [13] used Euclidean distance and correlation coefficient to measure the difference among battery characteristic curves. Duo [14] proposed a fuzzy-based algorithm to compare among the characteristic curves. However, the above researches are based on a full charge and discharge test on battery cells, which is very time-consuming. On the other hand, all the adopted methods intend to pick out two feature vectors with the minimum distance as the criterion for battery cell clustering in a certain sample,

[1)] Email: qing.wang@durham.ac.uk

which may bring considerable errors if the sample is very sparse.

This paper aims to improve the efficiency of battery testing and sorting to cluster battery cells with similar properties. A new method, named squeeze algorithm is proposed for a fast testing and sorting system. A database of complete charge and discharge curves are established as classifying criteria. For a battery cell to be tested, it will go through a short time charge and discharge test to generate a partial charge and discharge curve. Then the obtained partial curve would match with the database using the squeeze algorithm to be sorted. As a result, the corresponding tested battery cell gets classified. This method reduces the test time and improves efficiency for battery cell testing and sorting, which is fit for the industrial application.

II. EXPERIMENTAL

Experiments have been conducted using Matlab and Simulink. A battery test circuit model is built with ideal elements in Simulink. After a series of complete charge and discharge tests with this model, 1111 complete charge and discharge characteristic curves are obtained as the database, which will be the classification destinations of the clustering process. And then 100 LIBs are fast tested with an incomplete charge and discharge process to get part of the characteristic curves as the clustering samples. The sorting procedure is realized by applying the squeeze algorithm with Matlab code, to get the final clustering results.

A. Charge and Discharge Circuit Model

The charge and discharge circuit model is based on a battery module in Simulink Library Browser, as showed in Fig. 1. It has been validated that charge and discharge characteristic curves from the Simulink battery module resemble the real test curves quite well [15]. For better simulation results, the battery module needs to be set to fit the LIB characteristics and the charge and discharge process should be consistent with real battery test.

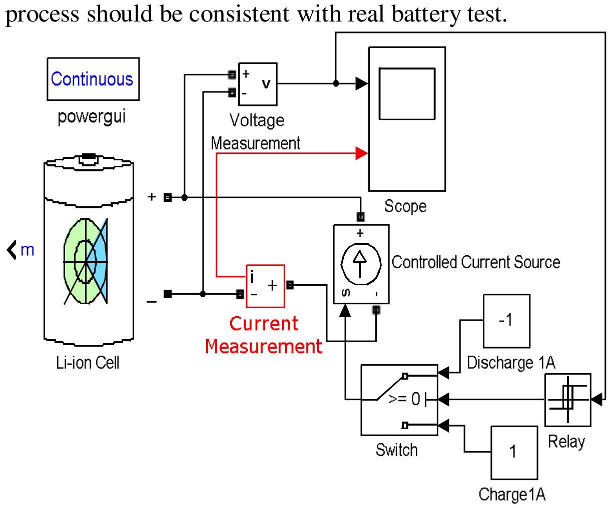


Fig. 1. Charge and Discharge Circuit Model.

In the LIB family, 18650 (18mm diameter × 65mm length) is the most popular battery type of the day and is widely used in laptops and other e-products. The nominal voltage of LIBs is 3.6 V or 3.7 V, varying with the manufacturer. The rated capacity ranges from 2 Ah to 3 Ah, with 2.6 Ah being the most commonly used currently. To have a better knowledge about the charge volume held by a battery, state-of-charge (SOC) is defined to measure the ratio of the temporary amount of charge at a particular moment to the fully charged volume [1]. For example, the SOC of a fully charged battery is 100% and it will decrease as the battery is discharged.

