STATE ESTIMATION AND THERMAL FAULT DETECTION FOR LITHIUM-ION BATTERY PACKS: A DEEP NEURAL NETWORK APPROACH

STATE ESTIMATION AND THERMAL FAULT DETECTION FOR LITHIUM-ION BATTERY PACKS: A DEEP NEURAL NETWORK APPROACH

BY

MINA NAGUIB, M.Sc.

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2022) McMaster University

Electrical & Computer Engineering Hamilton, Ontario, Canada

TITLE:	STATE ESTIMATION AND THERMAL FAULT DETECTION FOR LITHIUM-ION BATTERY PACKS: A DEEP NEURAL NETWORK APPROACH
AUTHOR:	Mina Naguib M.Sc. (Electrical Engineering) Ain Shams University, Cairo, Egypt
SUPERVISOR:	Dr. Ali Emadi Ph.D. (Texas A&M University) IEEE Fellow Canada Excellence Research Chair Laureate
NUMBER OF PAGES:	xxviii, 232

To my Family,

and my beloved wife Christine

Abstract

Recently, lithium-ion batteries (LIBs) have achieved wide acceptance for various energy storage applications, such as electric vehicles (EVs) and smart grids. As a vital component in EVs, the performance of lithium-ion batteries in the last few decades has made significant progress. The development of a robust battery management system (BMS) has become a necessity to ensure the reliability and safety of battery packs. In addition, state of charge (SOC) estimation and thermal models with high-fidelity are essential to ensure efficient BMS performance.

The SOC of a LIB is an essential factor that should be reported to the vehicle's electronic control unit and the driver. Inaccurate reported SOC impacts the reliability and safety of the lithium-ion battery packs (LIBP) and the vehicle. Different algorithms are used to estimate the SOC of a LIBP, including measurement-based, adaptive filters and observers, and data-driven; however, there is a gap in feasibility studies of running these algorithms for multi-cell LIBP on BMS microprocessors. On the other hand, temperature sensors are utilized to monitor the temperature of the cells in LIBPs. Using a temperature sensor for every cell is often impractical due to cost and wiring complexity. Robust temperature estimation models can replace physical sensors and help the fault detection algorithms by providing a redundant monitoring system.

In this thesis, an accurate SOC estimation and thermal modeling for lithium-ion batteries (LIBs) are presented using deep neural networks (DNNs). Firstly, two DNN-based SOC estimation algorithms, including a feedforward neural network (FNN) enhanced with external filters and a recurrent neural network with a long short-term memory layer (LSTM), are developed and benchmarked versus an extended Kalman filter (EKF) and EKF with recursive least squares filter (EKF-RLS) SOC estimation algorithms. The execution time of EKF, EKF-RLS, FNN, and LSTM SOC estimation algorithms with similar accuracy was found to be 0.24 ms, 0.25 ms, 0.14 ms, and 0.71 ms, respectively. The DNN SOC estimation algorithms were also demonstrated to have lower RAM use than the EKFs, with less than 1 kB RAM required to run one estimator.

The proposed FNN and LSTM models are also used to predict the surface temperature of different lithium-ion cells. These DNN models are shown to be capable of estimating temperature with less than 2 ºC root mean square error for challenging low ambient temperature drive cycles and just 0.3 ºC for 4C rate fast charging conditions. In addition, a DNN model which is trained to estimate the temperature of a new battery cell, is found to still have a very low error of just 0.8 ºC when tested on an aged cell.

Finally, an integrated physics, and neural network-based battery pack thermal model (LP+FNN) is developed and used to detect and identify different thermal faults of a LIBP. The proposed fault detection and identification method is validated using various thermal faults, including fan system failure, airflow lower and higher than setpoint, airflow blockage of submodule and temperature sensor reading faults. The proposed method is able to detect different cooling system faults within 10 to 35 minutes after fault occurrence. In addition, the proposed method demonstrated being capable of detecting temperature sensor reading offset and scale faults of ±3 ºC and ±0.15% or more, respectively with 100% accuracy.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Ali Emadi, for his inspiring leadership and continuous support throughout my Ph.D. journey. He is a role model to me, and I always learn from his passion and dedication in the way he leads our group. I am also very grateful to him for giving me the chance to be a part of one of the world's leading academic research programs in transportation electrification at McMaster Automotive Resource Center (MARC). This gave me the opportunity to work in a unique lab and testing environment and collaborate with distinguished colleagues from different expertise areas.

I would like to thank my supervisory committee members, Dr. Mehdi Narimani and Dr. Berker Bilgin, for their valuable comments and suggestions that always help me improve the quality of my Ph.D. thesis. I also would like to express my deepest appreciation to Dr. Phillip Kollmeyer for all his time and sincere support. We used to have a meeting weekly since I joined MARC, Dr. Phil was always present and eager to help in every part of my research work. I am also thankful to Dr. Ryan Ahmed for his support in sharing his knowledge and answering my questions at the beginning of my Ph.D. study. I would like to thank Dr. Carlos Vidal for his support, especially in establishing my research direction.

I also would like to thank my friends, Josimar Duque, Junran Chen, Fauzia Khanum, Jeremy Lambert and all my friends in the energy management system team and Cam Fisher for their support in the testing and the collection of the data.

Words cannot express my gratitude to my wife Christine for her dedication, patience, and sacrifice throughout our journey. She never showed less interest in hearing my presentations and helping me improve my work. I am also grateful to my parents and parents-in-law for giving me support and encouragement through my Ph.D. I also would like to thank my siblings, Fady, Hala, and Sandy, for the love they showed to my parents and me every time we needed it.

This research was undertaken, in part, thanks to funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Research Chair in Transportation Electrification and Smart Mobility.

Contents

Chapter 4 Application of Deep Neural Networks for Lithium-Ion Battery Surface Temperature Estimation Under Driving and Fast Charge Conditions 76

List of Figures

List of Tables

Notation

Abbreviations

ALBO	Adaptive Lemberger observer
BMS	Battery management system
CNN	Convolutional neural network
CRELU	Clipped rectified linear unit
DNN	Deep neural network
ECM	Equivalent circuit model
EKF	Extended Kalman filter
EKF-RLS	EKF with recursive least squares filter
EV	Electric vehicle
EVB	Microprocessor's evaluation board
FNN	Feedforward neural network
GRU	Gated recurrent unit
HPPC	Hybrid pulse power characterization
HWCUST	Customized highway drive cycle
HWFET	Highway fuel economy test
LA92	Unified dynamometer driving schedule
LIB	Lithium-ion batterie
LIBP	Lithium-ion battery pack
LP	Thermal lumped parameters

LP+FNN	Integrated physics and neural network-based thermal
LRELU	Leaky rectified linear unit
LS	least squares algorithm
LSTM	Long short-term memory layer
MAE	Mean absolute error
MAXE	Maximum error
MCU	Micro control unit
NARX	Non-linear autoregressive exogenous model
NN	Neural network
NPF	Nonlinear predictive filter
OCV	Open circuit voltage
PDE	Partial differential equation
PHEV	Plug-in hybrid electric vehicle
PIL	Processor in the loop
RAM	Random access memory
RLS	Recursive least squares
RMSE	Root mean square error
RNN	Recurrent neural network
SDA	Open serial debug adaptor
SHARCNET	Shared hierarchical academic computing network
SM	Submodule
SMC	Sequential Monte Carlo

SOC	State of charge
SOH	State of health
SPKF	Sigma point Kalman filter
TMS	Thermal management system
UART	Universal asynchronous receiver-transmitter
UDDS	Urban dynamometer driving schedule
UKF	Unscented Kalman filter
US06	Supplemental federal test procedure
WT	Waiting window

Chapter 1

Introduction

1.1 Background and motivation

Recently, lithium-ion batteries (LIBs) have achieved wide acceptance for various energy storage applications, such as electric vehicles (EVs) and smart grids. As a vital component in EVs, the performance of lithium-ion batteries in the last few decades has made significant progress. Lithium-ion battery packs (LIBPs) are composed of series and parallel configurations of lithium-ion cells. The cells, although from the same manufacturing batch, vary in electrical characteristics, which adds more complexity to the battery management system (BMS). The BMS has an important role in monitoring the cells and controlling the pack. Hence, the development of an efficient BMS has become a necessity to ensure the reliability and safety of battery packs.

The state of charge (SOC) of LIB is an essential parameter that should be reported to the vehicle electronic control unit and the driver. Inaccurate reported SOC impacts the reliability and safety of the LIBP and vehicle. The SOC cannot be measured directly using sensors; hence, a robust SOC estimator must be implemented on the BMS. Different algorithms are used to estimate the SOC of a LIBP, including measurement-based, adaptive filters and observers, and data-driven; however, there is a gap in feasibility studies of running these algorithms for multi-cell LIBP on actual BMS microprocessors.

Temperature sensors are usually utilized to monitor the temperature of the cells in LIBPs. Using a temperature sensor for every cell is often impractical due to cost, and wiring complexity. Besides, most smart thermal fault detection systems require redundant models that help in detecting and identifying thermal faults. Hence, robust temperature estimation models can replace the physical sensors and help the fault detection algorithms by providing a redundant monitoring system for the existing sensors. Several studies have investigated the development of accurate temperature estimation models. However, these studies have only examined a limited range of operating conditions. Hence, a comprehensive analysis of different battery chemistry, dynamic, fast charge, aging conditions, and ambient temperatures is always required to ensure the robustness of these models before they are deployed to the BMS.

LIBPs experience several faults during operation that lead to accelerated battery degradation and, in the worst case, trigger pack fire and explosion. The risk of the thermal runway of LIBPs is considered the major factor contributing to the decision not to buy an EV. BMS should be able to early detect and identify these faults before triggering any of the LIBP safety limits. Several methods are developed to analyze the measurements from LIBP and detect the faults by capturing anomalies in these measurements. However, the quantity and quality of the collected measurements from the pack are considered one of the most challenging aspects contributing to the inaccuracy of these methods. In addition, several studies utilize the measured voltage of each cell to detect most of the electrical faults and severe thermal faults. However, for moderate and less severe thermal faults resulting in moderate temperature rise, the anomalies in the voltage can be negligible, and it is unclear if these faults can be detected using existing methods. Most thermal faults occur in the cooling system often leads to poor heat dissipation in the LIBP which are not sufficient to trigger the maximum temperature limits. These faults include cooling system failure, coolant lower or higher than a setpoint, coolant flow blockage and temperature sensor faults.

1.2 Thesis contributions

Based on the previous discussion, accurate SOC and temperature estimation models that are feasible to run on the BMS are essential for the safety and reliability of the EV. Therefore, this research presents significant contributions to the area of SOC and temperature estimation models’ development for multi-cell LIBPs and deployment feasibility to BMS microprocessors. Besides, this research also contributes to thermal fault detection and identification development by developing an accurate method that can detect various thermal faults. Hence, the main contributions of this thesis are summarized as follows:

1- A comprehensive overview of the characteristics of multi-cell lithium-ion battery packs, including inconsistency analysis, SOC estimation algorithms and methods and balancing techniques

2- Development of methodology for a processor in the loop (PIL) for measuring the execution time and memory use measurements of a feedforward neural network (FNN) and recurrent neural network (RNN) with long short-term memory (LSTM) and benchmarking them versus traditional Kalman filter SOC estimation algorithms

3- Development of two types of deep neural networks (DNNs) to predict lithium-ion cell surface temperature, including a FNN and a recurrent neural network with LSTM layer

4- Investigation of the optimal frequencies applied to the inputs to the FNN temperature estimation model

5- Evaluation of the proposed DNN temperature estimation models using data from three different lithium-ion batteries at realistic, challenging operating conditions, including low ambient temperature, high temperature rise, fast charging, and aging

6- A comprehensive benchmarking of the proposed DNN temperature estimation models versus prior studies in the literature

7- Measurement of BMS microprocessor execution time and memory use for the proposed DNN temperature estimation models using a PIL platform

8- Development of a physics-based combined with neural network temperature estimation (LP+FNN) concept for multicell LIBPs

9- Comprehensive development stages of the proposed LP+FNN model, including thermal lumped parameters determination method and training of the neural network

10- Evaluation of the proposed LP+FNN using data collected from an air-cooled 72-cell LIBP at driving and fast charge conditions at two different ambient temperatures

11- Development of statistical-based method using the LP+FNN model to detect thermal faults before triggering the maximum temperature limit

12- Development of a fault identification scheme to identify different thermal faults

13- Validation of the proposed fault detection and identification method using various thermal faults, including cooling fan system failure, airflow lower or higher than setpoint, airflow blockage and temperature sensor reading faults

1.3 Thesis outline

This thesis aims to present accurate and computationally feasible SOC and temperature estimation models for lithium-ion batteries and packs. Besides, the thesis presents a comprehensive investigation of the different thermal faults of LIBPs and a new method to detect and identify them. The thesis is organized as follows:

Chapter 2 presents an overview of the characteristics of multi-cell lithium-ion battery packs. Besides, the chapter investigates battery pack SOC estimation methods along with the impact of cell inconsistency on pack performance and SOC estimation. A detailed consistency analysis for lithium-ion battery packs is presented, including reasons and impacts on the pack performance. Besides, four categories of pack SOC estimation methods are presented, including individual cell, lumped cell, reference cell, and mean cell and difference estimation methods. The SOC estimation methods are compared in terms of algorithm type, computational load, and engineering effort to help practitioners decide which method best fits their application. Cell balancing methods, which are necessary due to cell inconsistencies, are discussed as well.

Chapter 3 proposes a processor in the loop (PIL) platform that is used to assess the execution time and memory use of different SOC estimation algorithms. Four different SOC estimation algorithms are presented and benchmarked, including a feedforward neural network (FNN) enhanced with external filters, a recurrent neural network with long short-term memory (LSTM), an extended Kalman filter (EKF), and EKF with recursive least squares filter (EKF-RLS). The algorithms are deployed to two different NXP S32Kx microprocessors and executed in real-time to assess the algorithms' computational load. The algorithms are benchmarked in terms of accuracy, execution time, flash memory, and random access memory (RAM) use. In order to ensure the validity of running these models for multiple cells in the pack, the impact of increasing the number of instances to run each algorithm simultaneously is investigated as well.

Chapter 4 presents two deep neural network (DNN) modeling approaches that are used to predict the surface temperature of LIBs. The first model type is based on a FNN enhanced with external filters, while the second model is based on a recurrent neural network with a LSTM layer. These models are trained and tested using experimental data from three batteries, two cylindrical cells, and one pouch cell at a range of driving, fast charging and health conditions.

Chapter 5 introduces an integrated physics and deep neural network-based battery pack thermal model (LP+FNN). Besides, the model is benchmarked against the traditional thermal lumped parameters (LP) and FNN models. The parametrization and training of the proposed model are discussed, then the model is tested using drive cycles from an air-cooled LIBP at various thermal and driving conditions.

Chapter 6 proposes a method to detect and identify different thermal faults of LIBPs before triggering the maximum temperature limit. The method works by comparing the measured and the estimated temperatures from an accurate physics-based combined with a neural network model. The proposed fault detection and identification method is validated using various thermal faults, including cooling system failure, incorrect airflow, airflow blockage and temperature sensor reading faults.

Chapter 7 provides a summary of the thesis, the conclusions, and recommendations for future research.

Chapter 2

Lithium-Ion Battery Pack Characteristics and Challenges

Recently lithium-ion battery packs have gained significant interest, especially for electric vehicle (EV) applications. Hybrid and electric vehicle battery packs are composed of series and parallel configurations of lithium-ion cells. The utilization of series and parallel connections allows for essentially any pack voltage and energy to be achieved; however, this adds more complexity for the battery management system (BMS) which monitors the cells and controls the pack [1]. Vehicles may have a very high number of small cells, such as one configuration of the Tesla Model S which has 7140 2.9 Ah cylindrical cells arranged in 16 modules, with each module consisting of 74 parallel cells and six series cells (74P6S) [2].

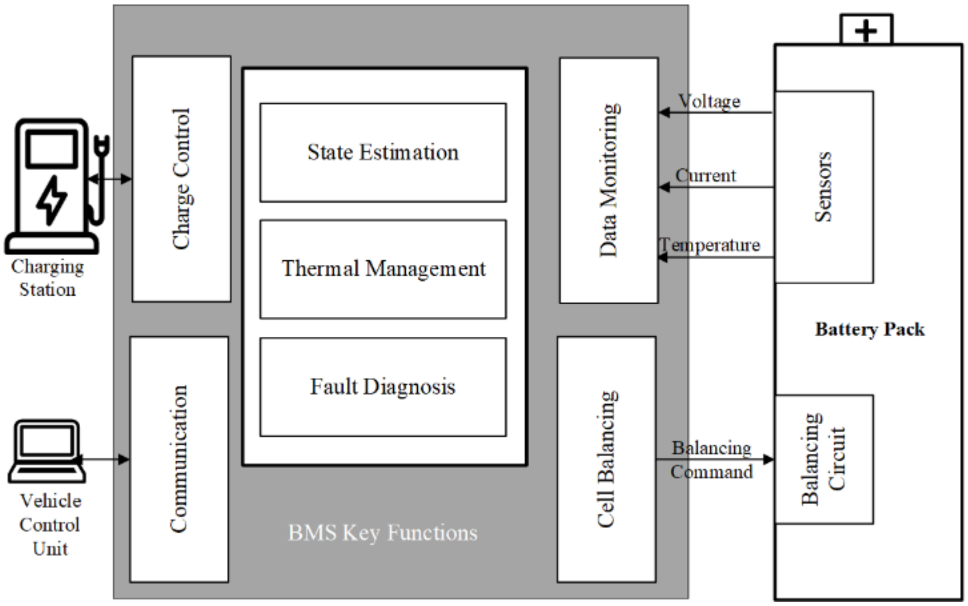


Figure 2-1: Battery management system key functions

They may also have a smaller number of larger cells, like one configuration of the BMW i3 which utilizes 50 Ah prismatic cells arranged in eight modules consisting of 2 parallel and 12 series cells each (2P12S) [3]. The BMS must be able to monitor the state of charge and ensure a balanced state of charge (SOC) for each series connected group of parallel cells in the pack. Figure 2-1 shows the main components and key functions of the BMS. Battery state of charge is defined as the ratio of coulombs of charge currently stored in the cell over the cell's total charge capacity. The SOC cannot be measured directly using sensors; hence, a robust SOC estimator must be implemented with the BMS to ensure accurate SOC values are reported to the driver [4]. Generally, the SOC of a battery can be estimated using different algorithms, including measurement-based, adaptive filters and observers, and data-driven algorithms. For the measurement-based algorithms, SOC is estimated based on measured physical quantities; typically integrated current [5], [6], open circuit voltage (OCV) [7], [8], or impedance [9], which are directly related to the cell SOC. Adaptive filters and observers estimate SOC using a battery model combined with measured physical quantities. Examples of these algorithms include the family of Kalman filters [10]–[14] and the particle filter [15], [16], least squares filter [17], [18], and adaptive Luenberger observer [19]. Data-driven algorithms, which are based on machine learning models, are often referred to as black-box models because they model LIB input-output relationships without the need for models representing the underlying physics or chemistry. Machine learning models are trained with measured data such as voltage, current, temperature, and state of charge. Recurrent and non-recurrent networks have been used to reliably and accurately estimate battery SOC [4], [20]–[22] and state of health (SOH) [4], [23] of the battery.

Battery cells, even when manufactured in the same batch, can have parameter variations of 1% or more. One study, for example, found resistance to vary by 0.3% and capacity to vary by 1.3% [24]. When cells are assembled in a pack bus bar, weld resistance can further exacerbate differences between cells. These factors contribute to inhomogeneous current, voltage, temperature, and cell characteristics during pack operation, and may cause further aging to the battery pack [10]. For example, in [25], 48 cells from the same batch were tested under identical conditions. A 10% capacity variation was found between the cells after 1000 cycles, and the cause of this variation was determined to be inconsistent manufacturing of the cells. Hence, some studies have presented a screening process for selecting homogeneous cells to group them in a pack [26], [27]. There are many methods for identifying inconsistencies in a battery pack from measured data, including signal processing, model, and data fusion-based methods which will be discussed in Section 2.1.

When estimating SOC for a battery pack, the SOC of each cell must be considered due to inevitable variances in cell characteristics. Pack estimation algorithms utilize the discussed SOC estimation algorithms and may include additional features to minimize computational complexity or increase the accuracy. The simplest class of pack estimation algorithms lumps all the cells into a single large cell and estimates SOC for the lumped cell, ignoring differences in SOC of individual cells. An individual SOC estimator can also be used for each cell, but the computational complexity may be too high for the BMS. To address this, a reference cell may be selected for the pack, and then a higher bandwidth, more accurate SOC estimation method can be used for this cell. Lower bandwidth, less accurate SOC algorithms can then be used for the remainder of the cells, reducing the computational requirements. Pack SOC estimation methods can also utilize difference models, which estimate the difference in SOC of each cell from the mean cell SOC. Details of each of these types of methods, including implementation, advantages, and disadvantages are discussed in Section 2.2.

Lithium-ion battery packs also require a means of adjusting or balancing individual cell SOC due to variations of the cells' characteristics and operating conditions. Cell balancing methods can be categorized into two main methods, namely, dissipative and non-dissipative methods. Dissipative methods typically discharge a cell by applying a resistor across it and tend to be slow acting but low cost and compact. Non-dissipative methods utilize power electronic circuits to transfer energy between cells. They may act more quickly and conserve energy but increase the cost and size of the BMS. Dissipative and non-dissipative balancing methods are discussed and compared in detail in Section 2.3.

Overall, this chapter presents state of the art and current challenges for developing robust SOC estimation algorithms for lithium-ion battery packs and considers cell inconsistency and balancing in relation to SOC estimation. The causes of inconsistent performance among cells in the pack along with their impact on the pack performance are discussed and modeled results are presented to depict how variances of cell capacity and resistance impact cell state of charge, current and voltage distribution. A detailed discussion of different battery pack SOC estimation methods is provided, including algorithms used and the theory of operation of each method. Cell balancing methods are also compared in terms of the active elements, advantages, and disadvantages of each method. The rest of the chapter is organized as follows; in section 2.1 the consistency analysis of lithium-ion battery packs is discussed and in section 2.2 battery pack SOC estimation algorithms and methods are presented. In section 2.3, the range of cell balancing methods are discussed and finally, the conclusions and recommendations are presented in section 2.4.

2.1 Consistency analysis for lithium-ion battery cells and packs

Ideally each battery cell leaving a manufacturing line would perform identically throughout its life if they were used under the same conditions. However, many factors cause cells in a pack to age at different rates though, such as variances in manufacturing processes and uneven temperature distribution. These factors tend to cause cells to age unevenly over time, as described in Figure 2-2, making managing a battery pack over its life difficult. In this section, the causes of inconsistent performance among cells in a pack are discussed along with their impact on pack performance. Modeling is also used in this section to depict how variances of cell capacity and resistance impact cell state of charge and voltage and current distribution.

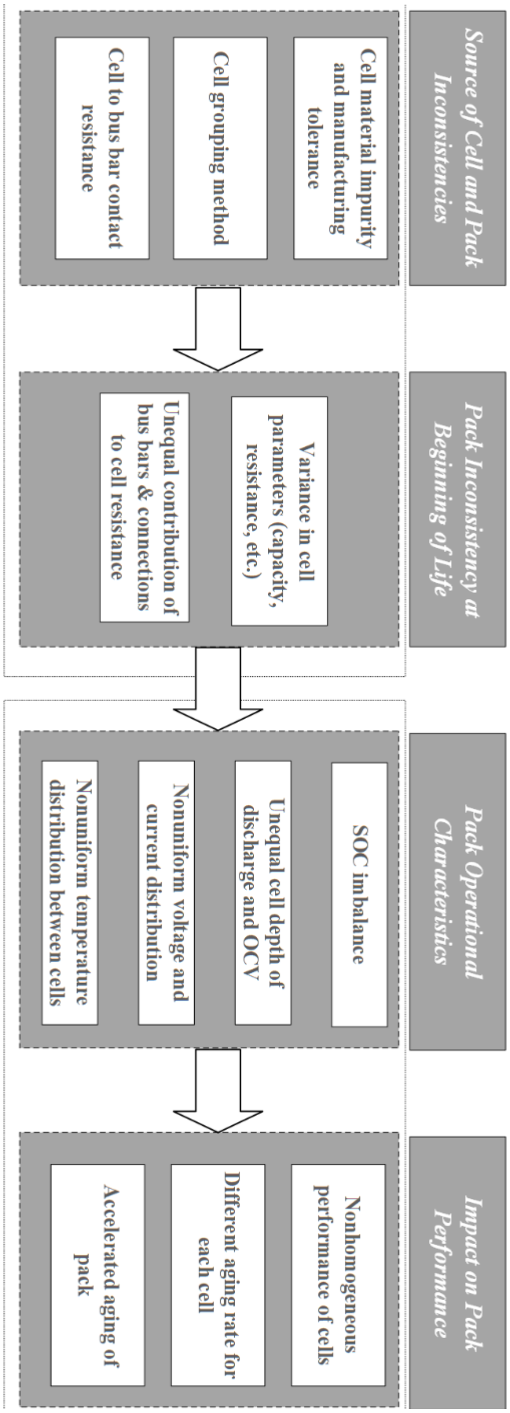


Figure 2-2: Lithium-ion battery packs inconsistency production and operational causes and effects

2.1.1 Inconsistency of cell manufacturing and welding progress

Inconsistent cell characteristics originate at the production stage. Impurities in active materials and tolerance in human and automated manufacturing processes contribute to cell performance variations [24], [28]–[30]. Typically, these factors result in each cell having slight differences in capacity and resistance. For example, in one very comprehensive study, quantity 1100 Sony US26650FTC1 3 Ah cylindrical LiFePO4-graphite cells were tested [24]. Although, the cells produced in the same batch, the standard distribution of discharge capacity and dc resistance was found to be around 0.3% and 1.3%, respectively. Diagnostic tests, such as the wavelet-based method proposed in [31], may be used quickly to measure cell parameters so cells can be sorted into similar groups prior to the assembly of packs. While the differences between cells are typically small, they may have a significant impact on pack performance over time as will be discussed in the next subsection.

When cells are assembled into a pack, series and parallel connections are made with conductive bus bars. The connection between the cells and the bus bars are most often made via a welding process. The welding process has a significant impact on the overall resistance of the pack. Since all welds are not identical, the welding process also contributes to differences in resistance between cells. In one study, cylindrical cell to bus bar connection methods including press contact, resistance spot welding, ultrasonic welding, laser beam welding, and soldering were investigated and found to have resistances of 0.154, 0.167, 0.169, 0.130, and 0.080

mΩ respectively [28]. Depending on the connection type, resistance can vary by as much as 0.02 mΩ, which would cause around a 0.1% variance in resistance for a cylindrical cell with 20 mΩ nominal resistance [28]. Unequal cell to bus bar resistance has been found to cause uneven heat generation in battery packs [29] and to contribute to unequal current sharing between parallel connected cells, especially at higher c-rates [30].

2.1.2 Impact of inconsistencies on pack performance

Inconsistencies in initial cell resistance and capacity, as well as resistance variance introduced by cell to bus bar connections, all contribute to inhomogeneous current, voltage, temperature, and aging in battery packs. Many studies have investigated the impact of these inhomogeneities and quantified their impact.

For example, a 3S3P configured pack with cell resistance and capacity varying by 5% was found to have an 8% SOC imbalance and 3 ºC variance in temperature across the pack after a full discharge [32]. After 310 days of cycling, the variance of capacity and resistance between the cells grew to 10% and 25% respectively, demonstrating that over time variance between cells tends to grow. In another study, a 20% mismatch in resistance and capacity of parallel connected cells was found to reduce pack lifetime by 40%, showing the importance of having consistent cell characteristics [33].

Differences in cell resistance also contribute to uneven current and voltage distribution among parallel and series connected cells. In [34], low temperatures were found to exacerbate this issue, with two parallel connected cells with a 13% difference in resistance having a 50% difference in current at 5 ºC. In [35], the current of parallel connected cells was found to be directly proportional to their relative capacities, leading to imbalanced state of charge during operation. Differences in cell capacity also cause nonuniform voltage and temperature distribution, as was observed for an 8S pack with 2.5% variation in capacity between cells which led to a 0.8% variation in terminal voltage between cells during a cycle [36].

The cooling system for a battery pack can also drive uneven temperature distribution. If the cooling media, typically air or liquid, significantly increases in temperature as it flows across the pack or does not cool some parts of the pack as effectively as others, there will be a temperature distribution across the cells. In [37], the liquid cooling system caused a 4 ºC temperature variation between cells in a 4S pack and resulted in a 1% variation in voltage between the cells during operation. Non-uniform temperature was shown to cause a 25% difference in current for two parallel connected cells in [38], and to cause 5% additional aging for a temperature distribution of 5 ºC in [39]. Cooling systems should therefore minimize the temperature variation across the pack, ideally keeping the hottest and coolest parts of the pack within a few degrees Celsius of each other.

2.1.3 Evaluation of pack inconsistencies

Inconsistencies in the pack will result in differences in voltage, current, and temperature between cells. Cell resistance, capacity, and other characteristics can be estimated from measured voltage, current and temperature data. There are many methods for inconsistencies evaluation in a battery pack from measured data, including model, and data fusion-based methods. For signal processing approaches, time domain voltage and current data, which is typically collected using lab-based tests, are used to extract pack inconsistency features [40], [41]. Model-based methods utilize adaptive filters to fit equivalent circuit models to measured data and to estimate important features of each cell during operation, such as resistance and capacity [42], [43]. Data fusion methods directly quantify the cell inconsistency using mathematical theories without the need for a cell model, and include information entropy [44], principal component analysis [45], and copula theory [46]. All of these methods can be helpful for managing battery pack performance over time, and for determining if a battery pack meets manufacturing consistency and performance specifications.

2.1.4 Modeled impact of resistance and capacity variation on cell current, voltage and SOC

To illustrate how resistance and capacity variation between cells affect the distribution of voltage and current in a battery pack, several different cases are modeled in this section. The cells are modeled with a simple equivalent circuit model (ECM) developed in Simulink, which includes state of charge dependent open circuit voltage in series with a resistance. Figure 2-3 and Figure 2-4 show the impact of the variance of cell resistance on the performance of battery packs with three series and three parallel connected cells. The modeled cells have identical capacity and OCV-SOC characteristics and a 20% resistance variance, as labeled in the figures. Figure 2-3 shows that a 20% resistance variance in a 3S battery pack leads to significant differences between terminal voltage even though SOC is the same at each point in time. Figure 2-4 shows that when the cells are connected in parallel, the resistance variance leads to imbalanced current and, therefore imbalanced SOC during the charge. Once the charge stops, the cells' SOC will equalize due to the self-balancing effect [10], but these circulating currents will cause some extra loss.

