

# Decentralized Non-Neighbor Active Charge Balancing in Large Battery Packs

Alexander Lamprecht<sup>1</sup>, Martin Baumann<sup>2</sup>, Tobias Massier<sup>1</sup>, Sebastian Steinhorst<sup>2</sup>

<sup>1</sup>TUMCREATE, Singapore, alexander.lamprecht@tum-create.edu.sg

<sup>2</sup>Technical University of Munich, Germany

**Abstract**—Recently, active charge balancing of the cells in battery packs has been gaining importance over state-of-the-art passive balancing solutions. The main advantage of active balancing lies in the ability to transfer charge between cells rather than dissipating it thermally. This enhances the overall efficiency and energy output of battery packs. In this paper, we develop a new class of strategies for decentralized operation of charge transfers between non-neighboring cells using appropriate balancing hardware architectures. While the benefits of the active balancing approach with a centralized controller have been discussed in literature extensively, the implementation of adequate strategies for scheduling charge transfers in decentralized battery management systems, which promise to be more robust and modular, have not been studied sufficiently so far. Furthermore, existing decentralized strategies only deal with charge transfers between neighboring cells. In order to compare our novel distributed non-neighbor balancing strategies to existing neighbor-only balancing strategies, we implement them in an open-source simulation framework for decentralized battery management systems. Our results show that we are able to improve the two most important metrics of balancing time and losses by up to 63 % and 51 %, respectively.

## I. INTRODUCTION AND RELATED WORK

Electric Vehicles (EVs) are of central importance for the future of sustainable mobility and reduced greenhouse gas emissions. However, the widespread acceptance and application of EVs is impeded by range anxiety and long charging times. This drawback is mainly induced by the limited capacity of current battery technologies. Fast charging of batteries, which leads to faster cell aging and requires more frequent battery replacements, is another disadvantage. Therefore, it is crucial to optimally use the given capacity of a battery pack.

A typical 18650 Li-Ion battery cell has a voltage of 2.5 V to 4.2 V with a capacity of 1500 mAh to 3600 mAh [1]. That means, in order to provide enough power and energy for EV applications, many cells need to be wired in parallel and series connection. While parallel-connected cells are always at the same voltage level and therefore the same SoC, cells which are connected in series do not necessarily have the same SoC (Fig. 1a). This effect is caused by slight differences between cells which result from variations in manufacturing, aging or environmental parameters such as temperature. When charging or discharging, the SoCs of series-connected cells hence drift apart. To accommodate for these variations, charge balancing is applied to battery packs.

This work was financially supported in part by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme. 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.

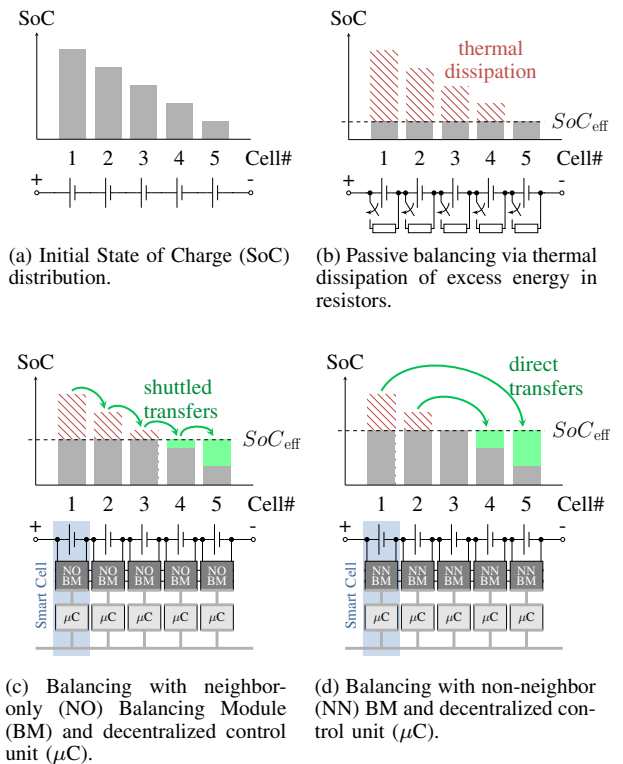


Fig. 1: Comparison of passive, neighbor-only and non-neighbor balancing for a given SoC distribution. The difference in effectively usable capacity  $SoC_{eff}$  is visible. The underlying modular smart cell platform with balancing module and microcontroller is displayed in 1c and 1d.