In the test model, the LIB module is set up using three parameters (nominal voltage, rated capacity and Initial SOC). The charge and discharge characteristic curve of the LIB is determined once nominal voltage and rated capacity are assigned with certain values. Initial SOC reflects the initial statement of the battery. In order to establish an abundant database of characteristic curves, the parameters nominal voltage and rated capacity of the database batteries are evenly-spaced divided. The nominal voltage is set with 11 values from 3.6 V to 3.7 V, with 0.1 V increment for each rank. And the rated capacity varies from 2 Ah to 3 Ah with 0.01 Ah as the increment, ending up with 101 ranks. Initial SOC is assigned with a random value from 1% to 100%. As a consequence, the database consists of 1111 battery cells, with their parameters different from each other and their charge conditions completely unknown.

It is required to charge and discharge the database battery cells to get complete charge and discharge characteristic curves from the test. As for the LIBs, no standard is provided to define a fully-charged point and an empty-capacity point. It is a widely used routine to regard 4.2 V as the full state and 3V or lower as the empty state [5]. Two methods exist in the charge and discharge process: constant current (CC) and constant voltage (CV). CC is mostly used in this process while CV is suitable for deep charging to fully charge a battery. Each database cell in the test model would firstly be charged to 4.2 V in the first step (discharged to 4.2 V if the initial voltage is over 4.2 V). The second step discharges the cell with 4.2 V as the starting point down to 2.8 V, which is taken as the end point of the discharge process. Another charge process follows to raise the voltage back to 4.2 V (all the charge and discharge processes are in constant current 1 A for convenience). Voltage data of the tested batteries is sampled in the discharge and recharge process to form as the feature vector, with the sample time of 20s. In consequence, a complete discharge and charge curve is obtained from step 2 and 3 (see Fig. 2).

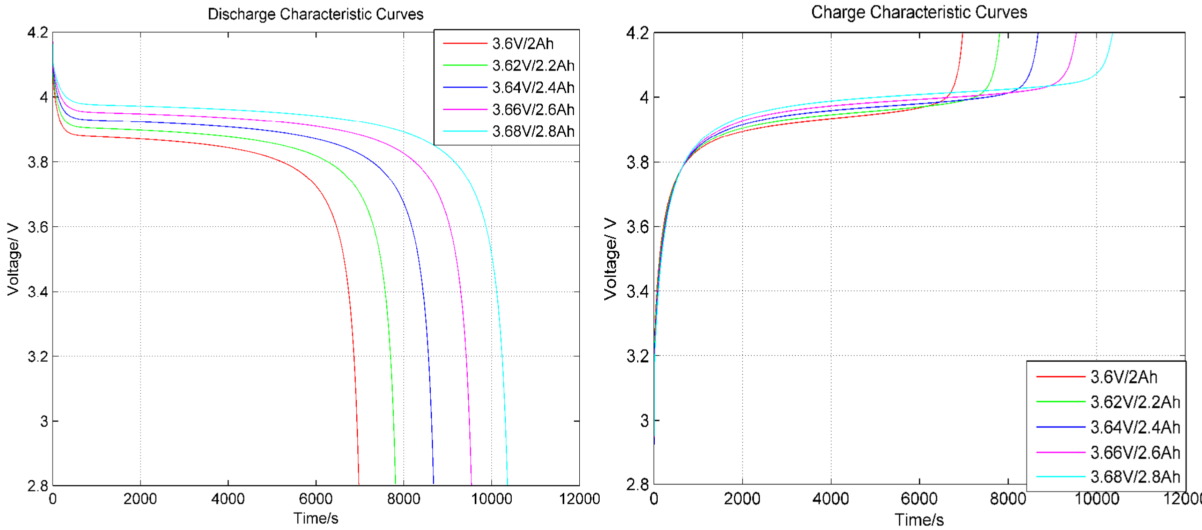


Fig. 2(a). Complete Discharge Characteristic Curves.

(b). Complete Charge Characteristic Curves.