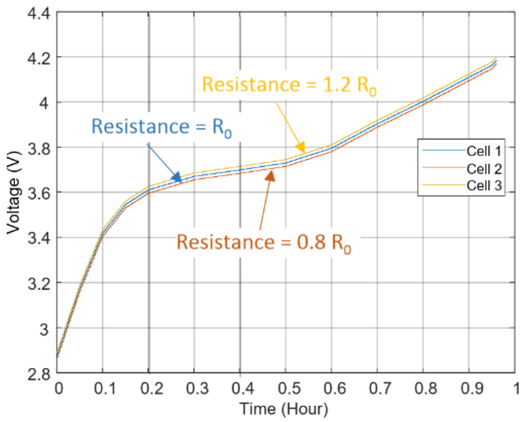


Figure 2-3: Impact of 20% internal resistance mismatch on voltage sharing of 3 series connected cells during constant current charge

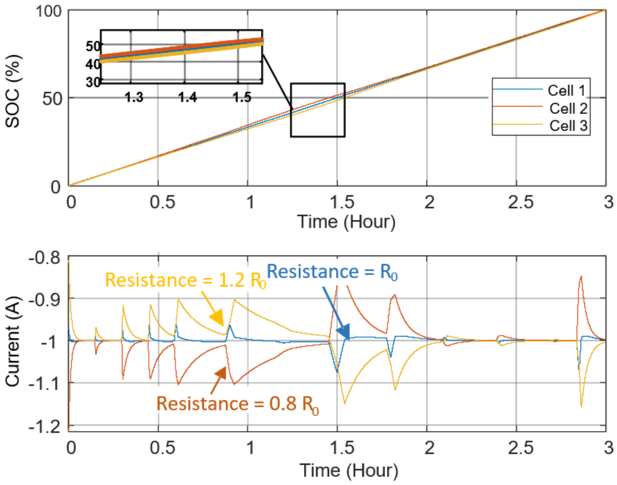


Figure 2-4: Impact of 20% internal resistance mismatch on SOC and current sharing of 3 parallel connected cells during constant current charge

Figure 2-5 and Figure 2-6 show the impact of the cell capacity variance on the performance of a 3S and a 3P battery pack, respectively. The cells are assumed to have identical internal resistance and OCV-SOC characteristics and a 10% capacity variance, as indicated in each figure. The figures show cell capacity variance has a significant impact on the SOC of series connected cells, an imbalance that would have to be corrected for through cell balancing methods like those discussed in Section 2.3.

The magnitude and types of cell inconsistency and their impact on the performance of the battery pack should be considered when developing battery pack estimation methods. Pack SOC estimation methods must be able to identify the SOC of each cell, even for example when the cells' terminal voltage is different due to resistance variation. Cell inconsistency may also lead to inhomogeneous degradation of the cells and accelerated aging of the pack.

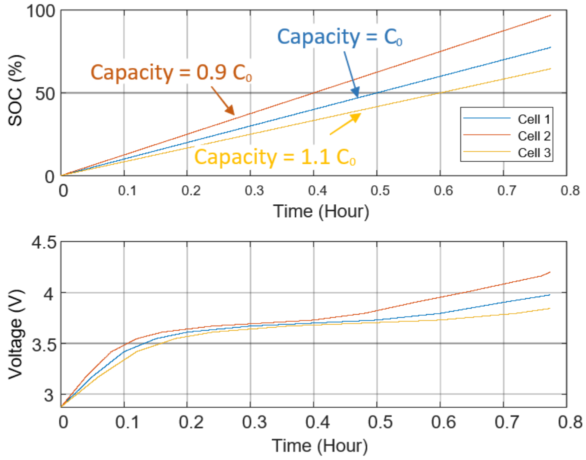


Figure 2-5: Impact of 10% capacity mismatch on SOC and voltage sharing of 3 series connected cells during constant current charge

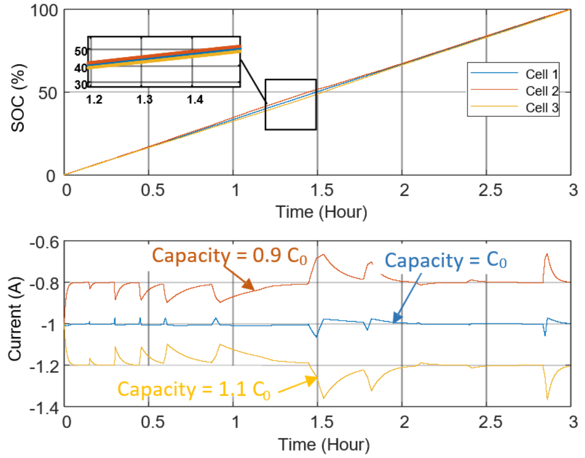


Figure 2-6: Impact of 10% capacity mismatch on SOC and current sharing of 3 parallel connected cells during constant current charge.

2.2 SOC estimation of lithium-ion battery packs

Estimating the state of charge for a lithium-ion battery pack is challenging because each series connection of parallel cells will have slightly different characteristics, such as capacity, resistance, or temperature. SOC estimation algorithms, which have been the subject of increasing interest in the literature (see Figure 2-7), must be able to account for differences in cell characteristics and report an SOC value for the pack. Several algorithms of estimating cell SOC are discussed in Section 2.2.1. These algorithms are then applied within pack SOC estimation methods like those which are compared and contrasted in detail in Section 2.2.2.

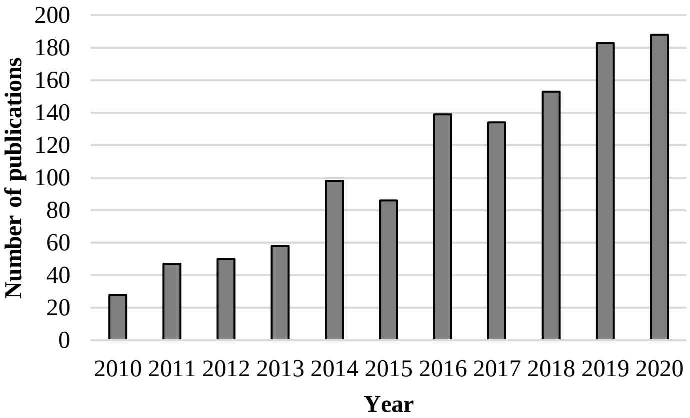


Figure 2-7: Number of publications each year for keywords "state of charge" and "Lithium-ion batteries" searched in the IEEE library

2.2.1 SOC estimation algorithms

Generally, SOC can be estimated using different algorithms, including measurement-based, adaptive filters and observers, and data-driven algorithms. In the measurement-based algorithms, SOC is estimated based on measuring some physical quantities which are directly related to the cell SOC namely; coulomb counting [5], [6], OCV [7], [8], and electrochemical impedance stereoscopy (EIS) [9]. For coulomb counting, SOC is estimated by integrating battery current [5], [6]. SOC can be estimated directly through open circuit voltage, but OCV can only be directly observed after an hour or more of non-use [7], [8]. Battery impedance measured with EIS, which applies a sinusoidal voltage or current to the battery and measures the response, can also be used to estimate the battery SOC [9].

Adaptive filters and observers estimate SOC as a function of measured terminal voltage, current, and temperature and are a more practical solution because they can estimate SOC during operation and correct for current sensor and other errors. These algorithms utilize a battery ECM or electrochemical model as part of the estimation process. The Kalman filter [10]–[14], particle filter [15], [16], least squares filter [17], [18], and adaptive Luenberger observer (ALBO) [19] are all commonly used to estimate battery SOC. The family of Kalman filters includes the extended Kalman filter (EKF) [10], fuzzy-based EKF [11], adaptive Kalman filter (AKF) [12], sigma point Kalman filter (SPKF) [13], and unscented Kalman filter (UKF) [14]. Generally, the Kalman filter estimates SOC via coulomb counting and an equivalent circuit model which is fit to the measured data. The filter is tuned to adjust how much it trusts the model and measurements. The particle filter or so-called sequential Monte Carlo (SMC) filter utilizes a Monte Carlo sampling method to extract particles from the posterior probability distribution, update their weights, and thus estimate battery SOC [15], [16]. The least-squares filter is a statistical algorithm in which regression analysis is used to determine the best-fit line for a given dataset. Least squares filters have been used with battery models to estimate battery SOC [17] and capacity [18]. Adaptive observers such as the ALBO, which updates observer gain each time step to match the stochastic nature of SOC estimation [19], can also be used to estimate SOC.

Another category of SOC estimation algorithms is based on data-driven algorithms such as neural networks, deep learning, and support vector machine algorithms. These algorithms treat the battery as a black box, and learn the relation between measured values and SOC. The models are trained with data recorded during operation, such as voltage, current, temperature, and state of charge which is the target. Recurrent and non-recurrent networks have been used to estimate SOC [20]–[22] and state of health (SOH) as well [4].

2.2.2 Battery pack SOC estimation methods

Battery pack SOC estimation methods must consider all the inconsistencies which are common in battery packs, while also not placing too much computational load on the BMS. Pack SOC estimation methods, therefore, aim to simplify the estimation process and improve accuracy by lumping the cells together as a single large cell, by estimating SOC of some cells at a lower update rate, or by estimating cell SOC difference compared to a mean cell. These SOC estimation methods, along with the simplest method of just estimating SOC for each individual cell, are discussed in the following sections.

2.2.2.1 Individual cell estimation

A straightforward method of pack SOC estimation is to implement a single SOC estimator for each individual cell, like those described in section 2.2.1. The pack SOC is then determined as a function of the individual cell SOCs, with for the simplest case the minimum cell SOC used to represent pack SOC during discharging and the maximum cell SOC used during charging, as described in equation (2.1) and shown in Figure 2-8. Because each SOC estimation algorithm instance may utilize significant computation resources, this method is not always practical. Other pack estimation methods aim to reduce computational load and potentially improve estimation accuracy as well.

$$SOC_{pack} = \begin{cases} SOC_{\min cell}, & during\ Discharging \\ SOC_{\max cell}, & during\ Charging \end{cases} \tag{2.1}$$

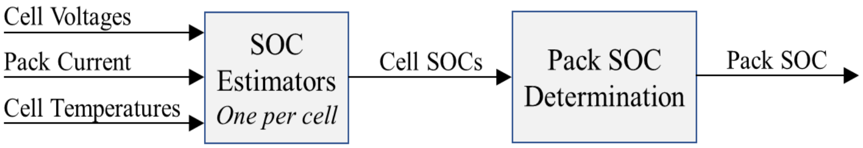


Figure 2-8: Individual cell estimation method

2.2.2.2 Lumped cell estimation

If the cells in a battery pack have similar characteristics, then it may be suitable to simply consider the pack to be one large cell and to estimate SOC as a function of the overall pack voltage, V_{pack}, and current, I_{pack}, as illustrated in Figure 2-9.

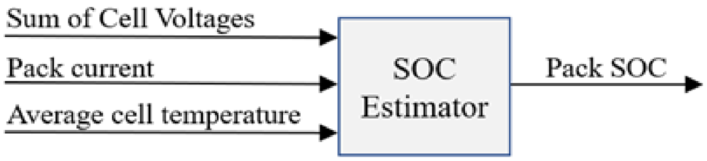


Figure 2-9: Lumped SOC estimation method

Many such lumped cell models have been proposed in the literature, such as in [10] where the parameters for a pack ECM were determined offline using a genetic algorithm and SOC was estimated each time step with an EKF. In [19], a similar approach was taken, and pack ECM parameters were determined with a hybrid pulse power characterization test, and an ALBO was used to estimate SOC. Incorporating aging into SOC estimation is also important, as was done in [11] using a fuzzy-based EKF for estimating SOC at various stages of aging. Since SOC is a direct function of open circuit voltage, it is also possible to consider algorithms which eliminate the battery pack current sensor, a costly component. The study in [47] proposed such a current sensor free solution using filtered terminal voltage and ECM, where reasonable SOC estimation accuracy was achieved for a pack. However, it significantly increased the computational load of the algorithm.

The main advantages of lumped cell methods are their simplicity, and they may be a good option for less dynamic applications where the SOC imbalance of cells is not expected to be large. However, this method can lead to accelerated aging of the weakest cell in the pack and of a poor estimation of pack SOC if cell characteristics vary too much.

2.2.2.3 Reference cell estimation

As an alternative to lumping all the cells together a single cell from the pack, referred to as the reference cell, can be selected to represent the pack performance. The SOC of the reference cell is then estimated using a higher bandwidth, more accurate SOC estimation method. The remaining cells may have a simpler, lower bandwidth SOC estimation method applied as shown in Figure 2-10, allowing for a good pack SOC estimate without the need to have a full performance estimator for each cell. The reference cell is typically chosen based on the weakest cell, i.e., the cell with the lowest voltage during discharge and highest during charging, as is done in [48]-[50].

In [48], the lowest voltage cell is selected as the reference cell and two proposed modified nonlinear predictive filters (NPF) are used, one executed at a higher frequency and a second at a lower frequency, to provide two different estimates of reference cell SOC. The proposed NPF is a modified optimal state estimator implemented for nonlinear systems. However, the process noise can take any form and is estimated with the optimal state. In [49] the lowest voltage cell was also selected as the reference cell, and an online variable factor recursive least squares filter was used to estimate the reference cell parameters for an EKF SOC estimation method. The work in [50] expands beyond that in the other studies by proposing a dual time-scale EKF method which estimates SOC of the weakest reference cell at a higher frequency and SOC of the remaining cells at a lower frequency. The SOC estimation of all cells can be used in equalizer circuit control and improve the accuracy of the SOC estimation in case the reference cells are changed.

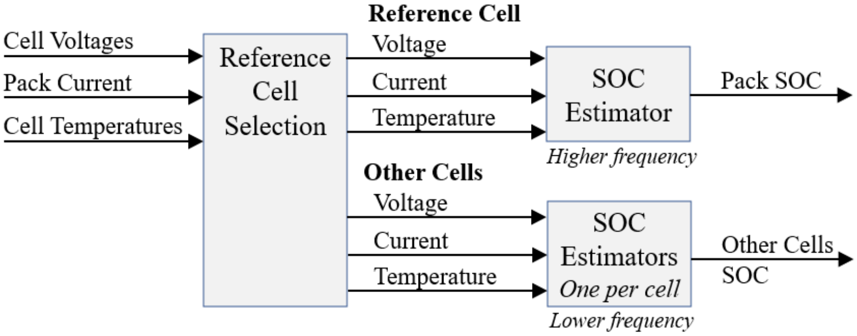


Figure 2-10: Reference SOC estimation method

2.2.2.4 Mean cell and reference estimation method

The individual, lumped, and reference cell methods all approach SOC estimation in a similar way, using cell or pack measurements as inputs and outputting cells or pack SOC. For mean cell and difference estimation another approach is taken. Mean cell SOC is estimated based on the mean of all the cell voltages and temperatures as shown in Figure 2-11. For each individual cell, the difference in SOC, ΔSOC, compared to the mean cell is estimated as a function of the difference between the individual and mean cell voltage, ΔVcell, and temperature, ΔT, using simplified cell difference models. An accurate, higher bandwidth method is used for estimating the mean cell SOC, and a simpler, lower bandwidth method is used to estimate the ΔSOC values. As a result, mean cell and difference estimation methods typically estimate cell SOC and thus the pack SOC as mentioned in equation (2.1), with good accuracy and low computational complexity compared to other methods.

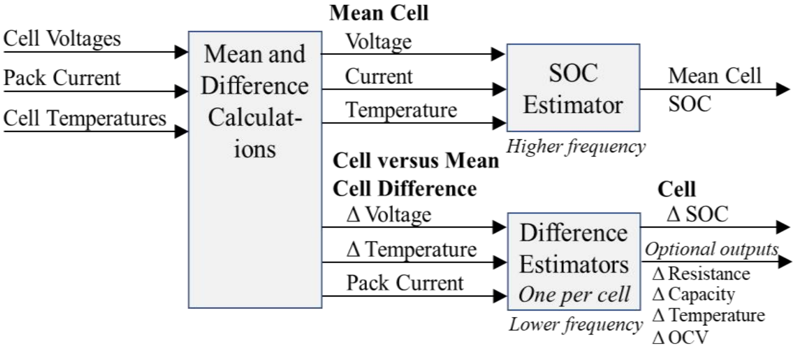


Figure 2-11: Mean and difference SOC estimation method

The cell difference models, which capture the difference in SOC between each cell and the mean cell, may also include the difference in internal resistance, capacity, temperature, polarization voltage, and OCV. Cell difference models are often equivalent circuit models with parameters such as delta open circuit voltage and delta resistance. The model parameters are determined by fitting the model to measured cell voltage difference data.

Many mean cell and difference SOC estimation methods have been proposed in the literature, including [13], [14], [42], [51]–[53] which utilize mean cell voltage and few estimators which alternatively select the most average cell to represent the mean cell [27], [54], [55] The work in [51] presented a dual timescale method using an EKF to estimate mean cell SOC at a higher frequency and a method using OCV to estimate cell SOC difference at a lower frequency. In [14] and [42], which use an UKF and the OCV respectively, the cell difference models also account for cell resistance deviation, leading to more precise SOC estimation. Cell charge and discharge resistance was considered in [52], an EKF was used to estimate both mean cell and cell difference SOCs. Another method in [13] utilized even more detailed cell difference models which include deviation in temperature, internal resistance, and capacity, and utilizes a SPKF to estimate mean cell SOC at a higher frequency and a second SPKF to estimate cell SOC difference at a lower frequency. An SPKF was also used to estimate mean cell SOC in [53], and a delta filter was used to estimate cell SOC difference via a difference model considering cell resistance and capacity differences.

The aforementioned studies all perform cell SOC difference estimation online, but this estimation can also be done using offline methods as in [54], [55]. Machine learning has been utilized as well, including in [54] and [55] which both use Kalman filters along with neural network based bias correction methods to determine cell to cell variations.

2.2.3 Comparison of pack SOC estimation methods

Table 2.1 presents a comprehensive comparison of the battery pack SOC estimation methods discussed in this section. The table lists the algorithm type, the cell variance characteristics considered, datasets used, and the corresponding SOC estimation root mean square error (RMSE), mean absolute error (MAE), and/or maximum error (MAXE). The variance in the cells' characteristics including resistance (*R*) and capacity (*C*) and their impact on the cells' temperature (*T*) and open circuit voltage (OCV). All of the algorithms are demonstrated to have quite reasonable error, RMSE a few percent or less or MAXE 5% or less. Error may be higher for a given application though, so it could be beneficial to evaluate several different algorithms and methods before selecting one.

The performance of the different pack SOC estimation methods is also compared qualitatively and quantitatively in Table 2.2. The individual cell and reference cell methods require the most individual SOC estimation algorithms, and therefore have a higher computational load for the BMS. The lumped cell method only has a single SOC estimation algorithm, and the mean cell and difference method utilize simplified difference models to estimate individual cell SOC, resulting in both methods having the lowest computational load for the BMS. The engineering effort to develop the algorithm is a qualitative value, indicating how much time and resources are needed to implement the algorithm. The individual and lumped cell methods require the least engineering effort to develop, since both only require standard SOC estimation algorithms while the reference and mean cell and difference methods require the development of specialized estimation methods for cells other than the reference or mean cell.

2.3 SOC balancing methods for lithium-ion battery packs

Because the cells in a battery pack have non-uniform properties, as was discussed in Section 2.1, it is necessary to have a method of balancing the pack to prevent cell SOC differences from growing over time. If the difference in SOC between cells becomes too large, the usable capacity will be substantially reduced due to the fullest cell limiting the maximum charge and the emptiest cell limiting the minimum charge, as illustrated in Figure 2-12. Cell balancing is typically a very slow process, with resistive balancing circuits dissipating a few hundred milliwatts of power from the most charged cells. Non-dissipative cell balancing circuits, which transfer energy from more charged to less charged or lower capacity cells, can also be used to extend the usable capacity of a battery pack.

Table 2.1: A comparison of SOC estimation methods in lithium-ion battery packs and the corresponding algorithms accuracy

SOC Method	**Pack Configurat-ion**	**Algorithm**	**Cell characteristics variance**				**Dataset**	**Error**
			R	**C**	**T**	**OCV**		
Lumped cell SOC estimation	3S [10]	EKF					DST^1	0.26 % RMSE
	16S [19]	ALBO					FUDS^2	2.5% MAXE
	120S [11]	Fuzzy-based EKF					NEDC^3 FTP^4	0.82% RMSE
	12S [47]	OCV	x				N/A	N/A
Reference cell SOC estimation	6S [48]	NPF	x				UDDS^5	2% MAXE
	4S [49]	EKF	x		x	x	FUDS	0.5 RMSE 20°C
	6S [50]	EKF	x	x			DST	2.34-2.37% RMSE
Mean Cell and Difference SOC estimation	96S [51]	EKF				x	Complex pulse current+ FTP	4% MAXE
	8S [14]	UKF	x			x	CC^6 Discharge	1.83% RMSE
	96S [42]	OCV	x			x	N/A	3% MAXE
	12S [52]	EKF	x			x	NEDC	2% MAXE
	12P7S [13]	SPKF	x	x	x		N/A	0.5 MAE at 20°C
	4S [53]	SPKF	x	x			UDDS	1% MAXE
	4S [27]	AEKF	x	x	x	x	UDDS+DST	1% MAXE
	6S [54]	EKF		x	x		UDDS+DST	2% RMSE 10°C
	12S [55]	AEKF		x			DST	2% MAXE

x: means covered in the study, DST1: Dynamic stress test, FUDS4: Federal urban driving schedule, NEDC3: New European drive cycle, FTP4: Federal test procedure, UDDS5: Urban dynamometer driving schedule, CC6: Constant current

Table 2.2: Performance comparison of SOC estimation methods

	Individual cell	**Lumped cell**	**Reference cell**	**Mean cell and difference**
Number of SOC algorithms	N	1	(1) + (N-1)	1 SOC + N SOC difference
Cell SOC estimation	Yes	No	Yes	Yes
Computational load for BMS	High	Low	Medium	Low
Engineering effort to develop algorithm	Low	Low	Medium	High

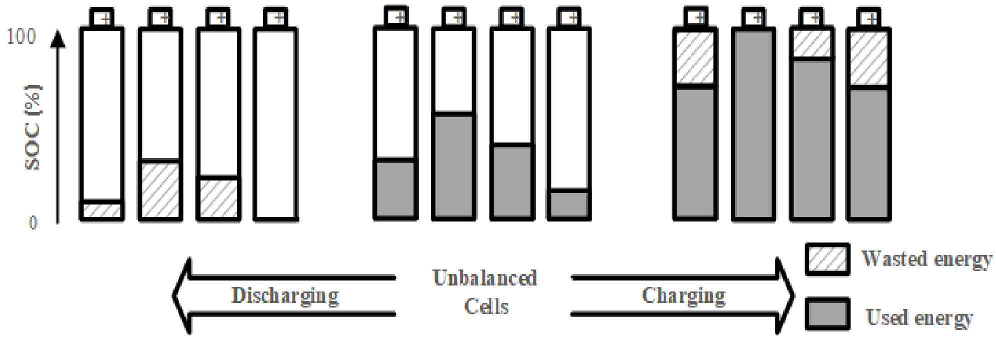


Figure 2-12: The impact of the SOC imbalance on 4-cell battery pack during charging and discharging scenarios

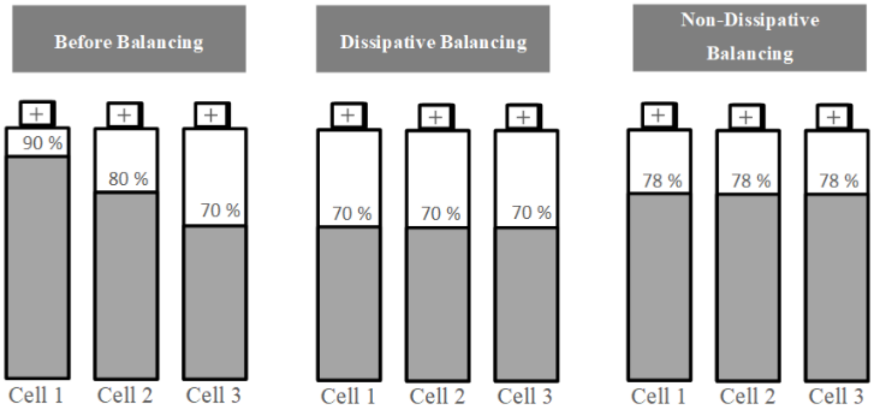


Figure 2-13: A comparison between dissipative and non-dissipative cell balancing methods

Non-dissipative balancing circuits are considerably more complex and expensive though and must be fairly high power to offer a meaningful improvement in battery pack performance. In cases of large SOC imbalance, as shown in Figure 2-13, non-dissipative balancing will prevent the need to simply convert excess charge to waste heat. Lithium-ion battery cells have such minimal self-discharge though that they typically require very little balancing over time, minimizing any efficiency benefits from non-dissipative methods. Despite this, there is still substantial interest in balancing methods for lithium-ion battery packs, including the dissipative and non-dissipative methods highlighted in the remainder of this section.

2.3.1 Dissipative balancing methods

Most EV manufacturers use dissipative resistive balancing circuits in their battery packs due to their reliability and simplicity. Switched shunt resistors [56] are most commonly used, but it is also possible to use fixed resistors which are always connected to the cells [57]. The fixed resistors cause the pack to naturally balance over time, since the highest voltage cells will have higher resistor current, but there are always losses even when the pack is fully balanced making it a rather undesirable method [58]. In the switched shunt resistor method [56], each cell is associated with a balancing resistor and a switch to connect it to the cell, typically a MOSFET. Most battery cell voltage measurement chips are able to directly control each balancing switch, and some even have the switches integrated into the chip and just require an external resistor. The BMS balances the pack by enabling the discharge resistors on the most charged cells.

2.3.2 Non-dissipative balancing methods

Non-dissipative balancing utilizes capacitor [56], [58], inductors [59], [60], transformers [61]–[63], and various common power electronic converter topologies [64]–[68] to transfer the energy among the cells within the pack. Energy is transferred from more charged to less charged cells, preventing the waste of energy present for dissipative methods. Non-dissipative balancing can achieve relatively high balancing speed [69] and high efficiency [70], which are the main advantages of this method. However, this method involves many components that add more cost and complexity to the balancing circuits [71].

2.3.2.1 Capacitor-based balancing

In this method, capacitors are utilized to transfer the energy between adjacent cells or from the pack to the cell, thus achieving cell balancing. All implementations are based of the same concept, a capacitor is charged while connected in parallel with a higher voltage cell and discharged while connected in parallel with a lower voltage cell. For double-tiered switched capacitor balancing, there is one capacitor per cell and two switches, and the capacitors are switched between adjacent cells at a 50% duty cycle to achieve equal voltage among cells [57]. For single switched capacitor balancing, a single capacitor is used with a group of cells along with five switches plus one switch per cell, allowing the capacitor to be connected in parallel with any of the cells [58]. The switches are controlled intelligently to move energy between cells until balancing is achieved. The double-tiered capacitor method has a faster balancing time and can more easily be modularized while the single switched capacitor method has fewer components [72].

2.3.2.2 Inductor-based balancing

For inductor-based methods, one or more inductors are utilized for cell balancing [61], [62]. The single-inductor balancing system utilizes one inductor to transfer the energy between the pack to the weakest cells [61]. The control system selects the weakest cell with the lowest SOC level to transfer the energy through activating the corresponding switches. The multi-inductor method utilizes N-1 inductor for balancing N cells, [62]. The controller senses the voltage difference of the two neighboring cells, then a control signal is applied to the switches with the condition that the higher cell must be switched on first to transfer the energy to the weakest cell. The inductor-based cell balancing methods have a relatively higher balancing speed and efficiency. However, they have higher switch current stress as compared to the remaining methods [74].

2.3.2.3 Transformer-based balancing

Transformers can be utilized to perform isolated transfer of power between cells and the pack and individual cells. The variations of this approach include the use of multiple transformers [63], transformers with multiple secondary windings [64], and a single transformer switched among cells [65]. The multiple transformers method [63] utilizes several transformers where all the primary windings are connected in parallel, and each of the secondary windings are connected to a separate cell via a diode. The primary winding is connected across the pack voltage via a switch, and power is transferred from the pack to the cells by switching at 50% duty cycle. For the multi-secondary winding transformer method [64], the multiple transformers used for the previous method are replaced by a single multi-secondary winding transformer and the balancing approach is the same. However, in this method, the number of cells is limited by the feasible number of secondary windings [74], [75]. For the single transformer method, the secondary winding is switched between cells to charge the weakest cells until balancing is achieved [65]. The switched transformer method is more compact, but to ensure good equalization between cells it requires a more complex control process than the other transformer-based methods [75].

2.3.2.4 Common converter topology-based balancing

Common dc-dc converter topologies can also be used for balancing, such as bidirectional buck-boost [64], bidirectional Cuk [65], bidirectional flyback [66], full-bridge [67], and quasi-resonant [68] converters. Typically, one converter per cell is utilized, and the converters transfer power between adjacent cells. Rather than simply allowing the voltage of cells to be matched like many of the prior methods discussed, the converters can control the flow of power in any way the BMS commands allowing more flexibility for managing SOC of the cells.

The bidirectional buck-boost converter is utilized in [64] to transfer energy between two adjacent battery cells. Another bidirectional converter, the Cûk converter, has the same principle of operation but utilizes capacitors as the energy transfer elements instead of inductors [65]. The bidirectional flyback converter, which is derived from the buck-boost converter [66], utilizes a transformer and fewer components to achieve cell balancing. Bidirectional converters have the advantage of transferring energy into or out of cells. The multi-module full-bridge converter is a fully controlled converter that transfers the energy from the cell to the adjacent cell or from the pack to the weakest cell [67]. This method has the advantage of it can be scaled for higher power applications. Zero-current quasi-resonant or zero-voltage quasi-resonant converters can also be used to achieve cell to cell balancing [68]. The resonant circuits are tuned to achieve zero switching current and voltage which reduces the switching loss. Overall, the converter-based cell balancing methods achieve high efficiency and good balancing speed; however, they are more expensive and require a more complex control system [72], [73].

2.3.3 Comparison of balancing methods

Table 2.3 presents a comparison between the different cell balancing methods, including the number of active elements and advantages and disadvantages of each method [71]-[75]. Overall, dissipative balancing is a reliable, lower cost, and simpler cell balancing method. However, this approach is inefficient as the energy is released in resistors in the form of heat without being transferred to other cells.

The non-dissipative cell balancing methods can be fast and energy efficient as compared to the dissipative balancing methods. The comparison between the different non-dissipative balancing methods shows that there is no single method that is clearly the best, since each method has a different combination of cost, balancing speed, control complexity, and overall simplicity. Common converter-based cell balancing methods may be very promising in the future if the power conversion and control circuits can be optimized to reduce the size and cost sufficiently.

