



Multidimensional Machine Learning Balancing in Smart Battery Packs

Fonso, Roberta Di; Sui, Xin; Acharya, Anirudh Budnar; Teodorescu, Remus; Cecati, Carlo

Published in:

IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society

DOI (link to publication from Publisher):

[10.1109/IECON48115.2021.9589698](https://doi.org/10.1109/IECON48115.2021.9589698)

Creative Commons License

CC BY 4.0

Publication date:

2021

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Fonso, R. D., Sui, X., Acharya, A. B., Teodorescu, R., & Cecati, C. (2021). Multidimensional Machine Learning Balancing in Smart Battery Packs. In IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society (pp. 1-6) <https://doi.org/10.1109/IECON48115.2021.9589698>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Multidimensional Machine Learning Balancing in Smart Battery Packs

Roberta Di Fonso
DISIM
University of L'Aquila
L'Aquila, Italy
roberta.difonso@graduate.univaq.it

Xin Sui
Department of Energy Technology
Aalborg University
Aalborg, Denmark
xin@energy.aau.dk

Anirudh Budnar Acharya
Department of Energy Technology
Aalborg University
Aalborg, Denmark
abac@energy.aau.dk

Remus Teodorescu
Department of Energy Technology
Aalborg University
Aalborg, Denmark
ret@energy.aau.dk

Carlo Cecati
DISIM
University of L'Aquila
L'Aquila, Italy
carlo.cecati@univaq.it

Abstract—Lithium-ion batteries have high energy density, lightweight and long life cycle, thus they are the choice for powering electric vehicles. The needed high voltage battery pack is achieved using series-connected cells, that ideally should be identical. However, parameter variations of cells in EVs, along with different working conditions can cause State of Charge (SoC) and temperature imbalances that shorten battery lifetime. Moreover, a series connection can potentially be exposed to single-cell failure. This paper proposes a redundant smart battery topology based on the series connection of individual cell modules. Each module is formed by a cell with an insertion/bypass circuit and a wireless processor that monitors cell states and communicates with a Master controller. To synthesize the nominal voltage, a battery pack with a small redundant number of cells is considered. In this way, it can be made both fault-tolerant and reach higher safety. The problem is then shifted to the control algorithm that at each sampling time has to select “n” cells out of the total cells, according to some goals. A Machine Learning-based control algorithm was developed and tested in Matlab to insert/bypass the cells and reach simultaneous balancing of both SoC and temperature. The new method based on the K-nearest algorithm has been compared in simulation with a conventional sorting balancing method and showed superior performance, especially in temperature balancing.

Index Terms—smart battery, K-nearest, Machine Learning, AI, balancing, SoC, SoT

I. INTRODUCTION

Thanks to their high energy density, lightweight, and long life cycle, Lithium-ion batteries are the most common power source of electric vehicles. Battery packs are assembled by long string of single cells connected in series and/or parallel as to meet the voltage and power requirements of the application [1]. At present, the battery pack is one of the most expensive components of an electric vehicle and is therefore essential to employ strategies to extend the battery lifetime, increase the driving range and assure optimal operation of the pack. The cells inside the battery pack are not perfectly equal. The cells can have different capacities and this could happen for various

reasons: it could be a result of the manufacturing process or the aging mechanism that affects the cells differently [2]. Moreover, leakage currents and different working conditions can impact on the cells, causing state of charge (SoC) imbalances that lead to different depth of discharge (DOD) of the cells [3]. Temperature imbalance of the battery pack is another important issue and can be the result of parameter variation of the cells (internal resistance variation, thermal resistance variation) and temperature gradient variation of the coolant of the EV [3]. As a result, the cells in the pack are not equally utilized, and the weakest cell in a series-connected battery can stop the whole operation due to overcharge or over-discharge, so it is not possible to use all the energy stored in the battery. Furthermore, the battery lifetime is heavily affected by SoC imbalance and high temperature. To circumvent these issues, strategies are needed for balancing SoC and limiting the temperature gradient. Classical active balancing techniques use additional circuits, like DC/DC converters to exchange energy between cells [4–9]. A more flexible and efficient option to achieve the active balancing of the battery pack is represented by the smart batteries (SB) [2], [10–13]. SB topologies introduce modularity and reconfigurability in the battery pack thanks to the capability to insert/bypass single cells. This feature can be exploited to reach multidimensional control, that is simultaneous balancing of SoC and limiting temperature gradient inside the battery. In this paper, a K-nearest-based control algorithm is presented. The algorithm chooses at sampling time a subset of the cells to maintain the SB nominal output voltage. The cells are selected based on their sampled SoC and temperature values. The algorithm allows to trade off the SOC and temperature spread of the cells during discharge. A lower temperature spread has also the effect to slow down the aging of the cells and thus extending the lifetime of the battery pack. The paper is organized as follows: Section II presents the smart battery topology adopted in

