Run-to-Run Control for Active Balancing of Lithium Iron Phosphate Battery Packs

Run-to-run control for active balancing of lithium iron phosphate battery packs

Xiaopeng Tang, Changfu Zou, *Member*, *IEEE*, Torsten Wik, Ke Yao, Yongxiao Xia, Yujie Wang, Duo Yang, and Furong Gao

***Abstract*—Lithium iron phosphate battery packs are widely employed for energy storage in electrified vehicles and power grids. However, their flat voltage curves rendering the weakly observable state of charge are a critical stumbling block for charge equalization management. This paper focuses on real-time active balancing of series-connected lithium iron phosphate batteries. In the absence of accurate in-situ state information in the voltage plateau, a balancing current ratio (BCR) based algorithm is proposed for battery balancing. Then, BCR-based and voltage-based algorithms are fused, responsible for the balancing task within and beyond the voltage plateau, respectively. The balancing process is formulated as a batch-based run-to-run control problem, as the first time in the research area of battery management. The control algorithm acts in two timescales, including time-wise control within each batch run and batch-wise control at the end of each batch. Hardware-in-the-loop experiments demonstrate that the proposed balancing algorithm is able to release 97.1% of the theoretical capacity and can improve the capacity utilization by 5.7% from its benchmarking algorithm. Furthermore, the proposed algorithm can be coded in C language with the binary code in 118,328 bytes only and thus is readily implementable in real-time.**

***Index Terms*—Battery management system, active battery balancing, lithium-ion battery, run-to-run control**

I. Introduction

A. Motivation & technical challenges

Lithium-ion batteries are playing a crucial role in electrified transportation and smart grid with renewable energy integration, leading to an efficient and environmentally adapted society [1], [2]. To enable such a revolutionary paradigm shift, the battery's safety, reliability, and cost are critical concerns. Amongst various battery types, lithium iron phosphate ($LiFePO_4$, LFP) batteries are superior in these three factors and thus are widely deployed in today's electromobility market [3]. To satisfy vehicle requirements on power and energy, hundreds or thousands of individual battery cells are needed and connected in series and parallel, forming battery packs. Due to variability/uncertainty in manufacturing and operating conditions, cells in a series-connected pack are inherently different [4], [5]. In turn, different heat generation rates and aging processes in heterogeneous cells can increase the non-uniformity. The overall pack performance is limited by the weakest cell which usually has the lowest actual rated capacity but the highest internal resistance. Such a cell in general is the first to reach and then dominate the cut-off conditions at charging/discharging operations, leading to a substantial amount of energy underutilized in other cells. This fact will inevitably significantly lower the cost-efficiency and incurs heavy and voluminous energy systems.

To address the above problem, a sophisticated battery balancing system to improve pack-level performance by appropriately transferring and coordinating energy among different cells is imperative but technically challenging from at least three aspects.

- To balance the charge of a battery pack, the cell state-of-charge (SoC), defined as the ratio of the remaining capacity and actual rated capacity, is usually required in real-time. However, such a state cannot be measured directly, which makes its online estimation necessary [6], [7].
- LFP batteries have very flat voltage curves in a large working range, e.g., 20-90% SoC levels, due to their electrochemical properties [8], [9]. Under this circumstance, it is difficult to estimate the SoC from noisy voltage measurements and imperfect battery models [10].
- Estimating each cell's states in real-time can cause a significant computational burden to microprocessors of battery management systems (BMSs) [11], [12] and may be infeasible for many applications.

B. Literature review

To overcome existing challenges, battery balancing techniques in the literature can be classified into two types based on hardware design, i.e., passive dissipation and active balancing. In passive dissipation based configurations, the portion of cell-level energy above that of the lowest cell is all consumed through resistors or transistors [13]. The balancing

Manuscript received October 25, 2018; revised March 05, 2019; and accepted May 24, 2019. This work was supported by the National Natural Science Foundation of China project (61433005), Hong Kong Research Grant Council (16207717), Guangdong Scientific and Technological Project (Grant 2017B010120002), and Swedish Energy Agency (Grant 42787-1). (*Corresponding authors are Changfu Zou and Furong Gao.*)

X. Tang and F. Gao are with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, and F. Gao is also with Guangzhou HKUST Fok Ying Tung Research Institute, Guangzhou 511458, China (xtangai@connect.ust.hk, kefgao@ust.hk).

C. Zou and T. Wik are with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg 41296, Sweden (changfu.zou@chalmers.se, tw@chalmers.se).

Y. Wang and D. Yang are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (wangyujie@ustc.edu.cn, yd1995@mail.ustc.edu.cn).

K. Yao and Y. Xia are with the Guangzhou HKUST Fok Ying Tung Research Institute, Guangzhou 511458, China (keyao@ust.hk, yxxia@ust.hk).

currents are typically limited by temperature constraints of hardware, and the energy efficiency is regarded as zero [14]. These configurations are still widely used because of their simple system structure and high reliability. By contrast, active balancing configurations transfer energy among battery cells until some equalization state is reached. Obviously, such a class of balancing approaches are energy-efficient and have the potential to achieve a balancing state quickly [11]. In this regard, various active balancing protocols have been proposed using different hardware components, including inductor based [15], capacitor based [16], and bi-directional DC-DC converter based [17].

Control algorithms are of great importance to use the hardware for balancing battery cells in an efficient way. Among a number of existing control schemes, SoC-based balancing is widely used by moving capacity from cells with higher SoC levels to the lower ones [18]–[20]. The effectiveness of SoC enabled equalization techniques have been validated against $LiNiMnCoO_2$ batteries [21], $LiNiCoAlO_2$ batteries [22], and NiMH batteries [23]. However, these techniques heavily rely on the accuracy of SoC estimation for each battery cell in the pack. Although considerable research efforts have been made to improve the state estimation performance, such as in [24], the weak observability and computational issues have not been addressed for LFP batteries rigorously and comprehensively.

To reduce the computational cost, voltage-based balancing strategies have been used to circumvent SoC estimation. Different to SoC-based strategies is that voltage-based strategies transfer energy among battery cells based on their voltage difference [25], [26]. Voltage-based strategies are widely used in both passive and active balancing structures, thanks to their capability in realizing charge equalization with simple hardware and quick response. However, it should be noted that these strategies are often inferior in balancing performance, such as capacity utilization, relative to SoC-based ones under the same condition. This is because cells with the same SoC can have different internal resistances and then maintain dissimilar terminal voltages.

Recently, some sophisticated control schemes, such as model predictive control (MPC), were emerged for battery equalization, in which both the current states (voltage and/or SoC) and prediction of the future ones are considered in a real-time optimization problem [27], [28]. In the case where states can be either accurately measured or estimated, these control schemes demonstrated enhanced performance at the sacrifice of massive computational resources and time. However, this work aims to balance LFP batteries in embedded systems with limited computational power and without requiring estimation of any internal battery state.

Repetitive processes are common in our daily life and industrial operations, for instance, thermoplastic injection molding that repeatedly makes the same plastic part [29]. By defining each repetitive process as a batch, batch control methodologies enhance system performance by using both the current information and information learned from previous batches via some feed-forward control schemes. Run-to-run control is a typical batch control algorithm and well suits processes with relatively bad repetitive nature (i.e., repeatability among a sequence of batches is low) [29]. It was initially proposed by Sachs et al. [30] in the early 1990s and then was extended with a number of variants. Comparisons of different run-to-run control algorithms can be found in [31]. These algorithms are especially motivated by the control processes in which the product qualities of interest cannot be measured in real-time, particularly during the batch run, but some interested states can be measured at the end of each batch. As a consequence, the control has to be operated in an open-loop fashion within each batch and gets update recursively using the end-of-batch information. Run-to-run control has been used in numerous areas successfully [29], [32], [33] and can be potentially useful to solve the battery balancing problem, where charging, discharging, and relaxed modes are operated recursively.

C. Key contributions

The primary purpose of this paper is to propose a computationally and energy efficient balancing strategy for LFP batteries connected in series. Four original contributions made in this work distinguish it from the relevant literature. First, the concept of balancing current ratio (BCR) is proposed as a battery state, alternative to SoC and terminal voltage, for battery pack balancing management. Next, BCR-based and voltage-based balancing algorithms are pertinently combined to realize battery charge equalization within and beyond the voltage plateau, respectively. Then, the overall balancing problem is formulated in the framework of batch-based run-to-run control, for the first time. Finally, the developed balancing control algorithm is experimentally implemented via hardware-in-the-loop facilities with a low-cost microcontroller unit (MCU) with 128kB flash memory only. Extensive battery tests are performed to examine the performance of the proposed algorithm and its comparison with the conventional voltage-based algorithm.

D. Paper organization

The remaining content is arranged as follows. Section II introduces some important concepts and the balancing hardware topology. The time-wise and batch-wise balancing control algorithms are formulated and explained in Section III. Experimental validation and discussions on the result are presented in Section IV, followed by a concluding summary in Section V.

II. Concept definition and hardware

This section first mathematically defines and explains concepts of the SoC, state-of-health (SoH), batches, and balancing current ratio to be used throughout this paper. Then, the topology of battery balancing hardware is introduced.

A. Concept definition

SoC and SoH. For series-connected battery cells in a pack, the SoC of cell j at the sampling step k can be derived by

$$SoC_j(k) = SoC_j(0) + \sum\nolimits_{l=0}^{l=k} \frac{\eta \Delta t \left[I(l) + i_j(l)\right]}{Q_j} \quad (1)$$

where $SoC_j(0)$ is the battery cell's initial SoC. I and i are currents of the battery pack and the balancing electric circuit, respectively. Δt is the sampling time and is specified as 1 second in this work. Q_j is the cell's actual rated capacity, and η is the coulombic efficiency. The currents are defined to be positive for charging and negative for discharging.