Table 2.3: A comparison of cell balancing methods in lithium-ion battery packs where *N* is number of cells in the pack

Method		**Active elements**	**Advantage**	**Disadvantage**
Dissipative balancing	**Fixed resistors** [74]	*N* resistors	Easy to implement, low cost	Low balancing speed, continuous heat dissipation and pack discharge
	Switched resistors [75]	*N* resistors, and *N* switches	high balancing speed, relatively lower loss	More cost, limited to low power due to need to dissipate loss
Capacitor-based balancing	**Double tiered switched capacitors** [57]	*N* capacitors, and 2*N* switches	Adequate balancing speed, modularity, simple control	High number of switches, high cost
	Single switched capacitor [58]	1 capacitor, 1 resistor, and *N*+5 switches	Fewer components, more efficient	Low balancing speed
Inductor-based balancing	**Single inductor** [59]	1 inductor, 2*N* switches, and 2*N* diodes	Satisfactory balancing speed, higher efficiency	Complex control, high cost
	Multi-inductor [60]	*N*-1 inductors, 2*N*-2 switches	Good balancing speed, less control complexity	High cost

Transformer -based balancing	**Multiple transformers** [61]	*N* transformers, 1 switch, and *N* diodes	Good modular design, good balancing speed	Very high cost, less efficient, large size
	Multi windings transformer [62]	1: *N* transformer, 1 switch, and *N* diodes	relatively compact	less efficient, limited number of cells
	Switched transformer [63]	1 transformer, *N*+6 switches, and 1 diode	Lower magnetic losses, relatively compact	High cost, complex control is needed
Common converter-based balancing	**Buck-boost converter** [64]	*N* converters	Good efficiency, satisfactory balancing speed.	Larger size, cost, complex control is needed
	Cuk converter [65]	*N*-1 converters	Good balancing speed, satisfactory efficiency	Complex control is needed, relatively large size
	Flyback converter [66]	1 converter, 2*N* switches, and 1 transformer	Fewer components, less complex control, fast balancing speed	Transformer needed
	Multi-module full-bridge converter [67]	*N* converters	Can be scaled to high power applications, good balancing speed	Large size, high cost, complex control is needed
	Quasi-resonant converter [68]	*N*-1 converter	Easy to implement, relatively higher efficiency	Higher cost, and size

2.4 Discussion and recommendations

A robust SOC estimation algorithm and BMS must handle the inconsistencies between cells which are inevitably present in a battery pack. Cells produced in the same batch may have capacity and resistance variation around 1%, and the method of connecting bus bars to cells may add a further 1% resistance variation. These inconsistencies may also be exacerbated due to thermal variations within the pack and aging processes, resulting in cell capacity and resistance varying by more than 10% at end of life. Cell sorting, advanced welding techniques, and improved bus bar and thermal design methods may be applied to reduce variation between cells throughout the life of the pack, reducing the challenge for pack SOC estimation algorithms and the need for higher power cell balancing.

SOC estimation algorithms, including measurement based, filters, observers, and machine learning methods, form the basis for pack SOC estimation methods. The most efficient pack SOC estimation methods estimate SOC at the fastest rate for a single reference or mean cell, and then estimate SOC or the SOC difference for the other cells with a simpler, slower updating algorithm. These methods require more engineering effort to develop than methods which simply apply a full performance SOC estimation algorithm to each cell. Importantly any pack SOC estimation method must handle the differing characteristics of each cell throughout the life of the pack. Mean cell and difference methods, which fit a cell difference model to the difference in voltage between a cell and the mean cell, show particular advantage in identifying small differences in SOC between cells and therefore have significant promise for improving the robustness of pack SOC estimation.

Cell balancing is necessary due to the differences in cell capacity which are present at the time of manufacture and throughout the life of the pack. Pack SOC estimation algorithms which estimate the SOC of each cell are needed so the BMS can command the balancing circuitry to equalize charge throughout the pack. Resistive balancing remains the most common method of balancing SOC, but there are also many different non-dissipative methods for transferring energy between greater and lesser charged cells. The non-dissipative methods may utilize capacitors, inductors, or transformer-based power converters, and could potentially transfer enough energy to a weak cell to extend the range of an electrified vehicle. More research is needed to quantify how non-dissipative methods could benefit battery pack performance over the life of the pack.

There are many opportunities to further improve the robustness of pack SOC estimation algorithms. Machine learning, for example, has been shown to offer significant potential for cell SOC estimation, but has not yet been optimized specifically for pack SOC estimation. Also, while studies have focused on developing pack SOC estimation methods which are more computationally efficient, these methods have not been deployed to a BMS and compared in a comprehensive manner, so it remains uncertain how much benefit reference cell or mean cell and difference models would have. Furthermore, it is difficult to fairly compare different pack SOC estimation algorithms since each study utilizes different datasets. A standardized dataset and evaluation method, which includes a realistic spread in cell parameters and has data to end of life, would help researchers compare algorithms more systematically.

2.5 Summary

There is significant variation in the capacity and resistance of cells in a battery pack due to cell manufacturing tolerances, welding or interconnect methods, and bus bar design in the battery pack. As the battery pack ages, differences between cells grow due to non-uniformities in the pack, including non-uniform temperature distribution. Battery pack SOC estimation algorithms must consider these differences between cells and report a pack SOC value which considers the most and least charged cells. A robust algorithm will estimate the SOC of each cell, typically by either estimating the SOC of each cell individually or through methods like mean cell and difference algorithms which estimate the SOC of one cell and the difference in SOC between the remaining cells. Pack SOC estimation algorithms should not only estimate SOC accurately but must also not be too computationally intensive or difficult to design and implement. The differences between cells also necessitates that some method of balancing be implemented. Resistive balancing, where energy of more charged cells is dissipated in a resistor, remains the most common method. Many different power electronics converter-based methods, which transfer energy between cells, have been investigated by researchers as well and will likely see increased adoption if they can be shown to increase pack life without adding significant size or expense to the BMS.

Chapter 3

Comparison of Microprocessor Time and Memory Use of Neural Network and Kalman Filter Battery State of Charge Estimation Algorithms

Lithium-ion battery (LIB) performance and cost have improved considerably over the last few decades, resulting in their acceptance for a variety of applications including electric vehicles (EVs) and grid-tied energy storage [76]. LIBs have many positive characteristics, including high power and energy density and a prolonged lifetime. LIBs, regardless of the chemistry of the cathode, have strong nonlinear, time dependent characteristics which make battery modeling and state estimation challenging. A battery management system (BMS) is responsible for monitoring and controlling LIBs to ensure reliable and safe battery pack operation [77]. One of the essential roles of the BMS is to accurately report the battery state of charge

(SOC). The SOC cannot be measured directly using sensors and instead must be estimated using algorithms.

Battery SOC is defined as the coulombs of charge stored in the cell divided by the cell's total charge capacity. SOC can be estimated directly from some physical measurements including coulomb counting [5], open circuit voltage (OCV) [7], and impedance [9]. However, these quantities require additional adaptive algorithms or external equipment to get an accurate estimate of SOC. Adaptive filters and observers typically utilize measured values and battery equivalent circuit models (ECM) to estimate the battery SOC. Examples of these algorithms include the family of Kalman filters (KFs) [11], particle filters [78], and sliding mode observers [79]. SOC estimation filters, especially Kalman filters, are widely used with LIB chemistries because they demonstrate a reasonable trade-off between accuracy and computational load [80]. However, exhaustive battery model parameterization processes are necessary to ensure the effectiveness of the aforementioned filters. Recently, machine learning algorithms have also been developed to model LIB input-output relationships regardless of the underlying physics or chemistry. Machine learning algorithms can accurately estimate the battery SOC by training machine learning models with measured data such as voltage, current, and temperature without the need for a battery ECM [80]. Recurrent and non-recurrent neural networks have been utilized in battery modeling and state estimation applications including state of health [81], battery voltage [82] and surface temperature [83].

It is not sufficient for an SOC estimation algorithm to have good accuracy though, it must also be not too computationally complex, so that it can be implemented in real-time on a microprocessor for a multitude of cells. Some studies have investigated the computational load of algorithms along with their accuracy [84]-[86]. The work in [84] compared the performance of an unscented Kalman filter (UKF) SOC estimator with two neural networks, including a non-linear autoregressive exogenous model (NARX) and a recurrent neural network (RNN) with a long short-term memory layer (LSTM). The LSTM model was shown to have the lowest root mean square error when tested using the UDDS and DST drive cycles. However, the NARX and UKF models showed a lower execution time when tested on Intel core-i7 9700 CPU. In order to achieve a better way to evaluate the computational load of the algorithms, a big O notation was introduced in [85] to determine the complexity of the different extended Kalman filter (EKF) SOC estimation algorithms. The results showed that the double forgotten factor adaptive extended Kalman filter has a slightly higher big O than traditional EKF. The work in [86] presented a processor execution time and memory use benchmarking for EKF, sliding mode and Leuenberger observers when deployed to a BMS microprocessor. The execution time and the random access memory (RAM) use were {4.8, 2.8, 3 milli seconds}, and {3.56, 2.3, 2.16 kB} for one estimator of EKF, sliding mode and Leuenberger observers respectively. The processor execution time and the memory use are considered the most practical and realistic factors to determine the computational load of the SOC algorithms and to ensure safely running of the algorithms in a real-time application.

In this chapter, four battery SOC estimation algorithm types are benchmarked and deployed to microprocessors. The algorithms include an EKF, an EKF with a least-squares filter (EKF-RLS), a feedforward neural network (FNN), and a recurrent neural network with an LSTM layer. A methodology for processor in the loop (PIL) testing of the algorithms is introduced and execution time and the memory use are measured for each algorithm. The algorithms are deployed to a low- and high-performance microprocessor, both of which are recommended for BMS use. The execution time and memory usage are measured for both a single algorithm and for multiple estimators of the algorithm running on the processor, as would be done when estimating SOC for multiple cells. In Section 3.1 the PIL methodology is introduced, in Section 3.2 the algorithms are described, and in Section 3.3 the performance benchmarking of the algorithms is presented.

3.1 Processor in the loop methodology

The processor in the loop testing in this study is performed using the MATLAB/Simulink environment, as shown in Figure 3-1.

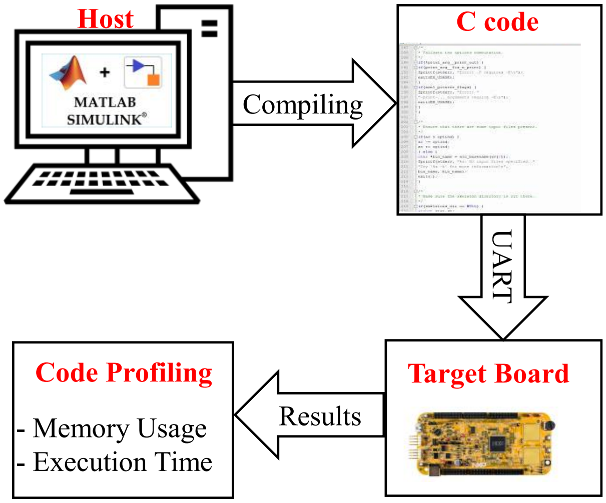


Figure 3-1: Description of the proposed PIL platform

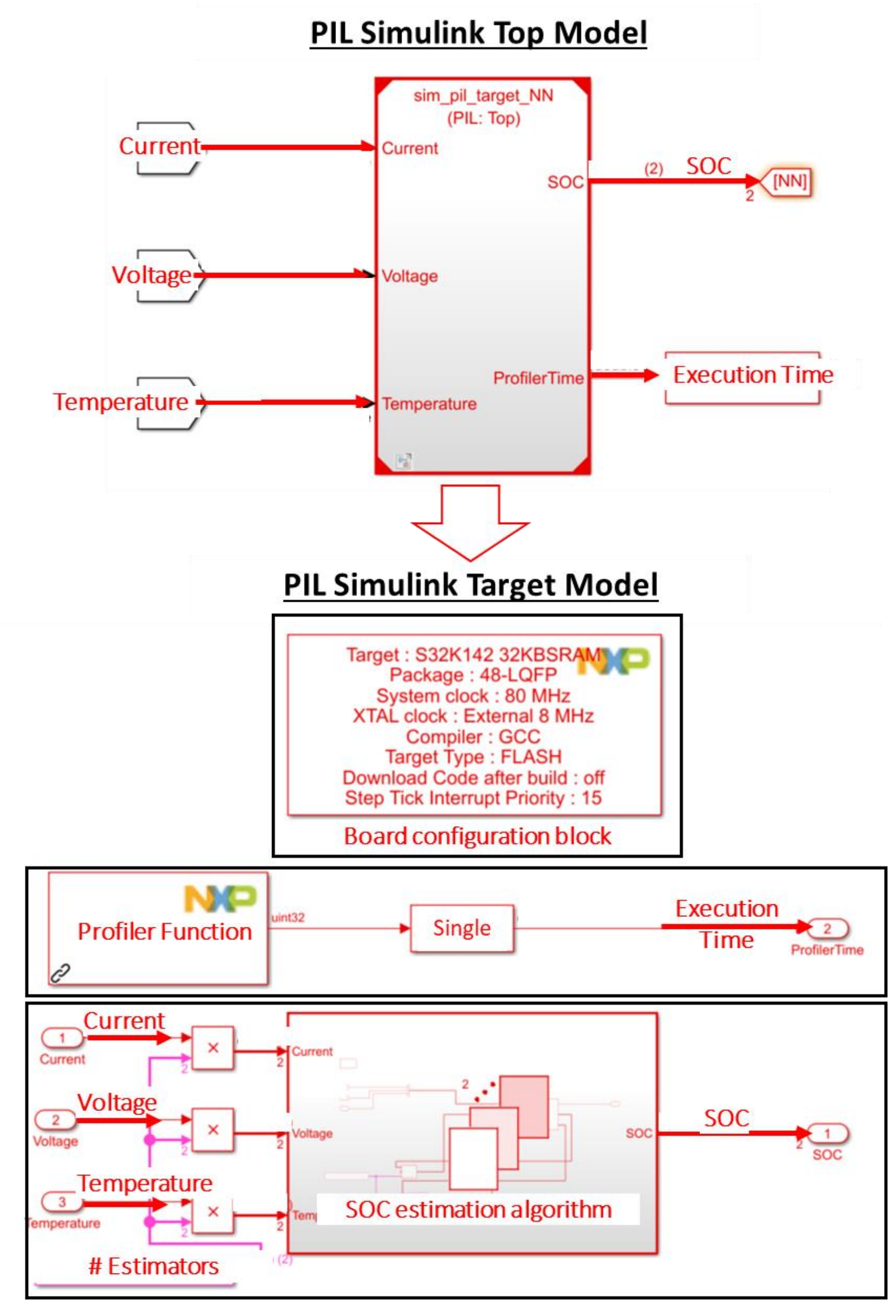
The SOC estimation algorithms are implemented in Simulink on a host computer. The microprocessor's evaluation boards (EVBs) are connected to the host computer using a universal asynchronous receiver-transmitter (UART). The Simulink models are compiled into C code using the MATLAB model-based design toolbox and deployed to the target board. A Simulink profiler block is utilized to measure the execution time and the memory use of the deployed algorithms. The models' SOC estimation output and the profiling results are sent back to Simulink via the UART once the code execution is complete. The details of the proposed PIL methodology are described in the following subsections.

3.1.1 Simulink model interface

A Simulink model is created which contains the SOC estimation model, the board configuration, and profiler blocks. This model is shown in Figure 3-2, which shows the PIL Simulink Top and Target Models. The model inputs are the measured battery voltage, current and temperature and the outputs are the code profiling values and the estimated SOC. The Target Model contains the SOC algorithm and is deployed to the microprocessor. When the Top Model is run in PIL mode, the Simulink Coder generates C source code for the Target Model to be deployed to the microprocessor. An open serial debug adaptor (SDA) is then used to deploy the generated C code from Simulink to the microprocessors using the host computer, which has an Intel core-i5 CPU and MATLAB 2020b installed. Then the results are sent back to the host computer using UART communication.

3.1.2 Microprocessor specifications

Two micro control units (MCUs) manufactured by NXP Semiconductors, which are from the S32K1 and S32K3 product families, are benchmarked in this study. The S32K1 family of MCUs utilizes 32-bit Arm Cortex-M4 processors, while the S32K3 family utilizes 32-bit Arm Cortex-M7 MCUs with single, dual, triple, or lockstep cores. The S32K142 processor, an 80MHz low-cost evaluation and development processor for general purpose automotive applications, is chosen from the S32K1 family and shown in Figure 3-3 [87]. The 160 MHz single core S32K344 MCU is selected from the S32K3 family and is recommended by NXP for battery management systems applications [87]. The specifications of both microprocessors are presented in Table 3.1.



Target : S32K142 32KBSRAM
Package : 48-LQFP
System clock : 80 MHz
XTAL clock : External 8 MHz
Compiler : GCC
Target Type : FLASH
Download Code after build : off
Step Tick Interrupt Priority : 15

Figure 3-2: Simulink top and target models description for S32K144 microprocessor

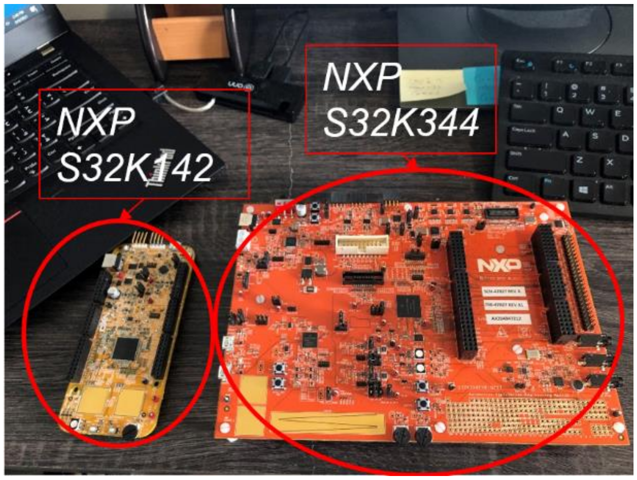


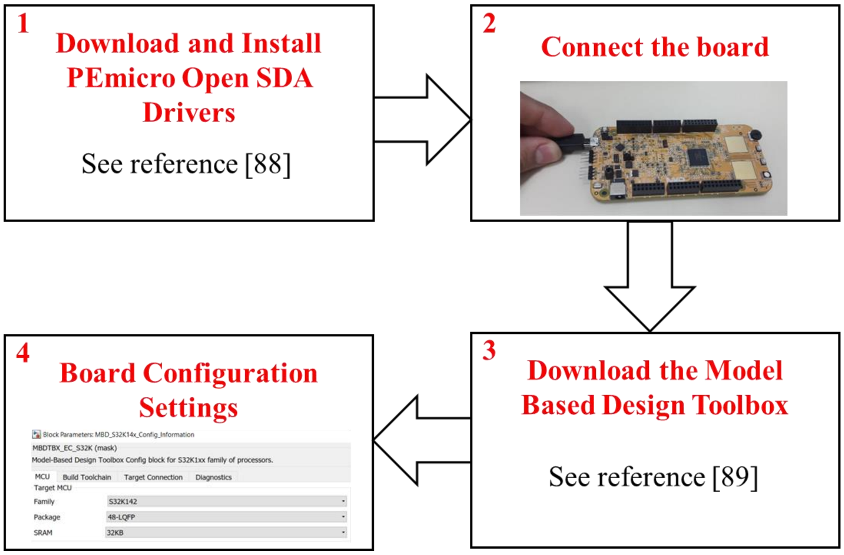
Figure 3-3: Two evaluation boards with the microprocessors used in the study

Table 3.1: Comparison between NXP 32K142 and S32K344 microprocessors

	NXP S32K142	NXP S32K344
Microprocessor Type	Arm Cortex-M4 Single Core	Arm Cortex-M7 Single Core
Frequency (MHz)	80	160
Flash Memory	256 kB	4 MB
RAM	32 kB	512 kB
Simulink Support	Yes	Yes
Production Stage	In Production	Pre-Production

3.1.3 Software setup and configuration steps

Figure 3-4 shows a summary of the software setup and configuration steps to install and run the proposed PIL models. The first step is to download and install PEmicro drivers for Open SDA communication [88]. Open SDA communication is essential to ensure safe deployment and debugging of the generated C code from Simulink Coder to the microprocessor. In the second step, the target board should be connected to the host computer using a USB/UART connection. If the target board is in its first use, flashing light with the three colors, including red, green, and blue will be noticed. Then, the MATLAB model-based design toolbox from NXP should be downloaded and installed [89]. This step also requires generating a user license from NXP by creating an account, including user and hard disc serial number information. The last step is to adjust the board configuration settings by choosing the microcontrollers' family, type of connection, and the communication port number. A step-by-step guide is provided by NXP for getting started with the NXP S324K142 in [90]. However, no training is yet available for the S32K344 as it is still in the preproduction stage.



Block Parameters: MBD_S32K14x_Config_Information

MBDTBX_EC_S32K (mask)

Model-Based Design Toolbox Config block for S32K1xx family of processors.

Target MCU

Family: S32K142

Package: 48-LQFP

SRAM: 32KB

Figure 3-4: A guide to software setup and configuration steps

3.2 SOC Estimation Algorithms

In this section, the four different SOC estimation algorithms which will be deployed to the MCUs are described. Two algorithms are based on Kalman filters, including an EKF and an EKF-RLS, both of which have a second order battery equivalent circuit model (2RC-ECM). The other two algorithms utilize neural networks, a non-recurrent FNN and a recurrent LSTM model. Table 3.2 lists the algorithms and the number of parameters for each. The number of parameters of EKF algorithms includes look-up tables containing the ECM and the covariance matrix values. The number of parameters is rather high because the algorithms are configured for temperatures ranging from -20 to 40 ºC. Besides, the number of parameters of the NN algorithms includes the weights and biases of the trained network.

Table 3.2: Description of the benchmarked SOC estimation algorithms

Algorithm	Acronym	Number of Parameters
Extended Kalman filter with 2RC-ECM	EKF	952
Extended Kalman filter with recursive least squares filter and 2RC-ECM	EKF-RLS	953
Feedforward neural network	FNN	522
Recurrent neural network with long short-term memory	LSTM	571

3.2.1 Extended Kalman filter

The Kalman filter is an optimal recursive data processing algorithm that works in a predictor-corrector fashion, and it is commonly used as an estimator for linear systems expressed in a state space form. The EKF is a special type of KF that is used for state estimation of non-linear systems. The filter states (x_k) are updated during the prediction stage while they are corrected during the measurement stage [11]. This is done by adjusting the value of the filter gain (*K*) as in (3.1)-(3.5).

3.2.1.1 Prediction stage

3.2.1.1.1 Predict the states ahead (a-priori)

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k) \tag{3.1}$$

3.2.1.1.2 Predict the error covariance ahead (a-priori)

$$P_{k+1|k} = AP_{k|k}A^T + Q_k \tag{3.2}$$

3.2.1.2 Correction stage

3.2.1.2.1 Kalman gain calculation

$$K_{k+1} = P_{k+1|k}C^T\left(CP_{k+1|k}C^T + R_{k+1}\right)^{-1} \tag{3.3}$$

3.2.1.2.2 Update state estimate with measurement (a-posteriori)

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - C\hat{x}_{k+1|k}) \tag{3.4}$$

3.2.1.2.3 Update error covariance matrix

$$P_{k+1|k+1} = (1 - KC)P_{k+1|k} \tag{3.5}$$

Where *P*, *Q*, and *R* are the parameter, process, and measurement covariance matrices, respectively. u_k is the input matrix at time *k* and the *A* and *C* are the state space matrices of the battery ECM.

The 2RC-ECM utilized in the study consists of an OCV in series with an internal resistance (R_o) and two RC parallel pairs with separate charge and discharge resistance as shown in Figure 3-5. The resistances and the OCV are functions of the SOC and the temperature. The ECM parameters are calculated using Hybrid Pulse Power Characterization (HPPC) test data. The ECM voltage and state of charge equations are expressed in discrete state space form [11] in (3.6) and (3.7) so they can be utilized in the EKF equations. The state variables (x_k) are SOC and the parallel branches voltages (V_1, V_2). The system input (u_k) is the battery current (I_k), and the system output (z_k) is the terminal voltage (V_k).

$$x_k = [SOC\ V_1\ V_2]; \quad u_k = [I_k]; \quad z_k = [V_k] \tag{3.6}$$

$$\begin{cases} A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-\Delta t/R_1 C_1} & 0 \\ 0 & 0 & e^{-\Delta t/R_2 C_2} \end{bmatrix} \\ B = \begin{bmatrix} \eta \Delta t / C_o \\ (1 - e^{-\Delta t/R_1 C_1}) R_1 \\ (1 - e^{-\Delta t/R_2 C_2}) R_2 \end{bmatrix} \\ C = [\delta OCV/\delta SOC \quad -1 \quad -1] \\ D = [-R_0] \end{cases} \tag{3.7}$$

Where η is the coulombic efficiency (assumed to be 1 for Li-ion batteries), Δt is the time step, and C_o is the cell nominal capacity. The parameters R_1, R_2, C_1, and C_2 are the resistances and capacitance of the parallel branches.

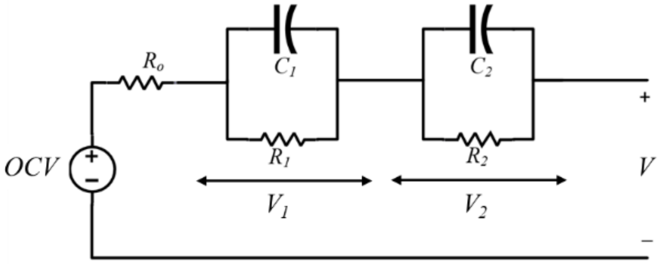


Figure 3-5: Second order battery equivalent circuit model

3.2.2 Extended Kalman filter with recursive least square errors filter

Recursive least squares (RLS) filter is a popular method to obtain the approximate parameter values of a static system by minimizing the sum of the squared errors between the measured data and the estimated values. Recursive techniques work well for estimating parameters which vary as system conditions change, such as battery parameters that vary with aging. In this work, the RLS is used to update ECM parameters so there is less error between the modeled and measured terminal voltage. In order to estimate the ECM parameters using the RLS algorithm, the discrete ECM voltage equations should be converted to the frequency domain by applying the bilateral z transform [91]. The RLS input matrix (Φ_k) contains two last time steps of the measured battery terminal voltage and current as in (3.8). The error (e_k) between the estimated and measured terminal voltage is updated as in (3.9). Hence, the parameters matrix (θ_k) is updated based on the error and the parameters' covariance (Y_k) as illustrated in (3.10)-(3.11). In the first time step of the RLS, the (θ_o) matrix is initialized with the unity vector. Once the RLS parameters are calculated, the ECM parameters are updated as illustrated in (3.12)-(3.13). These parameters are utilized to update the state space system equation in (3.7) which are input to the EKF.

3.2.2.1 Error calculation

$$\Phi_{k+1} = [1 \quad V_{k-1} \quad V_{k-2} \quad I_k \quad I_{k-1} \quad I_{k-2}] \tag{3.8}$$

$$e_{k+1} = V_{k+1} - \Phi_{k+1}^T \theta_k \tag{3.9}$$

3.2.2.2 Update the RLS covariance matrix

$$H_{k+1} = \frac{1}{\lambda} \left(Y_k - \frac{Y_k \Phi_{k+1} \Phi_{K+1}^T Y_k}{1 + \Phi_{K+1}^T Y_k \Phi_{k+1}}\right) \tag{3.10}$$

3.2.2.3 Update the parameters matrix

$$\theta_{k+1} = \theta_k + Y_{k+1} \Phi_{k+1} e_{k+1} \tag{3.11}$$

3.2.2.4 Obtain the updated ECM parameters

$$\theta_{k+1} = [(1 - a_1 a_2) OCV \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5] \tag{3.12}$$

$$\begin{cases} a_1 = \dfrac{R_1C_1}{R_1C_1+\Delta t} + \dfrac{R_2C_2}{R_2C_2+\Delta t} \\ a_2 = \dfrac{R_1C_1R_2C_2}{(R_1C_1+\Delta t)(R_2C_2+\Delta t)} \\ a_3 = -\left(R_0 + \dfrac{R_1\Delta t}{R_1C_1+\Delta t} + \dfrac{R_2C_2\Delta t}{R_2C_2+\Delta t}\right) \\ a_4 = \dfrac{R_0\Delta t}{R_1C_1+\Delta t} + \dfrac{R_0\Delta t}{R_2C_2+\Delta t} + \dfrac{R_1^2C_1\Delta t + R_2^2C_2\Delta t}{(R_1C_1+\Delta t)(R_2C_2+\Delta t)} \\ a_5 = \dfrac{-R_0R_1C_1R_2C_2}{(R_1C_1+\Delta t)(R_2C_2+\Delta t)} \end{cases} \tag{3.13}$$

Where λ is the RLS forgetting factor ($0<\lambda>1$) and is set to 0.8 in this study

3.2.3 Feedforward neural network

FNNs are a type of ANN where data is fed in a forward direction from the input to the output [92]. The simplest FNN has input and output layers where each layer has a certain number of neurons (*n*). Multiple intermediate layers (*l*) can be used in larger, more complex problems. The inputs fed to the first layer neurons are x_1 to x_{NI}, where *NI* is the number of inputs as shown in Figure 3-6. The inputs are multiplied with the respective weight, $\boldsymbol{w}_i^{l,n}$, summed, and added to the bias value ($b_i^{l,n}$) and output hidden state value $\boldsymbol{h}$. Each neuron also uses an activation function, $F^{l,n}$, to improve the performance of the network. Activation functions include the hyperbolic tangent, clipped rectified linear unit (CRELU), and leaky rectified linear unit (LRELU). Different activation functions are presented in [93] along with more details regarding the FNN. Eq. (3.16) is used to calculate the number of learnable parameters (LP_{FNN}) of a two-hidden layer FNN, like that used in this study. The learnable parameters include all weight and bias values and are a function of the number of inputs (*NI*) and number of neurons in the first and second hidden layer, N_{HL1} and N_{HL2}. During the backpropagation training process, the weights and biases of each layer are adjusted to minimize the summation of squared error (E) between the actual SOC and the estimated state of charge ($\widehat{SOC}_{\mathrm{k}}$) for D number of data points as in (3.14).

$$E = \sum_{k=1}^{D} \left(\widehat{SOC}_k - SOC_k\right)^2 \tag{3.14}$$

Unlike the LSTM, the FNN does not have a memory cell embedded in the model, and therefore does not encode information from the previous time steps. Several studies addressed improving the FNN performance by adding some filtered data to the inputs, such as voltage and current [93],[94]. In this study, the accuracy of the FNN is evaluated with one first-order low pass Butterworth filters applied to the voltage and current measurements. The Butterworth filter is selected due to its smooth frequency response roll-off and less phase delay. The frequency response ($H(f)$) of the filter is described in (3.15),

$$H(f) = \frac{1}{\sqrt{1 + \varepsilon^2 \left(\frac{f}{f_c}\right)^{2m}}} \tag{3.15}$$

where f_c is the chosen cut-off frequency, maximum passband gain ε is equal to 1, and the filter order m is equal to 1.

The network configuration used in this study consists of two hidden layers with 21, and 19 neurons and CRELU activation functions as shown in Figure 3-8 (a). The inputs of the network (Ψ_k) are battery current (I_k), terminal voltage (V_k), and battery surface temperature (T_k) and the output is estimated SOC ($\widehat{SOC}_k$). A Butterworth first order low pass filter with 5 mHz cutoff frequency is applied to the input voltage and current. The filtered inputs effectively add memory to the FNN over the bandwidth of the filter (roughly 200 seconds for a 5 mHz filter), increasing the accuracy of the network [94]. The network training process is repeated multiple times with different randomized initial parameters to help ensure the globally optimal training parameters are achieved.