Another 100 sample battery cells are fast tested to get partial characteristic curves as the clustering samples. The parameters of these sample cells are defined with random values in the limited ranges, which is similar to the case of real test. The fast test process first charges the cells to 4.2 V, the same with the first step of complete charge and discharge test. Then the discharge process follows. It continues for a limited time, which will stop long before the tested cell reaches the end of discharge point. And finally, the tested battery cell will be recharged back to 4.2 V. The voltage data acquisition process is the same as before. As there is a voltage step change at the beginning of the discharge and charge process, the beginning part of the partial curve may deviate from the standard characteristic curve. Therefore, the discharge and charge partial curve is chopped off at the head (200 s) and tail (500 s), left with the same length. With this incomplete discharge and charge process, a part of the characteristic curve could be obtained. These 100 partial curves from the short time test would be matched with the database complete characteristic curves to get clustered.

B. Sorting Procedure with the Squeeze Algorithm

The sorting procedure is to compare the partial characteristic curve with the database curves and find out an optimal clustering result. As a matter of fact, there lie several challenges in the whole procedure. First of all, by comparing with all the database curves, how to figure out the optimal one to be clustered with the partial curve is a big problem. Second, the length of the partial characteristic curve is different from the length of the complete characteristic curve. In the comparison process, the compared database curve needs to be cut down to a segment with the same length of the partial curve. Thus, to choose the start point of the segment part is another challenge. Third, how to define the distance between the partial curve and the database curve is also an important factor.

The squeeze algorithm is proposed to complete the sorting procedure. Fig. 3 outlines the flow chart for this process.

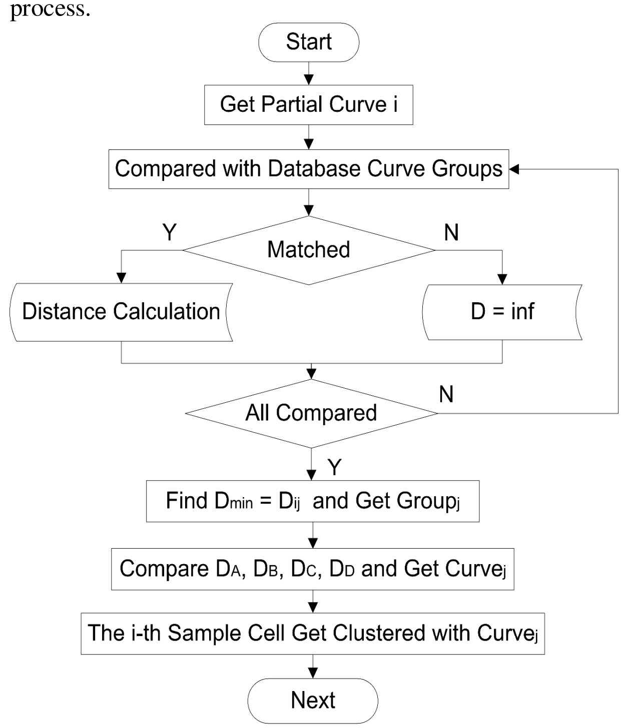


Fig. 3. Flow Chart for the Sorting Procedure.

For the battery cell in the fast time test, their parameters are set with random values in the limited ranges. As a result of the comparison process, the parameters of the matched database curve should approach the parameters of the partial curve. In order to increase the accuracy for the match results, the proposed algorithm intends to compare the partial curve to four adjacent database curves as a group at one time. The database curves are bunched into groups as indicated in Fig. 4, which amounts to 1000 groups totally.

Capacity \ Voltage	3.6 V	3.61 V	3.62 V	…	3.69 V	3.7 V
2 Ah	1 A B	2 A B	3	…	10 A B	11
2.01Ah	12 C D A B	13 C D	14	…	21 C D	22
2.02Ah	23 C D	24	25	…	32	33
…	…	…	…	…	…	…
2.99Ah	1090 A B	1091	1092	…	1109 A B	1100
3Ah	1101 C D	1102	1103	…	1110 C D	1111

Fig. 4. The Database Curves Divided into Groups

For example, if the nominal voltage and rated capacity of the partial curve are 3.674V and 2.385Ah respectively, then the optimal clustered group of the database curves should be database curve 426 (A: 3.67V/2.38Ah), 427 (B: 3.67V/2.39Ah), 437 (C: 3.68V/2.38Ah) and 438 (D: 3.68V/2.39Ah). The nominal voltage value of the partial curve falls between in the parameters of curve A and curve C (or curve B and Curve D), and the rated capacity value lies between the parameters of curve A and Curve B (or Curve C and Curve D). With this way, the discharge part of the partial curve is judged to be situated between curve A and curve D, while the charge part lies between curve B and curve C.