this work; Section III illustrates the artificial intelligence (AI) applications to cell state estimation; Section IV describes the multidimensional K-nearest approach; Section V presents the simulation model; Section V shows the results of simulations, which are carried out using Matlab, Simulink and Simscape and Section VI draws some conclusions and explores future work.

II. THE SMART BATTERY

Smart Battery (SB) [2], [11] is a new concept that combines advanced power electronics, wireless communication and artificial intelligence with the goal to develop the new generation of battery solutions for transportation and grid storage where the following new features are achieved: increased safety and reliability by fault-tolerant operation, user-controlled lifetime and software reconfiguration for 2nd lifetime applications. The structure of SB is shown in Figure 1 and consists of a battery cell, a switching device, and a slave controller. The cell is not directly connected to the battery string but

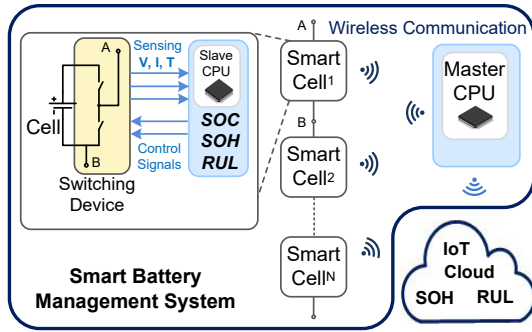


Fig. 1. Smart Battery Structure.

through the switching device which is implemented by a simple half-bridge MOSFET circuit and allows two operation modes: inserted or bypassed, as shown in Figure 2. Thus cell-

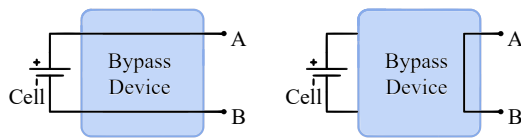


Fig. 2. Operation mode of the switching device: inserted cell on the left, bypassed cell on the right.

level load management is achieved. The slave controller can monitor the voltage, current, and temperature of the cell and also estimate SOC of the cell. All slaves are communicating wirelessly with a Master controller that is performing higher (package) level functions as: SOH estimation and prediction using AI, SOC&SOH balancing and lifetime control. The balancing process is done by bypassing one cell at a time and thus not affecting the load current. In contrast to other active balancing methods, this balancing method is not requiring the use of DC/DC converters and is therefore an ultra low energy balancing. The switching device is also used to permanently bypass faulted cells and thus adding fault tolerant operation

mode, increasing the safety and reliability at system level. Due to the switching operation, the SB might require an output LC low-pass filter if this is not already present in the load.

III. AI-BASED STATE ESTIMATION OF LITHIUM CELLS

AI is emerging and increasingly used for state estimation in batteries, such as state of charge (SOC), state of health (SOH), state of temperature (SOT) [14], remaining useful lifetime (RUL) prediction [15], and balancing control [16]. AI-based lifetime predictor can be trained based on laboratory data, then the mapping between features (i.e. the health indicators that contain the aging information extracted from measured V , I , and T) and target output (SOC, SOH, SOT, and RUL, etc.) can be obtained. The effective information can be extracted from the raw data automatically thanks to the ability of deep learning to process high-dimensional data. According to the estimated states, an AI-based lifetime controller can also be designed. Under this framework, the model after the rough initial training has the ability to learn independently from newly arrived data, which will greatly reduce the training cost. The diagram of using AI-based battery state estimation and multidimensional balancing in SB is shown in Figure fig:Control-Block-Scheme. The computational effort of the control algorithms increases significantly when the number of cells in the battery pack increases, thereby requiring a high-end processor that consumes high energy from the battery pack. This is typically not desired in automotive applications. The computational effort can be reduced using AI. The AI-based approach allows the implementation of complex algorithms in low-cost, low-energy microprocessors with dedicated AI units from vendors such as Texas Instruments, NVIDIA, NXP, etc., for automotive applications. The AI-based state estimation in SB provides flexibility and continuous training of the model, improving the performance of the specific battery pack used in the EV.