For cell j, its SoH is defined as the ratio between Q_j and the rated capacity Q_{rated} on the datasheet provided by its manufacturer, where Q_{rated} corresponds to the rated capacity at the beginning of service life. Consequently, there exists

$$SoH_j = Q_j/Q_{rated}. \tag{2}$$

Batches. Before defining batches, we distinguish the state of a battery pack by its average operating current $I_{avg}(M,k)$ over the time period $[k-M+1,k]$, namely

$$I_{avg}(M,k) = \sum\nolimits_{l=k-M+1}^{l=k} I(l)/M. \tag{3}$$

For a small positive real number ϵ, if $I_{avg}(M,k) > \epsilon$, the battery pack is in its *refueling mode*; if $I_{avg}(M,k) < -\epsilon$, it is in its *working mode*; and it is a *waiting mode* when $-\epsilon \leq I_{avg}(M,k) \leq \epsilon$. Here ϵ is used to compensate for uncertainties due to current sensor drifting and noise. M is a time window for evaluating the average applied/supplied current. Note that at the waiting mode, the battery is close to or already on a relaxed state.

Based on the above prescript modes, different batches can be defined for battery operation. Explicitly, a batch includes the current trajectory across a time period whose starting and terminal points are determined by the transition moment between two contiguous operating modes. For example, for a *working batch*, its starting point is defined as the time when the operation changes from a waiting/refueling mode into a working mode, and its endpoint is the time when the battery operation shifts to a waiting/refueling mode. The *refueling batch* and *waiting batch* are defined in a similar way. It is worth mentioning that these batch definitions are general so that applications in electrified vehicles and stationary energy storage can all be covered. In addition, the *end-of-batch voltage* and *end-of-batch SoC* are defined as the battery terminal voltage and SoC at the end of a batch, respectively. The problem to be solved in this work is to balance battery cells for refueling and working batches.

Balancing current ratio. The balancing current ratio (BCR) of cell j at time step k in a batch is defined as:

$$BCR_j(k) = \sum\nolimits_{l=0}^{l=k} i_j(l)/\left[\sum\nolimits_{l=0}^{l=k} I(l)\right]. \tag{4}$$

It is easy to know that for refueling and working batches, the denominator of the right-hand side of (4) will not be zero. This concept is introduced for battery balancing control, as an alternative to terminal voltage or SoC information.

B. Topology of battery balancing hardware

The topology of balancing hardware to be studied for lithium-ion batteries is illustrated in Fig. 1. In this structure, we aim to balance N series-connected battery cells through an isolated bi-directional DC-DC converter, MCU, and switch array. The balancing current i is not measured in most commercial BMSs to avoid increasing cost and system complexity. Currently, many commercialized battery equalizers support the output of constant currents, e.g., LTC3300, LTC8584, and EMB1499. With these two factors in mind, the converter's nominal output current denoted i_{nom} (a positive value) is used as the balancing current. Any single cell can be connected to the battery pack via the converter by controlling the switch array. The current direction can be decided flexibly to realize charge or discharge operation.

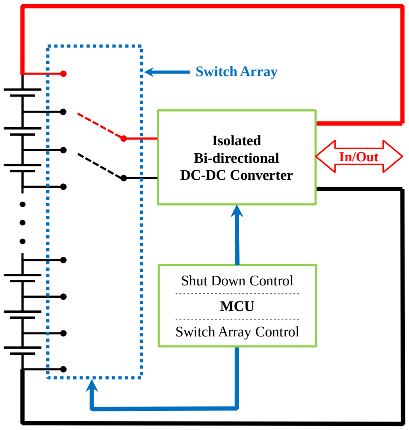


Fig. 1. Topology of the balancing hardware for lithium-ion batteries.

Suppose no energy loss and that the voltage of cell j is $1/N$ of the pack voltage, the summation of every cell's current for balancing is zero. For a cell being charged by the balancing hardware, it can be easily calculated according to Kirchhoff's current law that the branch balancing current is $i_{\text{nom}}(N-1)/N$. In this case, the balancing current supplied by the remaining cells is $-i_{\text{nom}}/N$. Similarly, for a cell being discharged, its branch balancing current is $-i_{\text{nom}}(N-1)/N$, and at the same time, the remaining cells absorb i_{nom}/N for charging. Without a doubt, when the hardware is not working, the balancing current for each cell is zero.

III. The Proposed Balancing Algorithm

This section exposes the main technical contributions of this paper. A novel battery balancing algorithm is proposed by leveraging a batch control methodology, in which control actions are implemented in two time-scales, namely time-wise control and batch-wise control. In particular, while the time-wise control is responsible for balancing batteries within a batch, the batch-wise control comes into play at the end of each batch. Each ingredient of this algorithm is motivated and explained step by step in the following.

A. Time-wise control

Time-wise control is realized within each batch through a hybrid balancing protocol, fusing voltage-based and BCR-based methods. The BCR-based method is proposed to balance LFP batteries inside the voltage plateau, where the SoC is weakly observable from voltage measurements. Once battery operation is beyond the voltage plateau, a voltage-based balancing method is employed.

For the hardware specified in Section II-B, the balancing current vector for all the N batteries is denoted $\mathbf{i} := [i_1, ..., i_N]$. To simplify the problem description, the state of the battery pack to control is denoted $\mathbf{x}$, which for example can be SoC, terminal voltage, and BCR. $\mathbf{x}^{ref}$ is the battery's reference state. Based on these notations, the general time-wise balancing control at time step k can be formulated as a constrained tracking problem

$$\min_{\mathbf{i}(k)} \left\|\mathbf{x}^{ref}(k) - \mathbf{x}(k)\right\|_\infty \tag{5a}$$

$$s.t.\ i_j(k) \in \{i_{\text{nom}}(1-N)/N, -i_{\text{nom}}/N, \\ 0, i_{\text{nom}}/N, i_{\text{nom}}(N-1)/N\} \tag{5b}$$

$$i_1(k) + i_2(k) + \cdots + i_N(k) = 0 \tag{5c}$$

where $j = 1, \cdots, N$, and $\|\cdot\|_\infty$ is the infinity norm. By using (5a), the difference between the weakest and strongest cells in the pack can be minimized. Now it is positioned to embody the general balancing algorithm based on certain real-time battery information. For the purpose of comparison and motivating other algorithms, the classical SoC-based balancing is introduced first.

1) SoC-based balancing algorithm:

Assumption 1: At each time step k, the SoC of each cell in the battery pack can be acquired accurately.

Once *Assumption 1* is valid, the state vector $\mathbf{x}$ in the general problem formulation (5) can be replaced by in-situ SoC information of all battery cells, namely, $\mathbf{x} = \mathbf{SoC} := [SoC_1, \cdots, SoC_N]$. The SoC tracking reference, $\mathbf{SoC}^{ref} = [SoC_1^{ref}, \cdots, SoC_N^{ref}]$, can be given by the average SoC of these battery cells

$$SoC_j^{ref}(k) = \sum\nolimits_{j=1}^{j=N} SoC_j(k)/N. \tag{6}$$

By implementing the above SoC-based balancing algorithm, the SoC values of different cells will equal to each other when the battery pack state converges to its quasi-steady state at time step $k+h$. Similar statements can be found, e.g., in [34]. We define this quasi-steady state as the ideal pack SoC, SoC_{pack}, which can be calculated in the form of

$$\begin{aligned} SoC_{pack}(k) &:= \sum\nolimits_{j=1}^{j=N} C_j(k) \left[\sum\nolimits_{j=1}^{j=N} Q_j\right]^{-1} \\ &= \sum_{j=1}^{j=N} SoH_j SoC_j(k) \left(\sum\nolimits_{j=1}^{j=N} SoH_j\right)^{-1} \end{aligned} \tag{7}$$

where $C_j(k)$ is the available capacity of cell j at time k. Then, there exists an optimal SoC, SoC_j^*, satisfying

$$SoC_j^*(k+h) = SoC_{pack}(k+h),\ \forall j \in [1, \cdots, N]. \tag{8}$$

Although the SoC-based algorithm has been introduced, it may not be applicable for LFP batteries because *Assumption 1* is hard to always hold within the voltage plateau. Indeed, this algorithm will be used as a valuable test bed and a starting point for other balancing algorithms.

2) Voltage-based balancing algorithm: As stated in Section I, the voltage-based balancing algorithm is preferred due to its simpleness in obtaining measurements and implementation. This algorithm is selected to balance batteries beyond the voltage plateau, and its underlying assumption is explicitly stated here.

Assumption 2: Batteries in a pack having the same terminal voltage maintain the same SoC.

Based on *Assumption 2*, the voltage-based balancing algorithm can approximate the SoC-based algorithm and finally leads to the steady state in (7)-(8). By using the general formulation in (5), the key idea and procedure of the voltage-based algorithm are presented to complete the overall balancing objective. The battery pack state is defined as $\mathbf{x} = \mathbf{V} := [V_1, ..., V_N]$, and the reference state, $\mathbf{V}^{ref} = [V_1^{ref}, ..., V_N^{ref}]$, is calculated as

$$V_j^{ref}(k) = \sum\nolimits_{j=1}^{N} V_j(k)/N,\ \forall j \in [1, \cdots, N]. \tag{9}$$

In this way, the battery balancing problem becomes a voltage tracking problem. By defining the tracking errors as

$$\boldsymbol{\beta}^V(k) := \mathbf{V}^{ref}(k) - \mathbf{V}(k) \tag{10a}$$

$$\beta_{\max}^V(k) := \max\{\boldsymbol{\beta}^V(k)\},\ \beta_{\min}^V(k) := \min\{\boldsymbol{\beta}^V(k)\} \tag{10b}$$

the procedure to implement the balancing control algorithm is proposed as follows:

Procedure 1:

1) Obtain the voltage tracking error $\boldsymbol{\beta}^V(k)$ and calculate $\beta_{\max}^V(k)$ and $\beta_{\min}^V(k)$;
2) If $\beta_{\max}^V(k) - \beta_{\min}^V(k) < 10$ mV, the balancing hardware does not come into work; else
3) If the battery pack is being discharged, then charge the cell corresponding to $\beta_{\max}^V(k)$; else
4) If the battery pack is being charged, then discharge the cell corresponding to $\beta_{\min}^V(k)$;
5) Increment k to $k+1$ and go to Step 1).

