Battery Pack Cell Balancing using Topology Switching and Machine Learning

Yuqin Weng
Electrical and Computer Engineering Dept.
Marquette University
Milwaukee, WI, USA
yuqin.weng@marquette.edu

Cristinel Ababei
Electrical and Computer Engineering Dept.
Marquette Universty
Milwaukee, WI, USA
cristinel.ababei@marquette.edu

***Abstract*—Battery packs are widely used in electric vehicles. Imbalance in the state of charge (SoC) of battery cells results in suboptimal operation of these packs due to their early stop of charging and discharging processes. This in turns leads to an increase in the number of charging cycles, which degrades the performance of the battery pack and results in shorter lifetime. To address cell imbalance, battery management systems (BMS) must employ cell balancing or equalization methods. In this paper, we propose a novel battery pack balancing technique, which uses a reconfigurable switching network to periodically change the pack topology in order to achieve cell balancing. The periodic reconfiguration is based on a machine learning (ML) algorithm that points the next topology that will get the pack into an operation mode in the next control period where cell imbalance is reduced. The proposed technique is verified with a custom simulation framework that integrates state-of-the-art extended Kalman filtering (EKF) models for state estimation for increased accuracy. Simulation results demonstrate that the proposed technique can successfully reduce battery pack cell imbalance.**

***Index Terms*—battery pack, cell imbalance, state of charge, battery pack reconfiguration**

I. Introduction

We are witnessing an increasing adoption of electric vehicles (EVs). For example, the state of California announced that it will ban the sale of new gas-powered vehicles starting in 2035 [1]. EVs are powered by battery packs formed by many battery cells that are connected in various configurations. The overall state of charge (SoC) of a battery pack is essential in battery management systems (BMS), whose role is to monitor the battery pack and to implement various optimizations aimed at improving performance, and increasing the traveled distance of EVs. Lithium-ion (Li-ion) batteries are largely used in today's EVs because Li-ion batteries have many advantages, including being rechargeable and offering fast charging times. However, Li-ion batteries pose challenges as well, and one of the challenges is cell imbalance in terms of SoC. That is because charging or discharging of individual cells is non-uniform and this results in variation of the actual SoC values. This happens due to unavoidable differences in cell characteristics such as capacity and internal resistance that depend on fabrication processes [2]. When cells become imbalanced for example during charging, the first cell that reaches full capacity stops the charging process early with potentially many other cells still only partially charged. Thus, the battery pack is not fully charged to capacity, the total driving distance is reduced, and the number of charges will increase. This in turn degrades performance and shortens the lifetime of the battery pack. Therefore, cell balancing techniques are desirable in order to improve the battery performance and to prolong lifetime of battery packs.

There have been several previous studies that looked at various aspects of battery cell balancing. Some excellent reviews of such techniques are presented in [3], [4]. Commonly, there are two major types of balancing techniques: passive and active balancing techniques [2]. The charge in higher energy cells is redistributed during cycles of charging and discharging when using active cell balancing. In passive cell balancing, the idea is to simply dissipate charge [5]. For example, the study in [2] proposed an equalization method that uses a grouping average technique based on the average level of battery terminal voltage [2]. The study in [6] proposed a method of using switch arrays for reconfigurable battery cells and designed an active SoC balancing. Similarly, the work in [7] proposed a battery pack balancing technique that also uses a reconfigurable battery pack structure with no additional SoC equalization circuit.

In this paper, we propose a novel battery cell balancing technique that uses a reconfigurable network of switches that we adopt from the work in [8]. To control the reconfiguration of the network of switches, the proposed technique uses accurate extended Kalman filtering (EKF) based SoC estimation techniques and machine learning (ML) based prediction. The objective of the proposed technique or algorithm is to periodically switch between different configurations or topologies of the battery pack, so that the battery pack operates in a mode where the variation of cells SoC is reduced, thereby keeping the charging or discharging as uniform as possible. In other words, the charging and discharging rates of all cells are kept as equal as possible and cell balancing or equalization is achieved. The decision as of what best topology to switch to - from a reduced set of predetermined topologies - is done with the help of an ML model. For training the ML model, we use a custom battery pack simulation framework to generate datasets that collect SoC values, max, mean and std of the SoC values. The ML model is trained to pick up the next best configuration from among the configurations that form a