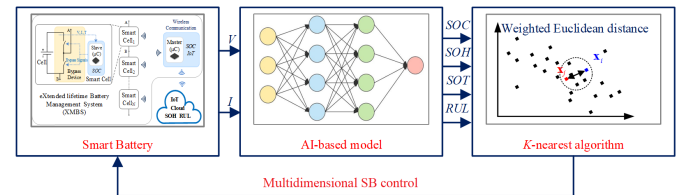


Fig. 3. Control Block Scheme.

IV. MULTIDIMENSIONAL K-NEAREST CONTROL ALGORITHM

The SB has more cells than strictly necessary, so only a subset $n \leq N$ of the cells is needed to synthesize the output voltage. The redundancy and the reconfigurability of the topology allow to dynamically choose among cells. A cell with low SoC (i.e. below 20%) can be bypassed in favour of cells with higher SoC; cells with out-of-range temperatures can be bypassed until they return to acceptable values. Since

the algorithm is dynamic, cells can be inserted or bypassed based on the working condition. In this way, the bypassed cells have time to rest until they are either inserted again by the algorithm or until the next charge. Furthermore in case of a cell fault the cell is permanently excluded and the SB keeps working. Given the N cells of the SB, the goal is to optimally choose the subset $n \leq N$ of cells to balance both the SoC and temperatures. The algorithm chooses the cells with highest SoC and lowest temperatures. The proposed control algorithm leverage on the K-nearest classification algorithm. Given a data-set of points and one or more test examples, the K-nearest algorithm calculates the distances from all the points in the data-set and the test example(s). These distances are then sorted and the k nearest neighbors (i.e the points with the minimum distances from the test examples) are returned. By properly choosing k , the test examples and the distance metric, we can use the K-nearest technique to optimally choose the n cells that need to be inserted in the SB. In this work the data-set is given by the N couples of SoC and temperatures $(SoC_i, T_i), i = 1, \dots, N$, the test example is given by the couple (SoC_{max}, T_{min}) , $k = n$ and the metric used for the distance is a weighted Euclidean distance. The control algorithm receives as input the voltage, SoC and temperature values of the N cells and outputs the control signals for the Half-Bridges of the N cell modules. The voltage values are used to calculate the number n of cells to synthesize the desired output voltage, while the SoCs and the temperatures are used to choose which cells need to be inserted to achieve balancing. The steps of the control algorithm are the following:

- given the SoC and temperature values of all cells create a virtual point v by choosing the highest SoC and the minimum temperature: $v = (SoC_{max}, T_{min})$;
- calculate the distance vector d :

$$d = \sum_{i=1}^N \left(k_S (SoC_i - SoC_{max})^2 + k_T (T_i - T_{min})^2 \right)$$

where: k_S, k_T are weighting factors for the SoC and the temperature;

- sort the distances d in ascending order and find the index vector of the corresponding cells;
- scan the index vector and add the voltages of the corresponding cells until the desired output voltage is reached;
- compute the binary vector to drive the h-bridges that insert/bypass the cells (1 to insert a cell, 0 to bypass it).

V. RESULTS

THE SIMULATION MODEL

The SB system is simulated in Matlab-Simulink-Simscape. In order to reduce the simulation time to acceptable levels, the SB model is built with only 25 modules, instead of 150 needed for a real EV battery (a model scaled down by 6). At each sampling time, the control unit connects in series the right number of cells in order to output an average of 60 V (instead of 360 V). The battery cell parameters shown in Table I are

from the preset model n. 6 from the Simscape Power System Library. With these cells, the parameters of the SB pack are shown in TableII.

TABLE I
CELL PARAMETERS.

Nominal voltage	3.6 V
Cell capacity	48 Ah

TABLE II
SMART BATTERY PACK PARAMETERS.

Controlled output voltage	60 V
Energy stored	4.32 kWh

Case Study

In order to get meaningful results, it is useful to think to the following case study. A car is parked in a garage for a time long enough to have all the cells at the same initial ambient temperature, which is 25°C. When the car starts for a test drive, the temperatures of the used cells will rise due to their internal resistance. The simulation is carried out thinking that the battery is contained in a box whose ambient temperature increases as the mean temperature of the cells. No cooling action has been considered. The initial SoC of the cells is set up between 80% and 90%, as shown in Fig. 4.