3.2.4 Recurrent neural network with long short-term memory layer

The LSTM layer is commonly used to overcome the vanishing or exploding gradient issue which can occur during the backpropagation training process for some RNN types [112]. An LSTM layer comprises a memory cell ($\boldsymbol{C_k}$), a memory candidate ($\tilde{\mathbf{C}}_k$), an input gate ($\boldsymbol{V_k}$), an output gate ($\boldsymbol{U_k}$), and a forget gate ($\boldsymbol{F_k}$), as shown in Figure 3-7. The memory cell is utilized to save information from past time steps, and the three gates control the flow of information to and from the cell. The

LSTM layer utilizes the input data at the present time step ($\boldsymbol{\Psi}_{\boldsymbol{k}}$) and hidden state values from the preceding time step ($\boldsymbol{H}_{\boldsymbol{k-1}}$) to update the current hidden states ($\boldsymbol{H}_{\boldsymbol{k}}$). The number of learnable parameters (LP_{LSTM}) for an RNN with a single hidden LSTM layer can be calculated as a function of the number of hidden units (HU) and the number of inputs using (3.17) [93].

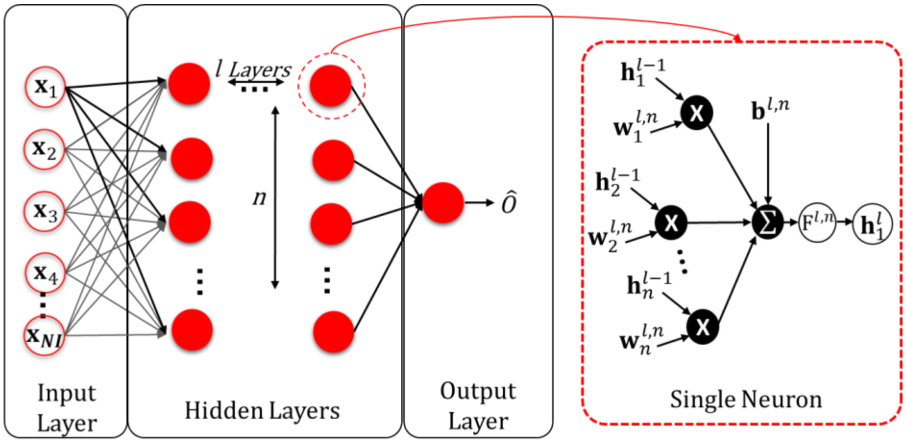


Figure 3-6: The structure of a typical multilayer feedforward neural network

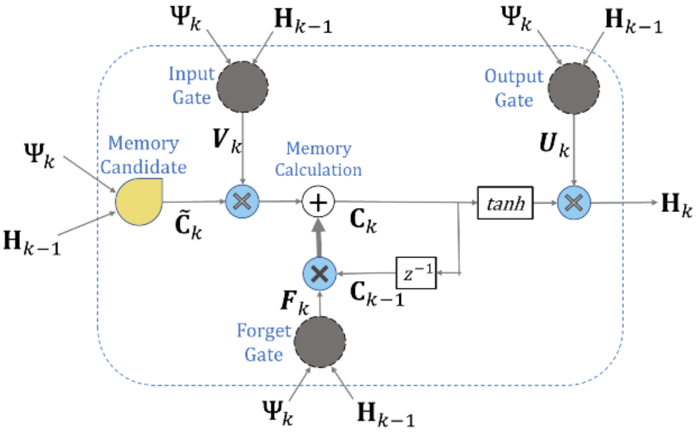
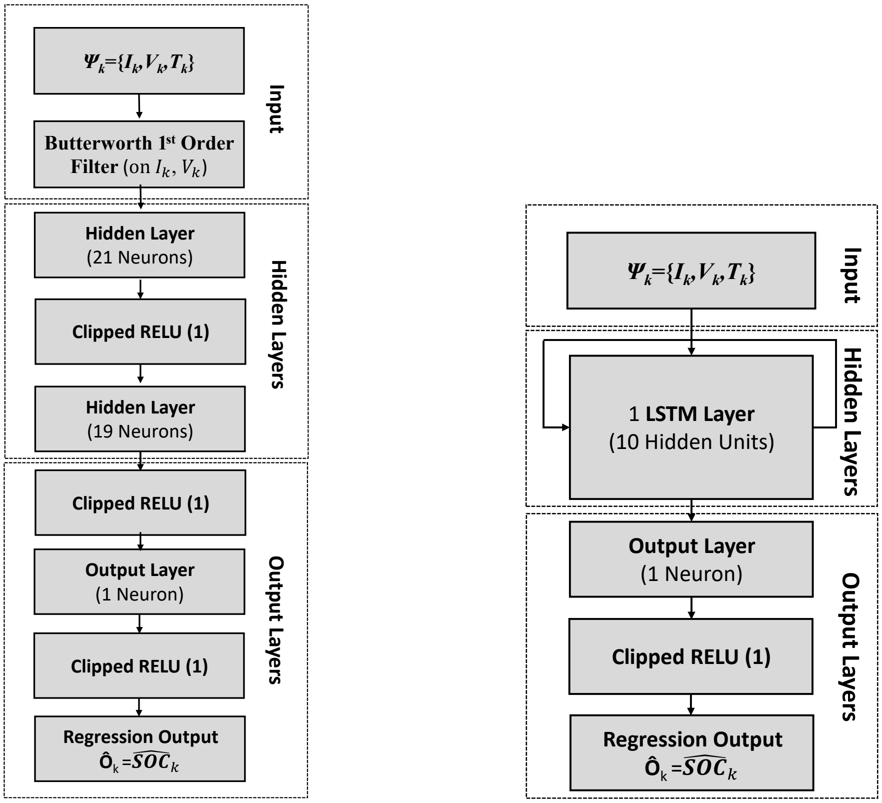


Figure 3-7: The structure of an LSTM unit

The LSTM SOC estimation model used in this study is composed of one LSTM layer with ten hidden units (*HU*) and an output layer with a fully connected layer with one neuron, as shown in Figure 3-8 (b). A CRELU activation function is used following the output layer. The inputs to the LSTM model are similar to the FNN including battery current, terminal voltage, and ambient temperature, and the output is the estimated SOC. The model parameters are proportionally adjusted based on the partial derivative of the loss function (*E*) during backpropagation process as in (3.14).



(a) Feedforward network (b) LSTM network

Figure 3-8: Feedforward and LSTM neural network SOC Estimation models

3.2.5 Determination of number of parameters for each algorithm

The ECM for the EKF models is fit to the HPPC test data at 5% SOC intervals from 0% to 20% and 80% to 100% SOC and at 10% SOC intervals from 20% to 80% SOC. As a result, the number of SOC breakpoints for each parameter of the ECM used in the EKF and the EKF-RLS algorithms is 14 points. Hence, the number of parameters at each temperature is 154 parameters (At each of the 14 SOC points there are 6 charge/discharge resistances, 4 charge/discharge time constants and one OCV). The parameter fitting process is repeated at each ambient temperature value, including -20, -10, 0, 10, 25, and 40 ºC. In addition, there are the R, and Q Kalman filter tuning matrices which include 14 points each and one forgotten factor value [82]. Hence, the total number of parameters used for the EKF and EKF-RLS are 952 and 953 parameters, respectively. The ECM parameters for the EKF are calculated at each time step by linearly interpolating between the model parameter values based on the temperature and SOC input values.

The FNN total number of learnable weight and bias parameters (LP_{FNN}) for the proposed model configuration, which are a function of the number of inputs, number of hidden layers, and the output layer, is 522 as in (3.16). The number of parameters for the LSTM model (LP_{LSTM}) is 571 (i.e., weight and bias values) as in (3.17),

$$LP_{FNN} = N_{HL1}\,(1 + NI + N_{HL2}) + 2N_{HL2} + 1 \tag{3.16}$$

$$LP_{LSTM} = 4(NI\ x\ HU + HU\ x\ HU + HU) + HU + 1 \tag{3.17}$$

where NI is the number of inputs to the model, N_{HL1}, N_{HL2} are the number of neurons of the first and the second layer, respectively, and HU is the number of hidden units of the LSTM layer.

3.3 Test setup and data acquisition

A lithium-ion pouch cell, as shown in Figure 3-9, is tested with a rated nominal capacity of 47.7 Ah and 3.7 V nominal voltage. The cell is the same as that used in the Chrysler Pacifica Hybrid PHEV. The cell was tested using twelve different automotive drive cycles and HPPC tests. The tests were performed with a Digatron Firing Circuits battery cycler with specifications presented in Table 3.3. The cell was tested using twelve different automotive drive cycles and HPPC tests presented in Table 3.4.

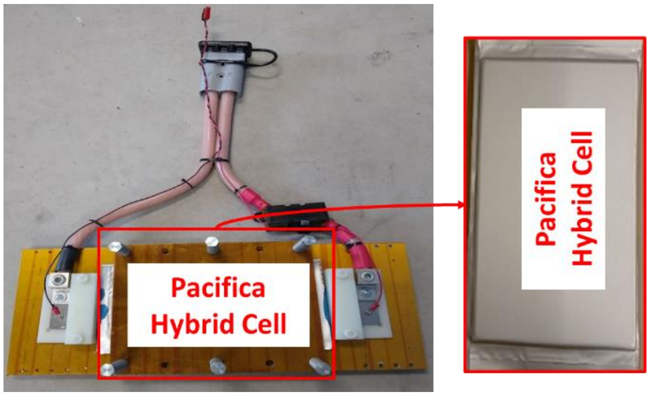


Figure 3-9: Pacifica Hybrid battery cell and test fixture

Table 3.3: Test setup specifications

Cycler Manufacturer	Digatron Firing Circuits
Test channel used	4 x 75 A, 0-5V
Voltage/Current accuracy	+/- 0.1% Full Scale
Data acquisition rate	1 Hz
Thermal chamber	Envirotronics M# SH16
Chamber size	8 cu. Ft.

Table 3.4: Description of tests performed on the battery for SOC estimation

Vehicle Modeled for Calculation of Drive Cycle Power Profiles	Pacifica Hybrid with 16 kWh 96s pouch cell pack
Temperatures Tested	-20, -10, 0, 10, 25, 40ºC
Characterization Test for Parametrizing ECM	HPPC with four current pulse magnitudes at each
Drive Cycles for Training of Neural Network Algorithms	Eight Mixed Cycles
Drive Cycles for Testing All Algorithms	UDDS, LA92, HWFET, US06

3.4 PIL methodology for validating the SOC algorithms: a case study

In this section the models are deployed to the two microprocessors and battery current, voltage, and temperature are the inputs, and the estimated SOC is the output as described in the previous sections. The SOC estimated by the four algorithms is presented for one drive cycle. The algorithms are also compared in terms of execution time and memory use on the two processors.

3.4.1 Demonstration of SOC estimated with the algorithms

In this subsection, the algorithms are deployed to one of the microprocessors and are tested using the US06 drive cycle at 40 ºC ambient temperature. The goal is only to demonstrate that algorithms are functioning properly, comparison of the algorithms in terms of accuracy is beyond the scope of this work.

Figure 3-10 shows measured and estimated time-domain SOC and the corresponding error, which is no greater than 5%, for the four algorithms. The US06 is selected because it is one of the most challenging and aggressive drive cycles for the algorithms. The EKF algorithms show better performance than the neural network algorithms. The RLS, by providing a better estimate of the ECM parameters, is shown to enhance the EKF SOC estimation accuracy. Overall, the four algorithms have reasonable SOC estimation accuracy, making them good candidates for comparison in terms of processor execution time and memory. With more effort in training and configuring the FNN and LSTM, their accuracy could likely be increased to be similar to the EKF and LSTM.

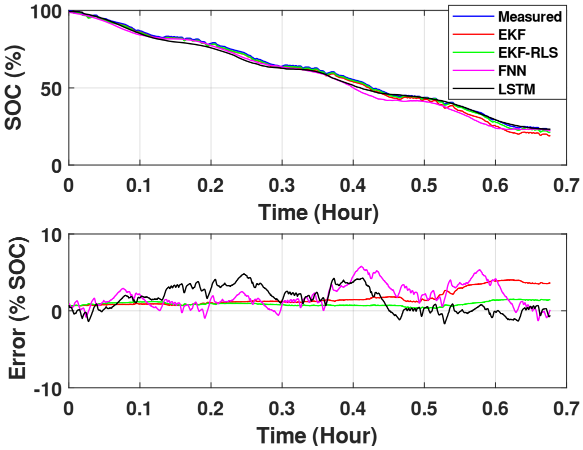


Figure 3-10: Estimated SOC and SOC estimation error in % SOC for each algorithm for the US06 drive cycle at 40 ºC

3.4.2 Algorithm’s execution time and memory use

This subsection presents the execution time and memory use to run one instance and multiple instances of each estimation algorithm on both processors. The execution time is the time required for the algorithm to execute once and is indicative of the computational complexity of the algorithm. The memory use quantifies the amount of flash memory and random access memory (RAM) used by each algorithm. Both execution time and memory use can restrict the number of estimators that can run simultaneously in a multi-cell series battery pack, where one estimator is required for each cell for which SOC is being estimated. It should be noted that the four models are built based on 32-bit single precision floating point variables.

Figure 3-11 shows the execution time for each microprocessor to run each algorithm for one timestep. The EKF, RLS-EKF, FNN, and LSTM took 1.7, 1.8, 0.8, and 4.2 ms respectively, to execute on the lower performance 80 MHz S32K142 processor. On the 160 MHz S32K344 processor execution time is reduced by 80% to 90%, to between 0.2 and 0.7 ms. This substantial decrease in execution time is due to the faster processor speed and also to the S32K344’s use of an Arm Cortex-M7 processor, which has a 42% higher CoreMark/MHz computational efficiency score than the S32K142’s Arm Cortex-M4 processor [96]-[97]. Adding the RLS to the EKF only increased the execution time of the EKF by a few microseconds. Overall, the FNN SOC estimation algorithm shows the lowest execution time while the LSTM algorithm shows the highest execution time, which is due to involving non-linear gate functions such as hyperbolic tangent and exponential. The LSTM can be configured with a lesser number of hidden units to improve its performance.

To determine the impact of executing multiple estimators of the algorithms, ten estimators of each estimation algorithm are deployed and executed simultaneously on the S32K344. The execution time of each algorithm is shown to increase linearly with the number of estimators running simultaneously. The algorithm execution time for a 100-cell pack (i.e., 100 simultaneous estimators), would therefore be 100 times the execution time for a single estimator, or 24 ms, 25 ms, 14 ms, and 71 ms for the EKF, RLS-EKF, FNN, and LSTM respectively. The execution time of the EKF, RLS-EKF, FNN and LSTM is fast enough that they could operate within a 100 ms (i.e., 10 Hz) update rate. The memory utilized by each algorithm may also be a limitation though, and is hence investigated next.

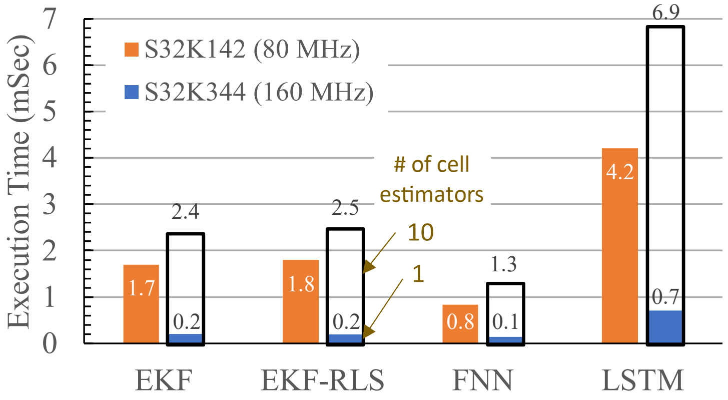


Figure 3-11: Algorithm execution time when deployed to S32K142 and S32K344 microprocessors

Figure 3-12 and Figure 3-13 show the flash and RAM memory use of each algorithm on both microprocessors. Flash memory use includes the code instructions size and model constant parameters' values, while the RAM includes the model variables that are calculated during code execution. The memory use only includes the SOC models, and excludes memory use for any of the supporting code. The EKF, EKF-RLS, FNN, and LSTM use 10, 12, 4, and 20 kB of flash memory respectively for one estimator of the algorithms on the S32K142 MCU. Each algorithm requires more flash memory on the S32K344 MCU, which is due to differences in each microprocessor’s architecture and compiler. The figure also shows that when running a hundred estimators on the S32K344 there is only a very slight increase in flash memory use, showing that the processor efficiently shares almost all of the code between each instance of the algorithm. For both processors the flash memory use is much less than the limit (256 kB for the S32K142 and 4

MB for the S32K344), showing the flash memory use is not a significant limitation for any of the algorithms.

The RAM memory, which is the fastest memory type and is rather limited in size, 32 kB for the S32K142 and 512 kB S32K344, is another potential limitation. Unlike for the flash memory, the EKF and the EKF-RLS algorithms have much higher RAM use than the neural network SOC estimation algorithms, as shown in Figure 3-13. To run 100 estimators on the S32K344, for example, the EKF requires 538 kB of RAM and the RLS-EKF requires 686 kB, while the FNN and LSTM only require19 and 32 kB or RAM respectively. The EKF algorithms require more than the 512 kB of memory, and therefore the compiled code could not even be deployed to the processor. Hence, unlike flash memory use, the RAM has a significant impact when deciding the number or type of estimators to use in a multi-cell battery pack.

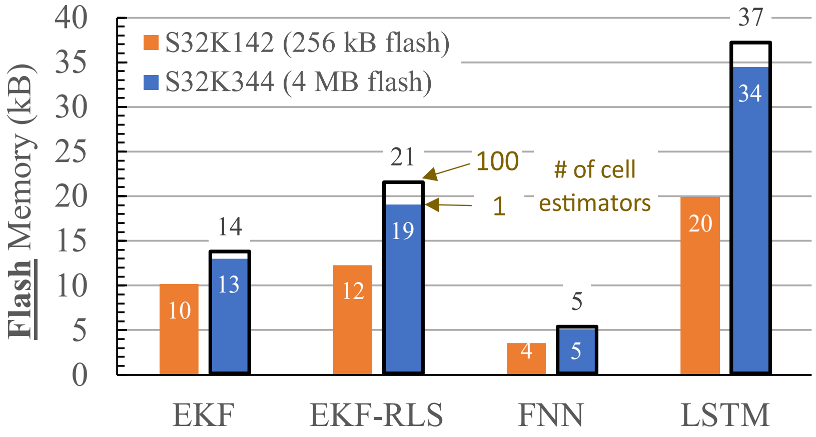


Figure 3-12: Algorithm flash memory use when deployed to S32K142 and S32K344 microprocessors

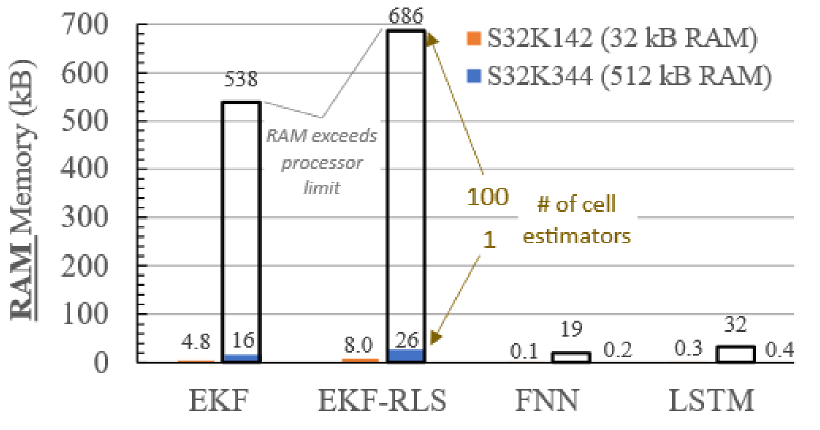


Figure 3-13: Algorithm RAM memory use when deployed to S32K142 and S32K344 microprocessors

In Summary, the traditional EKF, EKF-RLS SOC estimation algorithms showed reasonable execution time and flash memory use, but significantly higher RAM memory use. This could be sufficient for applications that require few estimators to run simultaneously on a low performance microprocessor such as the NXP S32K142. However, low- and high-performance microprocessors RAM may not be enough to run a hundred or more EKF and EKF-RLS simultaneous SOC estimators. It's also possible that RAM use could be reduced by implementing the code differently. The LSTM SOC estimator showed a considerably higher execution time and flash memory use, but lower RAM memory use. The computational complexity of the LSTM algorithm could be reduced by replacing some gate functions with less complicated approximated functions, or by reducing the number of LSTM hidden units. The FNN SOC estimation algorithm showed the lowest execution time and memory use and could even run two hundred estimators simultaneously at a reasonable speed without exceeding the microprocessor RAM or computational limits.

3.5 Summary

Four lithium-ion battery state of charge (SOC) estimation algorithms were presented and tested in this work on two different microprocessors. Two algorithms are based of the traditional Kalman filter family including an extended Kalman filter (EKF) and EKF with recursive least squares filter (EKF-RLS). The other two algorithms use feedforward (FNN) and recurrent with long short-term memory layer (LSTM) neural networks SOC estimation algorithms. The four algorithms demonstrated to have reasonable accuracy when tested on a microprocessor using a standard drive cycle. While the LSTM algorithm took the most time to execute, 0.71 ms on the higher power S32K344 processor, the FNN only had a lower execution time than the two EKF algorithms, 0.14 ms versus around 0.25 ms, showing that machine learning algorithms can have lower computational complexity than Kalman filters. The FNN and LSTM were also found to have less than 5% of the RAM use of the EKF algorithms, which required so much memory that 100 estimators could not be run simultaneously, an important limitation for high cell count battery packs. When evaluating an algorithm for an application it is therefore important to determine that its execution time and memory use do not exceed the capabilities of the BMS microprocessor for the desired number of cells and sample rate.

Chapter 4

Application of Deep Neural Networks for Lithium-Ion Battery Surface Temperature Estimation Under Driving and Fast Charge Conditions

One of the essential functions of the battery management system (BMS) is to monitor the performance of lithium-ion batteries (LIBs). Battery temperature is a critical factor that should be monitored to ensure safe and reliable battery operation [98]. The temperature of the battery also impacts the accuracy of battery state estimation, including state of charge (SOC) [77] and state of health [81]. Temperature monitoring of the battery pack is essential to avoid, in the worst-case, thermal runaway and destruction of the pack. Temperature sensors are widely used to measure the temperature of battery cells. However, it is often impractical to sense the temperature of every cell due to cost, number of sensors, and wiring complexity. Robust temperature estimation models can be used as an alternative to physical sensors or can perform as a redundant monitoring system for the existing sensors. In addition, these models can be applied to detect battery thermal and short circuit faults [99],[100] by monitoring the residual of the estimated and measured cell temperatures.

Several methods for estimating cell temperature have been investigated in prior research. These methods include electrochemical impedance spectroscopy (EIS) based methods [101]-[103], partial differential equation (PDE) thermal-based methods [104]-[107], and data-driven methods [83],[108]-[110]. In the EIS methods, one or several frequencies of AC current are injected into the battery. Then, based on the AC voltage response, the impedance of the battery is measured and is correlated with the temperature. The EIS method requires complex BMS hardware though, and cannot easily be performed during operation [101]. The PDE methods are multi-state models that capture the thermal behavior of the batteries with reasonable accuracy. These models emulate the generation and transfer of heat in the battery considering thermal boundary conditions. These methods are shown to be capable of modeling the temperature distribution of LIBs with lumped [104] or multidimensional 1D [105], 2D [106], and 3D [107] models. However, the more complex versions of these methods are not easily implemented in real-time on a BMS because they involve a high number of parameters and complex mathematical operations.

Recently, data-driven methods, which are based on machine learning, have been used to model nonlinear, time-dependent system behavior and are a promising alternative for battery temperature estimation. Machine learning-based models are trained with measured battery data, with inputs such as voltage and current and outputs such as SOC or temperature. Such models learn to mimic LIB behavior and have been used for terminal voltage modeling [82], SOC [77] and state of health (SOH) [81] estimation, and recently for temperature estimation [83],[108]-[110]. In [108], a recurrent neural network (RNN) model, with measured current, SOC, and ambient temperature as inputs, estimates battery cell voltage and surface temperature. The RNN model was shown to estimate the battery surface temperature during a portion of a low temperature drive cycle, where battery temperature was between 1 ºC and 8 ºC, with an error of less than 2 ºC. The work in [109] investigated different neural network (NN) model inputs to estimate battery surface temperature. The study concluded that feeding the model with measured voltage, current, and the prior time step of the predicted temperature led to predicting the battery surface temperature with a maximum estimation error (MAXE) of less than 3 ºC and less than 0.3 ºC root mean square error (RMSE). While this study does show very low error, it is not clear if unique data was used for training and testing the network, and only temperatures of 25 ºC or greater were investigated, so the proposed methodology may not achieve as good of results under more realistic scenarios. The work in [110] uses a radial basis function feedforward neural network (FNN) model with a Kalman filter to estimate the internal core temperature of a battery. The inputs to the model are measured surface temperature, voltage, and current, and a very low RMSE of less than 0.1 ºC and MAXE of less than 0.3 ºC is achieved for estimating internal core temperature. The reported results are excellent, but since measured surface temperature is an input to the model, it is not really comparable to algorithms which aim to estimate surface temperature like [108], [109], and the algorithms proposed in this study. In [83], a preliminary investigation of the application of non-recurrent and recurrent neural networks for battery surface temperature estimation was presented. Different combinations of network inputs were investigated, and around 3 ºC MAXE was achieved for automotive drive cycles.

While multiple studies have investigated neural networks for temperature estimation, these studies have only examined a limited range of operating conditions, and none have investigated the practicality of implementing the algorithms in a BMS microprocessor. Hence, the main contributions of this chapter are summarized as follows:

1) Development of two types of deep neural networks to predict LIB surface temperature including a feedforward neural network (FNN) and recurrent neural network (RNN) with long short-term memory (LSTM)

2) A methodology for improving the performance of the FNN by effectively adding memory of past time steps through the use of one or more filtered inputs

3) A comprehensive evaluation of the FNN and the LSTM surface temperature prediction models for realistic, challenging operating conditions, including low ambient temperature, high temperature rise, fast charging, and aging

4) A comprehensive benchmarking of the proposed models versus prior studies in the literature

5) Measurement of BMS microprocessor execution time and memory use for each algorithm using a Processor in the loop (PIL) platform.

The rest of the chapter is organized as follows: Section 4.1 provides an overview of deep neural networks. Section 4.2 presents the test setup and dataset. The structure selection for the surface temperature estimation models is detailed in Section 4.3. The temperature estimation for dynamic and fast charging conditions are presented in Section 4.4 and 4.5 respectively. The performance on a BMS microprocessor and testing the models' performance for an aged data set and versus other studies in the literature is presented in Sections 4.6 and 4.7. Finally, the conclusions are presented in Section 4.8.

4.1 Overview of deep neural networks

In recent years, ANNs have shown strong capability in extracting features from non-linear complex mathematical relationships. Deep neural networks (DNNs) are ANNs containing more than one hidden layer. There are several types of ANNs including FNNs, RNNs, and convolutional neural networks (CNNs). Each ANN type varies in structure, characteristics, and application. FNNs are commonly used in fault diagnosis of power systems [92] and pattern recognition [111]. RNNs were developed to model problems requiring memory so that past information is considered when calculating the current output. RNNs have shown strong performance in stock market forecasting [112], battery SOC estimation [77], and speech recognition [95]. For very large inputs, such as images, FNNs would have a large, complex structure for which determining the model weights and biases is challenging. CNNs, which utilize convolution on the inner layers, were developed to deal with image recognition and classification problems in an efficient and effective manner [113]. LIB modeling and state estimation are considered regression problems where FNNs and RNNs are commonly used. This section presents an overview of the FNN and the RNN- LSTM which are used in this study. The hyperparameters and framework for training the models are discussed as well.

The training of the proposed NNs and the corresponding hyper parameters are presented. The NN model parameters i.e., weights and biases, are updated iteratively during the backpropagation phase based on the loss (E) between the estimated and the actual output according to (4.1).

$$E = \sum_{k=1}^{k=D} \left(O_k - \hat{O}_k\right)^2 \tag{4.1}$$

where D is the total length of the input training data, O_k is the actual output which is the measured surface temperature of the battery at each time step k, and $\hat{O}_k$ is the estimated output which is battery surface temperature at time k in this study.

In this chapter, the initial learning rate for the training process, which specifies how much the weights and biases can be adjusted each epoch, is set to 0.01. The learning rate drop factor is 10%, and the patience is 1000 epochs, meaning that the learning rate is dropped by 10% each 1000 epochs. The training and testing datasets are resampled to 1 Hz and normalized so their values are between 0 and 1. All the training data is used as one minibatch for each update of the training parameters. The training process stops after the validation dataset accuracy does not improve for 300 consecutive epochs. Most trainings continued for around five thousand epochs, taking approximately three hours for the FNN and 13 hours for the LSTM models. The network's learnable parameters are randomly initialized at the start of each training. Each network configuration is trained between 3 and 20 times with different initial parameters each time, helping to ensure a global- or near-global optima solution is achieved. The network was created and trained with a similar script to that shared in [94] using MATLAB 2020b, the MATLAB Deep Learning Toolbox, and an Intel Core i7 CPU. Table 4.1 lists the different parameters used in the training process of the proposed NN models.

Table 4.1: Training parameters for temperature estimation models

Loss Function Optimizer	ADAM
Training Software	MATLAB 2020.b, MATLAB Deep Learning Toolbox
Training Platform	Intel Core i7 CPU
Initial Learning Rate	0.1
Learning Drop Factor	10%
Number of Mini batches	1
Validation Frequency	Each 10 epochs
Stopping Criteria	300 epochs with no loss improvement
# Training Repetitions	3 to 20

4.2 Test setup and dataset

Two batteries with different chemistries are used to train and validate the proposed models. The first battery is Panasonic 18650PF Li-ion battery cell with NCA chemistry and 2.9 amp-hours rated capacity, as shown in Figure 4-1. This battery is cycled at different dynamic conditions with a range of drive cycles at different ambient temperatures. One temperature sensor was placed on the outer surface to monitor the temperature of the cell. The second cell is a Kokam NMC pouch cell, which is tested at different fast charging rates in a liquid cooled module with three parallel (3P1S) cells as shown in Figure 4-2 (a). Between each cell is a 0.26 cm thick aluminum plate for cooling and fifteen T-type thermocouples are placed throughout the module. Seven of the thermocouple measurements are used in this study, including four placed on the face of the middle cell as shown in Figure 4-2 (b) and one on the coolant inlet and another on the outlet. A 200 W chiller with 15 L/min coolant flow rate is used to cool the module and is connected to copper cooling tubes attached to the edges of the fixture as illustrated in Figure 4-3. The thermocouple temperatures are logged in LabVIEW and measured with a National

Instruments NI-9213 thermocouple module. Table 4.2 presents the specifications of each cell [114]-[115] while the battery cycler and thermal chamber specifications are listed in Table 4.3.