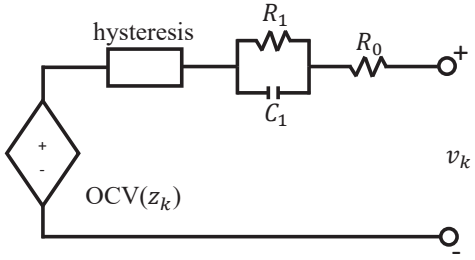


Figure 1. Equivalent electric circuit for a battery cell used by the ESC model [10].

Pareto front in the solution space.

II. Battery Cell Modeling, SoC Estimation and Reconfigurable Battery Pack Architecture

In this section, we present details on the battery cell models and SoC estimation techniques that are used in the custom simulation framework. We also describe the reconfigurable battery pack architecture that is assumed in this work.

A. Battery Cell Model

The popular study in [9] presented several battery cell models. The most sophisticated one is the enhanced self-correcting (ESC) model, and is the one that we implemented in the custom simulation framework, which we developed for this work. Compared to other models, in addition to capturing SoC, this model also captures the hysteresis voltage and diffusion processes into the state vector of the state space model. Next, we present only a brief description of this model, whose equivalent circuit model is shown in Fig. 1.

The state space representation for the ESC model can be described by the following equations [10]:

$$\begin{bmatrix} z_{k+1} \\ i_{R,k+1} \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & A_{RC} & 0 \\ 0 & 0 & A_{H,k} \end{bmatrix} \begin{bmatrix} z_k \\ i_{R,k} \\ h_k \end{bmatrix} + \begin{bmatrix} (\frac{-\eta_k \Delta t}{Q}) & 0 \\ B_{RC} & 0 \\ 0 & (A_{H,k}-1) \end{bmatrix} \begin{bmatrix} i_k \\ sgn(i_k) \end{bmatrix} \quad (1)$$

$$y_k = OCV(z_k) + Mh_k - R_1 i_{R,k} - R_0 i_k \quad (2)$$

Where, eq. 1 is called the state equation and eq. 2 is called the output equation. In these equations, y_k is the terminal voltage, as the output of the system, i_k is the cell instantaneous current which is treated as the input of the system, and R_0 is the internal resistance of the battery cell. The state vector x_k includes three states as follows:

$$x_k = \begin{bmatrix} z_k \\ i_{R,k} \\ h_k \end{bmatrix} \quad (3)$$

Where z_k represents the SoC state, $i_{R,k}$ is the diffusion process current state, and h_k is the hysteresis state. Matrix elements A_{RC}, B_{RC} and $A_{H,k}$ in eq. 1 are defined by the following expressions:

$$A_{RC} = exp(\frac{-\Delta t}{R_1 C_1}) \quad (4)$$

$$B_{RC} = 1 - exp(\frac{-\Delta t}{R_1 C_1}) \quad (5)$$

$$A_{H,k} = exp(-|\frac{\eta_k i_k \gamma \Delta t}{Q}|) \quad (6)$$

The SoC z_k is defined as 100% when the cell is fully charged and 0% when fully discharged. η_k is called the Coulombic efficiency; usually $\eta_k = 1$ when the cell is discharging and $\eta_k \leqslant 1$ when the cell is charging. Δt is a small sampling interval used to convert the continuous-time system to discrete-time. Q is the total capacity of the cell; it is the total amount of charge when charging the cell from SoC 0% to 100%. The open circuit voltage $OCV(z_k)$ is an ideal voltage source and is a function of SoC. In practice, $OCV(z_k)$ is found by cell testing and the testing data can be integrated as a look-up table. $R_1 i_{R,k}$ is used for representing the diffusion voltage [11]. While multiple RC pairs can be included in the ESC model, in this paper, we use only one $R_1 C_1$ pair, similarly to [10]. The third state is the hysteresis state h_k which also has the connection to the input i_k of the state equation. In eq. 6, γ is a small constant which can be used for modeling the voltage decay.