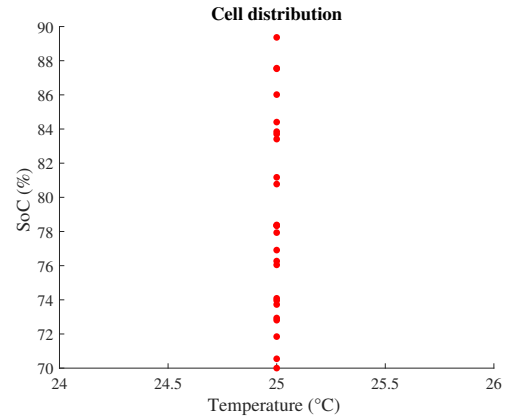


Fig. 4. Cell distribution at the beginning of the test.

SIMULATION RESULTS

The SB performance depends on the control algorithms used to select the right cells at each sampling time. Their selection can be based on a single dimension, like SoC, or on multiple dimensions, like SoC and temperature, or aging, or other parameters. The following results are related to two distinct control algorithms: (i) an SoC Sorting algorithm, based on cell SoC only and, (ii) the proposed Multidimensional K-nearest control algorithm (MKNA), based both on cell SoC and temperature. Both algorithms maintain the desired output voltage close to $V_{des} = 60 V$ using 16 or even more cells with discrete increments. The load is simulated by a 1.8 Ω resistor

resulting in an average discharge current of about 33 A.

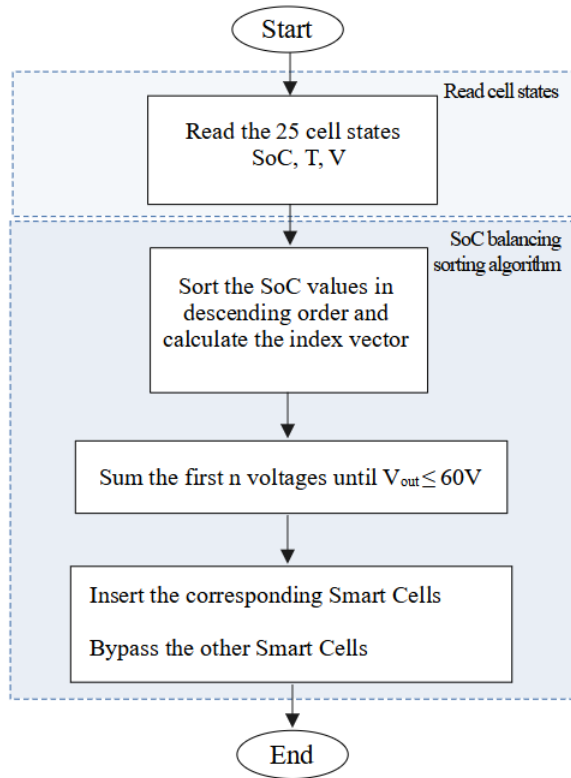
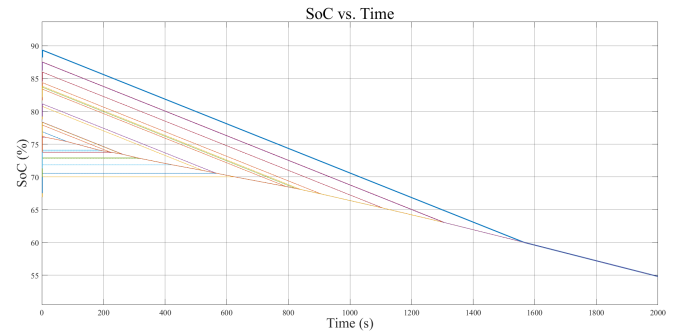


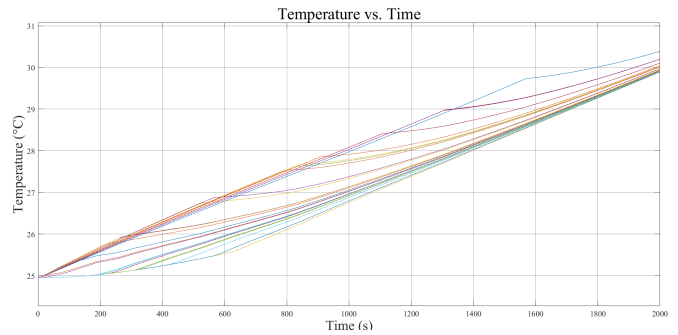
Fig. 5. SoC balancing sorting algorithm flow chart.