The state-of-the-art solution to this problem is to balance the battery pack passively. This means dissipating charge from cells with higher SoC thermally and therefore lowering the average SoC of the pack to the level of the cell with the lowest SoC. Fig. 1b visualizes this process. In contrast to passive balancing where energy is simply discarded, active balancing transfers excess charge from cells with higher SoC to cells with lower SoC (Figs. 1c and 1d). This allows for drastic reduction in losses but also requires additional circuitry to facilitate the charge transfers [2].

Different Active Charge Balancing (ACB) architectures have been discussed in literature [3] and can be divided into two fundamental categories; neighbor-only and non-neighbor balancing architectures. While neighbor-only archi-

tures only allow charge transfer between adjacent cells (Fig. 1c), non-neighbor architectures enable direct charge transfer between non-adjacent cells (Fig. 1d) [4].

The control of these balancing architectures is dictated by dedicated balancing strategies. These strategies are a fixed set of rules that aim to execute the balancing procedure as efficiently as possible. Conventional Battery Management Systems (BMSs) have a centralized computational unit that controls the balancing procedure. In this paper, we focus on a setup that uses decentralized controllers attached to each individual cell; a concept known as smart cells [5]. Each cell is equipped with a microcontroller, BM and communication interface as seen in Fig. 1c and 1d. These systems promise higher reliability and scalability due to their modular and decentralized nature and hence are in focus for many future energy storage solutions. Minimizing the computational overhead and communication effort through smart algorithms becomes an additional optimization task.

Our specific contributions in this paper are:

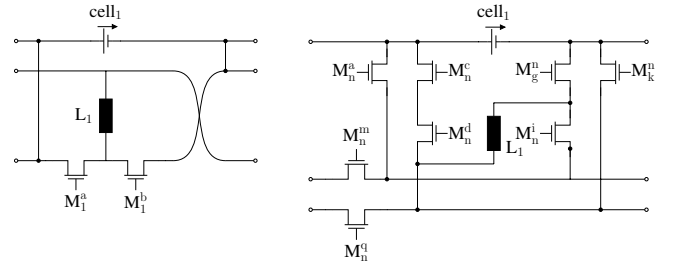
- Based on the architecture introduced in Section II, in Section III, we propose a novel decentralized non-neighbor balancing strategy which drastically improves balancing time, reduces balancing losses, and communication effort between cells over state-of-the-art neighbor-only strategies.
- In Section IV, we introduce a benchmarking methodology which allows for comparison of different balancing strategies based on balancing time and losses. The results of this benchmarking methodology (Section V) confirm the benefits in terms of time and losses arising from our proposed strategies.

Our methodology builds on top of the open source cyber-physical co-simulation framework presented in [6].

## II. NON-NEIGHBOR BALANCING ARCHITECTURE

In order to perform ACB, a suitable balancing circuit is required, which can transfer charge from one cell to another. Since two cells can't directly be connected due to their voltage difference, a temporary energy storage element is added to each cell. Architectures with different energy storage elements have been proposed in the literature [7]. Each of these approaches comes with its own advantages and disadvantages. In this paper, we focus on two established inductor-based architectures (see Fig. 2) which have proven to be highly efficient [8]. The difference between these two architectures lies in whether or not they are restricted to charge transfer between adjacent cells only. Architectures which fall into the first category are called neighbor-only architectures and typically require fewer components. Fig. 2a shows the corresponding circuit, from now on referred to as “*Narch*”.

Architectures which are able to facilitate direct charge transfer between non-adjacent cells are called non-neighbor architectures. Fig. 2b shows the corresponding circuit, from now on referred to as “*NNarch*”. This circuit contains a charge transfer bus which increases the transfer efficiency over larger distances. The direct charge transfer bus redundantizes inefficient consecutive transfers, shuttled through multiple neighboring cells, as it is necessary with the neighbor-only architecture for transfers over long distances.



(a) Neighbor-only balancing architecture “*Narch*” (b) Non-neighbor balancing architecture “*NNarch*”

Fig. 2: The charge balancing architectures “*Narch*” and “*NNarch*” [4].

In contrast to conventional BMS with only one central computational unit, we focus on smart cells with decentralized BMS. We chose the open-source cyber-physical co-simulation framework for smart cells in scalable battery packs from [6] to efficiently simulate large scale battery packs. We amended this framework to accommodate the non-neighbor balancing architecture in Fig. 2b and implemented our proposed non-neighbor balancing strategy which is explained in Section III.