B. SoC Estimation

Because the SoC of a battery cell is not a quantity that can be directly measured, estimation methods are typically employed. One of the most common used methods to estimate SoC of a battery cell is Kalman filtering. Kalman filter theory is a classic technique developed in the 1960's [12]. Kalman filters have been used in many application areas including control systems, signal processing and image processing due to its accuracy and robustness [13]. As a variation of Kalman filter theory, the extended Kalman filtering (EKF) is used for the estimation of non-linear state-space models. The idea of EKF is that a linearization can be used at around the current estimate by using partial derivatives of the state and output equations to compute estimates for each time step in discrete time [9]. Thus, EKF is a perfect choice for estimating battery cell SoC because the battery cell state-space model is a non-linear representation at time k. An illustration of how EKF is used for SoC estimation is shown in Fig. 2. The simulation tool, which we developed to be able to conduct the simulation experiments from this paper, implements the EKF approach for estimating SoC for all cells in a battery pack, which are then used to estimate the SoC of the entire pack for the best possible accuracy of results.

C. Reconfigurable Battery Pack Architecture

In this paper, we assume a custom reconfigurable battery pack, whose architecture is inspired from the work in [8], which used a similar reconfigurable network of switches in photovoltaic (PV) arrays. The main idea of this reconfigurable battery pack - illustrated in Fig. 3.a for a pack with four cells - is to use a network of programmable switches, which can be controlled to implement different pack topologies. Each cell in the pack has assigned a group of three switches ($S_{PT,i}$,

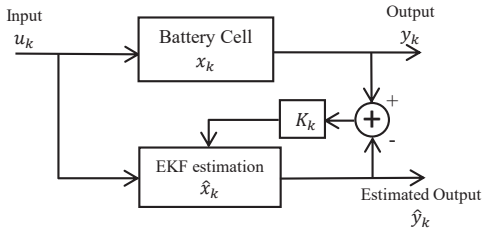


Figure 2. Illustration of how the ESC cell model would run in parallel with the true hardware cell. The model is solved at each time step k by the EKF algorithm that uses the Kalman gain K_k during the state correction step or measurement update.

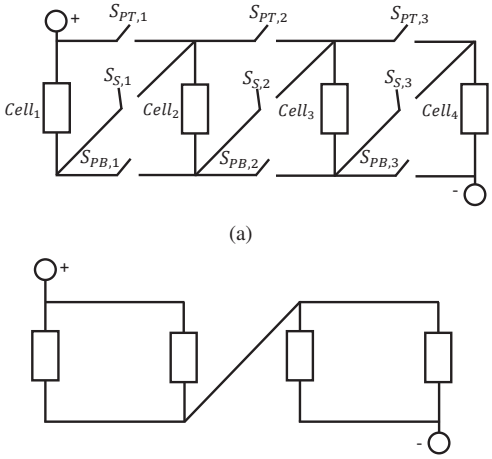


Figure 3. (a) Structure of battery pack reconfigurable architecture with 4 cells. (b) An example of a 2×2 battery pack topology.

$S_{PB,i}$ and $S_{S,i}$), which help connect the cell in various series or parallel configurations with other cells in order to form desired numbers of rows and columns in the pack. Note that for a given group of such switches, the top and the bottom level switches should always be in the same state, while the middle level switch must be in an opposite state [8].

By controlling the on/off state of these switches, different topologies can be realized. For example, a 2×2 topology can be realized as shown in Fig. 3.b. While this battery pack architecture has the advantage of being reconfigurable, it also has the downside of being more complex due to the network of switches, which adds to the overall cost of the battery.