SoC Balancing Sorting algorithm

This control algorithm is based on an SoC sorting. At each step, the algorithm sorts the SoC of all the cells in descending order and chooses the first n cells to be inserted in the SB string. The number n is dynamically defined in order to synthesize the desired output voltage. A flow chart of the algorithm is shown in Fig. 5. The chosen cells have the relative highest SoC, while the cells with the lowest SoC have time to rest. The algorithm action is best seen on an animated scatter plot (SoC vs. Temperature). Used cells are pushed down-right (SoC decrease and temperature increase) until all cells reach roughly the same SoC value (a flat line). From then on, the cells will be used more uniformly and remain horizontally aligned. The total simulation time is 2000 s. Fig. 8 shows four screen shots of the scatter plot at 1 s, 500 s, 1000 s and 2000 s. Fig. 6a and 6b show SoC and temperature vs. time of all the 25 cells. The temperature of each cell varies according to its usage. At the beginning of the test all cells are at 25°C, then the spread of temperatures increases as the temperature of the most inserted cells increases. The cells tend to reach the same higher temperature towards the end of the test. Fig. 8 shows how the cells to be inserted/bypassed are chosen by the algorithm at different time steps: a value of “1” corresponds to cell insertion, while a value of “0” corresponds to cell bypass.

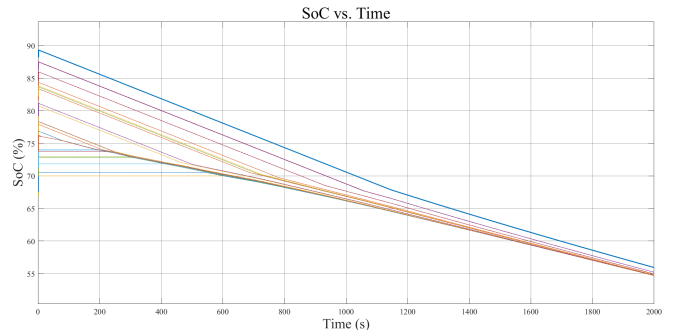


(a) SoC of the 25 cells.

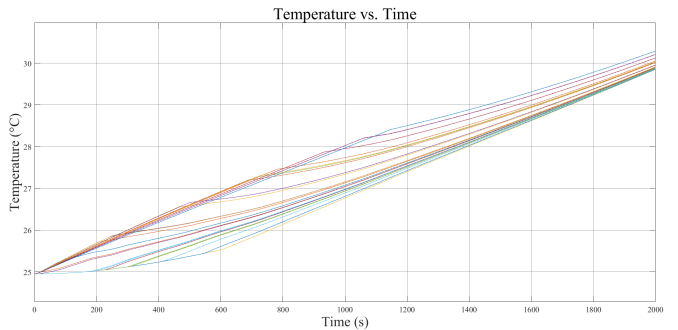


(b) Temperature of the 25 cells

Fig. 6. SoC balancing sorting algorithm simulation results.



(a) SoC of the 25 cells.



(b) Temperature of the 25 cells

Fig. 7. MKNA algorithm simulation results.

Multidimensional K-nearest control algorithm

The previous algorithm based only on SoC has the side effect of an uncontrolled spread of the cell temperatures. The idea here is to select cells both on their SoC and temperature

in order to reduce the maximum temperature spread. The K-nearest algorithm selects the n cells that are nearest to the dynamic virtual point SoC_{\max} , T_{\min} . The algorithm action could be best seen on an animated scatter plot (SoC vs. Temperature). Used cells are pushed down-right (SoC decrease and temperature increase) until all cells tend to converge on an arc centered on current SoC_{\max} , T_{\min} . This means that the maximum temperature spread is reduced at the expense of a small SoC spread (the ray of the arc). The weighting factors k_S , k_T can be used to give more weight to SoC or to temperature. The limiting cases are: (i) $k_T = 0$, the arc changes to an horizontal line and the cells are pushed to have the same SoC (as the previous SoC Sorting); (ii) $k_S = 0$, the arc changes to a vertical line and the cells are pushed to have the same temperature. Fig. 9 shows four screenshots of the scatter plot at 1 s, 500 s, 1000 s and 2000 s. Fig. 7a and 7b show SoC and temperature vs. time of all the 25 cells. Fig. 9 shows how the cells to be inserted/bypassed are chosen by the algorithm at different time steps: a value of “1” corresponds to cell insertion, while a value of “0” corresponds to cell bypass. Finally, Fig. 10 shows a comparison between