In the comparison process, the discharge part of the partial curve would be compared with curve A and curve D, while the charge part would be compared with curve B and curve C in a database curve group. As described in section I, the traditional method for the comparison process is to extract the feature vector from each curve and calculate the Euclidean distance between these vectors. However, those extracted characteristic vectors are the same length, which is totally different from the case in this paper. In consequence, it is necessary to cut down the database curve to the length of the partial curve.

Before distance calculation between partial curve and database curve group, it is necessary to judge if the partial curve matches with the given curve group. Taking the discharge partial curve for example, equal to head value of the partial discharge curve, there are corresponding start point 1 (s1) and start point 2 (s2) on curve A and curve D respectively (see Fig. 5).

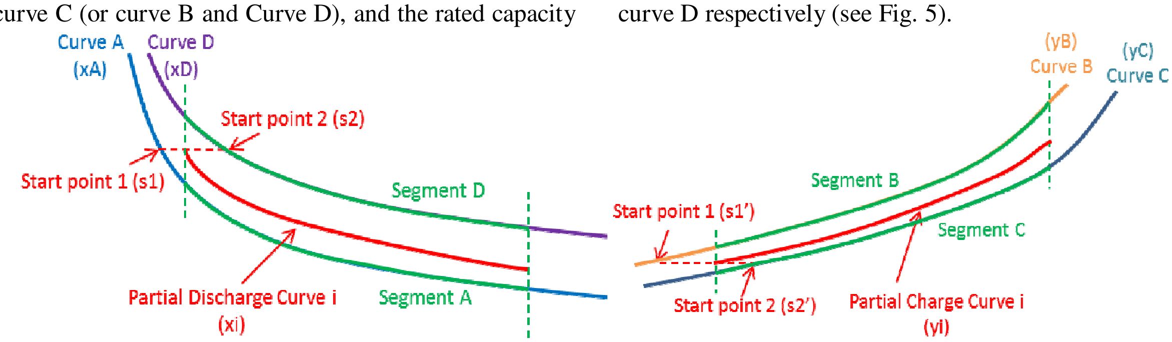


Fig. 5(a). Distance Calculation for the Partial Discharge Curve. (b). Distance Calculation for the Partial Charge Curve

With the head of the partial discharge curve sliding between s1 and s2, if the partial discharge curve exceeds the boundary of curve A or curve D, the partial curve could not be matched with this curve group. Thus, the distance between the partial curve and the related curve group would be set as infinite. While the partial discharge curve lies between segment A and segment D, the discharge distance is defined as the sum of the Euclidean distance between the partial discharge curve (vector xi) and segment A (vector xA) and the Euclidean distance between the partial discharge curve and segment D (vector xD). As it is a dynamic distance calculation process, a series of distance values could be obtained with the sliding step. The final distance of the discharge part would be the minimum value in these calculated results. For the distance of the charge part, the calculation process is similar (with vector yi, yB, yC). Calculation formulas are presented in (1) – (3) (n is the length of vector xi and yi). The distance between partial curve i and database curve group j is the sum of the discharge distance and the charge distance.