III. Proposed Cell Balancing Algorithm

A. Equalization via Switching Between Pack Topologies

The objective of the proposed algorithm is to achieve cells SoC equalization or balancing only by using the reconfigurable network of switches discussed earlier. We observed that operation of the battery pack in different topologies may result in different SoC variation profiles or traces, under given initial SoC conditions in each run. Therefore, the main idea of the proposed balancing algorithm is to periodically reconfigure the pack architecture and switch to topologies that are more likely to lead - in the next control period - to SoC variation profiles that reduce the imbalance, i.e., reduce the span of all SoC values at any given time. Extensive simulations

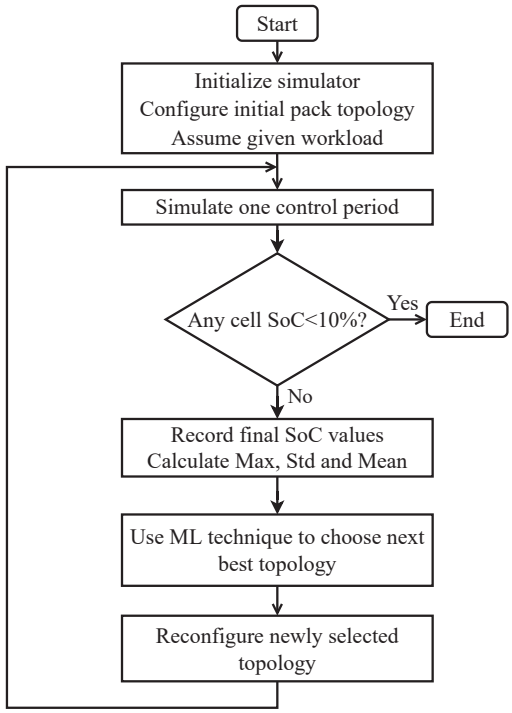


Figure 4. Flowchart of proposed algorithm for cell balancing via periodic (5 minutes) topology reconfiguration.

revealed that certain pack topologies may lead to an increase in imbalance while others do not. Additionally, we found that there is no one single topology that is always the best to use in all scenarios. Therefore, we propose to switch among different pack topologies, from a reduced set of predetermined topologies that were found to help reduce cells imbalance and which are acceptable in terms of output voltage levels.

Fig. 4 shows the flowchart of the proposed algorithm for pack topology or configuration switching. This algorithm is implemented and verified inside our custom simulation framework. The battery pack is simulated for example under a discharging workload profile starting from a predetermined or randomly selected initial state for all cells. Then, periodically, at times separated by *control periods*, the pack is reconfigured. To what other pack topology to switch to for the next control period is decided with an ML based technique, which may decide to actually keep the current pack topology for one more control period. The ML models will be described later on. However, these models require datasets for model training, testing and validation. It is the custom simulation framework that enables us to generate these datasets. We instrumented the simulator to simulate a given pack under many different initial SoC conditions for the selected number of predetermined pack topologies and to collect and store desired datasets. Simulations and dataset generation and collection is done in discharging scenarios until any of the cells reaches an SoC of 10%. The goal of the dataset generation is to generate rich datasets that will be used to train the ML models such that these models will be able to recognize/identify what is the best pack topology to reconfigure to, given the current status of all

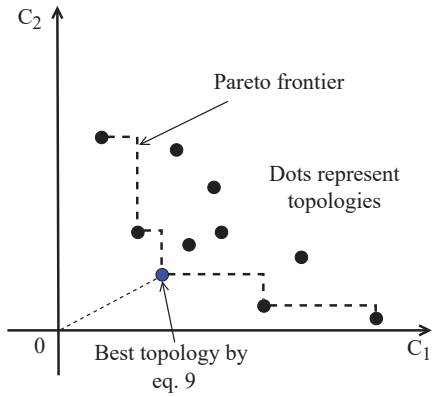


Figure 5. Illustration of the Pareto optimality. C_1 and C_2 are calculated with eq. 7 and eq. 8.

cells SoCs at the end of the previous control period. Details about the criteria used to choose the best configuration for the next control interval are provided in the next subsection.

