Efficient Decentralized Active Balancing Strategy for Smart Battery Cells

Nitin Shivaraman[1], Arvind Easwaran[1], Sebastian Steinhorst [2,3]
[1]Nanyang Technological University, Singapore, Email: nitin.shivaraman@gmail.com, arvinde@ntu.edu.sg
[2]Department of Electrical and Computer Engineering, [3]Institute for Advanced Study
Technische Universität München, Germany, Email: sebastian.steinhorst@tum.de

***Abstract*—Among series-connected cells in large battery packs, such as those found in electric vehicles, a charge imbalance develops over time due to manufacturing and temperature variations. Therefore, active balancing strategies can be employed in Battery Management Systems (BMSs) to attain a charge balance among cells by transferring charge between them, maximizing the usable capacity of the battery pack. Recently, decentralized BMS architectures with smart battery cells have been developed, in which balancing strategies can operate by local cooperation between the cells without requiring global coordination. In this paper, we propose a decentralized active balancing strategy for smart cells where we identify *boundary cells* having special properties. These boundary cells enable to divide the global balancing problem into independent subproblems, where local decisions on charge transfers eventually converge to a globally balanced battery pack. The proposed strategy is implemented in a simulator framework and compared with two decentralized state-of-the-art strategies. Our results show significantly improved performance and scalability of the proposed strategy in terms of charge transfer losses and communication overhead between cells, while maintaining a comparable time to balance.**

I. Introduction and Related Work

Cyber-Physical Systems (CPS) are growing in complexity, leading to an increased demand for power and energy. In the classic example of electric vehicles, this demand for energy directly stems from the need to increase the driving range so as to be competitive with respect to conventional vehicles. Lithium-ion batteries are generally preferred to meet this demand due to their efficiency and Size, Weight and Power (SWaP) characteristics. A typical Lithium-ion battery pack comprises cell modules arranged in a series connection, and each cell module can comprise several individual cells arranged in a parallel connection. As parallel-connected cells are electrically indistinguishable, in the remainder of this paper, we generally refer to any number of parallel-connected cells in such a module as a cell.

To extract maximum energy from a battery pack, it is important to effectively utilize the available capacity within its operational limits. Due to variations introduced by manufacturing, operating temperature and aging over time, an unequal State-of-Charge (SoC) develops across individual cells in the battery pack. The SoC of a cell, which can be estimated based on its open-circuit voltage, is a measure of the available energy stored. The battery capacity is dictated by the cell with the lowest SoC and the cell with the highest SoC[1]. This means that over time the battery pack could have some cells almost fully charged while some others almost fully discharged, thus significantly reducing its usable capacity.

For efficient utilization of a battery pack's energy storage capabilities, Battery Management Systems (BMSs) are introduced to control its operations. Among other things, the BMS can perform balancing operations in order to equalize the SoC of series-connected cells. Broadly speaking, there are two different approaches to balancing. The simpler approach is to remove excess charge from the cells with higher SoC through heat dissipation. This is achieved using passive elements such as resistors and the process is called *passive balancing*. It is very cost-effective from an integration perspective due to limited components and simplicity of design and control, and hence is a popular choice in the electric vehicle industry today. Problems with this method are containing the heat dissipated from the passive components and the losses in system energy efficiency.

The more complex approach is to transfer charge from cells with a higher SoC to cells with a lower SoC in order to equalize the stored energy of all cells. This process is known as *active balancing* and employs active components such as capacitors, inductors or transformers. Albeit being more complex and expensive in terms of cost, it is a more efficient way to increase the battery pack's usable capacity as long as losses during charge transfers are kept low.

The communication and control architecture used to realize the balancing operation of a BMS can be either centralized or distributed. In the centralized architecture, there is a single master controller that regulates the flow of charge by monitoring all cells. The controller has global knowledge of the individual SoCs and can therefore efficiently determine the charge transfers required for equalization. However, it suffers from all the disadvantages of a centralized architecture such as single point of failure, limited scalability, longer balancing time due to control signal generation bottlenecks at the master controller, etc. By contrast, in the distributed architecture, each cell has a local controller, termed as Cell Management Unit (CMU), to monitor and control the state of the cell and to make local decisions on charge transfers via communication with neighboring CMUs. We refer to such a cell with a local CMU as a *smart cell*. Although this decentralized architecture is more efficient in terms of its ability to parallelize decision

With the support of the Technische Universität München - Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n°291763. This work is supported in part by the Ministry of Education, Singapore, grant numbers RG21/13 and ARC9/14.