In the above procedure, the threshold of 10mV is specified for the equalization task by following [35]. This is to sidestep instability caused by too frequent switching between charge and discharge operations in the presence of voltage measurement noise.

3) BCR-based balancing algorithm: In the voltage plateau of LFP batteries, both *Assumptions 1* and *2* can be violated so that SoC-based and voltage-based balancing algorithms are not effective any longer. To resolve this problem, an algorithm armed with BCR information defined in (4) is sought to balance battery cells. By again following the general problem formulation in (5), the state of a battery pack in a batch is defined as $\mathbf{x} = \mathbf{BCR} := [BCR_1, ..., BCR_N]$, and the reference state is $\mathbf{x}^{ref} = \mathbf{BCR}^{ref} := [BCR_1^{ref}, ..., BCR_N^{ref}]$. The value of $\mathbf{BCR}^{ref}$ cannot be specified intuitively as the SoC- or voltage-based balancing methods. Its determination is guided by the following proposition.

Proposition 1: If each battery's SoH is available within a batch and if the referenced BCR, $\mathbf{BCR}^{ref}$, is equal to its optimal value, $\mathbf{BCR}^*(k) := [BCR_1^*(k), ..., BCR_N^*(k)]$, with each element satisfying

$$BCR_j^{ref} = BCR_j^* := \frac{N \cdot SoH_j}{\sum_{j=1}^{N} SoH_j} - 1 \tag{11}$$

$\forall j \in [1, \cdots, N]$, then the BCR-based balancing algorithm is equivalent to the SoC-based algorithm stated in *Section III-A1*.

Proof 1: By defining ΔC_j and ΔC_{pack} as the capacity changes of cell j and the battery pack from time step k to $k+h$ respectively, it can be inferred from (1), (7), and (8) that the pack SoC change is equal to the cell SoC change from time step k to $k+h$. This leads to

$$\frac{\Delta C_{pack}}{Q_{pack}} = \frac{\Delta C_j}{Q_j} \text{ and } \frac{\Delta C_j}{\Delta C_{pack}} = \frac{Q_j}{Q_{pack}} \tag{12}$$

where Q_{pack} is the pack capacity and is defined as the average value of the rated capacity of all cells in the battery pack.

The relationship between BCR_j^* and the two capacity changes (ΔC_j and ΔC_{pack}) can be found from the BCR definition of the balancing process given in (4). As a result, there exists

$$\begin{aligned} BCR_j^{ref} &= \sum\nolimits_{l=k}^{l=k+h} i_j(l) \left[\sum\nolimits_{l=k}^{l=k+h} I(l)\right]^{-1} \\ &= \frac{\Delta C_j - \Delta C_{pack}}{\Delta C_{pack}} = \frac{\Delta C_j}{\Delta C_{pack}} - 1. \end{aligned} \tag{13}$$

Substituting (2) and (12) into (13) results in

$$\begin{aligned} BCR_j^{ref} &= \frac{Q_j}{Q_{pack}} - 1 = \frac{Q_j / Q_{rated}}{\sum_{j=1}^{N} Q_j / (N \cdot Q_{rated})} - 1 \\ &= \frac{N \cdot SoH_j}{\sum_{j=1}^{N} SoH_j} - 1. \end{aligned} \tag{14}$$

Up to this point, the proof of *Proposition 1* is completed.

It can be known from *Proposition 1* that $\mathbf{BCR}^*$ and $\mathbf{BCR}^{ref}$ are independent of the time step k within each batch. Similarly to the voltage-based balancing method, the BCR tracking errors are defined as

$$\boldsymbol{\beta}^{BCR}(k) := \mathbf{BCR}^{ref} - \mathbf{BCR}(k) \tag{15a}$$

$$\beta_{\min}^{BCR}(k) := \min\{\boldsymbol{\beta}^{BCR}(k)\}. \tag{15b}$$

The following four-step procedure is used to implement the BCR-based algorithm:

Procedure 2:

1) Obtain the BCR tracking error $\boldsymbol{\beta}^{BCR}(k)$ and calculate $\beta_{\min}^{BCR}(k)$;
2) If the battery pack is being discharged, then charge the cell (whose BCR is negative) corresponding to $\beta_{\min}^{BCR}(k)$; else.
3) If the battery pack is being charged, then discharge the cell corresponding to $\beta_{\min}^{BCR}(k)$;
4) Increment k to $k+1$ and go to Step 1).

The BCR-based balancing algorithm has been presented, and its validity and practicability are discussed as follows.

Remark 1: The effectiveness of the proposed BCR-based algorithm depends on (11) in *Proposition 1*, where the central premise is to know every battery's SoH accurately. However, batteries, for example in EV and hybrid EV applications, are often only partially discharged/charged in daily usage due to vehicle range anxiety or conservative design for battery health protection. Due to this together with the noisy nature of current measurements, it is difficult to achieve the true SoH information if directly follow (1)-(2). Meanwhile, highly accurate estimation for SoH of each cell in the pack is an active research topic and can be technically challenging or computationally expensive.

Remark 2: As stated earlier, the balancing current is usually not measured in BMSs to keep a low cost and hardware complexity. Therefore, an engineering compromise is to use the nominal balancing current provided in the datasheet to calculate the BCR. In this way, errors will be introduced inevitably to the BCR calculation and then may degrade the balancing performance.

To circumvent the issues concerning the absence of SoH information and the uncertainty in BCR, a run-to-run control algorithm within the batch control framework is proposed for LFP battery balancing in the next subsection.

B. Batch-wise tracking algorithms

A batch-wise run-to-run control algorithm is proposed for $\mathbf{BCR}^{ref}$ so that it can converge quickly to the optimal value $\mathbf{BCR}^*$ defined in (11) without calculating SoC or SoH. In principle, run-to-run control is an open-loop based approach in time-wise but closes up the loop in batch-wise based on the repetitive nature of processes. A thorough exposition of run-to-run control can be found in [29], [31]–[33]. In this subsection, the main control theory and its underlying assumptions are presented first, followed by an application to solve the LFP battery balancing problem.

For a repetitive dynamic process, $\boldsymbol{u}$ is the control input vector, which typically has a low number of elements, denoted $\boldsymbol{u} \in \mathbb{R}^n$. $\boldsymbol{y}$ is the process output vector with the same size as $\boldsymbol{u}$, and a square dynamic system is considered.

Assumption 3: There exists a static map between $\boldsymbol{u}$ and $\boldsymbol{y}$.

Assumption 4: For the optimal point $(\boldsymbol{u}^*, \boldsymbol{y}^*)$ on the map $\boldsymbol{u} \to \boldsymbol{y}$, if $\boldsymbol{u} = \boldsymbol{u}^*$, then $\boldsymbol{y} = \boldsymbol{y}^*$.

Assumption 5: The value of $\boldsymbol{u}$ can be obtained at the beginning of each batch.

Assumption 6: $\boldsymbol{y}$ is available through measurements or observers at the end of each batch.

Theorem 1: Let *Assumptions 3-6* all hold, and if there exist a full-rank $n \times n$ matrix $\mathbf{G}$, a positive scalar α, and the control gain γ, and the following two inequalities hold

$$||\boldsymbol{y}^* - \boldsymbol{y}||_2 < \alpha(\boldsymbol{y}^* - \boldsymbol{y})^T \mathbf{G}(\boldsymbol{u}^* - \boldsymbol{u}), \ \forall \boldsymbol{u} \neq \boldsymbol{u}^* \tag{16a}$$

$$0 < \gamma < 2/\alpha \tag{16b}$$

where $||\cdot||_2$ is the 2-norm, then the batch-wise updating law for $\boldsymbol{u}$ with the form of

$$\boldsymbol{u}_{s+1} = \boldsymbol{u}_s + \gamma \mathbf{G}^{-1}(\boldsymbol{y}^* - \boldsymbol{y}_s) \tag{17}$$

can guarantee $\boldsymbol{u}_s \to \boldsymbol{u}^*$ and $\boldsymbol{y}_s \to \boldsymbol{y}^*$ as $s \to \infty$.

The proof and comprehensive interpretation of *Theorem 1* can be found in [33] and are ignored here for brevity. The emphasis is placed on its application to balance battery packs.

For the dynamic process of a battery pack across successive batches, $\mathbf{BCR}^{ref}$ is considered as the process input and the battery state of charge can be the process output. In this case, the goal is to steer $\mathbf{BCR}^{ref}$ to its optimal value $\mathbf{BCR}^{*}$ using the error between $\mathbf{SoC}$ and its optimal value $\mathbf{SoC}^{*}$ defined in (7)-(8). The initial time and terminal time of any batch are assigned to 0 and t_1, respectively. By following (17), for a given $\mathbf{BCR}_1^{ref}$, the referenced BCR for batch $s+1$ can be derived at the end of batch s in the form of

$$\begin{aligned}\mathbf{BCR}_{s+1}^{ref} =& \mathbf{BCR}_{s}^{ref} \\ &+ \gamma \mathbf{G}^{-1}\left[\mathbf{SoC}^{*}(t_1) - \mathbf{SoC}_s(t_1)\right]. \end{aligned} \quad (18)$$

Before using the run-to-run control, an additional assumption is applied, and then all these assumptions are justified or remarked on the battery system.

Assumption 7: The SoC changing range remains unchanged at different batches.

Justification 1 (for Assumption 3): The relationship of $\mathbf{BCR}^{ref}$ and $\mathbf{SoC}(t_1)$ can be derived from (1) and (4). As a result, for each cell j, there exists

$$\begin{aligned}SoC_j(t_1) = SoC_j(0) &+ \frac{\eta\Delta t}{Q_j}\sum\nolimits_{l=0}^{l=t_1} I(l) \\ &+ \frac{\eta\Delta t}{Q_j} BCR_j(t_1)\sum\nolimits_{l=0}^{l=t_1} I(l). \end{aligned} \quad (19)$$

Based on *Assumption 7* and (19), it can be concluded that there exists a static map between $\mathbf{BCR}^{ref}$ and $\mathbf{SoC}(t_1)$.