Two types of data sets were collected for the Panasonic battery, one of which has -20, -10, 0, 10, and 25 ºC fixed ambient temperature and the other of which has varied ambient temperature, as described in Table 4.4. The battery is cycled using four standard drive cycles – UDDS, LA92, HWFET, and US06 - and five mix drive cycles which are made of a randomized mix of the power profiles for the standard drive cycles. The maximum charge/discharge currents at 25 ºC for UDDS, LA92, HWFET, and US06 are 4.2/7.5 A, 9.7/10.5 A, 5.5/5.5 A, and 7.5/20.8 A, respectively. The mixed cycles are referred to as Mix#1 through Mix#5. The varied ambient temperature tests start at either -20 or 10 ºC, and the temperature rise throughout the tests is achieved by turning the cooling system off while leaving the circulating fans on. The varied ambient temperature tests are used to test how the temperature estimation models perform when the thermal conditions around a battery are changing. The drive cycle power profiles are generated from a model of an electric truck with a 35kWh battery pack and are scaled for a single 18650PF cell.

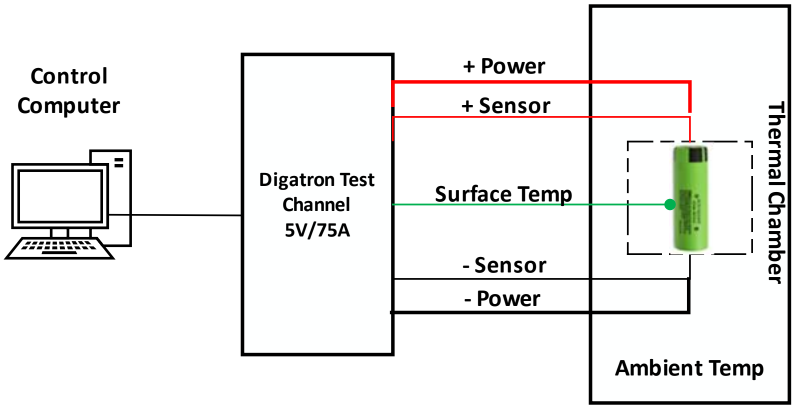
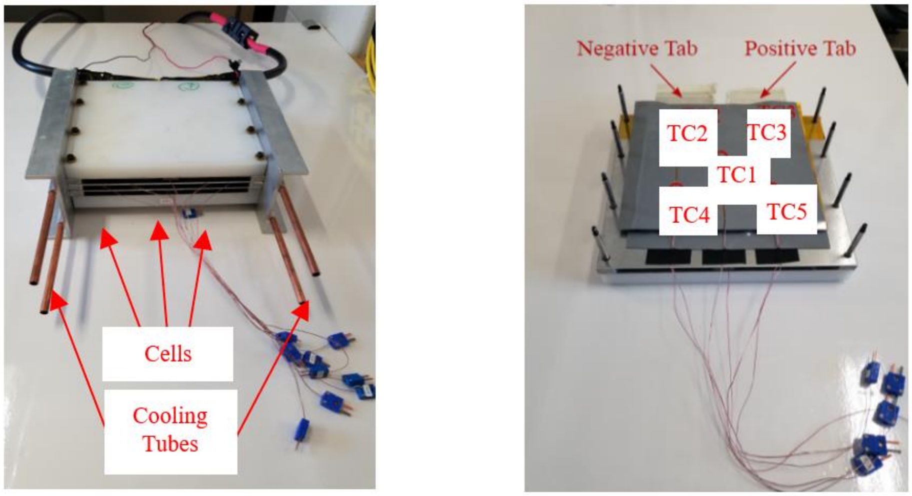


Figure 4-1: Panasonic test setup and data logging system



(a)Kokam Fixture (b) Thermocouples at middle cell

Figure 4-2: Kokam module fixture and thermocouples positions on the middle cell

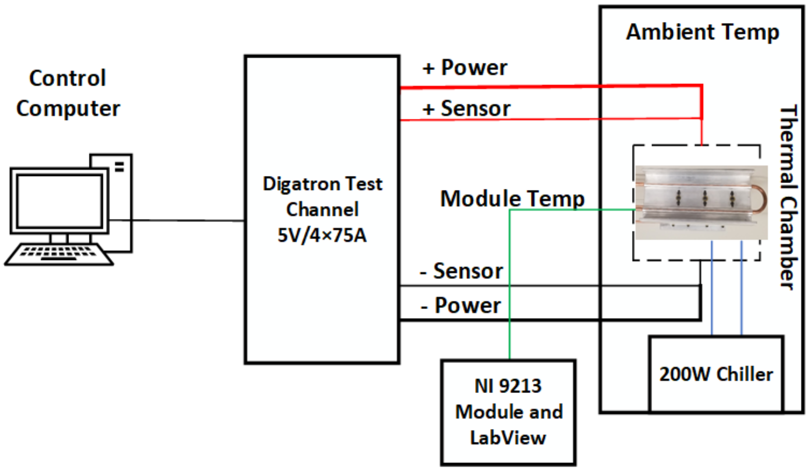


Figure 4-3: Kokam module test setup

Table 4.2: Battery specifications

Battery	Panasonic	Kokam
Model	18650PF	SLPB78216216H
Type	Cylindrical	Prismatic
Chemistry/Capacity	NCA (2.9 Ah)	NMC (31.0 Ah)
Nominal Resistance	36 mΩ	0.9 mΩ
Specific Power	1.7 kW/kg	2.8 kW/kg
Specific Energy	207 Wh/kg	160 Wh/kg

Table 4.3: Description of test setup and lab equipment

Cycler Manufacturer	Digatron Firing Circuits	
Test Channel	25 A, 0–18 V channel	4×75 A, 0–5 V channels
Cycler Data Acquisition	10 Hz	10 Hz
Voltage/Current Accuracy	±0.1% full scale	±0.1% full scale
Temperature Acquisition system	Embedded in cycler (±1.0 ºC)	National Instruments NI-9213 (±0.7 ºC)
Thermal chamber	Envirotronics Model SH16	Thermotron Model SE3000
Internal Volume	8 cu. Ft.	104 cu. Ft.
Temperature Regulation Accuracy	±0.5 ºC	±0.7 ºC

Table 4.4: Description of tests performed on batteries for temperature estimation models

Battery Datasets	Panasonic	3P1S Kokam module
Training	Mix 1 to 5 at -20 ºC, -10 ºC, 0 ºC, 10 ºC, 25 ºC	1C, 3C, 5C Fast Charge at 20 ºC
Validation (used to select the best trained models)	UDDS, LA92, HWFET, US06 at -20 ºC, -10ºC, 0 ºC, 10 ºC, 25 ºC "*Fixed Ambient Temperature*"	2C Fast Charge at 20ºC
Testing (done on the best trained models)	Mix 1-4 at -20 ºC, 10 ºC starting ambient temperatures "*Varied Ambient Temperature*"	4C Fast Charge at 20 ºC

The battery was fully charged at 25 ºC prior to each drive cycle test. Each drive cycle was repeated until the battery SOC reached between 10 and 30% depending on the temperature. This final SOC value was selected based on when the battery can no longer provide enough power to continue. The five mixed drive cycles at each fixed ambient temperature are used to train the proposed NN models and are labeled as *Training* data in Table 4.4. The four standard drive cycles with fixed ambient temperature are used for *Validation* and the mix drive cycles with varied ambient temperature are used for *Testing* the best trained NN models, and are referred to as the *Fixed Ambient Temperature* and *Varied Ambient Temperature* datasets respectively.

The Kokam module is charged at currents ranging from one to five C, which is 93 A to 465 A. For all the tests, the thermal chamber ambient temperature and the chiller coolant temperature are regulated to 20 ºC. Figure 4-4 shows the temperature collected from the thermocouples for a 5C charge. The thermocouple measurement which reaches the highest temperature, TC3, is used when training and testing the temperature estimation algorithms in this study. The 1C, 3C, 5C rate charge data is used for *Training* while the 2C charge is used for *Validation* and the 4C for *Testing*, as listed in Table 4.4. Figure 4-5 shows the battery temperature versus amp-hours discharged or charged for the training, validation, and testing datasets of both batteries. The figure shows that the training data covers the whole capacity range of the battery and that the NN model will need to in effect, interpolate between the trained temperature data in almost all test cases and extrapolate beyond 30 ºC for a few of the Panasonic tests.

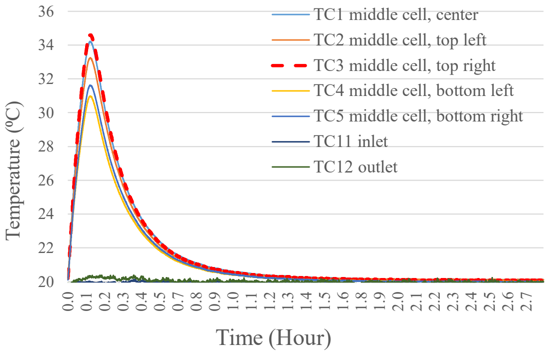
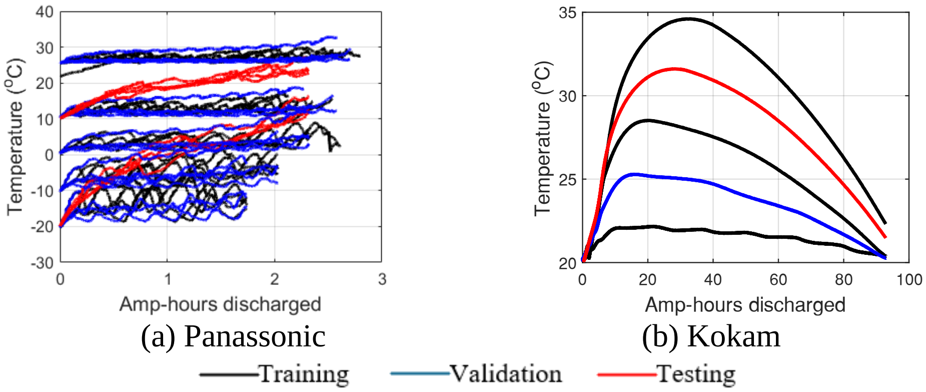


Figure 4-4: Kokam battery module temperature measurements for 5C charge



(a) Panassonic (b) Kokam

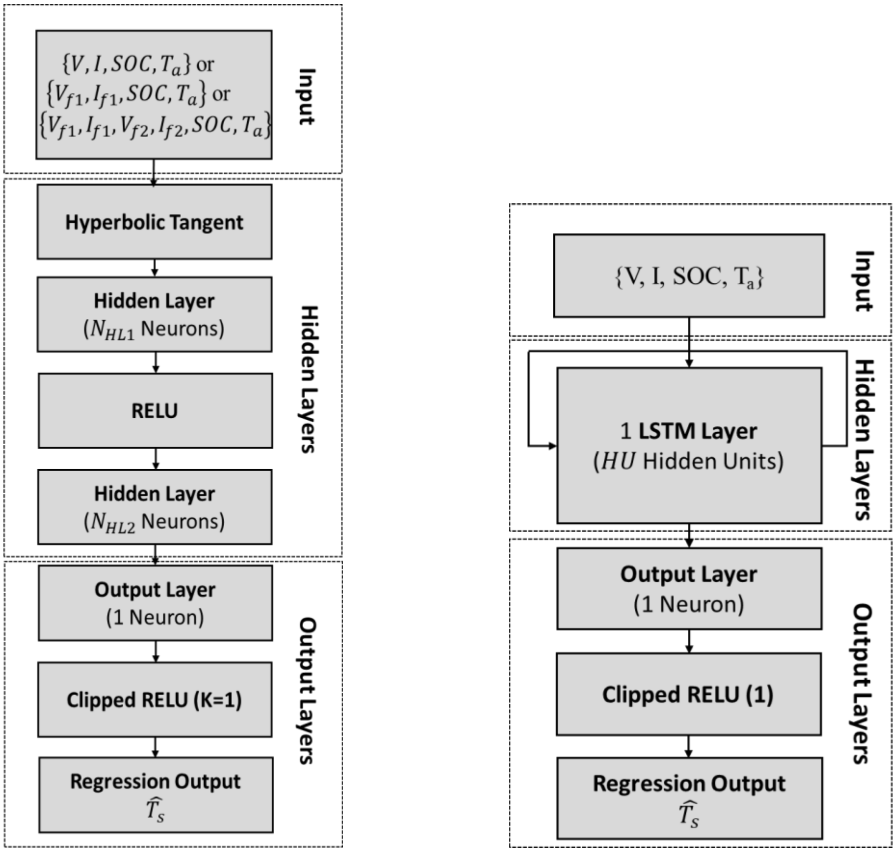
Figure 4-5: Cells’ temperatures for training, validation, and testing datasets

4.3 Neural network temperature estimation models structures and FNN filter frequency determination

In this section the structure of the investigated FNN and LSTM are presented. Additionally, a study is performed to determine the optimal filter frequency values used for the filtered voltage and current inputs to the FNN.

4.3.1 Determination of optimal corner frequency for filters on FNN input data

Figure 4-6 (a) shows the basic structure of the investigated FNN temperature estimation model, including inputs, layers, activation functions, and output. The inputs include SOC, ambient temperature (T_a), and voltage (V) and current (I) or filtered voltage (V_{f1}, V_{f2}) and current values $\left(I_{f1}, I_{f2}\right)$. Two hidden layers with fifty neurons each are used, which is expected to be a sufficient number of neurons and layers to achieve good accuracy [94]. The output layer consolidates all the neuron outputs to create a single output value for the network, estimated temperature ($\widehat{T}_s$). The investigation in this section focuses on determining the proper corner frequency for the optional filtered input values. A wide range of corner frequencies, for both one and two sets of filtered inputs, is considered.



(a) FNN model structure (b) LSTM model structure

Figure 4-6: Structure of FNN and LSTM battery surface temperature estimation models

First, a single set of filtered voltage and current inputs is investigated, such that inputs are $\{V_{f1}, I_{f1}, SOC, T_a\}$. Twenty-one networks are trained with the Panasonic cell *Training* data and filter frequencies varying from 0.01 mHz and to 100 mHz. These cover the range of frequencies over which the LIB electrical and thermal time constants are likely to occur. Average *Validation* dataset error is shown in Figure 4-7.

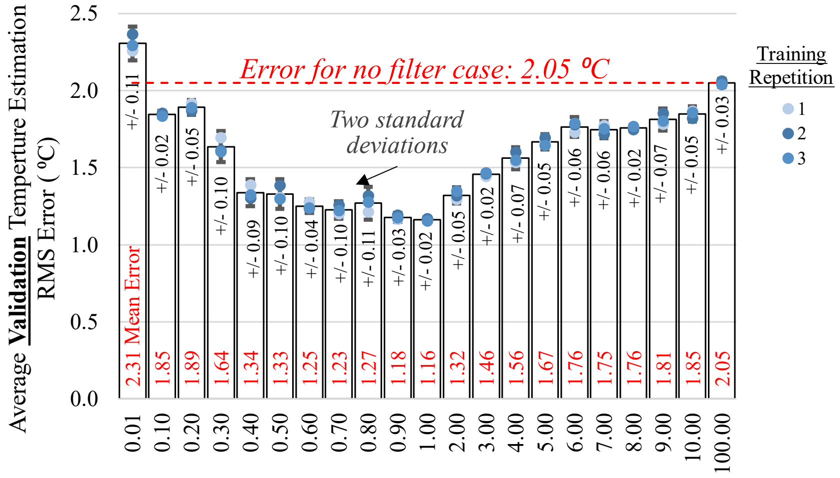


Figure 4-7: Panasonic validation data temperature estimation error versus filter frequency for FNN with a single set of filtered voltage and current inputs

The figure shows that the standard deviation of the three training repetitions is quite small, from 0.02 ºC to 0.11 ºC, verifying that three trainings are sufficient to ensure the best solution is achieved. With a filter frequency of 1 mHz the least error is achieved, just 1.16 ºC, which is 43% less than the 2.05 ºC error achieved for the FNN without filtered voltage and current as inputs, where inputs are instead $\{V, I, SOC, T_a\}$. The error is high for both low and high frequency filter values because with a high frequency filter the signal is similar to the unfiltered signal, and when the filter frequency is very low the output of the filter is essentially a small rising value and provides no meaningful additional information. The similar error for the studied high and low frequency filters also shows that a sufficiently wide range of filter values was investigated.

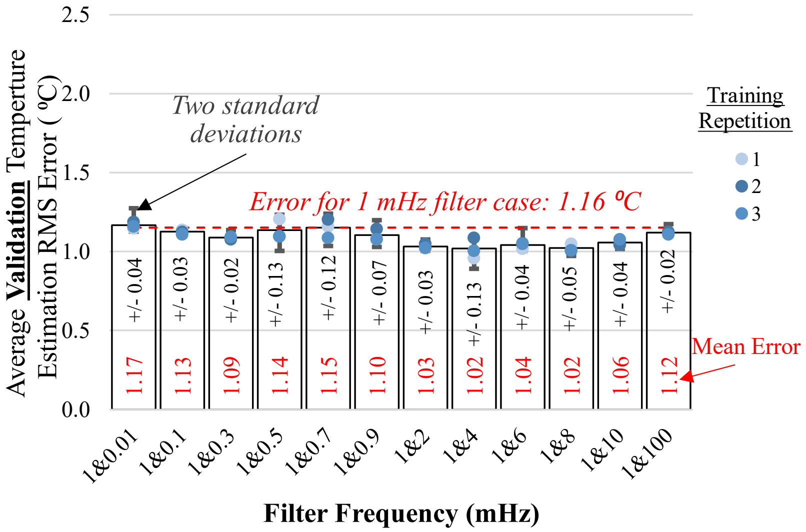


Figure 4-8: Panasonic validation data temperature estimation error versus second filter frequency for FNN with 1 mHz filters and a second set of filtered voltage and current inputs

Next, voltage and current filtered at 1 mHz, the filter value which achieved the best results with a single filter, are used along with voltage and current filtered at a second frequency, such that inputs are $\{V_{f1}, I_{f1}, V_{f2}, I_{f2}, SOC, T_a\}$. Networks are trained for twelve frequencies, ranging from 0.01 mHz to 100 Hz, and the results are presented in Figure 4-8. The addition of a second set of filtered values is shown to achieve a 0.14 ºC, 12% reduction in error for filter frequencies equal to 4 mHz and 8 mHz.

For the in-depth analysis in Sections 4.4 and 4.5, three FNN model structures are used based on the above analysis, the 4-input model with no filtered inputs, the 4-input model with 1 mHz filtered inputs, and the 6-input model with 1 and 4 mHz filtered inputs, as listed in Table 4.5. Importantly, the model filter frequencies are selected purely based on the analysis of the *Validation* dataset error above, and the *Testing* dataset which will be used in Section 4.4 is kept blinded from the selection of the optimal filter frequencies.

4.3.2 LSTM model structure

The LSTM has internal memory embedding information from past time steps, so it is not necessary to include filtered data as an input as was done for the FNN. Figure 4-6 (b) shows the LSTM temperature estimation model structure used in this study, including the layers and activation functions. Voltage, current, SOC, and ambient temperature are used as inputs. One hidden layer is used, and it includes 25 hidden units, where each hidden unit is the LSTM structure shown in Figure 3-7. Twenty-five hidden units are used so that the total number of learnable parameters, 3026 as listed in Table 4.5, is similar to the FNN.

In summary, the LSTM and the three FNN configurations from the previous section will be compared in the analysis for varied ambient temperature operation in section 4.4 and fast charging in section 4.5. The acronym, number of inputs and types, and number of parameters for each of the four models is presented in Table 4.5.

Table 4.5: Description of model configurations investigated in Sections 4.4 and 4.5

Model Structure	Acronym	# parameters	# Inputs	Type of inputs
FNN with no filters	FNN(N/F)	2851	4	Voltage, current, SOC, ambient temperature
LSTM	LSTM	3026	4	
FNN with 1 mHz filter	FNN(1mHz)	2851	4	Filtered voltage, filtered current, SOC, ambient temperature
FNN with 1 & 4 mHz filters	FNN(1&4mHz)	2951	6	

4.4 Temperature estimation for Panasonic cell electric vehicle drive cycles

In this section, the temperature estimation models with configurations discussed in Section 4.3 are trained with the Panasonic cell drive cycle *Training* data, as defined in Table 4.4. Each of the four models, three FNNs and one LSTM as listed in Table 4.5, are trained twenty times. The error of each of the trained models is calculated using the *Validation* dataset, which consists of the *Fixed Ambient Temperature* dataset. The trained model with the lowest validation error is then selected and evaluated using the *Testing* data, which consists of the *Varying Ambient Temperature* data which is more representative of conditions in an actual electric vehicle.

4.4.1 Training process with multiple training repetitions to select best trained model via validation data

Each of the four models is trained 20 times with unique initial parameters to ensure that the global optimum solution is reached and that a fair comparison between the model types is performed. The average of the RMS temperature estimation error for all the *Validation* cycles is calculated and is plotted for each model type and training repetition in Figure 4-9. The figure shows that it is important to perform multiple training repetitions, since the difference between the best and worst trained FNN with 1 and 4 mHz filters is quite significant, around 0.5 ºC. The difference is even greater for the LSTM, around 1.1 ºC. The trained model with the lowest *Validation* error – an error of 2.00 ºC for FNN (N/F), 1.12 ºC for FNN (1mHz), 0.96 ºC for FNN (1&4mHz), and 0.50 ºC for LSTM – is selected for evaluation in the next section. Notably, the error decreases substantially as more filtered inputs are added to the FNN, and the LSTM achieves about half the estimation error of the best FNN.

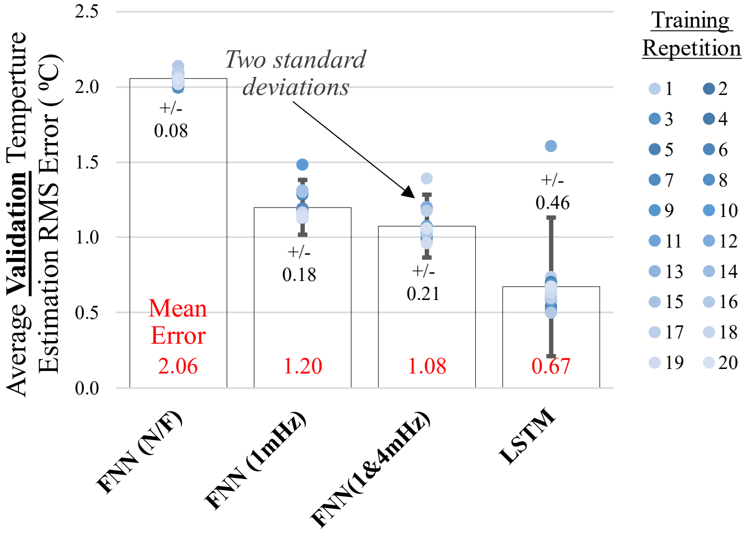
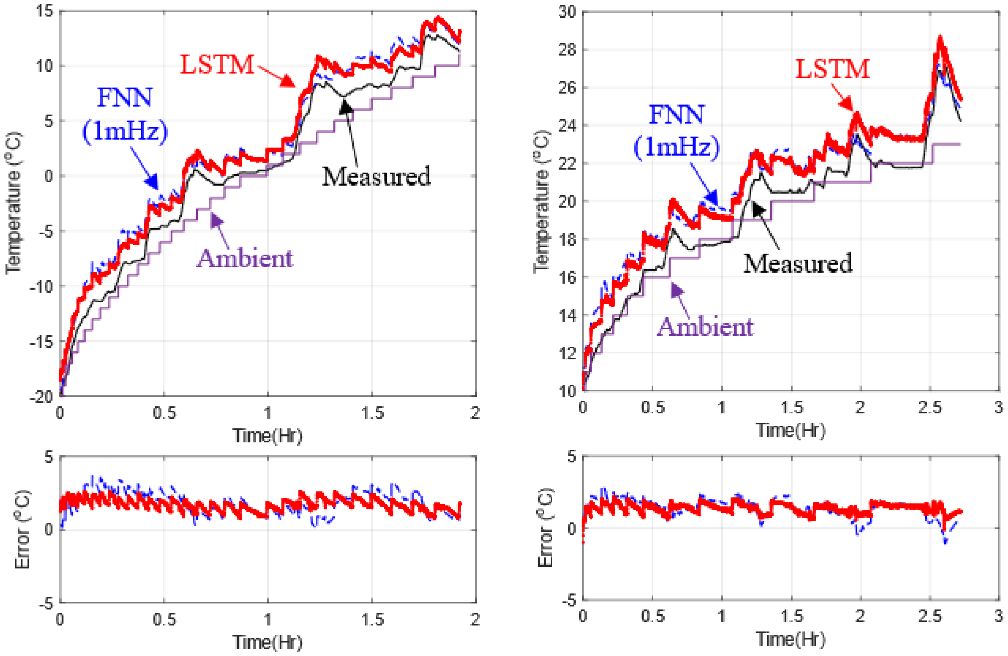


Figure 4-9: Temperature estimation error of each model for fixed ambient temperature Panasonic drive cycle validation data

4.4.2 Temperature estimation accuracy for best trained models and varying ambient temperature test data

In this subsection, the four best trained models are evaluated using the independent set of *Testing* data, the *Varying Ambient Temperature* drive cycle data as is described in Section 4.2 and Table 4.4. These drive cycles start at -20 ºC or 10 ºC and have a temperature rise of as much as 35 ºC. The temperature estimated by the FNN with 1 mHz filters and by the LSTM is plotted for two of the drive cycles in Figure 4-10. Even though the temperature rise is quite significant, the error is around 2.5 ºC or less for the LSTM and 4 ºC or less for the FNN for this Mix#1 cycle. The RMS error for each of the eight *Testing* cycles is then plotted in Figure 4-11, showing that the LSTM typically achieves around 5 to 10% lower error than the FNN. Besides, both models show a stable error for the testing drive cycles with the FNN average error of 1.8 ºC and the LSTM error of 1.6 ºC. The average RMS and maximum error of the eight *Testing* cycles for each model type are then plotted in Figure 4-12, showing that the LSTM still achieves the best accuracy, although by a smaller margin than was observed for the *Validation* data.

While the error for the *Testing* data is much higher than for the *Validation* data, nearly twice as high for the FNN and three times as high for the LSTM, the results demonstrate that the models can estimate temperature for data which is very different than the data which the model was trained with. This is an important characteristic for electric vehicle applications, where a wide range of operating conditions exist across different climates, terrain, and use cases. Overall, the accuracy of the temperature estimation is still quite reasonable, with less than 2 ºC RMS error for all but the FNN with no filters, less than 4.5 ºC max error for the FNNs with filters, and less than 4 ºC max error for the LSTM. Considering that low cost physical temperature sensors are often rated for several degrees Celsius of error, the machine learning algorithms appear to be an excellent alternative. Finally, one unexpected result was that the FNN error is slightly higher with two input filters, suggesting that when only marginal benefits of a second filter are observed during validation it may not be beneficial to include the additional filter.



(a) Mix#1 starting at -20 ºC (b) Mix#1 starting at 10 ºC

Figure 4-10: Temperature estimation for best FNN (1mHz) and LSTM models for Mix#1 varied temperature drive cycles

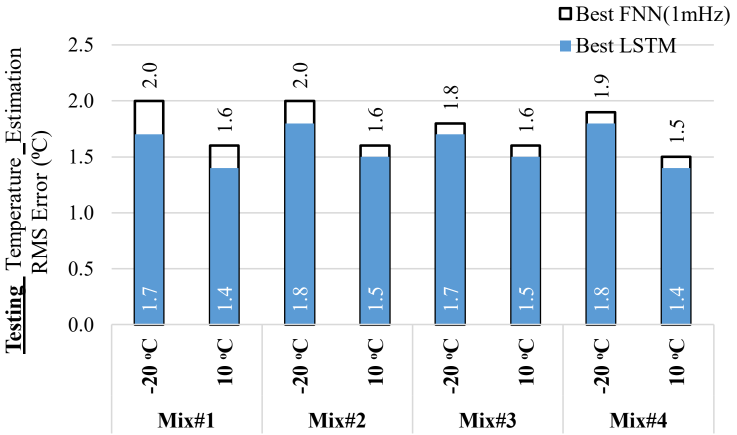


Figure 4-11: Temperature estimation error of FNN(1mHz) and LSTM models for each varied ambient temperature testing drive cycle

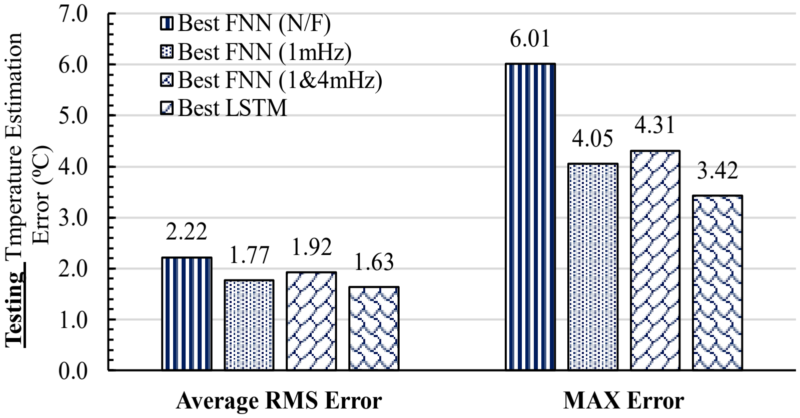


Figure 4-12: Temperature estimation error of each model for varied ambient temperatures drive cycle testing data

4.5 Temperature estimation for kokam cell fast charging

During fast charging, the battery is charged at a high rate for a short time, resulting in significant loss and temperature rise which is challenging to capture in battery thermal models. This section investigates and highlights the neural network model's capability to capture the temperature transients associated with fast charging. Like Section 4.4, each network is first trained multiple times, then the best trained networks are selected using the validation data, and finally, the error is evaluated for independent test data. Error is also presented as a function of the number of learnable parameters, giving insight into network sizing.

4.5.1 Training process with multiple training repetitions to select best trained model via validation data

Each model is trained twenty times with unique initial parameters to ensure the global optimum solution is reached, just like was done for the Panasonic drive cycle data. The model is trained with a 1C, 3C, a 5C fast charge, and a 2C fast charge is used for validation, as specified in Table 4.4. The validation error for each training repetition is shown in Figure 4-13. Each FNN training repetition has similar error while the LSTM error varies widely, again demonstrating the value of performing multiple training repetitions. The best trained model is selected from the training repetitions for use in the next subsection, with an error of 0.65 ºC for the FNN (N/F) model, 0.25 ºC for FNN (1mHz), 0.20 ºC for FNN (1&4mHz), and 0.29 ºC for LSTM. The FNN performance again improves as more filters are added, even though the filter frequencies were selected using an analysis of Panasonic cell data. This suggests that if the filter frequencies are close to the electrical and thermal time constants of the battery system, this should be sufficient to improve the performance of the FNN.