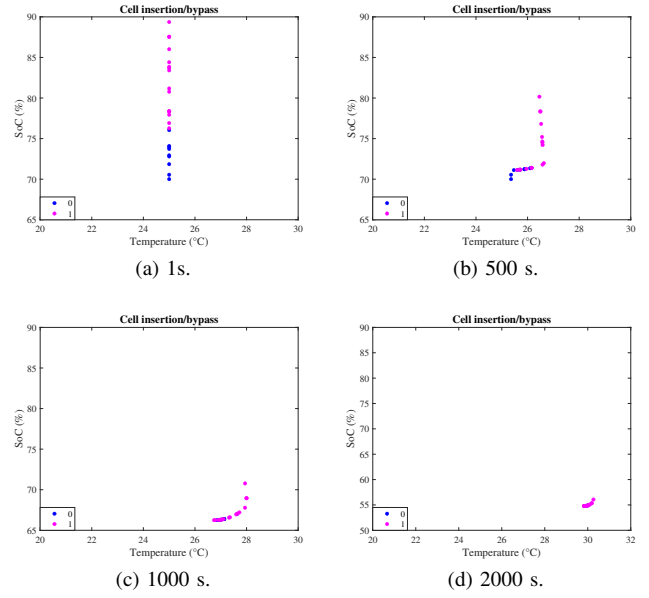


Fig. 9. Cell insertion/bypass choice - MKNA algorithm.

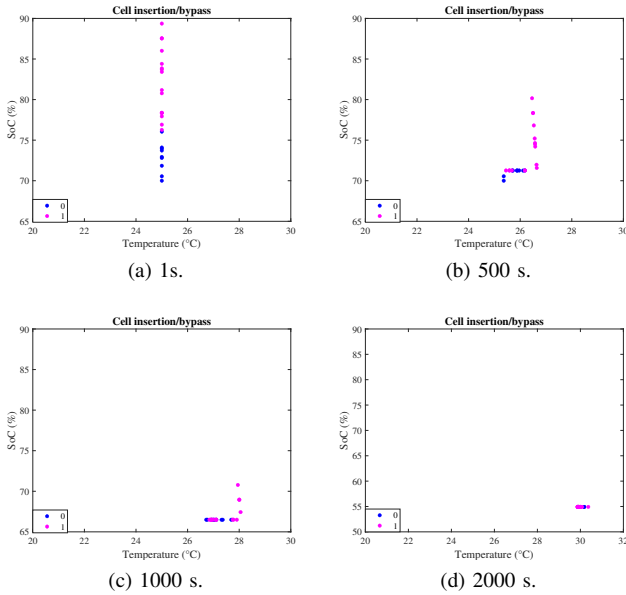


Fig. 8. Cell insertion/bypass choice - SoC balancing sorting algorithm.

the temperature spread of the two control algorithms vs. time. The SoC Sorting algorithm produces peak temperatures higher than the multidimensional algorithm.

VI. CONCLUSIONS AND FUTURE WORK

The proposed SB architecture, where at a given time only $n \leq N$ cells are series-connected, has many useful features like modularity, reconfigurability, cell fault tolerance, greater useful capacity. In order to achieve optimal balancing in operation for both temperature and SoC, a multidimensional K-nearest control algorithm (MKNA) was developed and

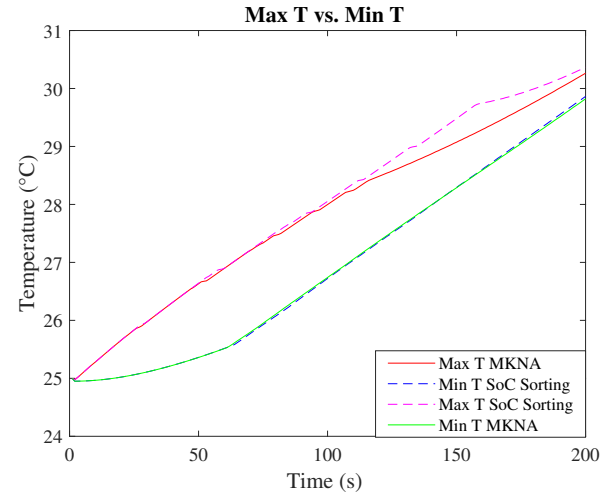


Fig. 10. Comparison of maximum and minimum temperature.