B. Choosing Topologies that are Pareto Optimal to Generate Datasets for ML Training

In the proposed topology switching approach, we need to periodically select possibly a new battery pack topology, which for the duration of the next control period will be more likely to improve two objectives: 1) reduce the inverse of the summation of all SoC values of all cells and 2) reduce the SoC values span or spread, measured as the difference between the maximum and minimum SoC values. This problem formulation is a multi-objective optimization. Therefore, we identify the next best topology to switch to from among those topologies that represent Pareto optimal solutions [14] in the solution space defined by the two objective functions. For example, Fig. 5 illustrates the solution space defined by the two objectives; each solution in this two dimensional space is a pack topology, and only some of these form what is called Pareto frontier. Pareto optimality of such solutions guarantee that the solutions are such that one cannot improve one objective without degrading the other [15]. The equations used for the two objective *cost* functions C_1 and C_2 are:

$$C_1 = SoC_{diff}^{max} = \max_{i\in[n]} SoC_i - \min_{i\in[n]} SoC_i \tag{7}$$

$$C_2 = 1/\sum_{i=1}^{n} SoC_i \tag{8}$$

Where we define $[n] = \{1, 2, .., n\}$ with n being the number of cells in the battery pack.

C. Prediction of Next Best Topology with ML

The challenge is to find a way to identify a Pareto optimal pack topology at the beginning of each new control period. We address this challenge by employing an ML model to do the prediction as of which topology to choose next. However, the quality of the ML based prediction depends on how well the model is trained to capture this type of prediction problem. This in turn depends on the quality of dataset - which we generate with our custom pack simulation framework. Essentially, we instrument the simulator to conduct exhaustive simulations of the type *what-if* scenarios - during which dataset is generated for all possible next topologies. Additionally, for each control period that is simulated and added to the dataset, we identify the Pareto optimal topology that is labeled as the next best topology. Looking again at Fig. 5, during the simulations for dataset generation, at the end of a simulated control period, there are N topologies evaluated via the cost functions C_1 and C_2; these topologies represent the dots in the figure. The best next topology will be identified as the one that results in the following minimization:

$$\min_{k\in[N]} \sqrt{(C_1^k)^2 + (C_2^k)^2} \tag{9}$$

Where we define $[N] = \{1, 2, .., N\}$ with N being the number of topologies that are predetermined to use only. C_1^k and C_2^k are the values of C_1 and C_2 for topology k as calculated with eq. 8 and eq. 7. This topology or configuration is recorded as the best one to switch the pack to and then continue the simulation for the next control period during the dataset generation process. Multiple simulations are run to collect data for what the best next topology is for different states of the battery cells. The process of conducting simulations for training datset generation is described with the pseudocode from Fig 6.

Algorithm: Dataset Generation based on Pareto Optimality

```
In: Reconfigurable pack architecture, workload
Out: Dataset for training and testing ML model
Assume discharging workload
for range_u :← [0.3 : 0.15 : 0.9] do
    for i ← 1 to M do // M: 100 or 200 simulations
        for j ← 1 to N do // N: number of topologies
            Configure pack to initial topology j
            Randomly select initial SoC from range_u
            for k ← 1 to N do
                Configure pack to next topology k
                Run simulation for next control period
                Record costs, eq. 7, eq. 8
                Find best next topology, eq. 9
                Record new datapoint: (Input features | Label)
                Input features: end SoC values, j, max, mean, std.
                Label: next best topology index
            end for
        end for
    end for
end for
```

Figure 6. Pseudocode of the simulation process used to generate datasets.

D. Machine Learning Models

We investigate two ML models which are trained to make predictions for what the best next pack topology is at any point during the pack discharging under any workload. These ML models are: random forest (RF) and k-nearest neighborhood (KNN). The RF model requires supervised learning and has been used in applications from face recognition and text

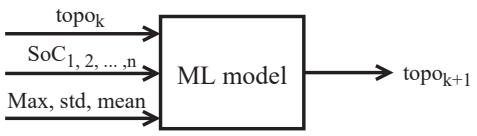


Figure 7. Selected input features and output labels in the ML model.

classification [16]. The KNN model is another popular model that also requires supervised learning. The simplicity and effectiveness makes KNN algorithm one of the most popular ML algorithms [17]. Its challenge is the selection of k, which impacts the quality of results. For both models, the input features used during training include (see Fig. 7): topology id from the previous control period, SoC values, max, mean and std of the SoC values at the end of the previous control period. We selected these input features based on a rigorous correlation analysis of a larger number of potential inputs; we retained those that are the least correlated among themselves in order to keep the model size as small as possible. The prediction accuracy of both models will be discussed in the next section.