[1]The cell with the lowest SoC determines the battery pack's discharging limit, whereas the cell with the highest SoC determines its charging limit.

978-3-9815370-8-6/17/$31.00

making, a key challenge is how to utilize such an architecture for solving a global balancing problem.

Contributions. In this paper, we propose a novel active balancing strategy called *BCAB* (short for *Boundary Cell based Active Balancing*) that utilizes the decentralized BMS architecture to balance cell SoCs via charge transfers. The key concept of this strategy is efficient identification of what we call *boundary cells*, which are cells having the property that no charge needs to be transferred "across" these cells in order to balance the battery pack. Although charge may need to be transferred to/from a boundary cell, no charge transfer is required from any cell above (likewise below) it to any cell below (likewise above) it in the pack. Thus, boundary cells essentially decompose the global balancing problem into several independent local subproblems, and hence allow us to effectively utilize the decentralized BMS architecture. It is important to also note that the number of boundary cells progressively increases with the balancing operation as more and more cells become balanced. Thus, even if there are not many boundary cells to begin with, eventually the balancing problem gets sufficiently decomposed into several subproblems.

Using a decentralized BMS simulator [1], we evaluate BCAB in terms of losses due to charge transfers, communication overhead and time to balance. In particular, we compare it with two existing strategies presented in [1]; *Below-Average*, in which each cell with a SoC below the pack average requests charge from neighboring cells and requests are acknowledged by the neighbors if their SoC is above the pack average, and *Min-Max*, in which all cells except the one with the highest SoC always request charge and requests are acknowledged by neighboring cells if the cell receiving charge does not end up with a higher SoC than the one sending charge for a defined transfer duration.

Our experiments show that BCAB significantly outperforms both existing strategies in terms of losses as well as communication overhead, without any impact on balancing time. We also show that BCAB scales much better with increasing number of series-connected cells in terms of communication overhead and losses when compared to the other strategies, while in terms of balancing time, it scales similarly. Thus, we can conclude that the proposed strategy effectively utilizes the decentralized BMS architecture.

Related Work. Several research studies have focused on developing circuits and architectures for improving the energy delivery of the battery. There are some notable works in developing algorithms for passive and active balancing. Song Ci et.al. [2] find optimal solution using dynamic programming by converting the balancing problem into Lagrangian equation. Raychev et.al. [3] adopt the fixed-time (discharge-rest combination), capacity-driven (switch between cluster till discharge up to certain level) and work-load driven (adapt to fixed-time based on load demand) strategies on autonomous battery clusters. Liang He et.al. [4] try to find disjoint paths by converting the balancing problem into path selection of a graph used to manage multiple loads. It greedily selects the load with largest battery discharge current and the path with least conflicts. Bouchhima et.al. [5] use cells with a high SoC to discharge first till they reach the SoC of other cells. The number of cells starts from a minimum limit and goes up to a maximum limit obtained by mapping the balancing algorithm into a network map. Dung et.al. [6] introduce an algorithm for the charging phase where rate of voltage change is monitored till a trigger point is reached. An equalizer is enabled after the trigger point to balance the charge among the cells. Hoque et.al. [7] focus on single highly/lowly charged cells and equalize using a forward/reverse fly back converter. However, the time to balance with this method is large. Most of these strategies are based on a centralized BMS architecture.

Recently, approaches to advanced modular active balancing architectures [8] and strategies [9] have been developed by Kauer et.al., providing an important building block of decentralized battery management concepts. With this trend towards decentralization and modularity in BMS design, our strategy focuses on a completely distributed smart cell architecture, initially introduced by Steinhorst et.al. [10].

II. Boundary Cell based Active Balancing Strategy

A. System model and balancing architecture

We consider the problem of balancing a battery pack comprising n cell modules or *cells* connected in series. Each cell c_i has a *State-of-Charge* (SoC) $z_i \in \mathbb{R}$. Let $\mathcal{C} = \langle c_1, c_2, \ldots c_n \rangle$ denote the ordered set of n cells in the battery pack, where c_1 (likewise c_n) denotes the topmost (likewise bottommost) cell in the series connection with respect to their potential towards the negative terminal of the pack. We use the notation $\bar{Z}$ to denote the average SoC of the battery pack (denoted in short form as *average pack-SoC*), i.e., $\bar{Z} = (\sum_{i=1}^{n} z_i)/n$. For each cell c_i, let $\bar{Z}_i^{\uparrow}$ denote the average SoC of the cells that are above c_i, i.e., $\bar{Z}_i^{\uparrow} = (\sum_{j=1}^{i-1} z_j)/(i-1)$, and $\bar{Z}_i^{\downarrow}$ denote the average SoC of the cells that are below c_i, i.e.,$\bar{Z}_i^{\downarrow} = (\sum_{j=i+1}^{n} z_j)/(n-i)$. The objective of the balancing strategy is to balance each and every cell in the pack, i.e., the balancing is considered complete when $z_i = \bar{Z}$ for each cell c_i[2].