simulated with the aim to limit the spread of cell SoCs and temperatures. The MKNA algorithm chooses the cells to be inserted/bypasses based on their “distance” from a virtual point that is dynamically computed at sampling time. The algorithm uses weighting factors to give more importance to SoC or temperature. It turns out that the MKNA algorithm reduces itself to the SoC Sorting algorithm when the temperature is weighted out. Simulation results show that the maximum temperature spread is reduced in comparison with a simple SoC sort. This operation has the potential to extend the lifetime of the battery pack, which will be evaluated in future work. The structure of the MKNA algorithm allows to easily add other dimensions to the control strategy, such as the internal resistance or the SoH of the cells.

REFERENCES

- [1] G. Du, W. Cao, S. Hu, Z. Lin, and T. Yuan, "Design and assessment of an electric vehicle powertrain model based on real-world driving and charging cycles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1178–1187, 2019.
- [2] M. Ricco, J. Meng, T. Gherman, G. Grandi, and R. Teodorescu, "Smart battery pack for electric vehicles based on active balancing with wireless communication feedback," *Energies*, vol. 12, no. 20, 2019.
- [3] F. Altaf, "Thermal and state-of-charge balancing of batteries using multilevel converters," Ph.D. dissertation, Mar. 2014.
- [4] M. Daowd, N. Omar, P. Van Den Bossche, and J. Van Mierlo, "Passive and active battery balancing comparison based on matlab simulation," in *IEEE Veh. Power and Propuls. Conference*, 2011, pp. 1–7.
- [5] 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.
- [6] Y. Li and Y. Han, "A module-integrated distributed battery energy storage and management system," *IEEE Trans. Power Electron.*, vol. 31, no. 12, pp. 8260–8270, 2016.
- [7] M. Einhorn, W. Roessler, and J. Fleig, "Improved performance of serially connected li-ion batteries with active cell balancing in electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2448–2457, 2011.
- [8] S. Narayanaswamy, M. Kauer, S. Steinhorst, M. Lukasiewicz, and S. Chakraborty, "Modular active charge balancing for scalable battery packs," *IEEE Trans. Very Large Scale Int. Syst.*, vol. 25, no. 3, pp. 974–987, 2017.
- [9] M. Lukasiewicz, M. Kauer, and S. Steinhorst, "Synthesis of active cell balancing architectures for battery packs," *IEEE Trans. Comput. Des. Int. Circuits Syst.*, vol. 35, no. 11, pp. 1876–1889, 2016.
- [10] A. Manenti, A. Abba, A. Merati, S. M. Savaresi, and A. Geraci, "A new bms architecture based on cell redundancy," *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 4314–4322, 2011.
- [11] X. Huang, A. B. Acharya, J. Meng, X. Sui, D.-I. Stroe, and R. Teodorescu, "Wireless smart battery management system for electric vehicles," in *IEEE Energy Conv. Congr. Expo. (ECCE)*, 2020, pp. 5620–5625.
- [12] B. Majmunovic, R. Sarda, R. Teodorescu, C. Lascu, and M. Ricco, "Highly efficient smart battery pack for ev drivetrains," in *IEEE Veh. Power Propuls. Conf. (VPPC)*, 2017, pp. 1–5.
- [13] X. Huang, X. Sui, D.-I. Stroe, and R. Teodorescu, "A review of management architectures and balancing strategies in smart batteries," in *IECON 2019 - 45th Ann. Conf. IEEE Industr. Electron. Society*, vol. 1, 2019, pp. 5909–5914.
- [14] X. Sui, S. He, S. B. Vilsen, J. Meng, R. Teodorescu, and D. I. Stroe, "A review of non-probabilistic machine learning-based state of health estimation techniques for lithium-ion battery," *Appl. Energy*, vol. 300, p. 117346, Oct. 2021.
- [15] M. H. Lipu, M. Hannan, A. Hussain, M. Hoque, P. J. Ker, M. Saad, and A. Ayob, "A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations," *J. Clean. Prod.*, vol. 205, pp. 115–133, 2018.
- [16] B. Wu, W. D. Widanage, S. Yang, and X. Liu, "Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems," *Energy and AI*, vol. 1, p. 100016, 2020.