Justification 2 (for Assumption 4): This assumption implies that there exist optimal SoC and BCR, and if $\mathbf{BCR}^{ref} = \mathbf{BCR}^{*}$, then $\mathbf{SoC}_s(t_1) = \mathbf{SoC}^{*}(t_1)$. While the existence of optimal values is naturally satisfied according to (11) and (8), the second condition is valid based on *Proposition 1*.

Remark 3 (for Assumptions 5-6): To satisfy *Assumption 5* according to (18) and satisfy *Assumption 6*, one has to estimate SoC and SoH of each battery cell for obtaining $\mathbf{SoC}_s(t_1)$ and $\mathbf{SoC}^{*}(t_1)$. However, it is important to avoid performing this task because estimating SoC and SoH gives rise to a large computational burden while the estimation itself may not be accurate.

Remark 4 (for Assumption 7): The SoC changing range of lithium-ion batteries is dependent on practical utilization and user habits. For cycling operation of batteries in electric vehicles (EVs) and hybrid EVs, similar driving schedules such as urban dynamometer driving schedule (UDDS) are often performed repeatedly. In a general sense, the users tend to recharge their batteries from the same lower level to the high one, e.g., 10-90%. Therefore, *Assumption 7* can be plausible and largely simplifies the control implementation. Furthermore, run-to-run control often works well in practice for processes with relatively bad repetitive nature [29].

In (18), γ and $\mathbf{G}^{-1}$ are two tuning parameters. Increasing their values makes the batch-wise control converge faster but more sensitive to the voltage error. The choice of these two parameters can be guided by (16). By setting $\alpha = 1$ and $\gamma = 1$ that satisfies (16b), the selection of $\mathbf{G}$ can follow the proposition:

Proposition 2: For a full-rank $\mathbf{G} = diag\{G_1, \cdots, G_N\} \in \mathbb{R}^{N\times N}$, and for each $j \in [1, \cdots, N]$, if there exist

$$G_j \geq \frac{\eta\Delta t}{Q_j}\sum\nolimits_{l=0}^{l=t_1} I(l), \text{ for refueling batches} \quad (20a)$$

$$G_j \leq \frac{\eta\Delta t}{Q_j}\sum\nolimits_{l=0}^{l=t_1} I(l), \text{ for working batches} \quad (20b)$$

then (16) can always be satisfied.

Proof 2: Based on *Assumption 3* and *Justification 1*, the following two equations can be established

$$BCR_j^{*} = \frac{SoC_j^{*}(t_1) - SoC_j(0) - \frac{\eta\Delta t}{Q_j}\sum_{l=0}^{l=t_1} I(l)}{\frac{\eta\Delta t}{Q_j}\sum_{l=0}^{l=t_1} I(l)} \quad (21)$$

$$BCR_j^{ref} = \frac{SoC_j(t_1) - SoC_j(0) - \frac{\eta\Delta t}{Q_j}\sum_{l=0}^{l=t_1} I(l)}{\frac{\eta\Delta t}{Q_j}\sum_{l=0}^{l=t_1} I(l)}. \quad (22)$$

Subtracting (21) from (22) results in

$$BCR_j^{*} - BCR_j^{ref} = \frac{SoC_j^{*}(t_1) - SoC_j(t_1)}{\frac{\eta\Delta t}{Q_j}\sum_{l=0}^{l=t_1} I(l)}. \quad (23)$$

For all $BCR_j^{ref} \neq BCR_j^{*}$, based on (23), we can easily derive

$$\begin{aligned}&\left(SoC_j^{*} - SoC_j\right) G_j (BCR_j^{*} - BCR_j^{ref}) \\ &\quad = \left(SoC_j^{*} - SoC_j\right)\frac{G_j\left(SoC_j^{*} - SoC_j^{ref}\right)}{\frac{\eta\Delta t}{Q_j}\sum_{l=0}^{l=t_1} I(l)} \\ &\quad \geq \left(SoC_j^{*} - SoC_j\right)^2. \end{aligned} \quad (24)$$

The right-hand side of (20) represents a battery cell's SoC change. Since the SoC change in a batch is always within the set $[-1, 1]$, G_j can be chosen as 1 for refueling batches and -1 for working batches to satisfy (20).

Based on *Remark 3*, the voltage tracking error and a full-rank matrix $\bar{\mathbf{G}}^{-1}$ are adopted to approximate (18) to decrease the computational cost, leading to

$$\mathbf{BCR}_{s+1}^{ref} = \mathbf{BCR}_{s}^{ref} + \gamma\bar{\mathbf{G}}^{-1}\left[\mathbf{V}^{ref}(t_1) - \mathbf{V}_s(t_1)\right]. \quad (25)$$

To ensure (25) is equivalent to (18), $\bar{\mathbf{G}} = diag\{\bar{G}_1, \cdots, \bar{G}_N\}$ can be specified as

$$\bar{G}_j = G_j\frac{V_j^{ref}(t_1) - V_j(t_1)}{SoC_j^{ref}(t_1) - SoC_j(t_1)}. \quad (26)$$

In the battery model-free approach, it is difficult to calculate $\bar{G}_j$ accurately from (26). Considering the open circuit voltage U is a static function of SoC, $dU/dSoC$ can be used to approximate $(V_j^{ref} - V_j)/(SoC_j^{ref} - SoC_j)$. Consequently, $\bar{G}_j$ is selected according to

$$\bar{G}_j \geq (dU/dSoC)_{\max} + \delta, \text{ for refueling batches} \quad (27a)$$

$$\bar{G}_j \leq -(dU/dSoC)_{\max} - \delta, \text{ for working batches} \quad (27b)$$

where δ is a positive parameter as a margin to compensate for the error introduced by the approximation. If one can select appropriate values for $(dU/dSoC)_{\max}$ and δ based on the static function $U(SoC)$, the computation of BCR via (25) becomes a trivial task. $\bar{G}_j$ will affect the designed control in both its convergence rate and sensitivity to voltage measurement noise. Increasing its magnitude will be instrumental to noise suppression but reduces the convergence rate, and vice versa.

When using the control algorithm (25), the integral overflow can be triggered if the balancing current required to track $\mathbf{BCR}^{ref}$ exceeds the limitation of hardware. To address this issue, the BCR calculated at the end of the corresponding batch via (4) can be used in (25). In this case, the batch-wise updating law for $\mathbf{BCR}^{ref}_{s+1}$ becomes

$$\mathbf{BCR}^{ref}_{s+1} = \mathbf{BCR}_s(t_1) + \gamma\bar{\mathbf{G}}^{-1}\left[\mathbf{V}^{ref}(t_1) - \mathbf{V}_s(t_1)\right]. \tag{28}$$

For the employed run-to-run control with the gain determined via (27), a rigorous analysis of its overall performance can be difficult to conduct. This is due to the employed approximation of (25) to (28) and potential violations of *Assumptions 5-7*. As a result, it is not straightforward whether the proposed algorithm can inherit the convergence property provided in *Theorem 1*. Alternatively, numerical examination through battery-in-the-loop experiments will be performed in Section IV.

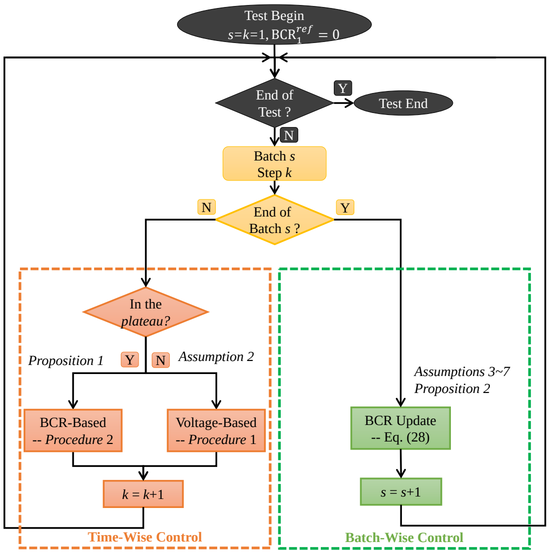


Fig. 2. Schematic of the proposed balancing algorithm for LFP batteries.

C. Summary

So far, the batch control based balance algorithm has been presented for LFP batteries, in which the control actions are taken both time-wisely and batch-wisely. This algorithm relies on BCR information within the plateau but voltage information outside of the plateau. It is completely model-free and does not depend on any battery's SoC and SoH, which significantly lowers the computational and memory requirements in BMS microprocessors. A flowchart to summarize the proposed balance algorithm is shown in Fig. 2.

IV. Experimental implementation

The proposed run-to-run control based balancing algorithm in the previous section is now implemented for real-world LFP batteries. Its effectiveness is characterized quantitatively, followed by discussions on the practicability.

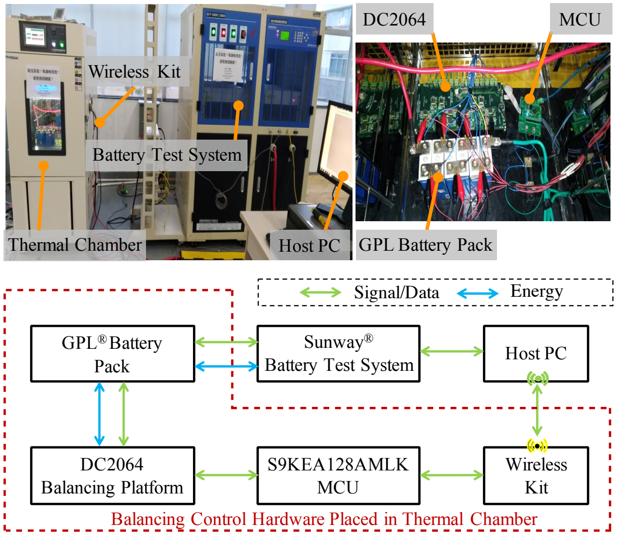


Fig. 3. Top: Photograph of the balancing hardware. Bottom: Schematic of the experimental battery tester.