We make the following assumptions regarding a generic smart cell architecture that we use in our strategy.

1) The balancing architecture of the Battery Management System (BMS) is decentralized with each smart cell comprising a Cell Management Unit (CMU) to make local balancing decisions as illustrated in Fig. 1. The cells have a communication network to exchange information and initiate requests for charge transfers.
2) Each cell c_i is aware of its position in the series-connected battery pack, i.e., it knows its index number i. This topology information can either be programmed during battery pack assembly, or a dynamic topology identification protocol such as the one proposed in [11] can be applied.
3) The charge transfers between cells are restricted to *neighbor-only* transfers using an active cell balancing circuit from the architecture class proposed in [12].

[2] In practice, due to imprecise SoC measurements and minimum limits on quantum of charge transfers, it is impossible to balance cells precisely. Therefore, error margins are generally defined to account for such variations. Although we consider such an error margin in our experiments as discussed in Section III, we ignore it here to keep the presentation simple.

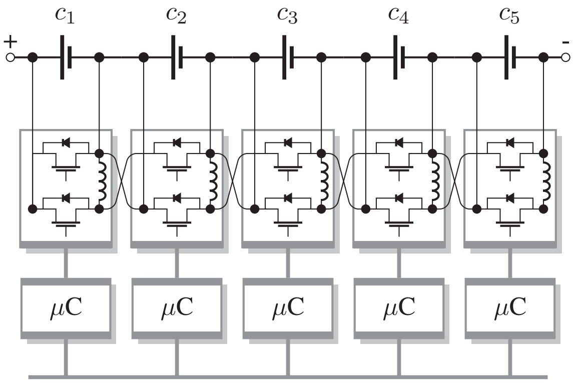


Fig. 1. Five series-connected smart cells, each consisting of the battery cell, the modular neighbor-only active balancing circuit, the microcontroller and the interface to the communication channel.

4) If two cells agree to transfer charge, the transfer is performed in a fixed cycle of time duration T_m. The amount of charge transferred and losses incurred during T_m can be measured by the difference in SoC of the cells after the transfer.
5) If a pair of cells, say c_i and c_{i+1}, is involved in a charge transfer during some time, then neither c_{i-1} nor c_{i+2} can be involved in charge transfers during the same time. We refer to this restriction as the *neighbor exclusion constraint*. We introduce this constraint for safety reasons such that a single balancing circuit module is not involved in two charge transfers at the same time.
6) The battery pack is neither discharging nor charging while the balancing strategy is being executed. Therefore, the only change in the total SoC of all cells is due to losses incurred during charge transfers. While we assume this to simplify the discussion in the remainder of this paper, the continuous tracking of each individual cell's SoC enables balancing decisions to be based on the observed system state at any time point. Hence, the strategy would also eventually converge during loads applied to the pack.

B. BCAB Strategy

The key concept used in our balancing strategy to decompose the global balancing problem into independent subproblems is the identification of what we call a *boundary cell*. We first formally define a boundary cell, and then discuss how it can be used to decompose the problem.

Definition 1 (Boundary Cell): A cell $c_i \in \mathcal{C}$ is defined as a boundary cell, if any one of the following three conditions are satisfied.

1) $z_i \geq \bar{Z}$, $\bar{Z}_i^{\uparrow} \leq \bar{Z}$ and $\bar{Z}_i^{\downarrow} \leq \bar{Z}$.
2) $z_i < \bar{Z}$, $\bar{Z}_i^{\uparrow} \geq \bar{Z}$ and $\bar{Z}_i^{\downarrow} \geq \bar{Z}$.
3) $i = 1$ or $i = n$.