## III. DECENTRALIZED NON-NEIGHBOR BALANCING

The two main optimization criteria for balancing strategies are balancing time and induced losses. The decentralized nature of the smart cell approach requires decision making on individual cell level. In this section, we therefore introduce a novel decentralized non-neighbor ACB strategy. In the following this strategy is referred to as “*NNAdapt*”

*NNAdapt*: The algorithm works as follows. A cell  $c_r \in C$ , where  $C$  is the set of all cells in the pack, requests charge from a determined transfer partner  $c_t$  if a set of conditions is fulfilled and the requesting cell's status is idle. A cell is considered idle if it is currently not participating in a charge transfer. If it is involved in a charge transfer process or is located between two transfer partners, it is considered blocked. The number of potential transfer partners is limited by a predefined maximum transfer distance  $\hat{d}$ . The transfer distance  $\hat{d}$  describes the distance between two charge transfer partners. A distance of  $\hat{d} = 1$  describes a transfer between adjacent cells.

Cells located between  $c_r$  and  $c_t$  cannot be involved in any other transfer due to the physical restrictions of the balancing architecture. After one charge transfer, blocked cells are made available for subsequent transfers.

Taking  $\hat{d}$  into account, a subset of reachable cells can be calculated for a given cell  $c_k$ , where  $k$  is the cell index with  $1 \leq k \leq n$  and  $n$  the total number of cells. The limiting indices of this subset  $k^+$  and  $k^-$  are calculated according to Equation 1.  $k^+$  stands for the cell closest to the positive terminal and  $k^-$  for the cell closest to the negative terminal.

$$\begin{aligned} k^+ &= k - \hat{d} - \frac{1}{2} \left[ \text{sgn} \left( k - \hat{d} - \frac{1}{2} \right) - 1 \right] (\hat{d} - k + 1) \\ k^- &= k + \hat{d} + \frac{1}{2} \left[ \text{sgn} \left( n - k - \hat{d} + \frac{1}{2} \right) - 1 \right] (k + \hat{d} - n) \end{aligned} \quad (1)$$

Equation 1 ensures that the limiting indices  $k^+$  and  $k^-$  of each cell  $c_k$  are between 1 and  $n$ . These limiting indices are used to determine the subsets of cells within the reachable distance  $\hat{d}$ . Subset  $C_k^+$  represents all cells closer to the positive terminal of the battery pack than cell  $c_k$ . Subset  $C_k^-$  contains all possible transfer partners closer to the negative terminal.

Whether to request charge from subset  $C_k^+$  or  $C_k^-$  is decided as follows. Let  $Z$  be the set of SoCs of all cells in  $C$ . Then  $Z_k^+$  is the set of SoCs of all cells in  $C_k^+$  and  $Z_k^-$  is the set of SoCs of all cells in  $C_k^-$ . Charge from a cell in subset  $C_k^+$  is requested if  $\max(Z_k^+) \geq \max(Z_k^-)$ . Otherwise, charge from a cell in subset  $C_k^-$  is requested. This is shown in Equation 2. The determined cell  $\hat{c}_k$  with SoC  $\hat{z}_k$  is requested if  $z_k < \bar{Z}$ , where  $z_k$  is the SoC of cell  $c_k$  and  $\bar{Z}$  is the average SoC of the battery pack.

$$\begin{aligned}\hat{z}_k &= \max(\max(Z_k^+), \max(Z_k^-)) \\ \check{z}_k &= \min(\min(Z_k^+), \min(Z_k^-))\end{aligned}\quad (2)$$

Second, if  $\min(Z_k^+) < \min(Z_k^-)$  then  $\check{c}_k$  is acknowledged for a charge transfer according to Equation 2. The transfer request from cell  $c_{\min.k}$  with SoC  $\check{z}_k$  is acknowledged only if  $z_k > \bar{Z}$ , where  $z_k$  is the SoC of the transmitting cell. The balancing procedure is terminated if  $\forall z_k \in Z : z_k \in [\bar{Z} - \varepsilon, \bar{Z} + \varepsilon]$ .

The aforementioned set of rules is an enhancement of the neighbor-only strategy min-max from [6] to utilize non-neighbor transfers and already makes for a converging balancing strategy. However, since this straightforward strategy is not utilizing the full potential of the non-neighbor balancing approach, we amend it in the following with a more extensive set of conditions. Nonetheless, we use this intermediate strategy for performance comparison and therefore will refer to it as “NNMinMax”. With it, balancing losses can be reduced due to the advantages of direct transfers. Balancing time however can be longer than with conventional neighbor-only balancing strategies due to the high number of blocked cells during long transfers.

To counteract this drawback, we propose the adaptive strategy “NNAdapt”. The fundamental idea of this strategy is to make decisions not solely based on SoCs, but to use the cells’ idle states in combination with SoCs instead. The smart cell approach allows us to broadcast the idle states of each cell to the entire pack. With this knowledge, each cell  $c_k$  is able to detect the subset of all idle cells  $C_k^{idle} \subset C$  it can be engaged in a transfer with. This means that there are no blocked cells within a potential charge transfer path.