A. Specifications

1) Hardware configurations: The balancing hardware platform and experimental schematic are shown in Fig. 3. The hardware included six components, including a battery pack, battery test system, host PC, balancing platform, microcontroller unit (MCU), and wireless kit. In details, the battery pack constituted 6 LFP battery cells connected in series, and each had 20Ah rated capacity and a 3.2V nominal voltage. The balancing platform, armed with a DC2064 demo board produced by Linear Technology [36], was a monitor and actuator controlled by a low-cost MCU. DC2064 is, in fact, powerful to balance multiple cells simultaneously but has been artificially designed to manipulate one cell only at a time in the experiments. By doing so, the obtained balancing hardware topology is equivalent to Fig. 1, that is typically used in commercialized BMSs thanks to its simplicity and low cost. The MCU communicated with the host computer through a blue-tooth based wireless communication kit.

To generate measurements, the module LTC6803 embedded in the DC2064 platform recorded the battery voltage, while the pack current was measured by a 2.5mΩ shunt and a current amplifier, INA199 produced by Texas Instruments (TI). The isolated DC-DC converters were controlled by the module LTC3300. The converters' nominal balancing current is 2.4A for discharging and 2.2A for charging, which was used to calculate the BCR in (4). The batteries, DC2064 demo board, MCU, and wireless kit were all placed in a thermal chamber with a fixed temperature of 25°C. The battery pack was charged and discharged through a Sunway battery test system, which was also controlled by the host computer.

The detailed specifications of the main hardware components are summarized in Table I.

TABLE I
HARDWARE SPECIFICATIONS.

Name	Model	Specifications
Battery test system	Sunway CE-7001	Pack: 500V, 100A, 0.1% accuracy; Single cell voltage: 5V, 0.1% accuracy.
Thermal chamber	HongZhan	25±2 °C fixed.
Battery pack	GPL Battery	6S1P LFP cells.
Balancing platform	Linear DC2064	Nominal balancing current: 2.4A for discharge, 2.2A for charge; Voltage sampling accuracy: ±1.5mV at room temperature; Nominal efficiency through the flyback converter: 92%.
MCU	NXP S9KEA128AMLK	Core: Cortex M0+ @48MHz; ADC resolution: 12bit SAR at 64 times over sampling = 16bit.
Voltage reference	Linear LTC6658	Channel 1: 5.0V±0.05% (trimmed) with source current of 150mA; Channel 2: 2.5V±0.05% (initial) with source current of 50mA.
Current sensor	ZYCN	2.5mΩ, 0.5% accuracy at full temperature range.
Current amplifier	TI INA199	>0.5% accuracy at room temperature (trimmed).

2) Load, parameters, and initialization: The safety voltage level for every single cell was set as 3.65V for charging and 2.5V for discharging. For pack operation, when any cell's cut-off voltage has been reached, the battery pack was considered to be fully charged/discharged. Following this rule, the battery pack was first fully discharged and then was fully charged at constant currents for ten times. This process resulted in ten refueling batches and ten working batches. As run-to-run control gets feedback of the measured voltage only at the batch end, regardless of the other voltage sampling points in the batch, using constant current profiles for verification will not lose generality. Furthermore, to examine the proposed balancing algorithm in more complex conditions, these load profiles have been specified deliberately to be different among some batches. The current profiles for all these batches are listed in Table II.

For the experimental implementation, the BCR was updated after each batch of the same type. In other words, the BCR for the next refueling batch was obtained from the previous refueling process, rather than the adjacent working process. The initial referenced BCR in (15) was $\mathbf{BCR}_{s=1}^{ref} = 0$. The element $\bar{G}_j$ in (27) was chosen as 5 for refueling batches and -5 for working batches. The voltage boundaries of the plateau were 3.1 and 3.5V. The coulombic efficiency η was treated as 1 and Q_{rated} was 20Ah based on the datasheet provided by GPL. For the specified load profiles, i.e. constant currents listed in Table II, $M = 1$ in (3) is selected to simplify the software implementation, and $\epsilon = 0.1$A is selected as a margin to account for the current amplifier error.

The initial SoCs of battery cells were all set as 100%. This situation aligns with real-world applications when the batteries are initially packed together or after regular maintenance. Note that the primary challenge for battery equalization over multiple cycles stems from unbalanced SoH levels. The controller's capability to address this challenge will be examined.

3) Software: The control algorithm was written using standard C language in a CodeWarrior development platform and then was downloaded to the MCU. After compilation, the generated binary code size was 118,328 bytes in total. Such a small code size indicates that the proposed algorithm can be implemented readily in real-time in commercial BMSs. It should be pointed out that in the experiments, a large amount of data has been sent back to the host PC for debugging, testing, and further data analysis. Most of the data is unnecessary to send and store in the host PC, implying appreciable reductions in the code size can be achieved. Further improvements in computational efficiency can be achieved by code optimization, but are not focused in the present work.

B. Experimental results: Battery calibration & evaluation

Before experimentally testing the proposed balancing algorithm, the battery cells are parameterized and their properties are studied.

The actual rated capacity of each cell is calibrated using a constant-current constant-voltage (CCCV) charging protocol under a current rate of 0.75C at 25°C, where voltage limits are 2.5 and 3.65V, the cutoff current is 0.05C. The current and voltage of battery packs are critical quantities in designing pack-level charging management algorithms, in which cell-level voltages are usually not controlled individually and thus can be dissimilar among cells. To avoid overcharging some cells, only the constant-current phase is used for battery pack charging operation. In this regard, "CC capacity" representing the total capacity delivered in the constant-current phase is introduced to evaluate the performance of balancing algorithms.

The obtained capacities of each cell are listed in Table III. Clearly, these cells are completely different in capacities. In details, the weakest cell, #03, has 16.95Ah actual capacity, corresponding to SoH level of 0.848. On the contrary, the actual capacity of the strongest cell, #05, is 18.41Ah and the SoH is 0.921. Furthermore, it can be calculated that each cell's CC capacity takes account of more than 90% of its rated capacity. When the 6 cells are assembled into a pack, the ideal CC capacity of the pack is 17.016Ah, which is the average CC capacity. If no balancing is applied, the pack CC capacity is termed as "static capacity" and is equal to that of the weakest cell, i.e., 15.351Ah.

In addition, discharging characteristics of these cells are investigated. Fig. 4 depicts their voltage trajectories under 0.75C discharge operation. In general, cells with lower voltage curves tend to have less rated capacities. However, here is a more complex situation with the considered battery pack.

TABLE II
LOAD PROFILES.

Batch Number	01	02	03	04	05	06	07	08	09	10
Discharging current (A)	-12.5	-17.5	-10	-15	-17.5	-15	-15	-10	-12.5	-15
Charging current (A)	15	15	15	15	15	15	15	15	15	15

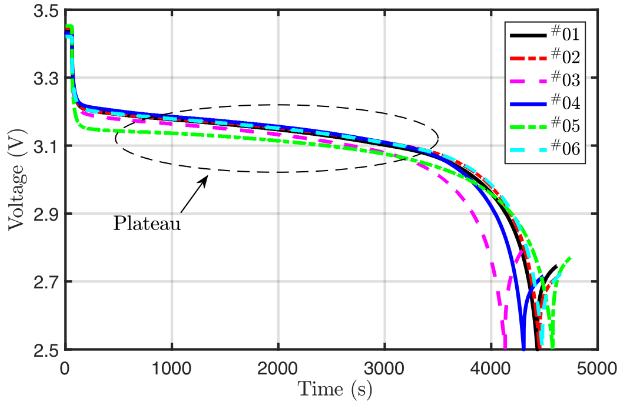


Fig. 4. Voltage evolution profiles at discharging for six selected battery cells.

TABLE III
ACTUAL RATED CAPACITY AND CC CAPACITY OF BATTERY CELLS.

Battery Number	Actual Rated Capacity (Ah)	CC Capacity (Ah)
01	18.215	17.262
02	18.304	17.373
03	16.954	15.351
04	17.682	16.806
05	18.819	17.836
06	18.410	17.467
Average	18.064	17.016

Specifically, cell #05 has the largest capacity, but the lowest voltage in the plateau. Meanwhile, the capacity of cell #03 is smaller than others, but its voltage in the plateau is not the lowest.

C. Experimental results: Battery balancing performance

Based on the above experimental specifications and batteries, the performance of the proposed balancing algorithm is comprehensively examined and analyzed. The ideal and static pack capacities are used to benchmark the proposed algorithm. For further comparison, a conventional voltage-based balancing algorithm following *Procedure 1* is also implemented based on identical configurations and conditions. By using the two algorithms in succession, the battery pack has been balanced during cycling operation.

1) Balanced voltage profiles: The voltage evolution profiles of all cells over the specified twenty batches are shown in Fig. 5, where the discharging curves in the first and tenth working batches are zoomed in with Sub-figures (c) and (d). As expected, the voltage trajectories are not only different among battery cells but also different for the same cell when two balancing algorithms are applied. Specifically, charging operations are dominated by the cell with the highest voltage, while discharging operations terminate when the weakest cell meets its low voltage bound. Second, it can be seen in Fig. 5(a)-(b) that the battery pack equipped with the proposed algorithm can work for 25.11h, which is longer than 23.86h performed by its counterpoint. In other words, by using the proposed algorithm, batteries can release and absorb more capacity in the entire process than its benchmark.