An important property of a boundary cell as defined above is that no charge needs to be transferred from any cell above (likewise below) the boundary to any cell below (likewise above) it in order to balance the pack. For example, if c_i satisfies condition 1, then its SoC is no less than the average pack-SoC, whereas the average SoC of cells above and below it are both no greater than the average pack-SoC. In this case, to balance the pack we may need to transfer charge from c_i to one or both of its neighbors, but no transfer is required across c_i because both $\bar{Z}_i^{\uparrow}$ and $\bar{Z}_i^{\downarrow}$ are below the average pack-SoC. In the BCAB strategy, no charge is transferred across such boundary cells, and this is one of the main reasons why it is able to significantly reduce transfer losses.

Using the concept of a boundary cell we now present the balancing strategy BCAB. Pseudocode for this strategy for a cell c_i is given in Algorithm 1. We split the entire balancing operation into one or more rounds of a *boundary cell detection phase* followed by a *charge transfer phase*. During a boundary cell detection phase, we use Definition 1 to identify all the boundary cells in the pack based on current SoC values (lines 2–3). In particular, we identify those boundary cells that are not balanced, i.e., $z_i \neq \bar{Z}$. In the charge transfer phase, unbalanced boundary cells identified previously are now eligible to initiate transfer requests. Additionally, all over-saturated cells[3], irrespective of whether they are boundary cells, will also be eligible to initiate transfer requests. This is to ensure convergence in some rare cases when there are several over- and under-saturated[4] cells in the pack. For each such cell c_i, we first determine the total amount of charge that must be transferred to/from cells above as well as below it. We identify the larger of the two transfer amounts and initiate a transfer request between c_i and its neighbor in that direction (lines 5–13). In the case of corner cells (c_1 and c_n), since there is only one neighbor, we can ignore this choice. In the following paragraphs we discuss some of the coordination and charge transfer details for this strategy.

Algorithm 1 BCAB strategy for cell c_i

```
1: /* Boundary cell detection phase */
2: Compute Z̄_i^↑, Z̄_i^↓ and Z̄ based on information received
   from the other cells.
3: Use Definition 1 to determine whether c_i is an unbalanced
   boundary cell.
4: /* Charge transfer phase */
5: if c_i is an unbalanced boundary cell or over-saturated then
6:     Compute z_i^↑ = (Z̄_i^↑ − Z̄) × (i − 1).
7:     Compute z_i^↓ = (Z̄_i^↓ − Z̄) × (n − i).
8:     if |z_i^↑| ≥ |z_i^↓| then
9:         Request charge transfer with c_{i−1}.
10:    else
11:        Request charge transfer with c_{i+1}.
12:    end if
13: end if
14: if c_i receives a request for charge transfer then
15:     Acknowledge request and initiate transfer if feasible.
16: end if
```

Cell coordination. Since we focus on a decentralized architecture, the cells need to communicate with each other in order to transmit information regarding their SoCs. We assume that each cell c_i is aware of its position in the series-connected pack. This means it is also aware of the number of cells that

[3]Cells with SoC value equal to the upper safe SOC threshold.
[4]Cells with SoC value equal to the lower safe SOC threshold.

are above $(i-1)$ and below $(n-i)$ it. Then, the only additional information it requires is $\bar{Z}_i^{\uparrow}$ and $\bar{Z}_i^{\downarrow}$. This is because it can easily determine $\bar{Z}$ based on these two averages as well as its own SoC. If the communication network is a broadcast medium, then each cell can transmit its SoC value to all cells in the pack. On the other hand, if it supports independent pairwise communication, we can transmit this information using two daisy-chain communication; one from c_1 to c_n in which each cell c_i transmits $\bar{Z}_{i+1}^{\uparrow}$, and one from c_n to c_1 in which each cell c_i transmits $\bar{Z}_{i-1}^{\downarrow}$. At the end of this communication phase, each cell in the pack can independently determine a) whether it is a boundary cell, and b) whether it is balanced.

Once a boundary cell say c_i is balanced, it essentially decomposes the balancing problem into two subproblems that can be handled independently; balancing of cell group $c_1, \ldots, c_{i-1}$ and cell group $c_{i+1}, \ldots, c_n$. Note that the balanced cell c_i also ensures automatic satisfaction of the neighbor exclusion constraint for any transfers within these groups, and therefore no further coordination is required between them.