IV. Simulation Results

A. Comparison of Machine Learning Models

To model practical variations in cell characteristics that result in differences in cells SoC values, selected cell parameters are assigned different initial values by sampling values from predetermined distributions. Based on the discussion from [11], the parameters selected are: initial values of z, Q, R_1, C_1, and R_0. The imbalance occurs when randomly initializing those parameters. The actual numerical values of these parameters are taken from [10], [11].

We investigate a battery pack with 8 cells, which could be arranged in 128 different topologies. However, only a selected subset of these topologies will be used to switch between in the proposed approach because some topologies are not necessarily practical; for example, we eliminate topologies where all 8 cells are connected in series or in parallel. Eliminating such topologies is done on practical considerations that involve desired voltage levels at the pack's output terminals. On the other hand, topologies that have 4, 5, 6 rows are kept in the subset; for example, a four row topology may be $[2, 1, 3, 2]$. In total, 91 configurations are selected initially. However, out of these 91 configurations only 10 are finally kept for use in the proposed switching algorithm. These ten final topologies are selected as those that are found to be consistent among those on the Pareto frontier (see Fig. 5) during the dataset generation process. During the process of generating datasets and during simulations experiments to evaluate the proposed approach, the control period is assumed to be 5 minutes.

To investigate the prediction accuracy of the two ML models, we generate four different datasets. The datasets are generated for: 1) constant workload and 2) combination of multiple urban dynamo-meter drive schedule ($UDDS$) drive cycles. The accuracy of the two ML models is reported in Table I. Based on these results, we concluded that the RF model is better than the KNN model and that working with larger datasets helps improve the prediction accuracy. Therefore, in the remaining simulations, we only use the RF model for predictions.

Table I
Accuracy of ML Models.

ML Models	Dataset1	Dataset2	Dataset3	Dataset4
KNN	49.8%	52.9%	49.9%	52.5%
RF	57.1%	58.4%	54.7%	58.8%

B. Results Obtained with Proposed Balancing Algorithm

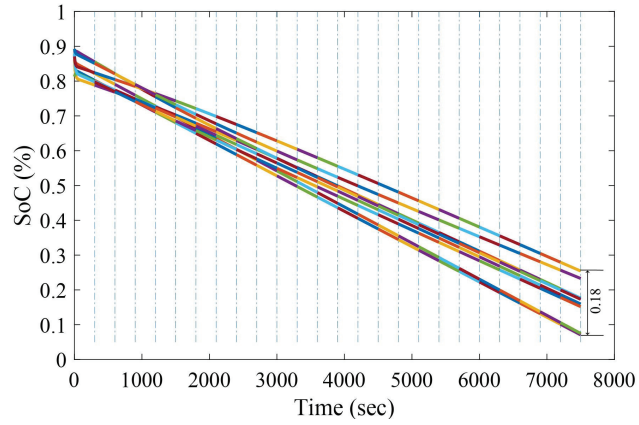
The proposed cell balancing algorithm is investigated in this subsection using the custom simulator. Simulations are conducted for both types of workload: 1) constant workload and 2) combination of multiple urban dynamo-meter drive schedule ($UDDS$) drive cycles.

The results for constant workload are presented in Fig. 8. In this figures, x axis represents simulation time while y axis represents the cells SoC. As a reference case we conduct a simulation where the pack topology with four rows and 2 columns $[2, 2, 2, 2]$ is used all the time; this topology is considered the default topology. The result of this simulation is shown in Fig. 8.a and it is run until any of the cells SoC reaches 10%, when the simulation stops - the stop time is an indicator of how long the pack can be used for driving - in this case 7,213 seconds. The result when the proposed balancing algorithm is employed is shown in Fig. 8.b, where the control periods (5 minutes each) are indicated with vertical dashed lines. The simulation stop time in this case is 8,000 seconds, which is with 10.9% longer compared to the reference case. This demonstrates that switching between different pack topologies can be an effective technique to achieve cells balancing, thereby prolonging the pack runtime.