After the partial curve i is compared with all the database curve groups, a distance array would be obtained. Finding the minimum one in the distance array, the optimal curve group j is selected out. Then, the choice of the clustered database curve would be decided by comparing the distance values between this partial curve and curve A, B, C, and D. At last, the partial curve gets clustered with the specific database curve.

$$D_{\mathrm{ij\_discharge}} = \min\left\{ \left(\frac{1}{n}\sum_{p=1}^{n} \left(\mathbf{xi}_p - \mathbf{xA}_{p+s}\right)^2 \right)^{1/2} + \left(\frac{1}{n}\sum_{p=1}^{n} \left(\mathbf{xi}_p - \mathbf{xD}_{p+s}\right)^2 \right)^{1/2} \middle| s1 \le s < s2 \right\}, \tag{1}$$

$$D_{\mathrm{ij\_charge}} = \min\left\{ \left(\frac{1}{n}\sum_{p=1}^{n} \left(\mathbf{yi}_p - \mathbf{yB}_{p+s'}\right)^2 \right)^{1/2} + \left(\frac{1}{n}\sum_{p=1}^{n} \left(\mathbf{yi}_p - \mathbf{yC}_{p+s'}\right)^2 \right)^{1/2} \middle| s1' \le s' < s2' \right\}, \tag{2}$$

$$D_{\mathrm{ij}} = D_{\mathrm{ij\_discharge}} + D_{\mathrm{ij\_charge}} \tag{3}$$

III. RESULTS AND DISCUSSIONS

A. Clustering Results for the Battery Cell Test

With the 100 battery cells from the fast test clustered to the 1111 database batteries, the accuracy of the proposed squeeze algorithm would be revealed from the experiment. The dynamic characteristic of the battery cell is reflected in the parameters nominal voltage and rated capacity. In consequence, if the parameters of the clustered database battery approximate the parameters of the battery cell from the fast test, the clustering result is taken as a successful one. Table 1 shows the clustering results for 10 battery cells from the fast test, with 9 success results. And the match result for the 5-th partial curve is reflected in Fig. 6. For the 100 battery cells as the clustering samples, the final success clustering rate is 86%, which shows that the proposed squeeze algorithm is feasible for the fast testing and sorting system.

TABLE I: THE CLUSTERING RESULTS FOR 10 BATTERY CELLS (THE LENGTH OF PARTIAL CURVE IS SET AS 2000S.)

Partial Curve No.	Clustered Curve No.	Test Cell Parameter (V, Ah)	Clustered Curve Parameter (V, Ah)	Clustering Result
1	884	[3.634, 2.804]	[3.63, 2.80]	Success
2	37	[3.625, 2.035]	[3.63, 2.03]	Fail
3	285	[3.692, 2.253]	[3.69, 2.25]	Success
4	419	[3.604, 2.377]	[3.60, 2.38]	Success
5	230	[3.671, 2.223]	[3.67, 2.22]	Success
6	1053	[3.668, 2.953]	[3.67, 2.95]	Success
7	368	[3.644, 2.326]	[3.64, 2.33]	Success
8	94	[3.653, 2.082]	[3.65, 2.08]	Success
9	449	[3.684, 2.401]	[3.68, 2.40]	Success
10	80	[3.619, 2.073]	[3.62, 2.07]	Success

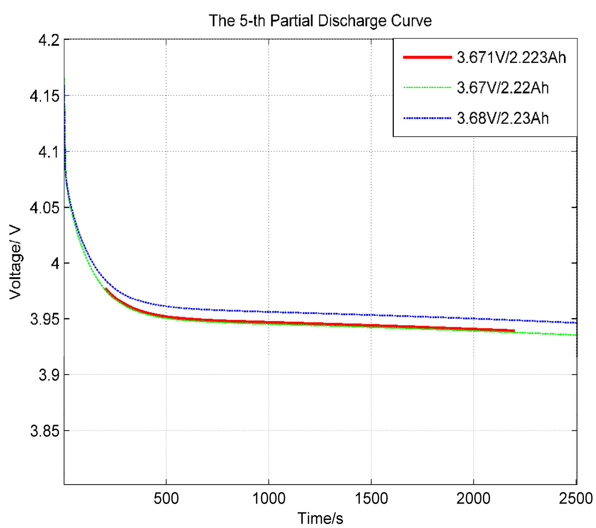
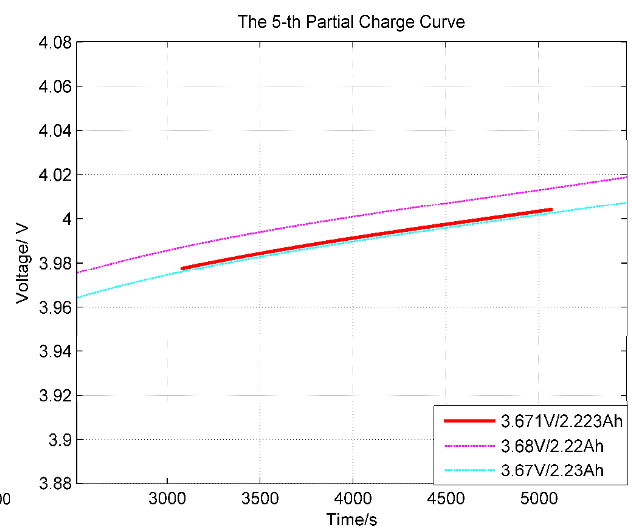