A cell  $c_k$  requests charge if its SoC  $z_k$  is below the battery pack’s average SoC  $\bar{Z}$  and its SoC equals the minimum SoC of the self-detected idle subset of cells  $\min(Z_k^{idle})$ . Cell  $c_k$  determines the index of the cell with the highest SoC  $\max(Z_k^{idle})$  within the reachable idle subset.  $c_k$  consequently requests charge from this determined cell  $c_t$ .

The requested transmitting cell  $c_t$  only acknowledges and initiates the requested transfer if its own SoC is above the battery pack’s average SoC, i.e.,  $z_t > \bar{Z}$ . Moreover, it checks whether its SoC is the maximum SoC in  $Z_k^{idle}$ , i.e., whether  $z_k = \max(Z_k^{idle})$ . A transfer will only be engaged with the cell that has the minimum SoC  $\min(Z_k^{idle})$  within the idle subset.

Furthermore, a cell  $c_k$  only requests charge if it belongs to the set of best possible transfer partners  $C_{bt}$  within a determined distance  $\hat{d}$  and the subset of idle cells  $C_k^{idle}$ . Hence, the transmitting cell only acknowledges a transfer if it also belongs to  $C_{bt}$ .

Both cells belong to  $C_{bt}$  if their indices are between  $k_{idle}^+$  and  $k_{idle}^-$  and their distance does not exceed the determined maximum transfer distance  $\hat{d}$ . In addition, there must not be any other pair of cells within this distance with a higher SoC difference than that between  $c_j$  and  $c_k$ .

The mathematical formulation of these conditions is given in Equation 3.

$$\begin{aligned}c_j, c_k \in C_{bt} &\iff j, k \in [k_{idle}^+, k_{idle}^-] \wedge |j - k| \leq \hat{d} \wedge \\ &\neg \exists l, m \in [k_{idle}^+, k_{idle}^-] : |z_l - z_m| > |z_j - z_k|\end{aligned}\quad (3)$$

Up to this point, this strategy already leads to a high number of concurrent transfers and therefore a favorable balancing time. The focus on maximum SoC deviation, however, could lead to transfers over long distances, effectively blocking large parts of the battery pack. To avoid this, the maximum allowed distance  $\hat{d}$  is reevaluated after each charge transfer based on the transfer’s calculated efficiency  $\tilde{\eta}$ . The transfer efficiency  $\tilde{\eta}$  is calculated based on the energy differential in the transmitting and receiving cells between the current time step  $(z_r, z_t)$  and the previous one  $(z_r^*, z_t^*)$ .

$$\tilde{\eta} = \frac{\Delta E_r}{\Delta E_t} = \frac{E_C(z_r) - E_C(z_r^*)}{E_C(z_t^*) - E_C(z_t)}\quad (4)$$

If  $\tilde{\eta}$  drops below a predefined efficiency threshold, the maximum permitted distance  $\hat{d}$  is decreased and broadcasted to all cells. This effectively splits the battery pack into local subsets and decreases the overall balancing time and losses.

The pseudo code in Algorithm 1 summarizes the NNAdapt strategy.

---

**Algorithm 1** Request and acknowledgment processes of request-driven NNAdapt strategy for cell  $c_k$ . Request policy and acknowledgment policy are true or false.

---

```

1: procedure REQUEST (to receive charge)
2:   if request_policy( $c_k$ )=true and status=idle then
3:     determine transfer partner  $c_t$  from best possible transfer
       candidates in  $C_{bt}$ ;
4:     request charge transfer from cell  $c_t$  within transfer dis-
       tance  $d$  and the idle subset;
5: procedure ACKNOWLEDGE (to send charge to  $c_r$ )
6:   if Acknowledge policy and status=idle then
7:     success=block_cells( $c_t, \dots, c_r$ );
8:     if success then
9:        $z_{tb}, z_{rb} = z_t, z_r$ ;
10:      acknowledge request from cell  $c_r$ ;
11:      transfer charge to cell  $c_r$  for time  $T_m$ ;
12:      unblock_cells( $c_t, \dots, c_r$ );
13:       $\tilde{\eta}$ =calculate_efficiency( $z_{tb}, z_{rb}, z_t, z_r$ );
14:      if efficiency  $\tilde{\eta} <$  efficiency threshold then
15:        decrease maximum transfer distance  $\hat{d}$  by 1;
16:        broadcast distance  $\hat{d}$  to other cells;

```

---

Table I summarizes the request and acknowledgment policies for both Strategies NNMinMax and NNAdapt. The search and identification of the best transfer  $C_{bt}$  within distance  $\hat{d}$  and the subset of idle cells is included in these policies.