Charge transfers. An unbalanced boundary cell may send a transfer request to one of its neighbors. If two neighboring cells are both unbalanced boundary cells (say c_{i-1} and c_i), then based on Definition 1 it is guaranteed that their requests for charge transfers cannot be conflicting. That is, if say c_i wants to transfer charge to c_{i-1}, it cannot happen that c_{i-1} wants to transfer charge to c_i in the same round. If c_i wants to transfer charge to c_{i-1}, it must hold that

$$\bar{Z}_i^{\uparrow} = \sum_{j=1}^{i-1} z_j/(i-1) < \bar{Z}. \tag{1}$$

Then,

$$\bar{Z}_{i-1}^{\downarrow} = \sum_{j=i}^{n} z_j/(n-i+1) > \bar{Z} \tag{2}$$

because

$$(i-1)\bar{Z}_i^{\uparrow} + (n-i+1)\bar{Z}_{i-1}^{\downarrow} = (n)\bar{Z}. \tag{3}$$

Hence, c_{i-1} cannot request for a transfer of charge to c_i in the same round, although it may request for a transfer of charge from c_i to itself which is consistent with c_i's request.

We follow the request-acknowledge method for charge transfers. All cells are required to acknowledge a transfer request and initiate the transfer if it is feasible to do so (lines 14–16). A cell acknowledges a request if and only if it is not already involved in another transfer, and acceptance of this request: 1) does not result in a violation of the specified lower and upper safe SoC thresholds for this cell, and 2) does not violate the neighbor exclusion constraint. If a cell acknowledges a request, the transfer cycle of time duration T_m can be initiated immediately between the pair. Upon completion of all charge transfers in this round, depending on the total amount of charge that was transferred, the remaining total SoC of all cells in the pack will be lower due to losses. This means for the next round, the average pack-SoC $\bar{Z}$ will also be lower.

Merging of subproblems. Due to losses during charge transfers, it is possible that a previously balanced boundary cell say c_i now becomes unbalanced because $\bar{Z}$ has reduced significantly. Then, the cell groups $c_1, \ldots, c_{i-1}$ and $c_{i+1}, \ldots, c_n$ can no longer be balanced independently, but rather must be merged together into a single group. This can be easily achieved after the SoC communication in boundary cell detection phase, because each cell, including a previously balanced cell, can determine the change in average pack-SoC.

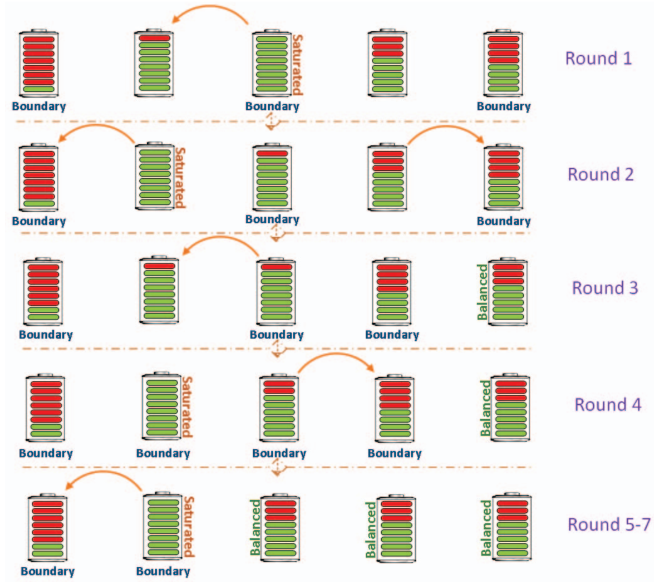


Fig. 2. Balancing example for a 5 cell battery pack with BCAB strategy.

Example. Figure 2 shows a simple example of a 5 cell battery pack being balanced over 7 rounds using the BCAB strategy. The initial SoC value of each cell is indicated by the number of green rectangles; these values for cells 1 through 5 are $1, 7, 8, 5$ and 4, respectively, thus resulting in a average pack-SoC of 5. We ignore losses during charge transfers to keep the presentation simple. In round 1, cells $1, 3$ and 5 are identified as unbalanced boundary cells, but the only transfer that occurs is $3 \rightarrow 2$ due to neighbor exclusion constraints. In round 2, again cells $1, 3$ and 5 are unbalanced boundary cells, whereas cell 2 is an over-saturated cell that can also initiate a transfer request. Transfers $2 \rightarrow 1$ and $4 \rightarrow 5$ occur in parallel in this round and cell 5 becomes balanced at the end. In round 3, $1, 3$ and 4 are unbalanced boundary cells, and the transfer $3 \rightarrow 2$ is completed. In round 4, $1, 2, 3$ and 4 are unbalanced boundary cells, transfer $3 \rightarrow 4$ is completed and both these cells become balanced. Finally, in rounds 5 through 7, transfer $2 \rightarrow 1$ is repeated until the entire pack becomes balanced.