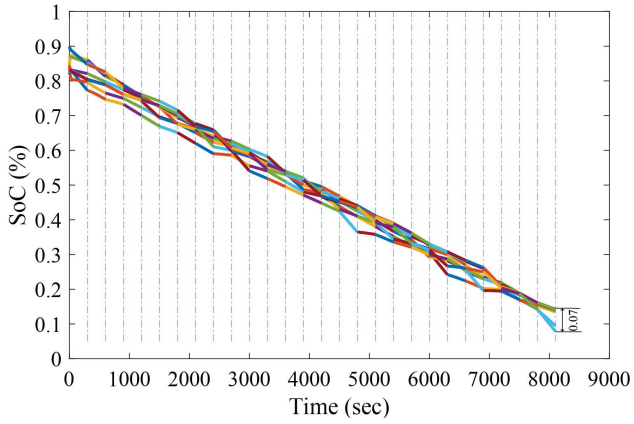
The results when the workload for the battery pack is a combination of multiple urban dynamo-meter drive schedule ($UDDS$) drive cycles are shown in Fig. 9. Similarly to the first type of workload, when the proposed cells balancing algorithm is used, the pack runtime increases from 31,967 seconds to 36,635 seconds - which represents an improvement of 14.6%.

V. Conclusion

We presented a novel cell balancing algorithm for battery packs. The proposed algorithm uses a reconfigurable network of switches that can be controlled to change the battery pack topology periodically. Changing the topology of the pack is done periodically during the normal operation of the pack with the help of a random forest machine learning based prediction technique. The proposed approach is verified with a custom simulation framework that employs state-of-the-art SoC estimation models. Simulation results conducted for an 8 cells battery pack for two different types of workloads demonstrated that the proposed cells balancing approach can increase the pack runtime with up to 14.6% compared to the reference case.

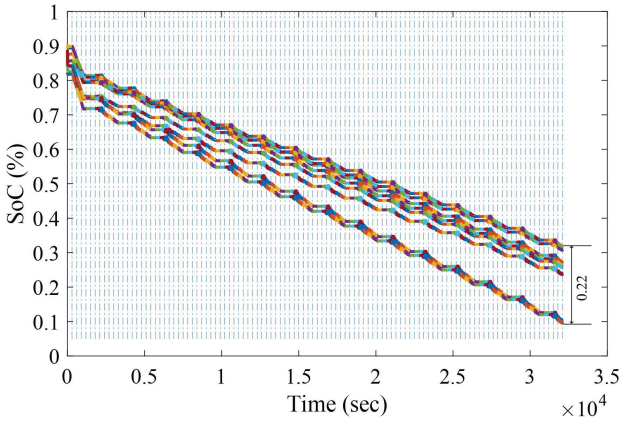


(a)

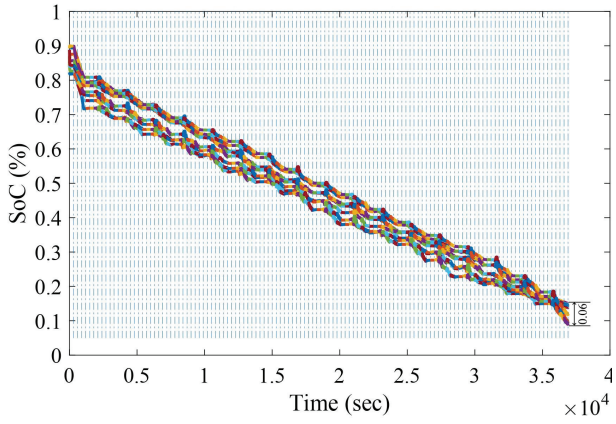


(b)

Figure 8. (a) Variation of cells SoC when reference topology $[2, 2, 2, 2]$ is used with constant workload. (b) Variation of cells SoC when the proposed cells balancing algorithm is used.