#### IV. BENCHMARKING METHODOLOGY

To allow a comparison of the results of the balancing simulations produced by state-of-the-art neighbor-only and our proposed non-neighbor strategies, we developed a sophisticated benchmarking method. Currently, to the best of our knowledge, there exists no standardized method to compare ACB strategies. Our method supplies a theoretical absolute baseline for balancing time and losses against which our results can be compared. The details of this method are explained below.

The main objective of ACB strategies is to decide which charge transfer combination out of the set of all possible transfers to select. This directly affects the energy dissipation and time of the balancing process which in turn are the main parameters to compare different balancing strategies. Due to resistive losses occurring with each transfer, the amount of energy dissipated depends on the total number of charge transfers during a balancing process. In neighbor ACB, charge transfers can only be conducted between adjacent cells, which increases the total number of transfers and consequently the losses. In non-neighbor ACB, charge can be transferred directly from one cell to another, which reduces the total number of transfers. Though the losses for each single non-neighbor transfer are higher than for neighbor transfers due to losses in the additional components (Fig. 2), non-neighbor charge balancing promises reduced overall losses as a neighbor-only strategy would require  $d$  transfers to bridge the distance, where non-neighbor strategies only require a single transfer [9].

Since it is not meaningful to compare the results of different balancing strategies relative to the point of origin, we formulate a method to obtain a theoretical absolute lower limit for balancing time and losses for a given random SoC distribution.

The theoretical limit calculation *does not* consider the blocking state of interjacent cells between two charge transferring cells and therefore is not reachable in practice. Each cell, however, is only allowed to participate in one charge transfer at a time.

*Theoretical Balancing Reference:* The general goal is to obtain theoretical baselines for balancing time and losses for a given SoC distribution. To dismiss trivial cases, we only take initial SoC distributions above a certain standard deviation into account. Since the details of the initial SoC distribution have a substantial influence on the performance of a given strategy, it is important to consider a statistically significant number of simulation runs and average the outcomes to obtain conclusive results.

First, we focus on obtaining the baseline for the balancing time. From the set of cells  $C$  we first pick the two cells with the largest positive and negative SoC deviation. We

TABLE I: Request and acknowledgment policies for the non-neighbor balancing strategies NNMinMax, and NNAdapt.

Strat.	Request $c_r \rightarrow c_t$		Acknowledgment $c_t \rightarrow c_r$	
	$c_r$	$c_t$	$c_t$	$c_r$
NNMinMax	$z_r < \bar{Z}$	$z_t = \hat{z}_k$	$z_t > \bar{Z}$	$z_r = \hat{z}_k$
NNAdapt	$(z_r < \bar{Z}) \wedge (c_r \in C_{bt})$	$c_t \in C_{bt}$	$(z_t > \bar{Z}) \wedge (c_t \in C_{bt})$	$c_r \in C_{bt}$

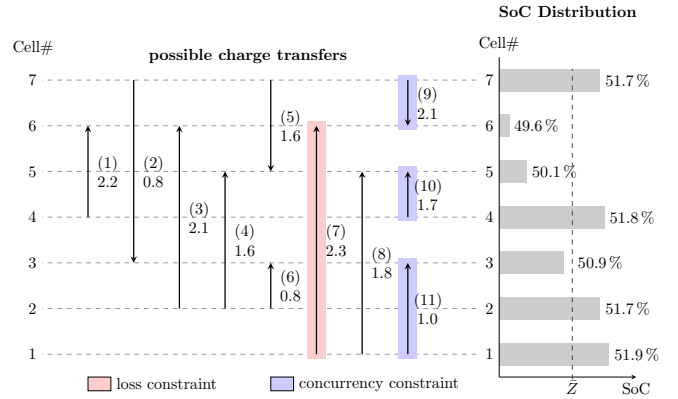


Fig. 3: Illustration of the realistic reference algorithm. Arrows represent potential transfers for an exemplary SoC distribution. Certain transfers are selected in order maximize performance in either balancing loss or time.

successively match the cells with the second largest deviation, third largest deviation and so on. Disregarding the resulting losses or blocking states of the cells, this method guarantees the fastest theoretically possible balancing process. We set the resulting balancing time as the baseline of our benchmark.

Analogously, we define constraints to obtain the lower limit of our balancing losses. Setting the number concurrent transfers to one and the maximum transfer distance to the total number of cells in the pack gives the baseline for the balancing losses.