Fig. 6(a). Match Result for the 5-th Partial Discharge Curve.



(b). Match Result for the 5-th Partial Charge Curve.

With different lengths of the partial curve, clustering results may have slight variations. It is analyzed that accuracy of the clustering results corresponds with the matched length, as a longer voltage curve brings a wider match part to get more accurate clustering results. The relationship between the length of the partial curve and the success clustering rate in the experiment is displayed in Table 2. It is obvious that the success clustering rate increases with the length of the partial curve. Taking into account both the fast test time and the success clustering rate, the option of 2000 s as the length of the partial curve is an optimal solution, leading to high efficiency and accuracy at the same time.

TABLE II: THE SUCCESS CLUSTERING RATE WITH DIFFERENT LENGTHS

Length of the Partial Curve (s)	Success Clustering Rate
1000	79%
2000	86%
3000	88%
4000	89%

B. Clustering Results for the Battery Cell Test

The clustering results for the battery cell test shows that the proposed squeeze algorithm works well on battery fast testing and sorting. In the 100 clustering samples, most battery cells would be clustered with the right database battery, while very few cells have small deviations in the clustering results. It is because the partial curve is sampled at the forepart of the discharge characteristic curve and the back of the charge characteristic curve, which are both high-voltage areas. In some cases, the differences among the characteristic curves are reflected at the low-voltage area. As a result, certain samples match with the wrong clustering group, ending up with the clustering error. Besides, if the parameters of the partial curve lie at the middle of the intervals, the distances to the 4 curves in a group are close to each other. It is very likely to choose the wrong curve. Those are the reasons for the clustering errors in the experiment.

The squeeze algorithm is based on an established database of battery characteristic curves. In the experiment, the parameters of the database batteries are set with specified values, which is different from the actual batteries. In real cases, the database characteristic curves could be set up by selecting a series of battery curves from numerous battery tests. The size of the database depends on the requirements of the non-discrepancies of the battery cells to be clustered. For a real battery test, the choice of the charge and discharge current may vary according to the demand of test time. Thus, the fast test time would also change correspondingly.

The whole experiment is conducted by simulation. Although the proposed squeeze algorithm is feasible for the simulation experiment, further work is necessary to verify the application significance of this algorithm. The differences between simulation and actual experiment should be taken into consideration. One is that the Simulink test model is built with ideal circuit elements to get the faultless characteristic curves, while the actual experiment system contains noises and all kinds of errors for the results (influenced by the temperature, for example). The algorithm may need some improvements for better effect. Another problem is that there are various kinds of LIBs in real life. The dynamic characteristics of battery cells may vary with the type or manufacturer, resulting in divergences on characteristic curves.

IV. CONCLUSION

In this work, the previous research on battery clustering has been compared and analyzed. To improve the efficiency, this paper proposes a new algorithm, the squeeze algorithm, to apply in a fast testing and clustering system. A battery test circuit model is established in Simulink to get a database of characteristic curves. For a battery cell to be clustered, it will be charged and discharged for a short time to get a partial curve and then matched with the database curves to get clustered.