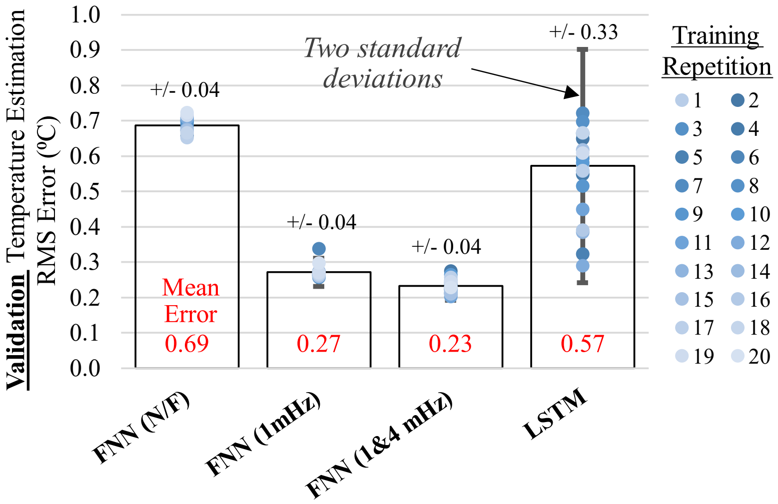
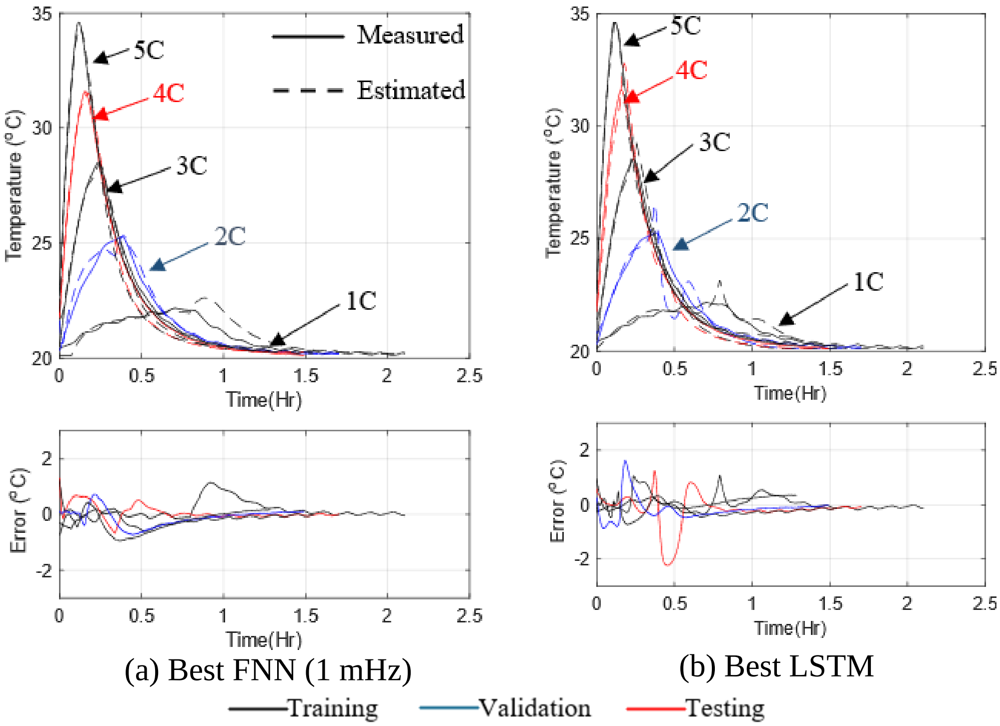


Figure 4-13: Temperature estimation error of each model for Kokam cell 2C fast charge validation data

4.5.2 Temperature estimation accuracy for best trained models and 4C fast charge test data

In this section, the best trained models as identified in the last section are evaluated for the 4C fast charge *Testing* data. Figure 4-14 shows the estimated temperature of the FNN (1mHz) and LSTM models for the 4C fast charge, as well as the other C-rate fast charges used for training and validation.



(a) Best FNN (1 mHz)

(b) Best LSTM

Figure 4-14: Temperature estimation for best FNN (1mHz) and LSTM models at 1C to 5C fast charging rates

The FNN error is exceptionally low for the 4C fast charge *Testing* case, never exceeding 1 °C, which is impressive considering that the 1C, 3C, and 5C fast charges used for training have a quite different temperature rise. The LSTM error is quite a bit higher, exceeding 2 °C at one point. Both models struggle to fit the 1C fast charge training data accurately, but this is likely due to fluctuations in the coolant and battery temperature caused by on and off cycling of the chiller when loss is very low which could be improved by adding the chiller status to the NN models inputs.

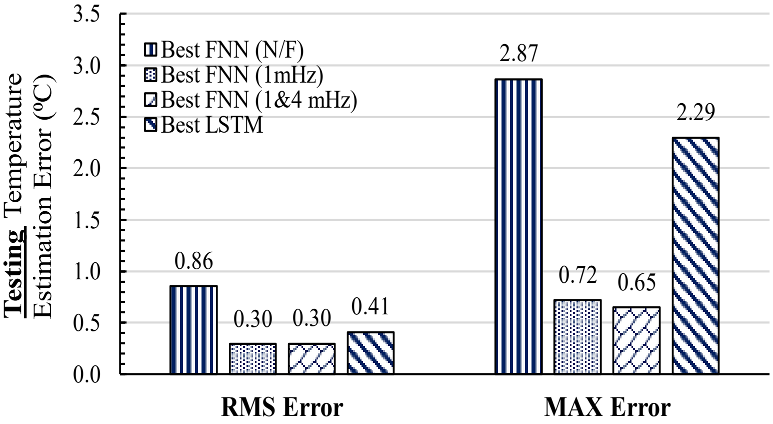


Figure 4-15: Temperature estimation error of each model for 4C fast charge testing data

Figure 4-15, shows the RMS error and maximum error of the four models for the 4C fast charge. Overall, the FNN with the 1 and 4 mHz filtered inputs has the lowest error, just 0.3 ºC RMS error and 0.65 ºC maximum error. This error is very low, which demonstrates that the FNN temperature estimation model could effectively be used along with a terminal voltage estimation model for predicting temperature for different fast charge profiles, enabling temperature to be regulated during fast charging with model predictive control for example. Overall, the fast charging dataset only contains constant current profiles make it less challenging for the NN to fit and contributing to the lower error of the NN compared to the error achieved for the drive cycles in Section 4.4. The LSTM, which performed best for the drive cycles tested, had a somewhat higher RMSE of 0.41 ºC and more than three times higher maximum error of 2.29 ºC. The fast charge dataset is smaller in size compared to the drive cycle, typically around 7% as long (29,341 data points for fast charge versus 434,797 data points for drive cycles). The LSTM performs worse in this case because the smaller amount of training data may not be sufficient to fully train the memory aspects of the LSTM. The fast charging voltage/current profiles are also less dynamic than the drive cycles which make the fast charge a less complex problem for a simple FNN to efficiently extract system features. There aren't many numerical studies comparing the FNN and LSTM, but the study in [116] showed that for modeling dynamic systems, the FNN shows better performance than a RNN when the inputs are noise-free. While the experimental data is essentially noise free due to the high accuracy of the battery cycler measurements, the drive cycle data could be considered similar to noise since it is highly dynamic while the output is a smooth signal. In this way, the research in [116] could be considered to support the LSTM performing better than the FNN for drive cycle data in Section 4.4.

4.5.3 Impact of number of learnable parameters on temperature estimation accuracy

In the prior sections, each temperature estimation model was configured with around 3,000 learnable parameters. To investigate whether this number of parameters is necessary to achieve the best accuracy, FNN and LSTM models with between around 150 and 10,000 learnable parameters are tested in this section.

Table 4.6: Number of learnable parameters for FNN (1mHz) and LSTM configurations investigated in Figure 4-16

FNN (1mHz)			**LSTM**	
N_{HL1}	N_{HL2}	LP_{FNN}	HU	LP_{LSTM}
10	10	171	4	149
25	25	801	15	1,216
50	50	2,851	25	3,026
75	75	6,151	35	5,236
100	100	10,701	50	11,051

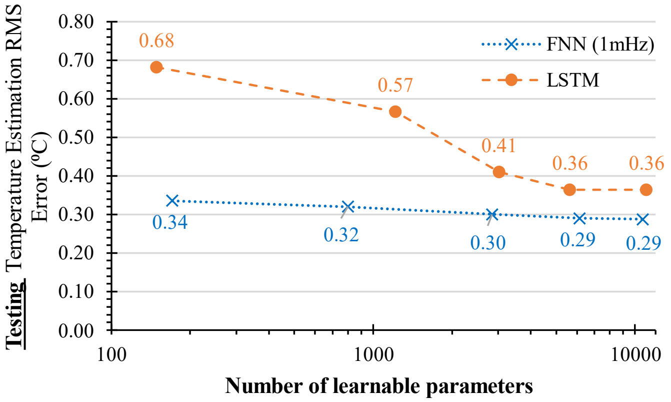


Figure 4-16: 4C fast charge testing temperature estimation RMS error of the FNN(1mHz) and LSTM models as a function of the number of learnable parameters

Similar to the previous cases, each model is trained 20 times and tested with the 4C fast charge data. The number of neurons, hidden units, and the corresponding number of learnable parameters of the LSTM and FNN model with 1mHz filters are presented in Table 4.6. The error versus number of learnable parameters is plotted in Figure 4-16 for the best model out of the 20 training repetitions, showing that both the FNN and LSTM error decreases as the learnable parameters increase. The FNN error decreases only slightly though, while the LSTM error decreases by about half as learnable parameters increase from 149 to 5,236. It is notable that even the smallest FNN, with just 171 learnable parameters, has better accuracy than any of the LSTM models. These results clearly show that the FNN is well suited for estimating the temperature of constant current, constant voltage fast charges, achieving higher accuracy than the LSTM with a much smaller network.

Overall, the FNNs with filtered data inputs are shown to be more accurate for the fast charging cases, while the LSTM is more accurate for the more complex dynamic, varied temperature drive cycles. In the next section, the execution time and memory use will be measured for each model when it is deployed to a BMS microprocessor.

4.5.4 Impact of SOC error on temperature estimation accuracy

The proposed models utilize SOC as one of the inputs, where SOC is calculated from current measured using an accurate battery cycler in the lab. However, SOC estimation algorithms implemented in the vehicle cannot be 100% accurate. Hence, a robustness test case with SOC error is investigated in this subsection. A ±5% offset in SOC is added to the 4C fast charge data. Temperature is estimated using the FNN with a 1mHz filter for the correct and offset SOC inputs, as shown in Figure 4-17. The RMSE obtained by the FNN (1mHz) for +5% and -5% offset in SOC are 0.35 ºC and 0.28 ºC respectively, and the error is 0.3 ºC with no SOC error. The error for the +5% SOC offset is just 15% greater than the error with the correct SOC, demonstrating that SOC error has a small impact on temperature estimation error. The maximum error obtained for both offset cases is still within 1 °C, which is quite acceptable given the fairly large SOC error. An accurate SOC estimation algorithm may achieve error of less than 2% for most cases, which would result in even less temperature estimation error.

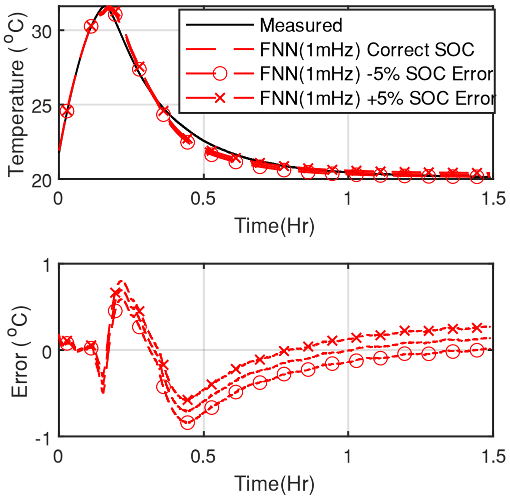


Figure 4-17: 4C fast charge testing temperature estimation RMS error for the FNN(1mHz) with ±5% offset error in the input SOC values

4.6 Microprocessor execution time and memory use

In this section, the three FNN models and the LSTM model with approximately 3,000 learnable parameters and an FNN model with 171 learnable parameters, as listed in Table 4.5 and Table 4.6, are deployed to a BMS microprocessor. The BMS microprocessor used in this study is an NXP S32K344 with a 32-bit 160 MHz Arm Cortex-M7 processor, as shown in Figure 4-18 [87]. The microprocessor has 4 MB of flash memory and 512 kB of random access memory (RAM). The

MATLAB/Simulink environment is used to generate C-code from the models which are implemented in Simulink. The C-code is generated using the MATLAB model-based design toolbox and is deployed to the microprocessor evaluation board using a universal asynchronous receiver-transmitter. The model execution time and memory use are then measured using a Simulink profile block. All the input signals are transferred from the host computer to the deployed algorithm executing on the microprocessor.

Figure 4-18 shows an example of the deployment of the FNN(1mHz) model with 2851 parameters. The resulting error for the FNN(1mHz) model is completely identical to the results attained with the algorithm running on a PC as presented in Figure 4-14 (a) and Figure 4-15. The microprocessor execution time to run one instance of each model is listed in Table 4.7. For the models with around 3,000 learnable parameters, the execution time is around 0.8 ms for the FNNs and 2.5 ms for the LSTM, indicating that despite the models having the same number of learnable parameters, the computational complexity of the LSTM, which includes nonlinear hyperbolic tangent and exponential gate functions, is higher. The results show that adding filtered inputs to the FNN results in a negligible execution time increase of a few microseconds. The execution time is much less for the FNN with 171 learnable parameters, just 0.09 ms, demonstrating that if slightly higher error is acceptable, a very computationally efficient algorithm can be used.

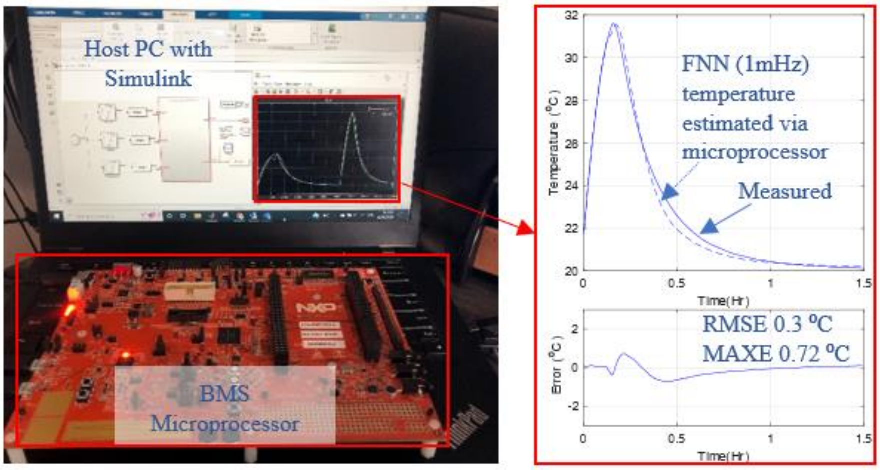


Figure 4-18: NXP S32K344 160 MHz BMS microprocessor with FNN(1mHz) deployment and testing for 4C fast charge case

Table 4.7: Model execution time and memory use for NXP S32K344 160 MHz BMS microprocessor

Model	# Parameters	Execution Time (μSec)	Flash (kB)	RAM (kB)
FNN(N/F)	2851	758	8.19	0.39
FNN(1mHz)	2851	763	8.62	0.41
	171	93	3.82	0.16
FNN(1&4mHz)	2951	771	9.00	0.48
LSTM	3026	2490	52.53	1.02

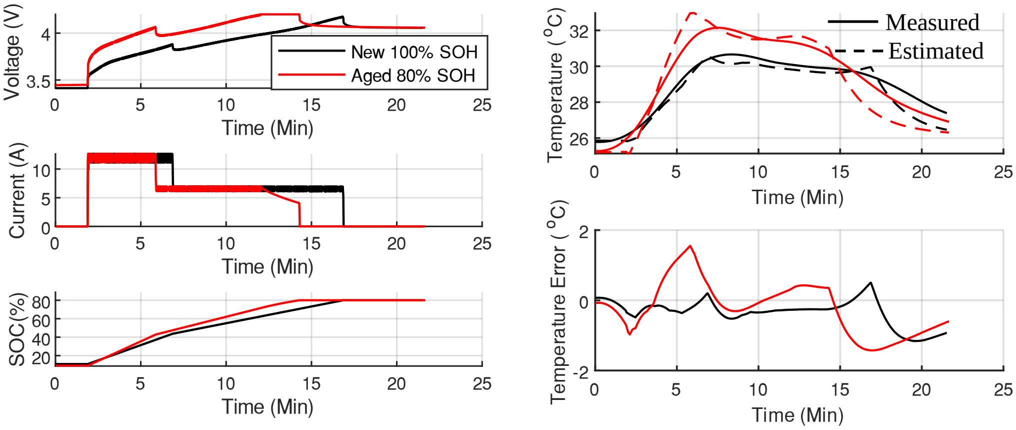
To implement one estimator of each algorithm on the microprocessor, between 4 kB and 9 kB of flash memory are needed for the FNNs while 53 kB is needed for the LSTM. This is just a fraction of the 4 Mb of flash memory available on the processor, demonstrating that the flash memory use of the proposed models is not a significant limitation to running them on a BMS. The models only require 0.2 to 1 kB of RAM, which again is just a small fraction of the 512 kB of RAM available on the processor and indicates that these algorithms could easily be implemented in a BMS.

In summary, execution time may be more of a limitation than memory use. Running the models one hundred times, as would be needed to estimate the temperature of one hundred cells, would consume between 8 and 25% of the processor time for a 1 Hz update rate and the 3,000 parameter models. Fortunately, for the fast charging case at least, a significantly smaller FNN model can be used which would reduce the overall execution time substantially, requiring just 0.8% of processor time to run 100 models each second. The smaller FNN could also easily be run a low-cost NXP S32K1 series microprocessor, which was observed in [117] to require about a factor of eight greater execution time compared to the NXP S32K3.

4.7 Aging study and models benchmarking versus studies in the literature

In the previous test cases, the proposed models were tested using experimental data from cells at the beginning of life (BOL). However, in any realistic scenario, the battery will experience aging while deployed in the vehicle. Therefore, in this test case, one of the proposed temperature estimation models is trained using charge profiles from a battery at BOL and is tested both at BOL and under aged conditions. Testing the neural network under aged conditions shows if the correlation between input voltage, current, and SOC and the temperature rise is consistent throughout the life of the battery. An FNN (1mHz) neural network is used, and is trained with experimental data from a 3 Ah capacity Samsung INR21700-30T cylindrical cell. The cell was subjected to an aging test performed at 25 ºC over six months until the cell reached 80% state of health (SOH). For the aging tests, the cell is discharged with drive cycle power profiles and then boost charged (4C rate followed by 2C rate).

Figure 4-19 (a) shows the measured voltage, current, and SOC during the boost charge for the cell at 100% and 80% SOH conditions. The measured temperature and the temperature estimated by the FNN(1mHz) model, and the corresponding estimation error is shown in Figure 4-19 (b). The FNN(1mHz) model error is quite low at 100% SOH, just 0.5 ºC RMS and 1.15 ºC max, and it only increases slightly at 80% SOH to 0.8 ºC RMS and 1.56 ºC max. This demonstrates that the model, which was only trained on new data, successfully learned the correlation between terminal voltage, current, SOC, and temperature. The model essentially translated the higher terminal voltage of the more resistive aged cell to higher temperature rise. To improve accuracy throughout the life of the cell, the proposed model can always be updated or calibrated either onboard in the cloud using data collected from a fleet of vehicles.



(a) Cell voltage, current, and SOC (b) Cell measured versus estimated temperatures

Figure 4-19: Samsung cell voltage, current, SOC, and temperature at 100% and 80% SOH conditions

The proposed models are also compared in Table 4.8 to other machine learning algorithms presented in the literature, including an LSTM, ANN, and FNN. The models are compared in terms of algorithm type, ambient temperature the algorithms are applied at, largest temperature rise case, and error. The maximum error of these models varies from 1.5 to 7 °C, which is greater than the 0.7 to 4 °C maximum error observed for the proposed models. Considering that the temperature rise reached up to 35 °C, which is twice as high as the studies in the literature. Besides, the proposed model showed around 35% increase in estimating the temperature of an aged battery as compared to [109], where the model showed a 130% higher error for an aged battery. The FNN in our study has the advantage of decoding some memory information than in [118] by adding some selected filter frequencies. Besides, the LSTM model with voltage as an input in this study shows better performance than the LSTM in [108] with only current, SOC, and ambient temperature. The table also shows the superiority of the proposed study over other studies due to including a variety of dynamic and fast charge conditions which challenge any modeling approach.

Table 4.8: Comparison of temperature estimation models with prior research

Model [Reference]	Li-Ion Battery {Capacity}	Testing Dataset {Studied ambient temperatures (°C)}	Maximum temperature rise[1](°C)	Error[1] (°C) {RMSE, MAXE}
LSTM [108]	N/A	Drive cycle {1 °C}	7	{N/A, 2}
ANN [109]	N/A {1.8 Ah}	Constant current constant voltage charge {25 °C}	≈17	New cell {0.3, 3} aged cell {0.15, 7}
FNN [118]	Sony {2.1 Ah}	Constant pulsating charge and discharges {5, 25, 45 °C}	13	{N/A, 1.5}
FNN, LSTM [Proposed]	Panasonic {2.9 Ah}	Drive cycles {-20 to 40 °C}	35	FNN {2, 4.05} LSTM {1.8, 3.42}
	Kokam {31 Ah}	4C Fast charge {20 °C}	13	FNN {0.3, 0.72} LSTM {0.41, 2.29}
	Samsung {3 Ah}	Fast charge {25 °C}	8	100% SOH {0.5, 1.15} 80% SOH {0.8, 1.56}

[1] Data are for most challenging cases

4.8 Summary

In this chapter, two deep neural network modeling approaches were proposed to predict the surface temperature of lithium-ion batteries. The first model type is based on a feedforward neural network (FNN) enhanced with external filters, while the second model is based on a recurrent neural network (RNN) with long short-term memory (LSTM). These models were trained and tested at a range of driving, charging and health conditions, with up to 15 ºC of temperature rise and 450 A of current for the fast charge tests. The models were also deployed to an NXP S32K344 BMS microprocessor to benchmark the models in terms of execution time and memory use. Including filtered voltage and current inputs to the FNN models considerably improved their performance, resulting in error close to or better than for the LSTM. The proposed FNN with 1 mHz filter model shown to be capable of predicting the temperature with a maximum error of no more than 4.5 ºC for challenging, low temperature drive cycles and no more than 0.3 ºC for 4C rate fast charges. Besides, the proposed FNN model showed around 35% increase in estimating the temperature of an aged battery than the beginning of life error. When running the models with around 3,000 learnable parameters on the BMS microprocessor, the FNNs required about 1/3 the execution time of the LSTM, showing that the LSTM is much more computationally complex than the FNN. The results also showed that both flash and RAM memory use of the FNN is much lower than the LSTM.

Overall, the results show that machine learning algorithms are very effective at learning the relationship between battery temperature and measured terminal parameters, reducing the need to create complex battery loss and thermal models. The machine learning algorithms can also easily be implemented to a BMS microprocessor, and do not require excessive execution time or memory.

Chapter 5

Lithium-Ion Battery Pack Thermal Modeling Via an Integrated Physics and Machine Learning Based Approach

Lithium-ion battery pack (LIBP) is an essential component in electrified transportation systems. The cost of LIBPs significantly impacts the electric vehicles (EVs) manufacturing cost, typically representing 30% to 40% of the electric vehicle's total cost [119]. Generally, LIBPs are configured from hundreds of batteries connected together to provide sufficient power and energy demands. Batteries in the pack experience major inconsistencies due to variations in the raw material, manufacturing and operational conditions. Several studies addressed optimizing the cell matching process during the manufacturing stage to avoid any potential problems resulting from cells' inconsistencies during operation. However, these studies concluded that these variations are found even in cells from the same manufacturing batch due to tolerance in the human, manufacturing and welding processes. The inconsistencies, although result from, they also contribute to the nonuniform temperature distribution over the cells, which impacts the vehicle range and safety. Thermal management system (TMS) is responsible for monitoring and controlling every cell temperature in the pack and detecting any potential faults. TMS is coupled with thermal models, which help in fault decision-making and add a redundant monitoring system to the existing physical temperature sensors. Thermal models are complex, especially for multi-cell battery packs, due to the involvement of several system components other than the cells. However, robust thermal modeling of LIBPs is always required to ensure safe and reliable pack operation.

Several methods have been introduced to model the thermal behavior of LIBs. A lumped or multi-dimension temperature distribution modeling requires accurate thermal parameters, including heat capacity and thermal conductivity of cells and other components in the pack. Each of these thermal parameters has a physical meaning and describes a thermal phenomenon that happens in the cell and the pack. The cell thermal lumped parameters can be calculated using a mass-weighted average of heat capacity and conductivity of cell components, including current collectors, tabs, and electrolytes. However, most battery manufacturers do not provide these thermal parameters and consider them confidential. Hence, several studies addressed thermal parameters' determination through direct measurement [120]-[123] and numerical methods [124]–[127].

Direct measurement methods utilize calorimeters [120], xenon flash sensors [121], and heat flux sensors [122] or inserting thermocouples inside the core of the battery [123]. These devices are used to directly measure the heat capacity and thermal conductivity of the cell. The direct measurement methods are accurate in measuring the battery parameters; however, they require complex preparation and expensive equipment. Other studies presented less complex methods to obtain the LIB thermal parameters [124]-[127] using simplified tests and nondestructive numerical procedures. The study in [124] proposed a simple test procedure followed by the least squares (LS) regression fitting to determine the thermal lumped parameters using the measured surface temperature during a current pulse test. Then a thermal lumped parameters (LP) model was tested to mimic the surface temperature of the battery using one drive cycle showing an error of less than 0.5 ºC. The study in [125] proposed an adaptive LS regression algorithm to estimate the battery lumped heat capacity and thermal conductivity of a cylindrical NCA battery in real-time. The LP model was investigated at ambient temperatures ranging from -20 to 40 ºC and was able to estimate the battery surface temperature with a root mean squares error of less than 0.05 ºC. The previous studies relied on the measured surface temperature in both parameters' determination and the model's validation. However, direct measurement validation should be provided to ensure the validity of the obtained thermal parameters using core temperature measurement and/or calorimeter. The study in [126] presented lumped heat capacity determination method for prismatic pouch, and cylindrical cells based on the LS regression using a simple constant power loss test at room temperature. Then the obtained heat capacity showed a 3.9% deviation from the heat capacity obtained by the calorimeters. The study in [127] expanded the work in [126] by performing parameter identification at multi-temperature, and three dimensions thermal parameters deduction from lumped parameters. Then LP and 3D temperature models were validated using temperature sensors and an infrared camera. The proposed LP and 3D models were able to estimate the battery temperature at 2C discharge rate with a maximum error of 1.6 and 2.2 °C, respectively. Based on the previous discussion, the numerical parameters' determination methods can be a sufficient and less complex alternative for obtaining battery thermal parameters. However, a validation of the adopted methodology is always preferred using direct measurement or benchmarking versus the given manufacturing thermal parameters. In addition, integrating additional components other than the cells in the pack should be considered when developing thermal models for multi-cell battery packs.

Recently, neural networks (NNs), derived from machine learning, are commonly used to model LIB behavior due to their strong capability in encoding the complex nonlinear behavior of the battery. Several studies addressed the application of the NNs for battery voltage modeling [82], and state of charge [77], and state of health [80] estimation. Recently, several studies discussed the possibility of using the NNs to model the battery thermal behavior, including core and surface temperature estimation [83],[131]-[132]. In [131], a recurrent NN based on a recurrent neural network with long short-term memory (RNN-LSTM) was utilized to estimate the battery surface temperature with an error of less than 1 ºC at room ambient temperature. While a feedforward NN (FNN) was benchmarked against RNN-LSTM model in [83] to estimate a cylindrical battery surface at different ambient temperature conditions ranging from -20 to 25 ºC. The proposed models in [83] were able to estimate the battery surface temperature with a maximum error of less than 3 ºC. The work in [132] presented a nonlinear autoregressive network with an exogenous inputs model to estimate the battery core temperature. The model was benchmarked against traditional FNN, and both models were able to estimate the battery core temperature with a maximum error of less than 1 ºC. The study in [110] proposed a hybrid FNN with an extended Kalman filter to enhance the core temperature estimation of a cylindrical battery. The proposed model was able to model a prismatic battery core temperature with a maximum error of less than 0.3 ºC. In [99], a RNN-LSTM was used for early detection of battery thermal runway by monitoring the residual of the estimated and measured cell temperatures. The proposed NN thermal model in [99] showed a temperature estimation mean absolute error of 0.01 ºC and a maximum error of 0.41 ºC when tested using non-faulty test cycles. Several studies demonstrated the capability of the NN to encode the thermal behavior of LIBs. However, the NN models’ parameters are abstract with no physical meaning, and they are highly dependent on the quality of the collected data and the nature of the training dataset.

A thermal model with a few degrees Celsius error is always required to ensure fast and safe fault detection and identification. In this chapter, an accurate integrated LP and FNN-based battery pack thermal model (LP+FNN) is introduced. Besides, the model is benchmarked against the traditional LP and FNN models. The parametrization and training of the proposed model are discussed, then the models are tested using drive cycles from an air-cooled LIBP at various thermal and driving conditions.

5.1 Battery pack thermal modeling overview

In this section, an overview of the LP, NN and integrated LP+FNN models is presented. Then, details of LP thermal of one and multi-cell models development are illustrated, including models' parameters and equations. Finally, multi-perception FNN and LP+FNN models are presented.

In this study, a robust thermal model that combines the benefits of both LP and NN models is introduced. A LP model consisting of cell thermal lumped parameters can be a sufficient alternative to model the thermal behavior of a LIB, which has been extensively utilized in several studies in literature. For multicell LIBP, a LP model should consider several aspects, including a variation of the airflow rates, heat transfer between the cells, and thermal mass of the pack components other than the cells. Figure 5-1 shows the LP model used in this study; the model is coupled with an electrical equivalent circuit model (ECM) to calculate the cells' power losses. Although LP model accuracy could be sufficient in design stages, for fault detection purposes, a few degrees Celsius error would cause a delay in fault detection and can, in the worst case, lead to thermal runway.

On the other hand, a multi-perception feedforward NN (FNN) model, shown in Figure 5-1, utilizes measured cells’ voltage, current, and SOC and maps them to the cells’ surface temperatures. NN models are developed using data collected in laboratories or from a fleet of vehicles on the cloud. The NN models are accurate enough when tested using various conditions and do not require experts to develop them. However, the NN parameters are abstract, and these models' accuracy relies on the quality of the data. For automotive applications, data is usually acquired and logged with low-cost sensors, which can challenge the FNN estimation accuracy. To overcome these challenges, integrating physics with FNN-based models can gather the benefits of both models. In this study, the concept of integrating a LP model and FNN, shown in Figure 5-1, is proposed to develop an accurate thermal model of multicell LIBP. The proposed integrated model utilizes the output temperature of the LP model and some measured battery parameters to estimate cells’ temperatures with the best possible accuracy. In the following sections, the development stages of LP, FNN and the LP+FNN models are discussed.