*Realistic Balancing Reference:* In order to obtain a realistic balancing reference, we make one adjustment to the aforementioned algorithm. Instead of executing  $n$  charge transfers, we only consider transfers which do not violate the blocking conditions. We also take the architecture and the distance between possible transfer partners into account. For calculating realistic references, the algorithmic evaluation of a given SoC distribution is important.

Fig. 3 shows an exemplary battery pack consisting of seven cells. The black arrows represent potential charge transfers. Visualized potential transfers follow the convention that they should transfer charge from a cell with above average SoC to a cell with below average SoC. Moreover, charge should be transferred towards the direction of a subset with lower average SoC. Analogous to the theoretical balancing reference, we implement two transfer selection procedures. The **loss constraint** procedure aims to minimize balancing losses while having a low number of concurrent transfers, which leads to longer balancing time. To achieve that, all potential transfers are sorted in descending order by SoC difference. A transfer is characterized by the cell number of the transmitting cell, the cell number of the receiving cell, the local distance and the difference in SoC between them. Transfers with high SoC gradients are preferred. Transfers sorted by SoC are gradually added to a list if they can be executed at the same time. In the example in Fig. 3, the loss constraint selection only chooses transfer (7) without any concurrent transfers.

By contrast, the **concurrency constraint** selection method achieves as many concurrent transfers as possible without

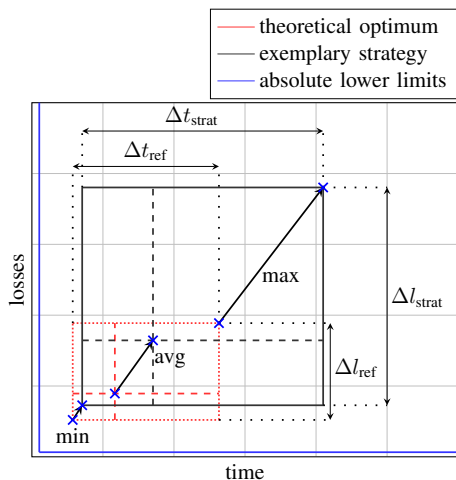


Fig. 4: Exemplary evaluation of balancing strategies.

considering the efficiency of each single transfer. Thus, this selection method focuses on low balancing time. To achieve this, all possible transfers are sorted in ascending order by the charge transfer distance in cells between transfer partners. A potential overlapping transfer substitutes another transfer in the execution list, if its SoC difference is higher. Naturally the loss constraint and the concurrency constraint are mutually exclusive.

*Comparison of Strategies:* Our benchmarking method allows us to evaluate different balancing strategies executed on various architectures. Since the initial SoC distribution substantially influences the performance of a given balancing strategy, comparisons are made on the basis of a statistically significant number of randomly generated initial SoC distributions. Simulation results are compared regarding balancing time and losses. For a set of initial distributions, the results yield five significant values: the four extrema for minimum and maximum balancing time and losses as well as the resulting average balancing time and losses. Fig. 4 shows an exemplary visualization of these values in a 2D scatter plot of time and losses. The solid lines represent the time and loss limits, containing all calculated results for a given set of initial distributions. Dashed lines give the coordinates of the calculated average values for balancing time and losses. The dotted red box visualizes the theoretical optimal limits for a given architecture. The simulation results for a given strategy are contained within the black lines. The deviation of the simulation results from the optimal solution becomes manifest in the surface area of the containing boxes.

Each strategy can therefore be characterized sufficiently by its minimum ( $t_{\min}/l_{\min}$ ), average ( $t_{\text{avg}}/l_{\text{avg}}$ ) and maximum ( $t_{\max}/l_{\max}$ ) points.

## V. RESULTS

With the help of the cyber-physical co-simulation framework the balancing process for an EV-typical battery pack with 96 cells in series and a capacity of 23 kWh, as it is used in the 2011 Nissan Leaf [10], is simulated. 100 randomly generated initial SoC distributions with a maximum variation of 3% are generated and the balancing process simulated