Note that in each round several combinations of transfers are possible, but only one combination is completed. This depends on the order in which transfer requests are received by cells. In general, it might be possible to reduce the number of rounds in BCAB by carefully selecting a combination that maximizes parallelism in each round. But this would require further coordination among cells once requests are received, and it is not clear whether the resulting reduction in balancing time would be substantial enough to compensate for this additional overhead.

III. Experimental Evaluation

A. Experiment Setup

Experiments are performed using the Cyber-Physical Co-Simulation Framework (CPCSF) initially presented in [1]. The simulator models a decentralized BMS architecture with smart battery cells on several layers, allowing an accurate system-level analysis without abstracting away behavior of the individual layers. It models the cell layer, the sensing and active cell balancing layer, the computation layer and the communication layer of the architecture using the Python-based discrete-event simulation framework *SimPy*. The actual balancing strategies are implemented in an extensible fashion, so that the BCAB strategy could be integrated into the simulator and compared with existing strategies introduced in [1].

Our experiments are performed on a complete electric vehicle battery model consisting of up to 96 smart cells connected in series. Each smart cell consists of 24 parallel-connected SAMSUNG INR18650-25R cells, with a nominal voltage of 3.75 V and a nominal capacity of 2.5 Ah per single cell. Consequently, for 96 series-connected smart cells, the pack has a nominal energy storage capacity of 21.6 kWh, which is similar to battery packs found in the BMW i3 or Nissan Leaf.

To obtain different scenarios for balancing, we implement a randomized initial homogeneous cell SoC starting point and add a random variation among the individual cell SoCs, both reproducible and controlled by a common random seed value. Therefore, an initial homogeneous cell SoC between 40% and 60% is randomly chosen using a uniform distribution. Furthermore, to create a SoC variation between the cells, we add an individual SoC value between 0% and 3% randomly chosen by a uniform distribution per cell to the common SoC value. For the presented experiments, we use 100 complete battery pack balancing simulation runs per strategy until all cells are equalized. We use the same random seed values for the runs of each strategy such that the SoC variation scenarios are comparable. Hence, while we determine both the initial cell SoC starting point and the individual SoC variation randomly to create 100 sufficiently diverse scenarios, the evaluated strategies compete on the same SoC variation per scenario for comparability of the results.

We compare the proposed BCAB strategy with two existing distributed active balancing strategies proposed in [1]: *Below-Average* and *Min-Max*. In the Below-Average strategy, a cell c_r is requesting charge from a neighboring cell c_s if the SoC of c_r is below the average pack-SoC, i.e., $z_r < \bar{Z}$. A cell c_s agrees to send charge to c_r if its own SoC is above the average pack-SoC, i.e., $z_s > \bar{Z}$. In the Min-Max strategy, requests from c_r to c_s are made by each cell except the one with the highest SoC in the pack. Requests are acknowledged as long as the requirement is satisfied that after the transfer, the SoC of c_s will not become smaller than c_r. Additionally, requests are acknowledged if the requester is the cell with the lowest charge or the requested cell has the highest charge in the pack, which ensures that transfers are always feasible. In both strategies, to determine for a cell $c_r (= c_i)$ requesting charge whether to request charge from its upper or lower neighbor, it uses the decision variable $\lambda = \mathrm{sgn}(\bar{Z}_i^{\downarrow} - \bar{Z}_i^{\uparrow})$ such that the cell c_s requested to send charge to c_r is determined by $c_s = c_{i+\lambda}$.

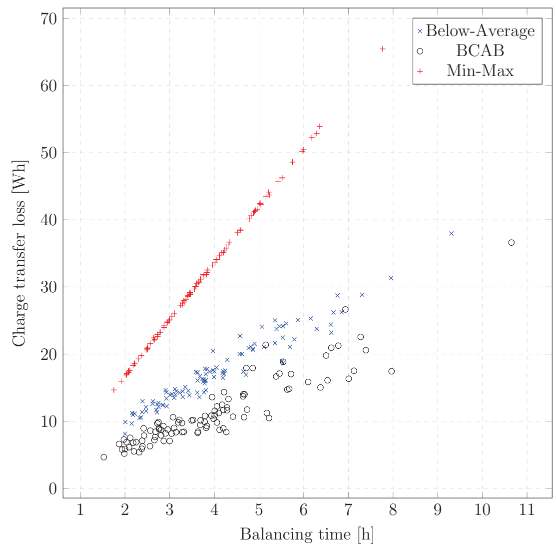


Fig. 3. Scatterplot of charge transfer loss versus balancing time.