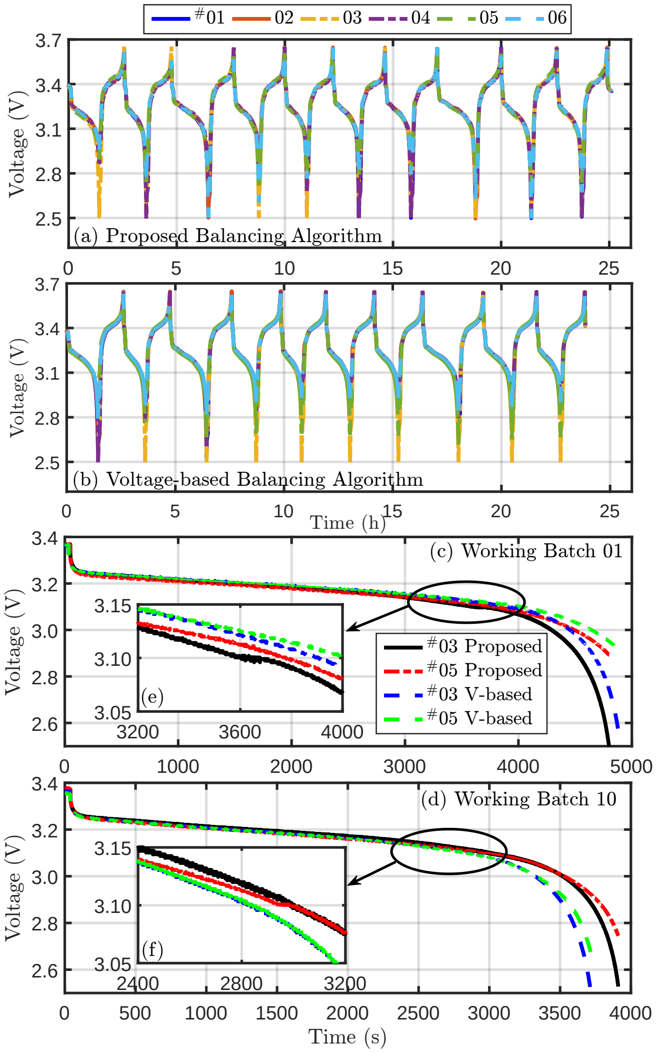


Fig. 5. Comparison of the proposed balancing algorithm and conventional voltage-based algorithm over ten working batches, in terms of the terminal voltages of each cell.

Fig. 5(c) shows that in the first working batch, the voltage curves of #03 and #05 in the proposed algorithm, particularly when outside the defined plateau, drop more quickly than those in the voltage-based algorithm. However, the opposite situation can be observed in Fig. 5(d). These phenomena imply that the

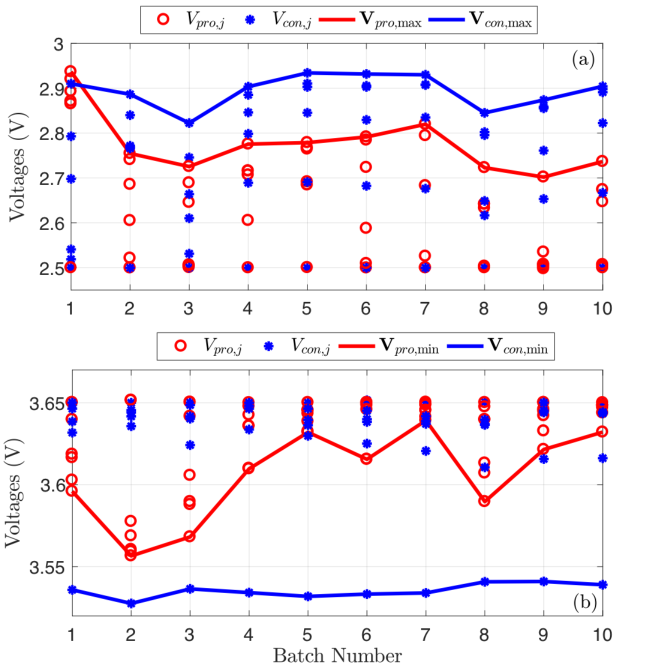


Fig. 6. Voltages at the end of the working batches (a) and refueling batches (b). $V_{pro,j}$ and $V_{con,j}$ are the voltages of cell j in the proposed and conventional balancing algorithms, respectively. $\mathbf{V}_{\max}$ and $\mathbf{V}_{\min}$ represent the maximum and minimum voltages of each cell in the battery pack, respectively.

voltage-based algorithm offers more capacity in the first batch but less in the tenth, compared to the proposed algorithm. For the situation in the first batch, it is because the referenced BCR in the proposed method has been specified as zero, and then no balancing is performed in the plateau.

The BCR- and voltage-based balancing parts in the proposed algorithm, following *Procedures 1-2* respectively, are switched at boundaries of the voltage plateau. As a result, using the BCR-based strategy, the cell whose voltage first reaches 3.1V will stay there for a while (due to charging by the balancing hardware), until another cell's voltage becomes this value, as shown in Fig. 5(e)-(f).

Additionally, the voltages at the end of batches are presented in Fig. 6. To facilitate the description, the minimum (maximum) voltage of all battery cells in the pack is named as "pack discharge (charge) voltage". It can be seen that the proposed approach ends up with lower pack discharge voltage but higher pack charge voltage than the conventional approach.

2) Capacity improvements: Capacity utilization of the two balancing algorithms is examined quantitatively. Corresponding to the voltage profiles in Fig. 5, the delivered capacities of each working batch are shown in Fig. 7(a). Clearly, the proposed balancing approach can release more capacity than the conventional one from the second batch to the end. In these nine batches, the average discharging capacity of the conventional and proposed approaches is 15.636Ah and 16.520Ah, respectively. Therefore, the proposed approach is able to extract 97.1% of the theoretical capacity and improves the capacity utilization 5.7% and 7.6% from its benchmark and static capacity, respectively.

The proposed approach is able to accommodate its performance by learning from the previous batches. For example, when using the conventional algorithm, the 6, 7, and 10th batches having the same load profile provide almost the same capacity. In the proposed algorithm, due to effects of the 9th batch, the discharging capacity of the 10th batch is smaller than those of the 6 and 7th batches. The released capacity of the proposed method can increase gradually when load profiles among contiguous batches are similar. This process can be evidenced from the 4–7th batches. As a result of insignificant load changes in these batches, the discharging capacity from the 5–8th batches based on the updating law in (28) can achieve substantial growth. However, no appreciable improvements are seen in the benchmarking method.

Indeed, the proposed method's performance may degrade or improve in the next batch if the load has a significant change. As can be seen, with a large load change in batch 8, the capacity decreases appreciably in batch 9. However, a different case is observed from batch 4 to 5. This uncertainty is potentially due to violations of *Assumption 2*. That is, under severely variant currents, batteries with the same SoC can be very different in their terminal voltages, because of voltage variations caused by internal resistance and charge-transfer reaction, for example.

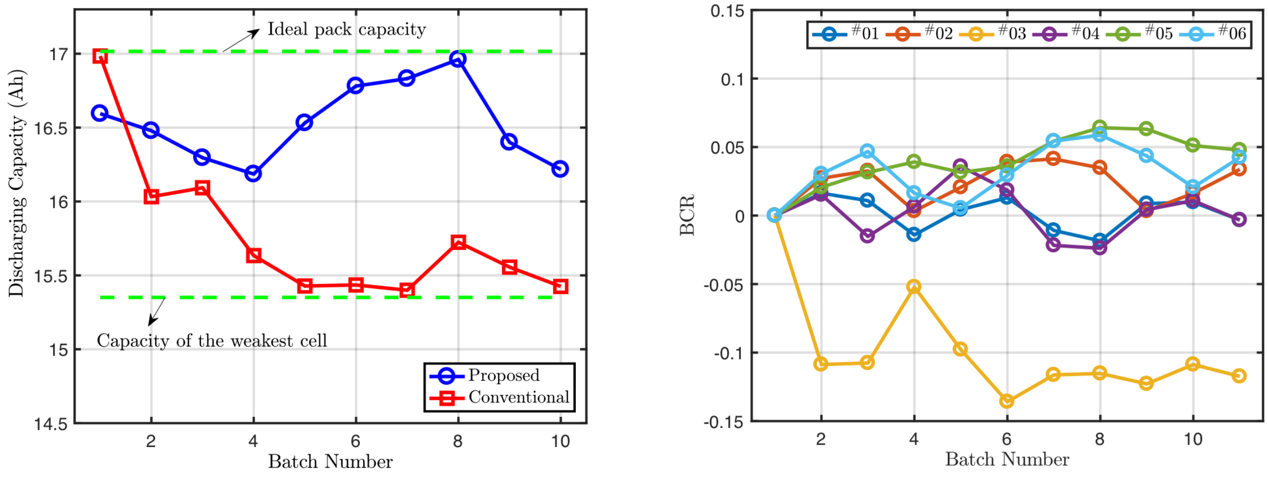
3) Actions of balancing hardware: The real-time balancing actions regarding which cell is being operated by hardware are studied. Experimental results of the first and ninth working batches are illustrated in Fig. 8, where the circle at each time step means that the corresponding cell is being charged by the hardware, and charging cell 0 means that the hardware is waiting.

When using the conventional algorithm, cell 05 is charged by the balancing hardware very frequently in the plateau for both the first and ninth batches. However, this cell with the largest capacity should be discharged to enlarge the overall pack capacity. Again, such charging actions are due to the invalidity of *Assumption 2* and can cause fault operations.

Using the proposed algorithm in the first batch, all cells have been charged only at the boundary of the voltage plateau. In the plateau of the ninth batch, cell 03 who has the lowest capacity is charged most frequently, but cell 05 is never charged. Therefore, the proposed algorithm equipped with the BCR information can help the hardware to balance the pack properly.

In the conventional algorithm, the balancing actions shift higher frequently among cells than those of the proposed one, leading to more energy loss in the balancing circuit and converter. The switching frequency can increase along with voltage measurement noise. Increasing the threshold in *Procedure 1* can efficiently address this problem but implies a larger tolerance to cell imbalance.