(a)



(b)

Figure 9. (a) Variation of cells SoC when reference topology $[2, 2, 2, 2]$ is used with $UDDS$ workload. (b) Variation of cells SoC when the proposed cells balancing algorithm is used.

REFERENCES

[1] Office of Governor, "Office of Governor, CA, Governor Newsom Announces California Will Phase Out Gasoline-Powered Cars & Drastically Reduce Demand for Fossil Fuel in California's Fight Against Climate Change," [Online]. Available: *https://www.gov.ca.gov/2020/09/23/governor-newsom-announces-california-will-phase-out-gasoline-powered-cars-drastically-reduce-demand-for-fossil-fuel-in-californias-fight-against-climate-change*, 2020.

[2] F. Yan, G. Hui, and S. Lu, "Research on equalization method of battery management system," *Chinese Automation Congress (CAC)*, 2019.

[3] M. Daowd, M. Antoine, N. Omar, P. Lataire, P. Van Den Bossche, and J. Van Mierlo, "Battery management system—balancing modularization based on a single switched capacitor and bi-directional DC/DC converter with the auxiliary battery," *Energies*, vol. 7, no. 5, pp. 2897-2937, Sep. 2014.

[4] M.M. Hoque, M.A. Hannan, A. Mohamed, and A. Ayob, "Battery charge equalization controller in electric vehicle applications: A review," *Renewable and Sustainable Energy Reviews*, vol. 75, pp. 1363-1385, 2017.

[5] K. Scott and S. Nork, "Active battery cell balancing," [Online]. Available: *https://www.analog.com/media/en/technical-documentation/tech-articles/Active-Battery-Cell-Balancing.pdf*, 2022.

[6] K.-M. Lee, D.-H. Kim, J.-H. Shin, T.-D. Goh, J.-H. Park, and Y.-H. Ryu, "Reconfigurable battery system with active balancing circuit," *Annual Conf. of the IEEE Industrial Electronics Society*, 2019.

[7] F. Chen, H. Wang, W. Qiao, and L. Qu, "A grid-tied reconfigurable battery storage system," *IEEE Applied Power Electronics Conf. and Exposition (APEC)*, 2018.

[8] Y. Wang, X. Lin, Y. Kim, N. Chang, and M. Pedram, "Architecture and control algorithms for combating partial shading in photovoltaic systems," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 6, pp. 917-930, June 2014.

[9] G. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs. Part 1. Background," *Journal of Power Sources*, vol. 134, no. 2, pp. 252-261, 2004.

[10] ——, "ECE5720: Battery management and control, University of Colorado Colorado Springs," [Online]. Available: *http://mocha-java.uccs.edu/ECE5720/index.html*, 2015.

[11] ——, "ECE4710/5710: Modeling, simulation, and identification of battery dynamics, University of Colorado Colorado Springs," [Online]. Available: *http://mocha-java.uccs.edu/ECE5710/index.html*, 2018.

[12] W. Greg and B. Gary, "An introduction to the Kalman filter," [Online]. Available: *chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf*, 2006.

[13] Y. Weng, "Detection and characterization of actuator attacks using Kalman filter estimation," *M.S. thesis, Marquette University*, 2019.

[14] K. Deb, "Multi-objective optimization using evolutionary algorithms: an introduction," KanGAL Report Number 2011003, 2011.

[15] Y. Akishita, Y. Ohsita, and M. Murata, "Network power saving based on Pareto optimal control with evolutionary approach," *Int. Conf. on Computing, Networking and Communications (ICNC)*, 2017.

[16] H. Lan and Y. Pan, "A crowdsourcing quality prediction model based on random forests," *Int. Conf. on Information, Intelligence, Systems and Applications (IISA)*, 2021.

[17] M. Papanikolaou, G. Evangelidis, and S. Ougiaroglou, "Dynamic k determination in k-NN classifier: A literature review," *IEEE/ACIS Int. Conf. on Computer and Information Science (ICIS)*, 2019.