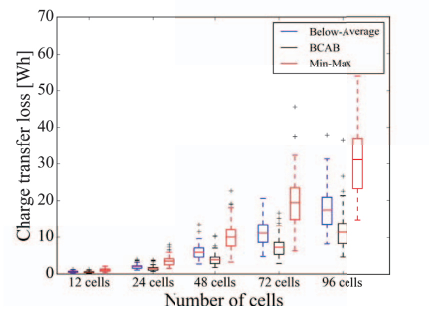
This ensures that charge is always requested from the direction where the substring of cells has the higher average SoC.

Transfer duration T_m is fixed at 30 seconds and error margin ϵ for determining whether a cell/pack is balanced is set at 0.2%. This margin is chosen to maintain stability and ensure convergence for all the strategies, taking into consideration the accuracy which can be provided by SoC estimation approaches. For Below-Average and Min-Max, ϵ is used as a stopping criterion; if the difference between the highest and the lowest SoC values in the pack is no larger than ϵ, then the balancing operation is considered to be complete. For the BCAB strategy, we use ϵ to determine whether an individual boundary cell is balanced; a cell c_i is assumed to be balanced if z_i, $\bar{Z}_i^{\uparrow}$ and $\bar{Z}_i^{\downarrow}$ are all within $\epsilon/2$ distance of $\bar{Z}$.

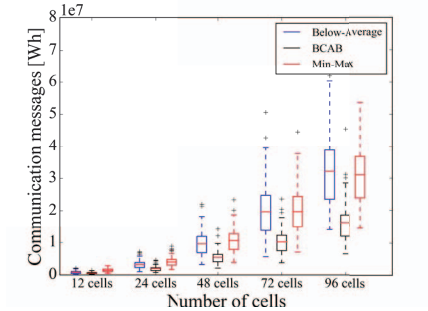
Simulations were performed on a machine with a 12-core Intel CPU and 16 Gigabytes of memory. A batch run of 300 simulations (100 for each strategy) for a 96 cell configuration took 742 minutes to complete, yielding about 2.5 minutes per balancing run.

B. Results

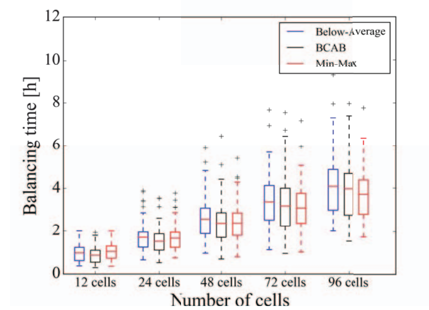
Figure 3 shows the total loss of charge incurred during transfers versus total balancing time for each of the 100 different SoC variations, when we consider a 96 cell battery pack. Balancing time indicates the time required for all the cells of the pack to achieve the desired SoC. A key observation is that BCAB incurs significantly lower transfer losses when compared to the other two strategies, with comparable balancing times. The reason for this performance is that in BCAB all initiated charge transfers are *necessary* to balance the pack; no unnecessary transfers are performed because no charge is transferred across boundary cells. Despite the improved efficiency, balancing times are comparable to the state-of-the-art strategies, hence making BCAB a favorable decentralized strategy for efficient active cell balancing.



(a) Charge transfer loss



(b) Communication overhead



(c) Balancing time

Fig. 4. Scalability of charge transfer loss, communication overhead and balancing time with respect to number of cells.

Figure 4 shows the scalability of the three analyzed strategies in terms of transfer losses, communication overhead and balancing time with increasing number of cells in the battery pack. For each value on the x-axis (number of cells), we present a box-plot for each strategy and performance parameter for the SoC variations created by the 100 seed values. The horizontal red lines in the boxes denote mean values for the respective strategy and parameter. The ends of the box represent the first (25%) and the third (75%) quartiles. The whiskers are at 1.5 times the interquartile range (difference of first and third quartile) both above and below the box. The points beyond the whiskers are outliers having extreme values for experiments of certain seeds. As can be seen from Figures 4(a), 4(b) and 4(c), BCAB scales significantly better than the other two strategies in terms of losses during charge transfers as well as communication overhead, while the balancing time scales similarly. The main reason for the reduction in communication overhead under BCAB is that as boundary cells proactively initiate transfer of charge to/from neighbors, very few transfer requests are not acknowledged when compared to the existing strategies. The acknowledging cells are not involved in any decision making except for transfer feasibility.