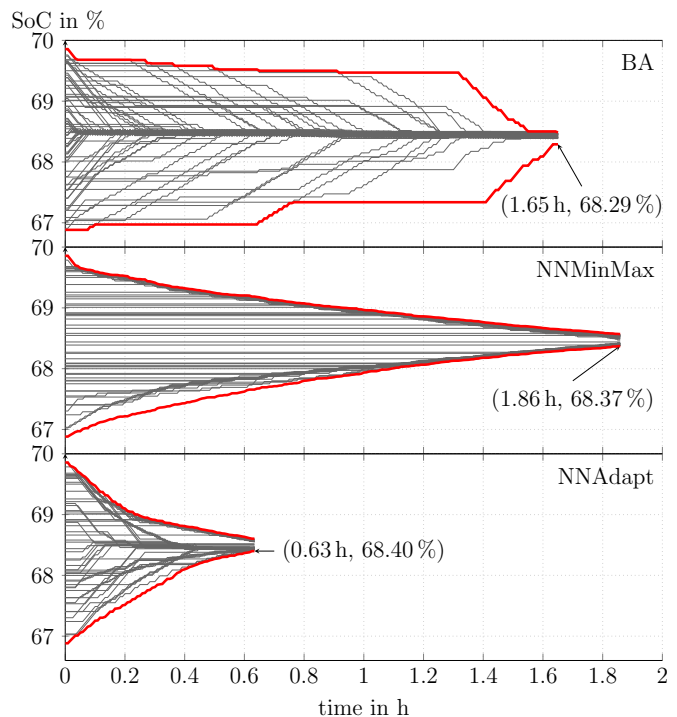


Fig. 5: Balancing process for an exemplary initial SoC distribution in a battery pack for strategies BA, NNMinMax and NNAdapt.

for different strategies. To evaluate our proposed strategy NNAdapt we chose the three best performing decentralized neighbor-only strategies from [6] as basis for the comparison. These strategies are BelowAverage (BA), Boundary (Bou) and MinMax. For fairness, The simulations for the neighbor-only strategies are conducted on architecture Narch while the non-neighbor strategies utilize NNarch (See Section II). For both architectures the same constant balancing current is assumed.

A detailed view of the results of the simulation of the balancing procedure is displayed in Fig. 5. The graphs show the development of the cell SoCs for an exemplary initial SoC distribution over time using the strategies BA, NNMinMax and NNAdapt.

While the BA strategy focuses on successively bringing the SoC of particular cells to the average SoC the NNMinMax strategy mainly facilitates charge transfers between the extrema of the current SoC distribution, resulting in a low number of concurrent transfers.

The overall results are a balancing time of 1.65 h and a final pack SoC of 68.29% for the BA strategy and a balancing time of 1.87 h and a final pack SoC of 68.37% for the NNMinMax strategy. While the NNMinMax strategy can utilize the advantages of the non-neighbor architecture in reducing losses, it does it at the cost of a very low number of concurrent transfers, resulting in longer balancing times than the BA strategy. In contrast, the NNAdapt strategy achieves considerably more concurrent transfers due to its sophisticated set of rules. Hardly any SoCs remains constant for long. NNAdapt balances the pack to the termination threshold of

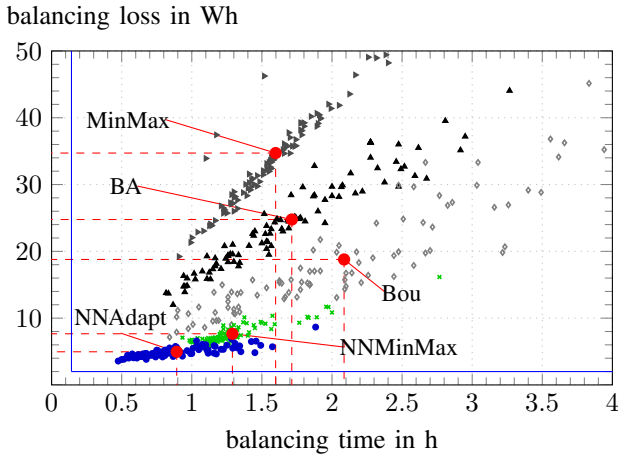


Fig. 6: Balancing loss and time of strategies MAX, BA, Bou, NNMinMax and NNAdapt. The simulation results arise from 100 initial SoC distributions with a maximum deviation of 3%. The red dots represent the average balancing time and losses for each strategy.

0.2% in 0.63 h and achieves a final SoC of 68.40%, while remarkably reducing the balancing time by 66% compared to NNMinMax and by 61% compared to BA.

TABLE II: Comparison of balancing times and losses of non-neighbor ACB strategies with the NNarch balancing architecture.

strategy	time in h			losses in Wh		
	min	avg	max	min	avg	max
NNMinMax	0.93	1.29	2.77	5.61	7.66	16.16
NNAdapt	<u>0.48</u>	<u>0.89</u>	<u>1.89</u>	<u>3.58</u>	<u>4.96</u>	<u>8.66</u>

We conduct the simulations for a total of 100 random initial SoC distributions. Each simulation results in a tuple of overall balancing time and losses, characterizing the used strategy. All resulting tuples are plotted in the scatter plot in Fig. 6. Table II summarizes the simulation results for our proposed strategies NNMinMax and NNAdapt with the help of the minimum, average and maximum balancing times and losses. The table underlines that strategy NNAdapt results in lower average balancing time and comparable losses to strategy NNMinMax, while the variance of these values is greatly reduced.