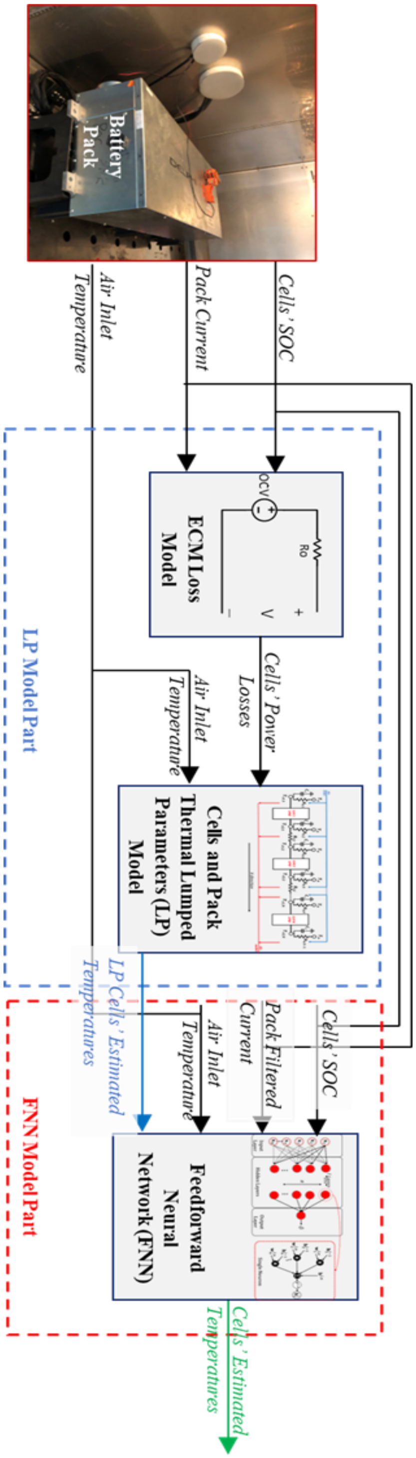


Figure 5-1: Battery pack temperature estimation models overview

5.1.1 Overview of the thermal lumped parameters model

5.1.1.1 Cell lumped parameters model

In this study, a LP cell model shown in Figure 5-2 is adopted to model the thermal behavior of one cell, where the thermal mass and the generated heat are assumed to be concentrated in the center of the cell. The heat is generated at the core and then transferred from the core to the surface of the battery by thermal conduction. The heat generation, absorption and transfer can be described by the heat balance equations described in (5.1)-(5.4). The core thermal resistance is divided equally between the cell surface into two halves to mimic the left and right sides of a prismatic cell. In lithium-ion batteries, the heat is generated from two sources, including irreversible and reversible heat losses [133]. The irreversible heat losses represent the ohmic losses of the internal cell components, including electrodes, tabs, and chemical reactions. They can be represented by an equivalent electrical resistance that consumes power in the form of heat as in (5.2). The reversible power losses represent the change in the entropy of the chemical reactions. The reversible heat losses can be calculated by multiplying the rate of the change of the battery open circuit voltage (OCV), battery current (I) and core temperature (T_c) as in (5.3). Then, the summation of these heat components is assumed to be transferred by conduction to the surface of the cell.

$$P_{loss} = P_{loss_{irr}} + P_{loss_{rev}} \tag{5.1}$$

$$P_{loss_{irr}} = I^2 R_{Ch,Dch} \tag{5.2}$$

$$P_{loss_{rev}} = -IT_c \frac{dOCV}{dt} \tag{5.3}$$

$$m_b C_b \frac{dT_c}{dt} = P_{loss} + (T_c - T_{s1})/(2R_{c,eq}) + \ (T_c - T_{s2})/(2R_{c,eq}) \tag{5.4}$$

Where $R_{Ch,Dch}$ is the electrical equivalent charge or discharge resistance, OCV is the cell open circuit voltage, m_b is the weight of the battery, C_b is the specific heat capacity of the cell, T_c is the core temperature and T_{s1}, T_{s2} are the battery two largest area sides' surface temperatures. Besides, P_{loss} is the total power loss and $R_{c,eq}$ is the thermal lumped core thermal resistance of the battery.

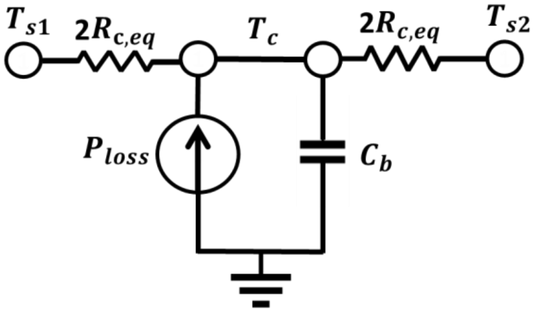


Figure 5-2: Cell thermal lumped parameters model

5.1.1.2 Battery pack lumped parameters model overview

In this subsection, a thermal LP model for an air-cooled multicell pack is developed using the thermal parameter of each component in the pack, including cells, tabs, and airflow. Besides lithium-ion batteries' applications, the LP models have been extensively used to model the temperature of different electrical applications, including electric motors [134],[135], and power electronic converters [136],[137]. The LP in this study is developed for an air-cooled N series cells module with air running through cooling channels in air gaps between cells, as shown in Figure 5-3. The heat is generated at the core of each cell and transferred from the core of the cell to the surface through thermal resistance by conduction, as discussed in the cell LP model shown in Figure 5-2. Then the heat is assumed to transfer from the two largest area surfaces (A_x) to the air by convection, as shown in Figure 5-3. The other cells' surfaces (A_y and A_z) are isolated with plastic casing and PCBs, and the heat transfer through these surfaces is neglected. The heat transfer is initiated by each cell and transferred to the adjacent cell by conduction and to the airflow by convection means which are presented by lumped contact (R_{cc}) and channels (R_h) thermal resistances, respectively as shown in Figure 5-3. The core and contact thermal lumped resistance is considered fixed for all cells assuming identical cells properties and connections, while the channels thermal resistances vary due to the variation of the airflow between cells. In addition, the heat capacity of the pack components other than cells is lumped and is represented by two shunt thermal masses ($m_c C_c$) added to each two cell sides. Hence, a thermal LP model shown in Figure 5-3 is built based on the aforementioned assumptions to model the temperature distribution over the cells in an N series cells module. The governing thermal equations describing the heat generation and transfer between every two adjacent cells can be written as in (5.5)-(5.7). The parameters determination method of the proposed model is discussed in Section 5.3.

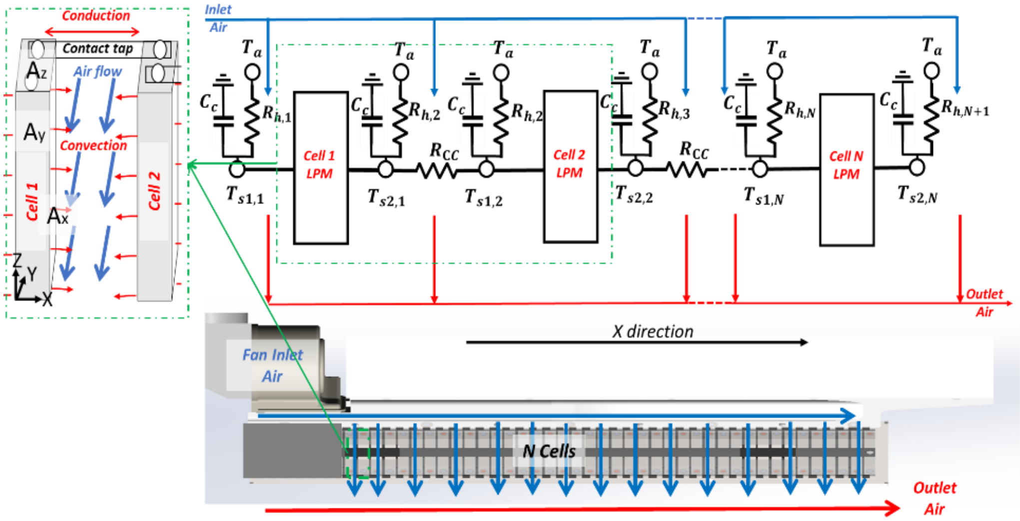


Figure 5-3: The proposed thermal LP model of N series cells module

$$m_b C_b \frac{dT_{c,i}}{dt} = P_{loss_i} + \frac{T_{c,i} - T_{s1,i}}{2R_{c,eq}} + \frac{T_{c,i} - T_{s2,i}}{2R_{c,eq}} \quad i\,\epsilon N \tag{5.5}$$

$$Side1\colon m_c C_c \frac{dT_{s1,i}}{dt} + \frac{T_{s1,i} - T_{c,i}}{2R_{c,eq}} + \frac{T_{s1,i} - T_a}{R_{h,i}} + \frac{T_{s1,i} - T_{s2,i-1}}{R_{cc}} = 0 \tag{5.6}$$

$$Side2\colon m_c C_c \frac{dT_{s2,i}}{dt} + \frac{T_{s2,i} - T_{c,i}}{2R_{c,eq}} + \frac{T_{s2,i} - T_a}{R_{h,i+1}} + \frac{T_{s2,i} - T_{s1,i+1}}{R_{cc}} = 0 \tag{5.7}$$

Where $m_c C_c$ is pack distributed lumped heat capacity of the pack components other than cells in J/K. $R_{h,i}$ is the lumped channel resistance of the cell#i. R_{cc} is the equivalent lumped thermal resistance of the tap connecting two adjacent cells and N is the total number of cells in one module, T_a is the inlet air temperature which is always equal to the chamber ambient temperature.

5.1.2 Overview of the feedforward neural network model

Feedforward neural networks (FNNs), commonly called multi perception neutral networks, are inspired by the human brain. FNNs are commonly used in fault diagnosis of power systems [92] and pattern recognition [111]. The basic structure of traditional FNNs contains neurons, layers, and activation functions all connected together to map the network inputs to outputs. Two FNNs are used in this study as presented in Figure 5-1, the first FNN utilizes the measured cells' parameters as inputs to the model, while the LP+FNN utilizes the estimated temperatures from the LP model as an additional input besides the cells' measured parameters. Both models have the same structure as the simple feedforward neural networks shown in Figure 3-6, and both output the cells' surface temperatures. A correlation analysis is performed first on cells' measured parameters to select the highly correlated inputs with the measured temperature. Once the models' inputs and structures are selected, the models are trained where their weights and biases are proportionally adjusted based on the partial derivative of the loss function (E) as in (4.1). Details about the development of the FNN and the LP+FNN models, including structure, inputs selection and training, are presented in Section 5.3.

5.2 Cells and pack specifications, test setup and data collection

An air-cooled LIBP extracted from a Plug-in Hybrid vehicle (PHEV) is utilized in this study. The battery pack consists of 72 series 72s SB Limotive cells, each 5.2 Ah nominal capacity. The pack is divided into two modules; each is 36 series cells, as shown in Figure 5-4. The pack has a total energy of 1.3 KWh and 266 V nominal operating voltage. Besides, ten dummy cells are placed at the beginning of each module, blocked with duct tapes to optimize the airflow across the cells. The pack is equipped with a fan to provide fresh air, which is conducted through a duct placed in the middle of the two modules. The cells are separated with 8 mm cooling channels which permit the fresh air to cool the cells down.

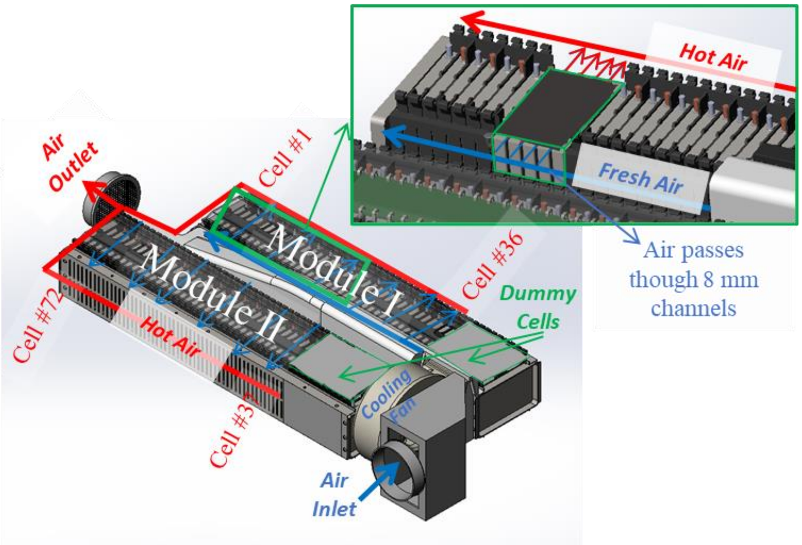


Figure 5-4: An air-cooled PHEV 72S1P battery pack layout

Table 5.1: Cell and pack specifications

	Cell	Pack
Cell Manufacturer	SB Limotive	72 series connected cells
Cell type	Prismatic	
Cell chemistry	NMC	
Nominal Capacity	5.2 Ah	
Nominal/min/max voltage	3.7/2.8/4.2 V	266/202 /302 V
Max discharge current	200A	
Specific Energy/Power	87 Wh/Kg/2887 W/Kg	Pack total energy (1.3 KWh) Pack 10-sec power (40 KWh)
Nominal resistance	2.6 mΩ	200 mΩ
Weight and dimensions ($L_x \times L_y \times L_z$)	220 g/ 12×80×120 mm	55 Kg/ 979×415×138 cm
Surface Area (A_x, A_y, A_z)	9600, 1440, 960 mm^2	N/A

Seventy-two temperature sensors, one sensor placed on top of each cell, are used to record the surface temperatures of each cell and to parameterize/train the lumped and neural network models. One cell from the pack, shown in Figure 5-5, is characterized and cycled at three different ambient temperatures, including 0, 10, and 25 ºC. A hybrid power characterization test (HPPC) is used for electrical equivalent resistance determination, which is used in power loss calculations.

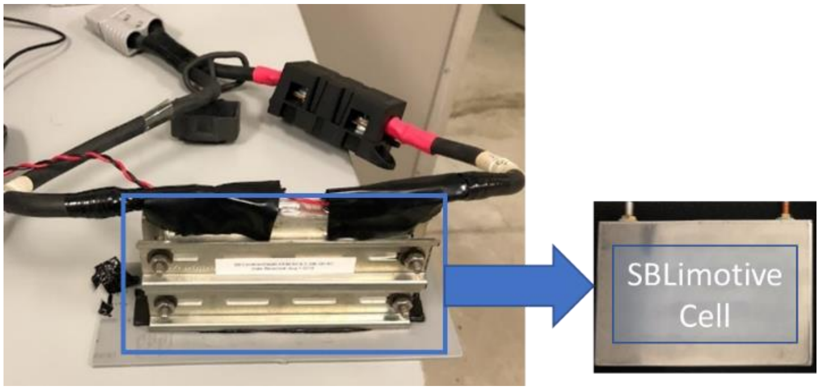


Figure 5-5: SB Limotive cell test fixture

Besides, a 10C constant power loss test is performed at 0, 10, and 25 ºC ambient temperatures on one cell and is used to determine the cell thermal lumped parameters, including the core resistance and heat capacity, as will be discussed in Section 5.3. The cell is placed in a chamber from Envirotronics, and the ambient temperature is controlled by controlling the cooling system of the chamber. Besides, one channel from Arbin firing circuit is used for this test with 60 A. At each test, the voltage, current, SOC, and battery surface temperature are collected using the Arbin tester at a 0.1 Sec log rate.

Similarly, the battery pack was tested using characterization and drive cycle tests at 15, and 25 ºC ambient temperatures, which is used as the inlet air temperature. The pack tests are performed using a D&V Firing Circuits battery tester and placed in a Thermotron chamber with specifications presented in Table 5.2. An 8C constant power loss test was performed on the pack to determine the convection coefficient and the heat capacity of the remaining pack components other than the cells. The pack was cycled using four standards, six *Mix* drive cycles and four different constant C-rates charges. During all test cases, the fan is assumed to work at the nominal speed, assuming that the safe temperature range of the cells is 15 to 40 ºC which does not require a varying fan speed. The four standard drive cycle tests are UDDS, LA92, HWFET, and US06, while the *Mix* cycles are made of randomized portions of the standard drive cycles and are used to train the NN models. The drive cycles power profiles are generated for a similar PHEV with 3.9

KWh which are scaled to match the tested pack power and energy capabilities. Pack total voltage, cells' voltages, current, and amp-hours are logged using Orion Original BMS [139] in 0.1 Sec log rate, and the cells’ temperatures are collected using Orion thermistor expansion module [140] in 1 Sec log rate. These data are used to train and test the proposed models. The characterization and drive cycle tests performed on the cell and pack are also summarized in Table 5.3.

Table 5.2: Test setup specifications for the cell and pack

	Cell	Pack
Tester manufacturer	Arbin Instruments	D&V Electronics
Test channel used	60 A, 0-5V	400 A, 0-500 V
Voltage/Current	+/- 0.04% Full	+/- 0.1% Full Scale
Data acquisition rate	10 Hz	10 Hz for voltage and current and 1 Hz for
Thermal chamber	Envirotronics M# SH16	Thermotron SE-3000-6
Chamber size	16 cu. Ft.	104 cu. Ft.
Collected data log acquisition accuracy	Same as cycler	Orion BMS and thermistor expansion

Table 5.3: Summary of tests performed on the cell and the pack

	Cell	Pack
Vehicle modeled for calculation of drive cycle power profiles	Minivan PHEV with 3.9 kWh	
Ambient temperatures tested	0, 10, 25 ºC	15, 25 ºC[1]
Characterization tests	HPPC test, and 10C constant power loss	8C constant power loss test
Dynamic tests	N/A	Ten drive cycles including Mix#1-Mix#6 and UDDS, LA92, HWFET, US06, and 4-10C fast charges

[1] The pack is placed in the chamber, so this temperature is considered the inlet air temperature as well

5.3 Temperature estimation models development

In this section, the parameters determination method of the LP model at different ambient temperatures is discussed. In addition, the obtained cell lumped parameters are benchmarked against parameters obtained using different methods for similar prismatic cells in the literature. Then the rest of the battery pack lumped parameters are obtained using tests performed on the pack. On the other hand, the FNN and LP+FNN models' development is presented, including the selection of the inputs, network structure and training framework.

5.3.1 Lumped parameters model development

In this section, the thermal lumped parameters determination of the cell and the pack are proposed. The parameters include cell specific heat capacity, core thermal resistance, pack components distributed lumped heat capacity, thermal contact, and channel thermal resistances.

5.3.1.1 Cell thermal parameters determination

A simplified cell LP model is utilized in this work, as shown in Figure 5-2. The cell is placed in a thermal chamber, maintaining it at a fixed ambient temperature, as shown in Figure 5-6 (a). The cell is tested at constant ambient and convection rate by adjusting the thermal chamber cooling/heating system and circulators. Five thermocouples are placed on the surface of the cell in the locations shown in Figure

5-6 (b) to improve the accuracy of the parameters' determination process. The heat balance equations in (5.1)-(5.4) are utilized to model the heat generation and transfer in the cell. In this study, reversible heat losses are neglected since irreversible power losses are the dominant source of heat generation in LIB [127]. The parameter determination method proposed in [126] and [127] are used in this study. The cell is placed in the chamber, and the air is assumed to flow uniformly all over the cells with no barriers. Hence, Eq. (5.4) can be reformulated to (5.8), where cell temperature (T_s) is represented by the average reading of the five sensors. Besides, the heat is assumed to transfer to the ambient by convection means only. Hence the heat flow from the surface to the ambient can be represented by an equivalent lumped resistance ($R_{o,eq}$) as described in (5.9).

$$m_b C_b \frac{dT_c}{dt} = P_{loss} + \frac{T_c - T_s}{R_{c,eq}} \tag{5.8}$$

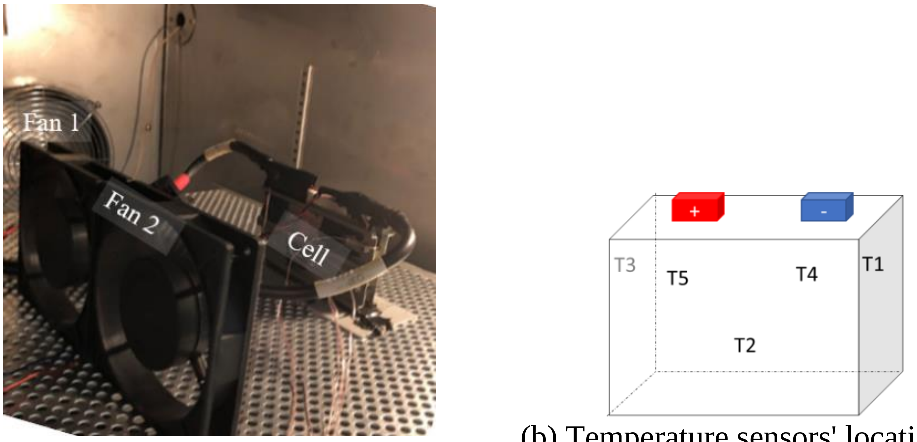
$$\frac{T_c - T_s}{R_{c,eq}} = \frac{T_s - T_a}{R_{o,eq}}, \ hence \quad T_c = \frac{R_{c,eq}}{R_{o,eq}}(T_s - T_a) + T_s \tag{5.9}$$

Since the ambient temperature does not change during the test; therefore, equation (5.9) is substituted in equation (5.8) as shown in equation (5.10).

$$mC_b' \frac{dT_s}{dt} = P_{loss} + \frac{T_a - T_s}{R_{o,eq}}, \qquad C_b' = C_b(1 + \frac{R_{c,eq}}{R_{o,eq}}) \tag{5.10}$$

The output resistance can be obtained from (5.10) at the steady state when $\frac{dT_s}{dt} = 0$. Besides, C_b' can be obtained by fitting measured surface temperature rise temperature using least squares fitting as in (5.10). In order to separate the core thermal resistance and cell specific heat capacity, two C_b' values are required. Hence, the test is repeated at two different airflow rates assuming negligible cell parameters change with the variation of the airflow.

Stacked or jelly-rolled batteries have two thermal conductivities according to the direction of the heat flow, including in-plane and cross-plane. Once the lumped core thermal resistance is obtained, the cell cross-plane (K_x), and in-plane thermal conductivities (K_y, K_z) can be calculated from the corresponding thermal resistivity as in (5.11) and (5.12) [127].



(a) Cell in the chamber

(b) Temperature sensors' locations on the cell

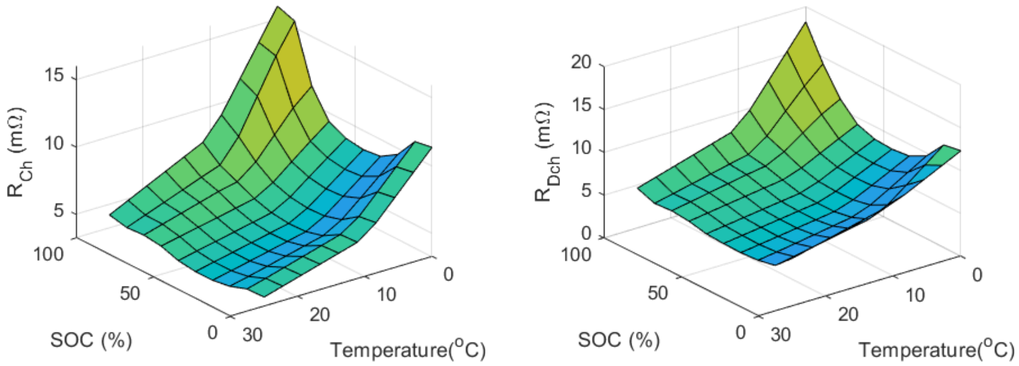
Figure 5-6: Test setup for determination of the cell thermal lumped parameters

$$R_{c,d} = R_{c,eq}\left(\frac{A_t}{A_d}\right) \qquad d \,\epsilon\, x, y, z \tag{5.11}$$

$$K_d = \frac{L_d/2}{A_d R_{c,d}} \quad d \,\epsilon\, x, y, z \tag{5.12}$$

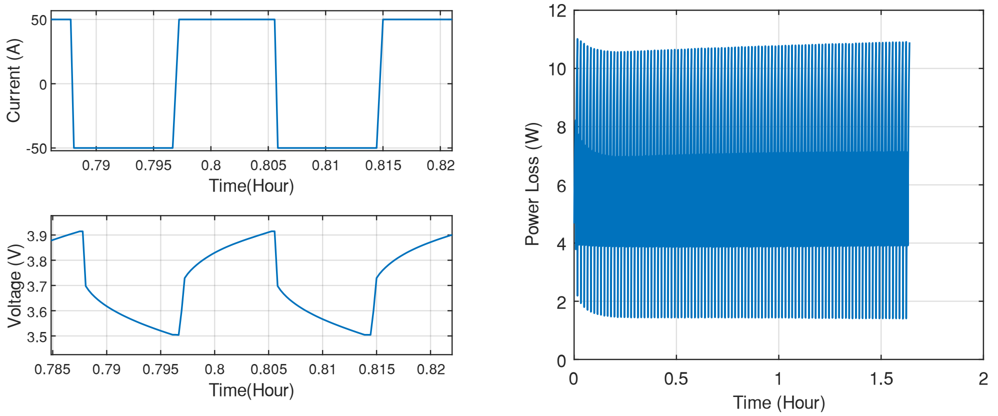
Where A_t is the total surface area of the battery, A_d is the surface area of the case at x, y, z directions, and L_d is the length in x, y, z directions.

A constant power loss test is designed for the parameters' determination process, which lasts for a long time, ensuring the cell temperature reaches steady-state conditions. The test is repeated two times, using only chamber fans, and another test with both chamber and external fans running. The test is repeated at three different ambient temperatures, including 0, 10, and 25 ^{o}C. At each test, the cell is fully charged at 25 ^{o}C then it is discharged to half its nominal amp-hours capacity prior to each test, and the ambient temperature is kept fixed during the whole test. The battery charge and discharge electrical equivalent resistance values are obtained using the HPPC test at three ambient temperatures, including 0, 10, 25 ^{o}C covering the whole temperature range. A look-up table of the charge and discharge resistances at three ambient temperatures versus the SOC, shown in Figure 5-7, is used in power loss calculation. Figure 5-8 (a) shows the recorded current and the voltage response for 100 seconds of the 10C constant power loss test at 25 ^{o}C while running the chamber fans only (low convection heat transfer rate). Figure 5-8 (b) shows the cell calculated power loss at 25 ^{o}C using the chamber fans only.



(a) Charge resistance (b) Discharge resistance

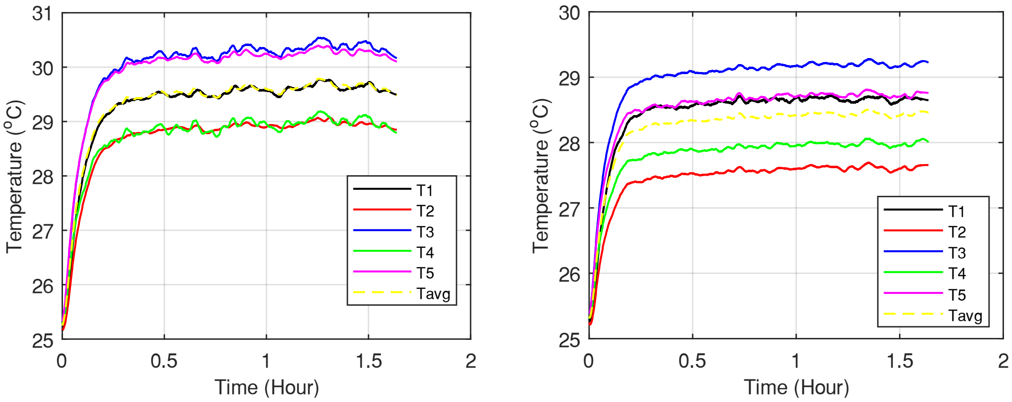
Figure 5-7: Cell charge and discharge electrical equivalent resistances at different SOC and temperatures



(a) Zoomed voltage and current (b) Calculated power loss

Figure 5-8: Periodic 10C constant power loss test and the corresponding cell power loss at 25 °C

Figure 5-9 shows the surface temperature readings of the five temperature sensors when the cell is tested at 10C constant power loss test when running at one and two fans at 25 °C ambient temperature. The figure shows that the differences in the temperature between the sensors over the cell are within 1 to 1.5 °C. This confirms the assumption of uniform temperature distribution over the cell surfaces. Besides, the maximum temperature rise is always shown near the cathode which is made from aluminum causing higher losses as compared to the copper anode. The average temperature rise using one fan and two fans at 25 ºC is 4.5 and 3.2 ºC, respectively, ensuring sufficient two convection resistance values. Moreover, using two fans helps the battery to achieve steady state conditions faster than using only one fan. Figure 5-10 shows the average temperature rise of the cell at the same 10C power loss test when repeated at different studied ambient temperatures and convection rates.



(a) One fan (b) Two fans

Figure 5-9: Five sensors readings at 10C constant power loss test at 25 ºC using one and two fans

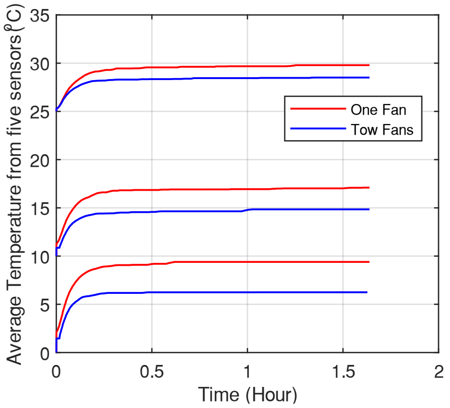
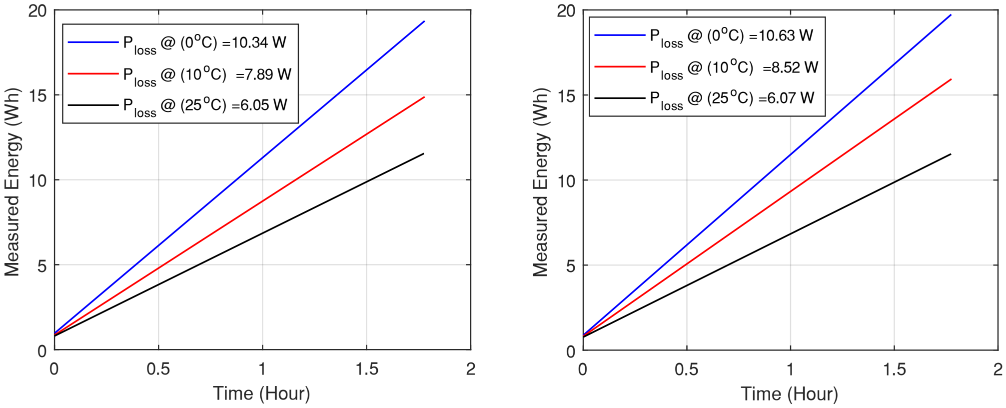


Figure 5-10: Cell average measured surface temperature for 10C constant power loss tests using one and two fans

The calculated average power loss values from the ECM using equation (5.1) at 0, 10, and 25 ºC ambient temperatures are {11.023, 8.920, 6.835} and {11,461, 9.368, 6.836} for one fan and two fans respectively as presented in Table 5.4. To validate the proposed power loss model, the power loss is measured directly from the slope of measured energy during each constant power loss test. Figure 5-11 shows the measured power loss from 10C constant power loss test at three different ambient temperatures and two heat convection rates. The results show that the error between the calculated power loss and the measured power loss is reasonable, between 6% to 13% for all tests. The accuracy of the power loss model used in this study is important since the calculated power loss for drive cycles and the charges are used and contribute to the accuracy of the LP model. The calculated power loss is used for the rest of the chapter, including the parameters determination and testing of the LP model.