The experiment is conducted with 100 partial curves from the fast test and 1111 database complete characteristic curves using the squeeze algorithm. The final success clustering rate reaches 86%, which reflects the feasibility for the proposed algorithm. Compared to the traditional battery clustering methods, this method could save test time to a large extent, with high efficiency and accuracy at the same time.

REFERENCES

[1] G. Pistoia, *Lithium-ion batteries: advances and applications*: Newnes, 2013, pp. 2-19, 346-359.

[2] D.-I. Ra, and K.-S. Han, "Used lithium ion rechargeable battery recycling using Etoile-Rebatt technology," *Journal of Power Sources,* vol. 163, no. 1, pp. 284-288, 2006.

[3] J. Nan, D. Han, and X. Zuo, "Recovery of metal values from spent lithium-ion batteries with chemical deposition and solvent extraction," *Journal of Power Sources,* vol. 152, pp. 278-284, 2005.

[4] T.-S. Dao, C. P. Vyasarayani, and J. McPhee, "Simplification and order reduction of lithium-ion battery model based on porous-electrode theory," *Journal of Power Sources,* vol. 198, pp. 329-337, 2012.

[5] I. Buchmann, and U. h. b. g. d. b. i. Y. Cadex Electronics Inc, *Batteries in a Portable World: A Handbook on Rechargeable Batteries for Non-engineers*: Cadex Electronics, 2001.

[6] P. Raspa, L. Frinconi, A. Mancini, M. Cavalletti, S. Longhi, L. Fulimeni, P. Bellesi, and R. Isidori, "Selection of Lithium Cells for EV Battery Pack Using Self-Organizing Maps," *Automotive Safety and Energy Technology,* vol. 2, pp. 32-39, 2011.

[7] M. Sitterly, L. Y. Wang, G. G. Yin, and C. Wang, "Enhanced identification of battery models for real-time battery management," *IEEE Transactions on Sustainable Energy,* vol. 2, no. 3, pp. 300-308, 2011.

[8] F. Sun, R. Xiong, H. He, W. Li, and J. E. E. Aussems, "Model-based dynamic multi-parameter method for peak power estimation of lithium–ion batteries," *Applied Energy,* vol. 96, pp. 378-386, 2012.

[9] H. He, X. Zhang, R. Xiong, Y. Xu, and H. Guo, "Online model-based estimation of state-of-charge and open-circuit voltage of lithium-ion batteries in electric vehicles," *Energy,* vol. 39, no. 1, pp. 310-318, 2012.

[10] J.-B. Shen, Y.-G. Tang, Y.-J. Li, and Z.-H. Xie, "A new method based on unsupervised clustering for lithium-ion battery classification," *Computers and Applied Chemistry (Chinese),* vol. 24, no. 3, pp. 305-308, 2007.

[11] W.-R. Jin, J. Pang, L. Tang, and Z. Ding, "Research progress in evaluation methods of consistency of li-ion power battery," *Battery Bimonthly (Chinese),* vol. 44, no. 1, pp. 53-56, 2014.

[12] J.-Y. Wang, Z.-C. Sun, X.-Z. Wei, and H.-F. Dai, "Research on power battery cell classification method for electric vehicles," *Chinese Journal of Power Sources (Chinese),* vol. 36, no. 1, pp. 94-98, 2012.

[13] H. Wang, and Y. Zheng, "Storage battery capacity judgment method based on mode identification technology," CN, CN 102221675 A, October 19, 2011.

[14] Z.-H. Duo, G.-C. Li, H. Zhang, and H.-Z. Yan, "Cell classification system based on automatic curve-recognition," *Chinese Journal of Power Sources (Chinese),* vol. 24, no. 2, pp. 99-102, 2000.

[15] Uk.mathworks.com, (2015). *Implement generic battery model - Simulink*. [online] Available: http://uk.mathworks.com/help/physmod/sps/powersys/ref/battery.html?searchHighlight=battery [Accessed 30 March 2016].