Table III shows a comparison between our proposed strategy NNAdapt, the intermediate strategy NNMinMax, and the state-of-the-art neighbor-only strategies regarding the resulting average balancing losses and times and the corresponding variance. This direct comparison is meaningful, because of the theoretical limits obtained from the benchmarking methodology introduced in Section IV. All values are normalized to these absolute theoretical limits, which are a balancing time of 8.5 min and a balancing loss of 2.00 Wh. Table III shows that our proposed strategy is not only significantly superior in terms of balancing losses and time, but also result in a much more consistent balancing process. NNAdapt improves the efficiency of non-neighbor balancing, with only a factor of less than 2.5 between the simulated loss results and the

TABLE III: Comparison of characteristic values of the discussed neighbor-only strategies BA, MAX, Bou and the proposed non-neighbor strategies NNMinMax and NNAdapt, normalized to the theoretical limits.

strategy	time		losses	
	avg	spread	avg	spread
BA	12.07	15.40	12.38	14.34
MinMax	11.25	10.64	<b>17.36</b>	16.23
Bou	<b>14.69</b>	<b>20.75</b>	9.39	<b>20.51</b>
NNMinMax	9.08	10.40	3.83	4.25
NNAdapt	<b>6.28</b>	<b>6.98</b>	<b>2.48</b>	<b>1.85</b>

theoretical absolute minimum.

## VI. CONCLUSION

In this paper we introduced a novel non-neighbor ACB strategy for battery packs with decentralized BMSs. We validated our results with the help of an open source simulation framework for decentralized battery management systems. We extended this framework to support non-neighbor architectures and defined a benchmarking method to calculate a theoretical baseline of balancing losses and time. This baseline serves as foundation for comparing the results from both neighbor-only and non-neighbor strategies. We show that our proposed strategy reduces the average balancing time by up to 63% and average balancing losses by 51% compared to state-of-the-art neighbor-only balancing solutions. Furthermore we show that our strategy approaches the theoretical limits especially with regard to balancing losses, making it highly suitable for application in EVs.

## REFERENCES

- [1] Samsung Group, "Specification of product: Lithium-ion rechargeable cell for power tools, model name: Inr18650-25r," 2014.
- [2] W. C. Lee, D. Drury, and P. Mellor, "Comparison of passive cell balancing and active cell balancing for automotive batteries," in *2011 IEEE Vehicle Power and Propulsion Conference*, pp. 1–7, Sept 2011.
- [3] T. Szalai, U. Schwalbe, M. Schilling, F. Endert, T. Heidrich, and S. Ivanov, "Design of an active battery management system for electric vehicles," in *PCIM Europe 2014: International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, pp. 1–8, May 2014.
- [4] M. Lukasiewicz, M. Kauer, and S. Steinhorst, "Synthesis of active cell balancing architectures for battery packs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 35, no. 11, pp. 1876–1889, 2016.
- [5] S. Steinhorst, M. Lukasiewicz, S. Narayanaswamy, M. Kauer, and S. Chakraborty, "Smart cells for embedded battery management," in *2014 IEEE International Conference on Cyber-Physical Systems, Networks, and Applications*, pp. 59–64, Aug 2014.
- [6] S. Steinhorst, M. Kauer, A. Meeuw, S. Narayanaswamy, M. Lukasiewicz, 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, pp. 62:1–62:26, 5 2016.
- [7] J. Cao, N. Schofield, and A. Emadi, "Battery balancing methods: A comprehensive review," in *2008 IEEE Vehicle Power and Propulsion Conference*, pp. 1–6, Sept 2008.
- [8] S. Narayanaswamy, M. Kauer, S. Steinhorst, M. Lukasiewicz, and S. Chakraborty, "Modular Active Charge Balancing for Scalable Battery Packs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 974–987, mar 2017.
- [9] F. Baronti, R. Roncella, and R. Saletti, "Performance comparison of active balancing techniques for lithium-ion batteries," *Journal of Power Sources*, vol. 267, pp. 603 – 609, 2014.
- [10] J. Smart and S. Schey, "Battery electric vehicle driving and charging behavior observed early in the ev project," *SAE International Journal of Alternative Powertrains*, vol. 1, no. 1, pp. 27–33, 2012.