4) Tracking error of the batch-wise control: The tracking performance of the batch-wise control with the updating law (28) is examined. Fig. 7(b) demonstrates $\mathbf{BCR}^{ref}$ values obtained before each discharging batch. $\mathbf{BCR}^{*}$ can be calculated by using (11) and CC capacity. A comparison of $\mathbf{BCR}^{*}$ and $\mathbf{BCR}^{ref}$ is carried out and the results are shown in Table IV.



(a) (b)

Fig. 7. (a): Discharging capacity of each batch; (b): The BCR evolution profiles over different batches.

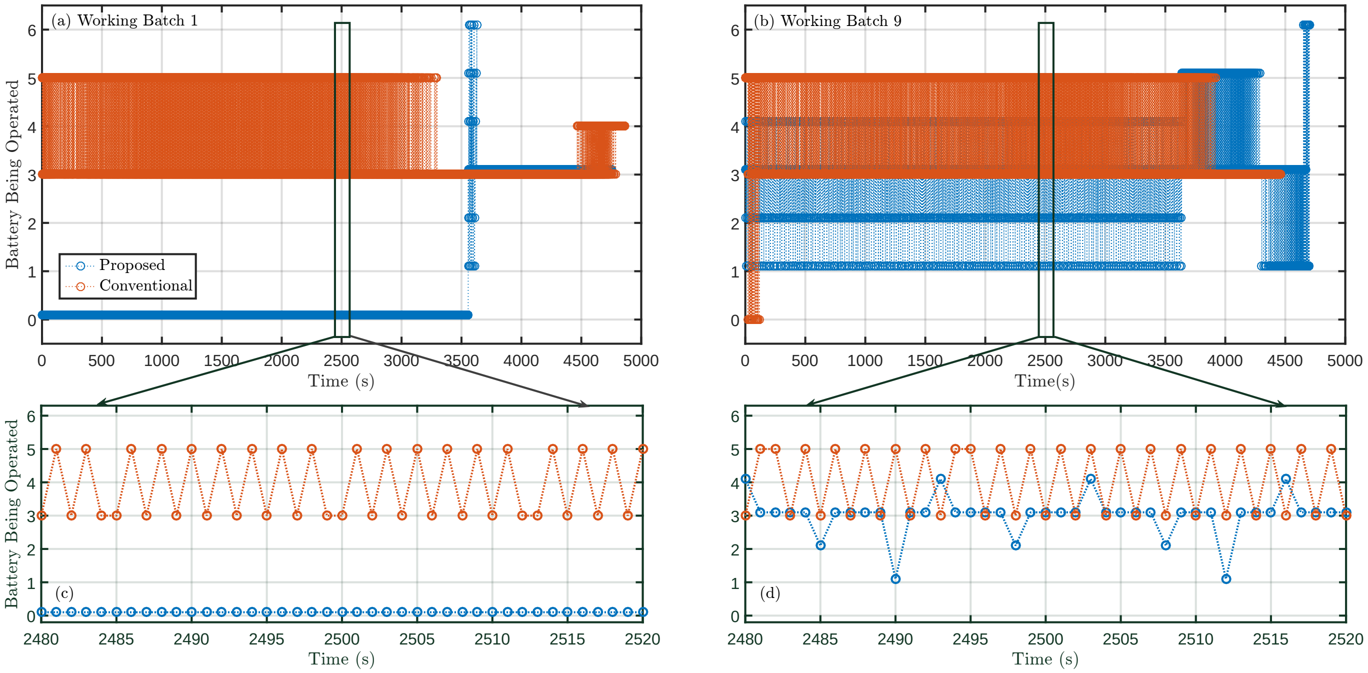


Fig. 8. Real-time actions of balancing hardware toward battery cells in the pack.

TABLE IV
BATCH-WISE BCR TRACKING RESULTS.

Battery Number	01	02	03	04	05	06
$\mathbf{BCR}^*$ (%)	1.45	2.10	-9.78	-1.23	4.82	2.65
$\mathbf{BCR}_{11}^{ref}$ (%)	-0.33	3.37	-11.74	-0.33	4.78	4.24
Error (%)	1.78	-1.27	1.96	-0.90	0.04	-1.59

Obviously, the batch-wise BCR tracking error is bounded by 2% within ten iterations. It can be inferred from (19) and (23) that the 2% error in BCR gives a less than 2% error in SoC. As known from start-of-the-art SoC estimation techniques in [8], [37]–[39], only carefully designed model-based SoC estimators can stand a chance to achieve 98% accuracy. Moreover, the proposed batch-wise control for battery balancing achieves such high accuracy in the presence of time-varying balancing currents and conversion efficiencies. When a more accurate initial referenced BCR can be calculated (e.g., via (11) and initial SoH of battery cells), even better tracking performance is likely to obtain.

D. Further Discussions

The nominal balancing current (i_{nom}) and the number of cells (N) in the pack are constant parameters in the problem formulation. Quantitatively speaking, more powerful hardware with larger balancing currents can generally lead to faster control convergence, associated with better balancing results.

Whereas this means more hardware costs and energy losses. Although six cells have been considered in the exemplary implementation, the developed control algorithm is sufficiently general for charge equalization of different numbers of cells in the topology of Fig. 1.

Future research opportunities exist in extensions of the proposed algorithm to incorporate effects of temperature and battery aging. Code optimization to further improve computational efficiency is another direction towards commercial BMSs in more applications.

V. Conclusions

This paper addressed the weak observability problem inside the voltage plateau and the computational issue caused by real-time state estimation for balancing management of lithium iron phosphate batteries. The technical novelties first arise from the introduction of a balancing current ratio (BCR) based algorithm and its combination with a voltage-based algorithm, separately responsible for the balancing task within and beyond the voltage plateau. The battery balancing process was formulated as a batch-based run-to-run control problem, in which the control policy was implemented in two timescales, i.e., time-wise and batch-wise. To the best of our knowledge, this is the first application of run-to-run control in the field of battery management. In addition, assumptions underlying each control algorithm are explicitly stated and remarked/justified on the battery system.

The proposed balancing algorithm was coded in C language with the binary code in 118,328 bytes only. Consequently, this model-free and state-of-charge independent algorithm is readily real-time implementable in many embedded systems. By implementing it on battery-in-the-loop facilities with a low-cost MCU, extensive results were obtained for battery balancing during different batches. It has been demonstrated that the proposed algorithm is able to extract 97.1% of the theoretical pack capacity and is 5.7% more efficient than the conventional voltage-based algorithm.

Acknowledgment

The authors would like to thank Prof. Zonghai Chen from University of Science and Technology of China for supporting the experimental implementation, and thank NXP and Linear Technology for providing free MCU and voltage reference samples.

References

[1] X. Hu, C. Zou, C. Zhang, and Y. Li, "Technological developments in batteries: A survey of principal roles, types, and management needs," *IEEE Power Energy Maga*, vol. 15, no. 5, pp. 20–31, 2017.

[2] J. Deng, C. Bae, J. Marcicki, A. Masias, and T. Miller, "Safety modelling and testing of lithium-ion batteries in electrified vehicles," *Nature Energy*, vol. 3, no. 4, p. 261, 2018.

[3] L. Wang, Y. Cheng, and X. Zhao, "A $\mathrm{LiFePO_4}$ battery pack capacity estimation approach considering in-parallel cell safety in electric vehicles," *Appl Energy*, vol. 142, pp. 293–302, 2015.

[4] Y. Zheng, W. Gao, M. Ouyang, L. Lu, L. Zhou, and X. Han, "State-of-charge inconsistency estimation of lithium-ion battery pack using mean-difference model and extended kalman filter," *J Power Sources*, vol. 383, pp. 50–58, 2018.

[5] X. Tang, C. Zou, K. Yao, G. Chen, B. Liu, Z. He, and F. Gao, "A fast estimation algorithm for lithium-ion battery state of health," *J Power Sources*, vol. 396, pp. 453–458, 2018.

[6] C. Zou, C. Manzie, D. Nešić, and A. G. Kallapur, "Multi-time-scale observer design for state-of-charge and state-of-health of a lithium-ion battery," *J Power Sources*, vol. 335, pp. 121–130, 2016.

[7] L. Zheng, L. Zhang, J. Zhu, G. Wang, and J. Jiang, "Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model," *Appl Energy*, vol. 180, pp. 424–434, 2016.

[8] X. Tang, Y. Wang, and Z. Chen, "A method for state-of-charge estimation of $\mathrm{LiFePO_4}$ batteries based on a dual-circuit state observer," *J Power Sources*, vol. 296, pp. 23–29, 2015.

[9] Y. Wang, C. Zhang, and Z. Chen, "A method for joint estimation of state-of-charge and available energy of $\mathrm{LiFePO_4}$ batteries," *Appl Energy*, vol. 135, no. C, pp. 81–87, 2014.

[10] G. Dong, J. Wei, C. Zhang, and Z. Chen, "Online state of charge estimation and open circuit voltage hysteresis modeling of $\mathrm{LiFePO_4}$ battery using invariant imbedding method," *Appl Energy*, vol. 162, no. 1, pp. 163–171, 2016.

[11] S. Li, C. C. Mi, and M. Zhang, "A high-efficiency active battery-balancing circuit using multiwinding transformer," *IEEE Trans Ind Appl*, vol. 49, no. 1, pp. 198–207, 2013.

[12] Y. Hua, A. Cordoba-Arenas, N. Warner, and G. Rizzoni, "A multi time-scale state-of-charge and state-of-health estimation framework using nonlinear predictive filter for lithium-ion battery pack with passive balance control," *J Power Sources*, vol. 280, pp. 293–312, 2015.

[13] M. Daowd, N. Omar, P. V. D. Bossche, and J. V. Mierlo, "Passive and active battery balancing comparison based on MATLAB simulation," in *IEEE Vehicle Power Propulsion Confer*, 2011, pp. 1–7.

[14] Y. H. Hsieh, T. J. Liang, S. M. O. Chen, W. Y. Horng, and Y. Y. Chung, "A novel high-efficiency compact-size low-cost balancing method for series-connected battery applications," *IEEE Trans Power Electr*, vol. 28, no. 12, pp. 5927–5939, 2013.