IV. Conclusions and Future Work

In this paper we presented a new decentralized active cell balancing strategy for smart battery cell architectures called BCAB, and demonstrated its superior performance in terms of reduction in losses during charge transfers and communication overhead when compared to two state-of-the-art strategies. The key idea used in this strategy is that of a boundary cell, which essentially decomposes the global balancing problem into several independent subproblems and as a consequence eliminates unnecessary charge transfers. We also showed that this strategy scales significantly better in terms of losses during charge transfers and communication messages when compared to the existing strategies with increasing number of cells, while it scales similarly in terms of balancing time. Reduced charge loss conserves energy in the pack, contributing to a higher energy output and better system efficiency.

Since BCAB does not initiate any unnecessary charge transfers, we conjecture that it is optimal in terms of minimizing the losses. A formal proof of this, even in the presence of average pack-SoC changes due to losses, would be a natural future direction of work. Improving BCAB towards further reducing balancing time is another direction that we plan to explore. In fact a detailed analysis of the correlation between the three parameters transfer losses, communication overhead and balancing time would be very valuable.

Acknowledgement

We sincerely thank Martin Lukasiewycz for his contributions in the initial discussions on the BCAB strategy.

References

[1] S. Steinhorst, M. Kauer, A. Meeuw, S. Narayanaswamy, M. Lukasiewycz, and S. Chakraborty, "Cyber-physical co-simulation framework for smart cells in scalable battery packs," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 4, 2016.

[2] S. Ci, J. Zhang, H. Sharif, and M. Alahmad, "Dynamic reconfigurable multi-cell battery: A novel approach to improve battery performance," in *APEC, 2012 27th IEEE*, Feb 2012, pp. 439–442.

[3] D. Raychev, Y. Li, and W. Shi, "The seventh cell of a six-cell battery," *Proc. of WEED*, 2011.

[4] L. He, L. Gu, L. Kong, Y. Gu, C. Liu, and T. He, "Exploring adaptive reconfiguration to optimize energy efficiency in large-scale battery systems," in *Proc. of RTSS, 2013 IEEE 34th*, Dec 2013, pp. 118–127.

[5] N. Bouchhima, M. Schnierle, S. Schulte, and K. P. Birke, "Active model-based balancing strategy for self-reconfigurable batteries," *Journal of Power Sources*, vol. 322, pp. 129 – 137, 2016.

[6] L.-R. Dung and Y.-S. Peng, "Adaptive battery equalization algorithm for capacitor-based battery management system," *Universal Journal of Electrical and Electronic Engineering*, 2016.

[7] M. Hoque, M. Hannan, and A. Mohamed, "Voltage equalization control algorithm for monitoring and balancing of series connected lithium-ion battery," *Journal of Renewable and Sustainable Energy*, vol. 8, no. 2, p. 025703, 2016.

[8] M. Kauer, S. Narayanaswami, S. Steinhorst, M. Lukasiewycz, S. Chakraborty, and L. Hedrich, "Modular system-level architecture for concurrent cell balancing," in *Proc. of the 50th Design Automation Conference (DAC 2013)*, 6 2013, pp. 155:1–155:10.

[9] M. Kauer, S. Narayanaswamy, S. Steinhorst, M. Lukasiewycz, and S. Chakraborty, "Many-to-many active cell balancing strategy design," in *Proc. of the 20th Asia and South Pacific Design Automation Conference (ASP-DAC 2015)*, 3 2015, pp. 267–272.

[10] S. Steinhorst, M. Lukasiewycz, S. Narayanaswamy, M. Kauer, and S. Chakraborty, "Smart cells for embedded battery management," in *Proc. of the International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2014, pp. 59–64.

[11] S. Steinhorst and M. Lukasiewycz, "Topology identification for smart cells in modular batteries," in *Proc. of the Conference on Design, Automation and Test in Europe (DATE 2015)*, 3 2015, pp. 1–4.

[12] N. H. Kutkut, "A modular nondissipative current diverter for ev battery charge equalization," in *Proc. of Applied Power Electronics Conference and Exposition, 1998. APEC'98*, vol. 2. IEEE, 1998, pp. 686–690.