[15] H. Dai, X. Wei, Z. Sun, and D. Wang, "A novel dual-inductor based charge equalizer for traction battery cells of electric vehicles," *Int J Elec Power*, vol. 67, pp. 627–638, 2015.

[16] Y. Ye, K. W. E. Cheng, Y. C. Fong, X. Xue, and J. Lin, "Topology, modeling, and design of switched-capacitor-based cell balancing systems and their balancing exploration," *IEEE Trans Power Electr*, vol. 32, no. 6, pp. 4444–4454, 2017.

[17] J. Gallardo-Lozano, E. Romero-Cadaval, M. I. Milanes-Montero, and M. A. Guerrero-Martinez, "Battery equalization active methods," *J Power Sources*, vol. 246, no. 3, pp. 934–949, 2014.

[18] N. Bouchhima, M. Schnierle, S. Schulte, and P. B. Kai, "Active model-based balancing strategy for self-reconfigurable batteries," *J Power Sources*, vol. 322, pp. 129–137, 2016.

[19] W. Han and L. Zhang, "Mathematical analysis and coordinated current allocation control in battery power module systems," *J Power Sources*, vol. 372, pp. 166–179, 2017.

[20] W. Han, C. Zou, C. Zhou, and L. Zhang, "Estimation of cell SOC evolution and system performance in module-based battery charge equalization systems," *IEEE Trans Smart Grid, in press,* , DOI 10.1109/TSG.2018.2867017, 2019.

[21] S. Wang, L. Shang, Z. Li, H. Deng, and J. Li, "Online dynamic equalization adjustment of high-power lithium-ion battery packs based on the state of balance estimation," *Appl Energy*, vol. 166, pp. 44–58, 2016.

[22] Q. Ouyang, J. Chen, J. Zheng, and Y. Hong, "SOC estimation based quasi-sliding mode control for cell balancing in lithium-ion battery packs," *IEEE Trans Ind Electron*, vol. 65, no. 4, pp. 3427–3436, 2018.

[23] L. Maharjan, S. Inoue, H. Akagi, and J. Asakura, "State-of-charge (SOC)-balancing control of a battery energy storage system based on a cascade PWM converter," *IEEE Trans Power Electr*, vol. 24, no. 6, pp. 1628–1636, 2009.

[24] Z. Zhang, X. Cheng, Z. Y. Lu, and D. J. Gu, "SOC estimation of lithium-ion battery pack considering balancing current," *IEEE Trans Power Electr*, vol. 33, no. 3, pp. 2216–2226, 2018.

[25] Y. S. Lee and M. W. Cheng, "Intelligent control battery equalization for series connected lithium-ion battery strings," *IEEE Trans Ind Electron*, vol. 52, no. 5, pp. 1297–1307, 2005.

[26] P. A. Cassani and S. S. Williamson, "Design, testing, and validation of a simplified control scheme for a novel plug-in hybrid electric vehicle battery cell equalizer," *IEEE Trans Ind Electron*, vol. 57, no. 12, pp. 3956–3962, 2010.

[27] W. Li, L. Kang, X. Guo, and Y. Yao, “Multi-objective predictive balancing control of battery packs based on predictive current,” *Energies*, vol. 9, no. 4, p. 298, 2016.

[28] S. M. Salamati, S. A. Salamati, M. Mahoor, and F. R. Salmasi, “Leveraging adaptive model predictive controller for active cell balancing in Li-ion battery,” in *IEEE North American Power Symp*, 2017, pp. 1–6.

[29] Y. Wang, F. Gao, and F. J. D. Iii, “Survey on iterative learning control, repetitive control, and run-to-run control,” *J Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[30] E. Sachs, R. S. Guo, S. Ha, and A. Hu, “On-line process optimization and control using the sequential design of experiments,” in *VLSI Technol Symp*, 1990, pp. 99–100.

[31] W. J. Campbell, S. K. Firth, A. J. Toprac, and T. F. Edgar, “A comparison of run-to-run control algorithms,” in *American Control Confer*, 2002, pp. 2150–2155.

[32] E. Del Castillo and A. M. Hurwitz, “Run-to-run process control: Literature review and extensions,” *J Qual Technol*, vol. 29, no. 2, pp. 184–196, 1997.

[33] G. François, B. Srinivasan, and D. Bonvin, “A globally convergent algorithm for the run-to-run control of systems with sector nonlinearities,” *Ind Eng Chem Res*, vol. 50, no. 3, pp. 1410–1418, 2011.

[34] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, “A review on the key issues for lithium-ion battery management in electric vehicles,” *J Power Sources*, vol. 226, no. 3, pp. 272–288, 2013.

[35] R. D. Koski and P. D. Lindquist, “Cell balancing battery pack and method of balancing the cells of a battery,” Oct. 2009, US Patent 7,598,706.

[36] “DEMO MANUAL DC2064A - LTC3300-1/LTC6803-2 Bidirectional Cell Balancer,” 2013.

[37] C. Zou, X. Hu, S. Dey, L. Zhang, and X. Tang, “Nonlinear fractional-order estimator with guaranteed robustness and stability for lithium-ion batteries,” *IEEE Trans Ind Electron*, vol. 7, no. 65, pp. 5951–5961, 2018.

[38] X. Tang, B. Liu, Z. Lv, and F. Gao, “Observer based battery SOC estimation: Using multi-gain-switching approach,” *Appl Energy*, vol. 204, pp. 1275 – 1283, 2017.

[39] X. Tang, F. Gao, C. Zou, K. Yao, W. Hu, and T. Wik, “Load-responsive model switching estimation for state of charge of lithium-ion batteries,” *Appl Energy*, vol. 238, pp. 423–434, 2019.

Xiaopeng Tang received the B.S. in automation from the University of Science and Technology of China, He Fei, China, in 2015.

He is currently working towards the Ph.D. degree in chemical and biological engineering with the Hong Kong University of Science and Technology, Hong Kong, China. His research interests include modeling and control of energy storage systems.

Mr. Tang has been a recipient of the National scholarship in 2015, and Hong Kong PhD Fellowship Scheme in 2017. He is also serveing as the reviewer for IEEE Transactions on Industrial Informatics, Applied Energy, and Energy Conversion and Management.

Changfu Zou (M’16) received the B.E. degree in automotive engineering from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree from the Department of Mechanical Engineering, The University of Melbourne, Parkville, Victoria, Australia, in 2011 and 2017, respectively.

He was a Postdoctoral Researcher and is currently an Assistant Professor with Automatic Control Group at the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. He was a visiting student researcher with the Energy, Controls & Applications Lab, University of California, Berkeley, USA, from 2015 to 2016. His research interests include modelling and control of energy storage systems, electrified vehicles and transport systems.

Dr. Zou was Guest Editor for IEEE Transactions on Industrial Informatics and IET Intelligent Transport Systems, and the recipient of Excellent Graduate Award of Beijing, Melbourne Research Scholarship, Scholarship of National ICT Australia, and Engineering & IT Melbourne Abroad Travel Scholarships.

Torsten Wik received the M.Sc. degree in chemical engineering (majoring in applied mathematics), the Licentiate of Engineering degree in control engineering, the Ph.D. degree in environmental sciences (majoring in automatic control), and the Docent degree in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 1994, 1996, 1999, and 2004, respectively. From 2005 to 2007 he worked as a Senior Researcher with Volvo Technology, Gothenburg, Sweden, in control system design for combustion engine test cells, and combined reformer and fuel cells.

He is a Professor and the Head of the Automatic Control group within the Department of Electrical Engineering, Chalmers University of Technology. His current research interests include optimal control, model reduction, and systems with model uncertainties, with applications to energy storage, environmental, and biological systems.

Ke Yao received the B.S. and M.S. degrees from the Northeastern University, Shenyang, China in 1999 and 2002, respectively, and received the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, China in 2008.

From 2008 to 2014, he was a research associate in HKUST Fok Ying Tung Graduate School. He has been acting as the manager of Advanced Manufacturing and Industry Automation Division in Guangzhou HKUST Fok Ying Tung Research Institute since 2015. His research interests include modeling, control, monitoring, and optimization of the industrial processes, especially in the area of polymer processing and renewable energy.

Yongxiao Xia received the B.S. in mechanical and electronic engineering from Beijing Institute of Technology, Beijing, China, in 2009.

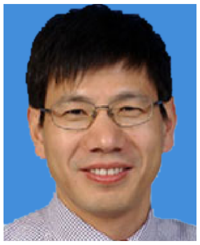
He served as a senior engineer in Beijing Nenggao Automation Technology Co., Ltd. from 2009 to 2013, and in Guangdong South China IoT Research Institute Co., Ltd from 2013 to 2015. He has been acting as the technical manager in Guangzhou HKUST Fok Ying Tung Research Institute since 2015, mainly responsible for the development of renewable energy systems such as solar-wind energy systems and battery management systems.

Yujie Wang received the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2017.

He is currently a Postdoctoral Fellow in the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China. His research interests include modeling, state estimation and control of battery, ultracapacitor and fuel cell systems with applications to vehicles and renewable energy storage.

Duo Yang received a B.S. in electronic science and technology from the University of Science and Technology of China, He Fei, China, in 2016. She is currently working towards the Ph.D. degree in control science and engineering with the University of Science and Technology of China.

Her research interests include complex system modeling and control, such as state estimation, nonlinear control, and health management for energy storage battery and fuel cells.



Furong Gao received the B.Eng. degree in automation from the East China Institute of Petroleum, Beijing, China, in 1985, and the M.Eng. and Ph.D. degrees in chemical engineering from McGill University, Montreal, QC, Canada, in 1989 and 1993, respectively.

He was a Senior Research Engineer with Moldflow International, Melbourne, Australia, from 1993 to 1995. He is currently a Chair Professor with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong, China. His research interests include process monitoring and fault diagnosis, batch process control, polymer processing control, and optimization.

Dr. Gao received a number of best paper awards, and is on editorial boards of a number of journals of his area.