Table 5.4: Summary of obtained cell thermal parameters at three ambient temperatures

Test Temp	Fan Status	Average calc. P_{loss}	$R_{o,eq}$	C'_b	$R_{c,eq}$(K/W)	C_b (J/Kg.K)
0 ºC	One Fan	11.023	0.660	1857	**0.638**	**943.9**
	Two Fans	11.461	0.407	2423		
10 ºC	One Fan	8.920	0.653	2113	**0.613**	**1089.9**
	Two Fans	9.368	0.416	2695		
25 ºC	One Fan	6.835	0.665	2185	**0.609**	**1141.6**
	Two Fans	6.836	0.414	2639		



(a) One fan (b) Two fans

Figure 5-11: Measured power loss from 10C constant power loss test at three different ambient temperatures

The average surface temperature of the five sensors shown in Figure 5-10 is utilized to obtain the cell thermal lumped parameters. At each temperature, the output resistance is calculated at steady state while LS fitting algorithm is utilized to obtain the C'_b. Once two C'_b values are obtained for each ambient temperature, the battery core thermal resistance and specific heat capacity are separated using (5.10). Table 5.4 summarizes the parameters obtained at each test and ambient temperature. The obtained core lumped thermal resistance and specific heat capacity at 25 ºC are

0.613 K/W, and 1141 J/Kg.K, respectively, while the output resistance changes from 0.66 to 0.41 from one fan to two fans, respectively.

The obtained cell thermal conductivity shows a slightly positive correlation with the temperature. This trend is most likely from the positive correlation of the thermal conductivity of the tabs and the current collectors with rising ambient temperature [141]. There are contradictory results regarding the influence of temperature on the thermal conductivity of LIBs. The studies in [142],[143] concluded a declining trend in thermal conductivity with rising temperature, while the studies in [127],[144] showed a rising trend in the thermal conductivity with the temperature increase. However, no dependency of thermal conductivity was shown on the temperature as presented in [145].

The obtained cell specific heat capacity shows approximately a 17% increase over the temperature rise range. The positive correlation of the cell specific heat capacity with the temperature results from the change in the heat capacity of the porous material, i.e., cathode, anode, and electrolyte. The specific heat capacities of the porous material are a function of thermal expansion, molar volume, and compressibility, which all are a function of temperature [146]. However, nonporous materials, including electrodes, current collectors, and the casing, have a negligible correlation with temperature change.

A summary of measured specific heat capacity and cross-plane thermal conductivity for similar prismatic batteries presented in the literature is summarized in Table 5.5. The cross-plane thermal conductivity of the tested cell in this study is calculated using (5.12) in the *x* direction and the cell dimension mentioned in Table 5.1. Only cross-plane thermal conductivity is benchmarked since there is a lack of the in-plane thermal conductivities presented in the literature. The table shows that the thermal parameters' determination methods range from the mass-weighted average of cell forming materials, using calorimeter, or inserting thermocouples inside the cell. The specific heat capacity for the prismatic cells presented in the table ranges from 900 to1142 J/kg.K at nominal temperature. This variation could be due to the variation in cathode chemistry, nominal capacity, and the accuracy of determination methods. The obtained specific heat capacity of the tested cell in this study lies within the range of the values in the literature.

Unlike the heat capacity, there is significant variation shown in the cross-plane thermal conductivity between the studies presented in the table. This variation could be from the change in the chemistry and cell structure. The obtained cross-plane conductivity of the investigated cell is slightly lower than the range of the values presented in the literature; however, such uncertainty is acceptable for the proposed simple method, which does not require a complex or expensive test procedure or destruction to the cell.

Table 5.5: Comparison of obtained thermal parameters versus prior research

Cell [ref]	Cell (Ah)	Cathode chemistry	Determination Method	C_b (J/Kg.K)	K_x (cross plane) (W/m.K)
[126]	5	NMC	Calorimeter	1025	N/A
[147]	104	NMC	Adiabatic chamber	1033	N/A
[149]	43	NMC	Infrared camera	978	N/A
[150]	42	LFP	EIS measurement	1142	N/A
[151]	20	LFP	Calorimeter	1048	N/A
[152]	2.8	N/A	Infrared camera	973	N/A
[149]	23	LTO		1146	N/A
[153]	60	LFP		N/A	0.485
[143]	6.8	LCO	Mass-weighted average	N/A	0.821
[142]	34	NMC	Inserting core thermocouple	N/A	1.1
[154]	25	NMC		N/A	1.1
[155]	37	NMC	Calorimeter	901	1.034
[148]	8	LFP	Heater + core thermocouples	1093	0.54
[127]	43	NMC	Lease squares fitting + Infrared camera	933	0.82
Proposed	5.2	NMC	Least squares fitting	**1141**	**0.403**

5.3.1.2 Pack thermal parameters determination

This subsection presents the rest of the pack thermal parameters determination, including contact thermal resistance, components distributed thermal heat capacity, and thermal resistance of the cooling channels, as listed in Table 5.6.

Table 5.6: Overview of pack thermal parameters

Parameter	Definition	Determination method	Value
R_{cc}	Contact thermal resistance	Direct calculation	0.21 K/W
R_h	Cooling channels thermal resistances	Genetic algorithm	See Figure 5-15
$m_c C_c$	pack distributed lumped heat capacity		180/branch and 25920/pack J/K

Firstly, the contact tap that connects the cells is made from aluminum and is 20 mm in length, and the conductivity of the aluminum k_{Al} is 239 W/m.K at 25 °C [156].

Hence, the contact thermal resistance (R_{cc}), can be calculated using equation (5.13), and is 0.21 K/W assuming that the variation of this resistance over the temperature is negligible.

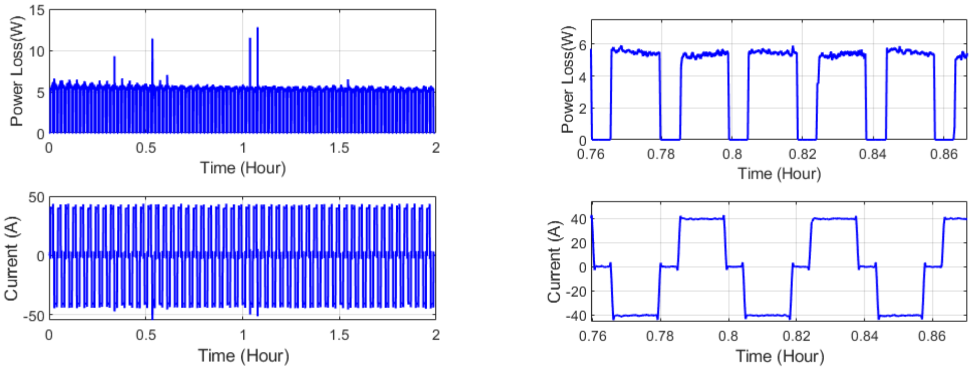
$$R_{cc} = \frac{1}{L_c \times k_{Al}} \tag{5.13}$$

Where L_c the length of the contact tap.

The convection rate of the cooling channels is non-uniform due to the non-uniform distribution of the air viscosity and pressure over the air path and cooling channels. An 8C constant power loss test shown in Figure 5-12 at 25 ºC ambient temperature (inlet air temperature) is utilized to determine the pack the distributed heat capacity $m_c C_c$ and channels lumped thermal resistances ($R_{h,i}$). For each module, the number of the cooling channels is N+1. The calculated average power loss of the cells from the test varies between 3.78 to 3.85 W. Figure 5-13 shows the steady state temperature of the 72 cells at the end of the constant power loss test. A maximum of 5 ºC variations of the maximum temperatures between the cells is shown. The middle and close to air inlets cells experience lower airflow rates as the viscosity of the air path is higher, and pressure drop is lower near the air inlet than the outlet. Hence, these middle and close to air inlets cells, i.e., cells 26 to 33, and 39 to 48, experience higher temperature rise than others.

The air channels lumped resistances are obtained at steady state by minimizing the error function (f) between the steady state measured cell surface ($T_{m,i}$) and the lumped parameters model estimated temperatures ($\hat{T}_{LP,i}$) in (5.14) using the genetic algorithm. It should be noted that the estimated surface temperature of each cell is obtained using the average temperature of the cell sides. The studied pack LP model is built based on MATLAB Simscape shown in Figure 5-14 and is used to obtain the LP estimated cells' temperatures.

$$\min_{R_h} f = \sum_{i=1}^{i=2N} \left(\hat{T}_{LP,i}(R_{h,i}) - T_{m,i}\right)^2$$
$$where\ \hat{T}_{LP,i} = \frac{(T_{s1,i} + T_{s2,i})}{2} \quad R_{h,i} \epsilon\ R_{h,1} \to R_{h,2N+2} \tag{5.14}$$



(a) One cell calculated power loss and current (b) zoomed cell calculated power loss and current

Figure 5-12: Periodic 8C constant power loss test performed on the pack and the corresponding power loss at 25 ºC

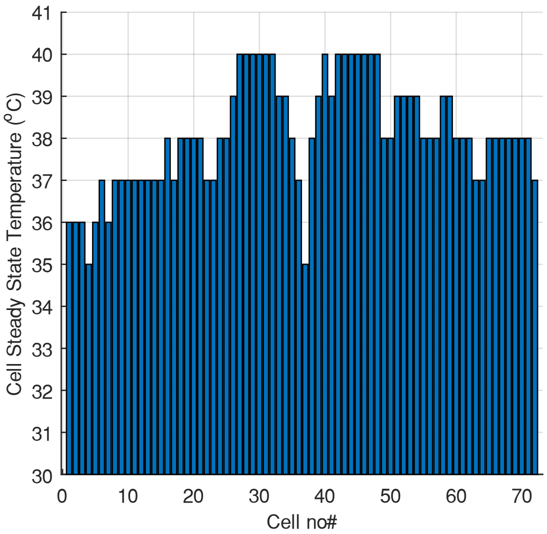


Figure 5-13: Steady state temperature achieved for each cell during 8C constant power loss test at 25 °C

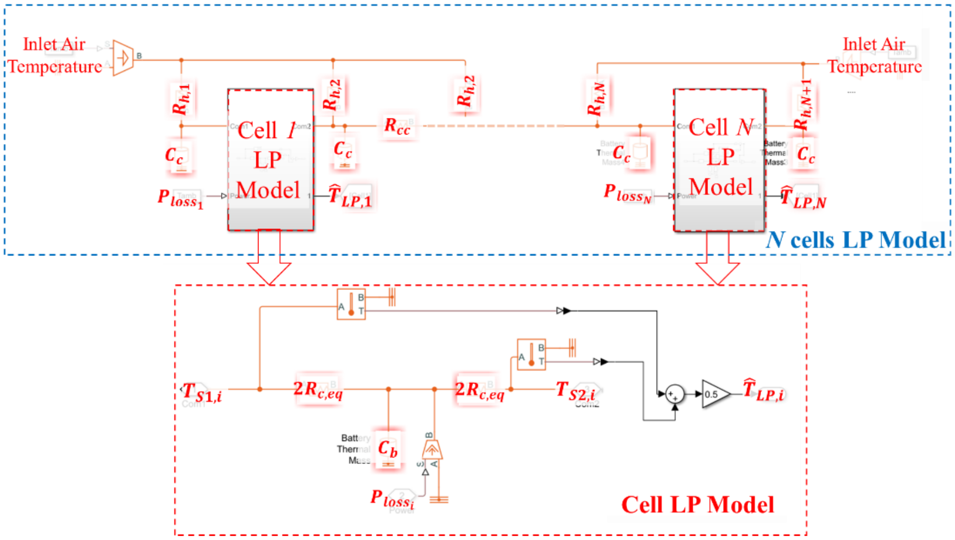


Figure 5-14: MATLAB Simscape lumped parameters model for one module of the tested pack

Figure 5-15 shows the cooling channels' thermal resistances obtained from the genetic algorithm ranging from 5.2 to 8.5 K/W. The obtained resistances coincide with the steady state temperature rise shown in Figure 5-13.

On the other hand, the pack distributed lumped heat capacity $(m_c C_C)$ is calculated by minimizing the error function (f) between the estimated and measured cell surface temperatures during dynamic temperature rise using the genetic algorithm after updating the model with the optimized convection resistances. The obtained branch pack distributed lumped heat capacity, $m_c C_C$, is 180 J/K per branch. Hence, the total heat capacity of the pack components other than cells is 25920 J/K which represents approximately 145% of the total heat capacity of the 72 cells in the pack.

Figure 5-16 shows the estimated surface temperature from the pack LP model versus the measured temperature for cell#1, 11, 40, and 69 at the 8C constant power loss test after obtaining the optimal convection resistances and pack distributed lumped heat capacity. It should be noted that the measured temperatures are collected in 1 ºC precision, which is the limitation of the Orion data logging system. The selected cells have different temperature rise ranges from 11 to 15 ºC. The results showed that obtained cell and pack thermal lumped parameters from LS and genetic algorithm are sufficient to mimic the pack temperature with less than 2.5 ºC maximum error (MAXE) for the 8C constant power loss test. However, further test cases are added in the following sections to illustrate the strengths and weaknesses of the LP model.

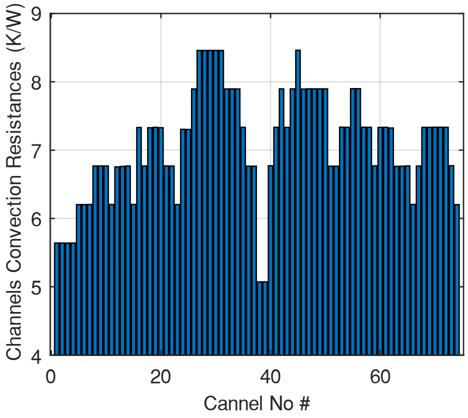
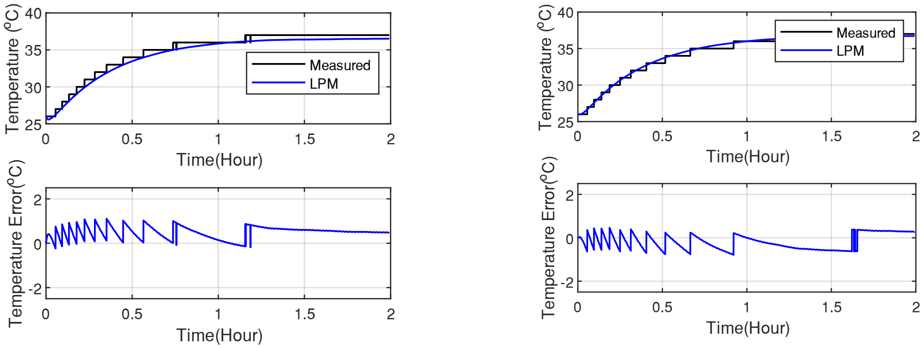
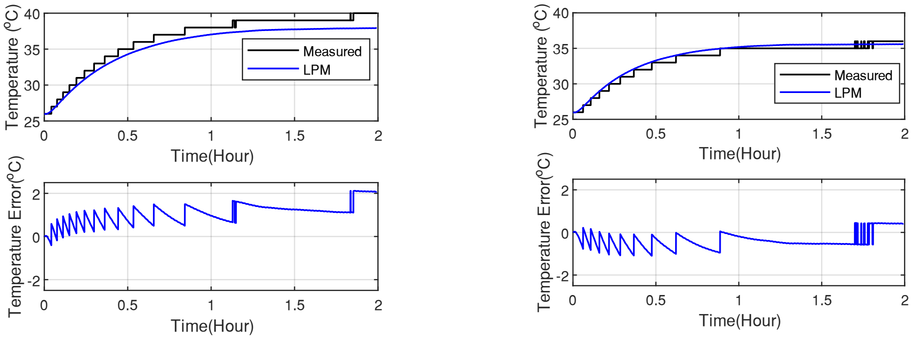


Figure 5-15: Thermal resistance for cooling channels of two module battery pack extracted from 8C constant power loss test at 25 ºC



(a) Cell #1

(b) Cell #11



(a) Cell #40

(b) Cell #69

Figure 5-16: Estimated LP model temperature versus measured for four cells at 8C constant power loss test at 25 ºC

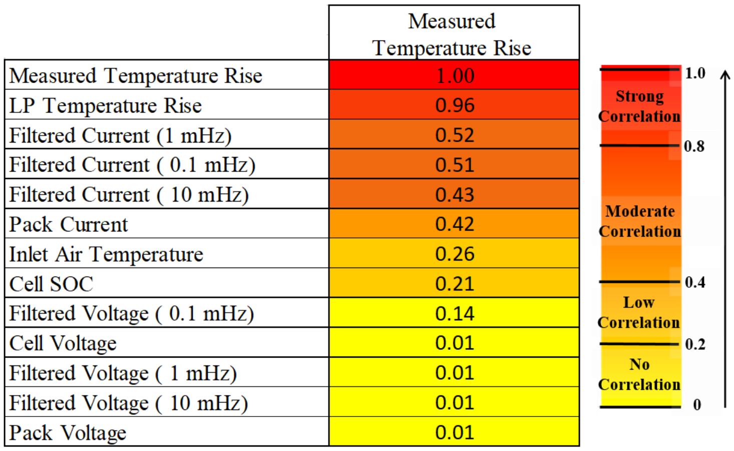
5.3.2 Neural network models development

In this subsection, the FNN and LP+FNN temperature estimation models are developed. A Spearman's rank correlation analysis is performed to determine the inputs to each model. Besides, the structure, numbers and types of layers of each model are presented. Finally, the training process and parameters are presented.

5.3.2.1 Selection of neural network model inputs

Different measured parameters are collected from the pack during operation, including cell voltages, pack voltage, pack current, cells' SOC, cells' temperatures rise, and inlet air temperature. Different combinations of these measurements, except the cells' temperature rise, can be inputted to the FNN. Using all measurements will impact the complexity of the training process, and in some cases, it leads to overfitting problems [157]. Hence, Spearman's rank correlation coefficient is performed between the different inputs and the measured temperature rise for one cell [158] to obtain the best features. Spearman's correlation assesses both linear and non-linear correlation between two variables by looking for monotonic relationships.

The correlation coefficients for each measurement with the cell surface temperature rise are listed in Figure 5-17. Coefficients from 0 to 0.2 are considered as no correlation, from 0.2 to 0.4 are considered low correlation, from 0.4 to 0.8 are considered moderate correlation, and from 0.8 to 1 are considered strong correlation. The figure shows that there is a strong correlation between the cell temperature rise and the current. However, no correlation is found between cell temperature rise and cell and pack voltages. Recently, several studies addressed improving the FNN performance by adding some filtered data to the inputs, such as voltage and current [93]. In this study, one low-pass first-order Butterworth filter is applied to the measured voltage and current and included in the correlation analysis. The Butterworth filter is selected due to its smooth frequency response roll-off and less phase delay. Three corner frequencies of Butterworth filters are applied to the cell voltage and current, including 0.1, 1, and 10 mHz. The current with 1 mHz filter corner frequency seems to have a better correlation with the cell temperature rise, while the filtered voltages tend to have no correlation. In addition, the air inlet temperature and SOC show a low correlation to the temperature rise. Besides, the estimated cell temperature rise from the LP model shows a strong correlation with the cell measured temperature rise. Based on the correlation study, the filtered current with 1 mHz (I_f) corner frequency, the cell SOC, and air inlet temperature (T_a) are selected for the FNN model inputs. In addition, the integrated LP+FNN model shares a similar structure and inputs to the FNN except by adding the estimated LP model cells' temperature rise ($\Delta\hat{T}_{LP}$). Figure 5-18 shows the structure of the investigated NN temperature estimation models for one cell, including inputs, layers, activation functions, and output. Two hidden layers with ten neurons each are used (N_{HL1}, N_{HL2} = 10), which is expected to be a sufficient number of neurons and layers to achieve good accuracy [93]. The output layer consolidates all the neuron outputs to create a single output value for the network, cell estimated temperature rise ($\Delta\hat{T}_{FNN}, \Delta\hat{T}_{LP+FNN}$). Once, the estimated temperature rise of each model is obtained they are summed to the cell initial temperature to obtain the estimated cell temperature ($\hat{T}_{FNN}, \hat{T}_{LP+FNN}$).



	Measured Temperature Rise
Measured Temperature Rise	1.00
LP Temperature Rise	0.96
Filtered Current (1 mHz)	0.52
Filtered Current (0.1 mHz)	0.51
Filtered Current (10 mHz)	0.43
Pack Current	0.42
Inlet Air Temperature	0.26
Cell SOC	0.21
Filtered Voltage (0.1 mHz)	0.14
Cell Voltage	0.01
Filtered Voltage (1 mHz)	0.01
Filtered Voltage (10 mHz)	0.01
Pack Voltage	0.01

Figure 5-17: Correlation coefficients of different measurements and cell's surface temperature

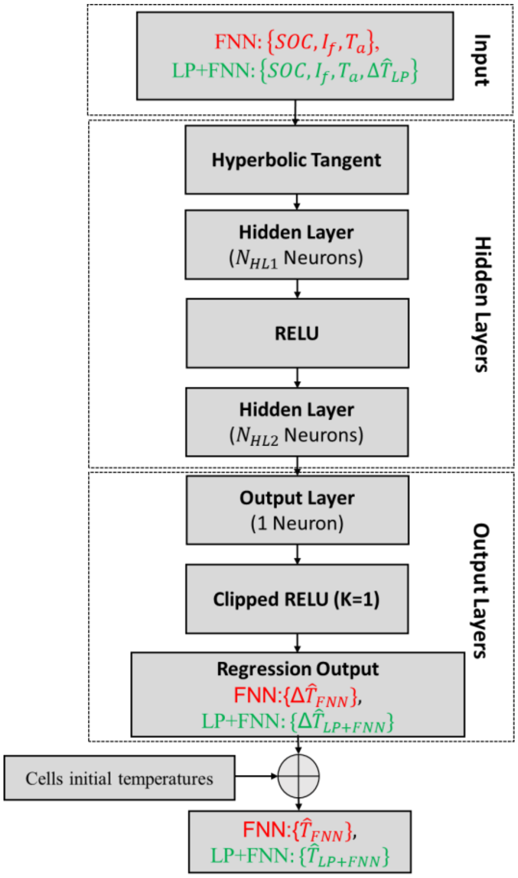


Figure 5-18: Structure of the FNN and LP+FNN cell models

5.3.2.2 Neural network models training using mix cycles

The FNN models parameters i.e., weights and biases, are updated iteratively during the backpropagation phase based on the loss (E) between the estimated and the actual output according to (4.1). The FNN and the LP+FNN models for each cell were trained using five *Mix* cycles, validated with one *Mix* cycle and tested using four standard drive cycles at two different ambient temperatures, as listed in Table 5.3 and shown in Figure 5-19. The temperature rise of the 15 ºC ambient temperatures cycles is slightly higher than the 25 ºC tests. The temperature rise of most test cases is in the range of the training dataset; however, in few cases, the models have to extrapolate. Besides, robustness test cases are added to investigate more scenarios that are not included in the training dataset.

The training initial learning rate of 0.01 is selected for the training process, which constrains how much the weights and biases can be adjusted for each epoch. The learning rate drop factor is 10%, and the patience is 1000 epochs, meaning that the learning rate is dropped by 10% each 1000 epochs. The training and testing datasets are resampled to 1 Hz and normalized, so their values are between 0 and 1. All the training data is used as one minibatch for each update of the training parameters. The training process stops after the validation dataset accuracy does not improve for 300 consecutive epochs. The network's learnable parameters are randomly initialized at the start of each training. Each model is multiple times with different initialization conditions to ensure a global- or near-global optimal solution is achieved. The network was created and trained using MATLAB 2021.b, the MATLAB Deep Learning Toolbox running on the shared hierarchical academic computing network (SHARCNET) with GPU. Table 5.4 lists the different parameters used in the training process of the proposed neural network models. The training process is repeated five times for each cell model, and the model with the lowest error for each cell is selected as the best-trained model. A total of 72 best-trained models for each of the FNN and LP+FNN models are developed and benchmarked against the estimated temperature from the LP model.

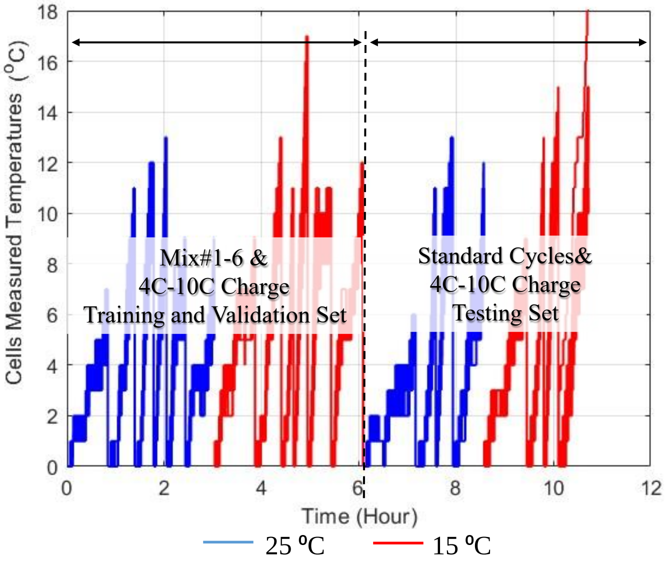


Figure 5-19: Training and testing temperature rise targets for FNN and LP+FNN models

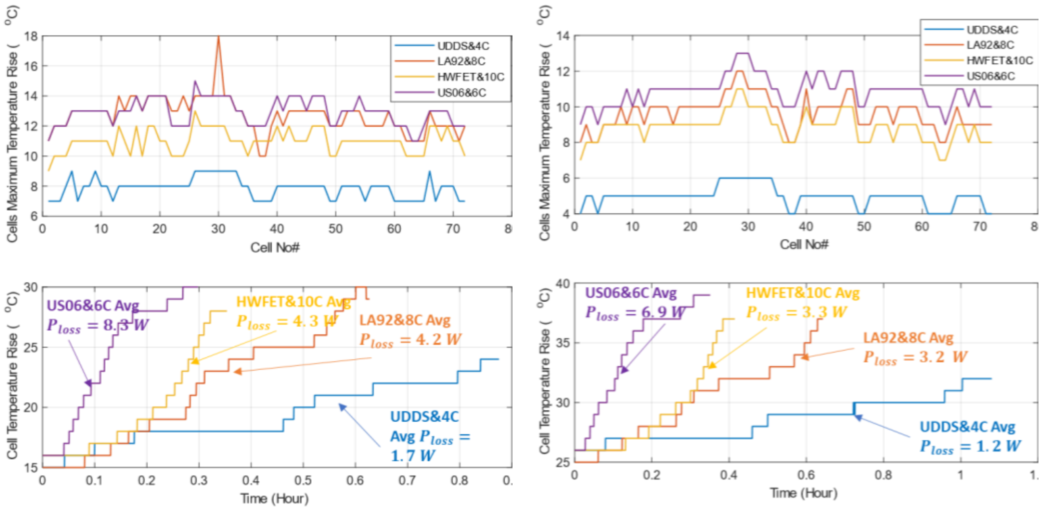
Table 5.7: Feedforward neural network temperature estimation models training parameters

Loss Function Optimizer	ADAM
Training Software	MATLAB 2021.b, MATLAB Deep Learning Toolbox
Training Platform	SHARCNET
Initial Learning Rate	0.1
Learning Drop Factor	10%
Number of Mini batches	1
Validation Frequency	Each 10 epochs
Stopping Criteria	300 epochs with no loss improvement
# Training Repetitions	5

5.4 Temperature estimation models testing using standard drive cycles and charge profiles

In this section, three LIBP temperature estimation models are benchmarked, including LP, FNN, and LP+FNN models using standard and different C-rate charges, which were not included in either characterizing or training the models. The proposed thermal models are tested using four test cycles, including UDDS, LA92, HWFET, and US06 and charge profiles range from 4 to 10 C-rate at 15 and 25 ºC ambient temperatures. These testing drive cycles range from city and less aggressive to urban and aggressive driving conditions. Figure 5-20 shows the maximum temperature rise achieved at the end of each drive and charge profile for each cell, and the corresponding average power loss. This temperature rise is proportional to the cell power loss and the duration of each testing profile. The figure shows that the maximum temperature rise for 15 ºC ambient temperature test profiles is slightly higher than the 25 ºC cycles due to increased battery internal resistance at lower temperatures. The figure also shows that the maximum temperature rise achieved by the less thermally aggressive profiles such as UDDS&4C is almost half the maximum temperature rise achieved by aggressive test cases such as US06. Besides, the temperature difference between the maximum temperatures of the cells ranges from 2 to 5 ºC depending on the generated power loss and the location of the cell. Hence, the studied temperature estimation models will be challenged by large temperature rise and variations of temperature between cells. Figure 5-21 shows a summary of the average root means squares error (RMSE) of the temperature estimation for each model for the aforementioned testing profiles. This average error is the mean temperature estimation error for the 72 cells. The average RMSE ranges from 0.78 to 1.74 ºC, 0.89 to 3.38 ºC and 0.32 to 0.57 ºC for LP, FNN and LP+FNN models, respectively.

It can be seen that the LP model is able to accurately capture the surface temperature of the cells at a range of thermal conditions with reasonable average RMSE, while the FNN model shows significantly higher RMSE for all test cases except UDDS at 15 ºC test. However, the LP+FNN shows the lowest average RMSE with more than half the error of the LP and the FNN for every test case.



(a) 15 ºC ambient temperature (b) 25 ºC ambient temperature

Figure 5-20: Cells' maximum and one cell time-domain temperatures for different test profiles and ambient temperatures

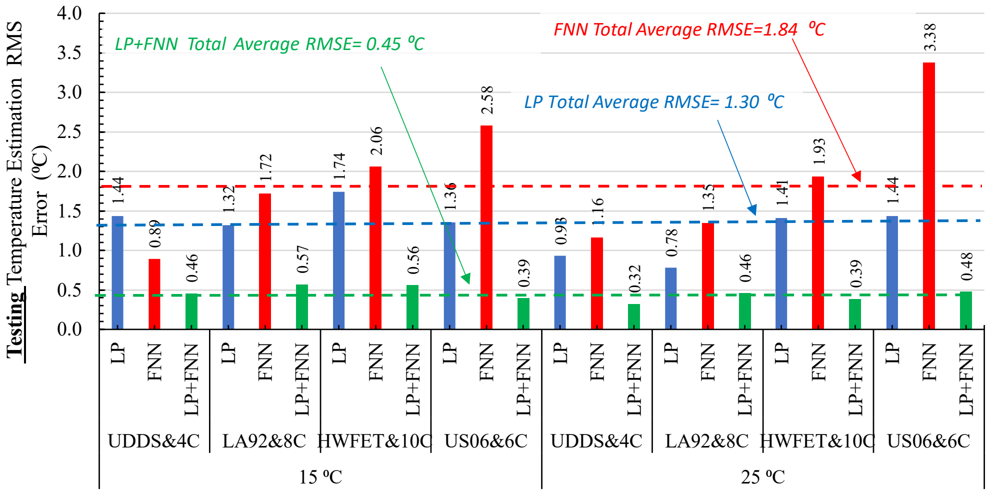
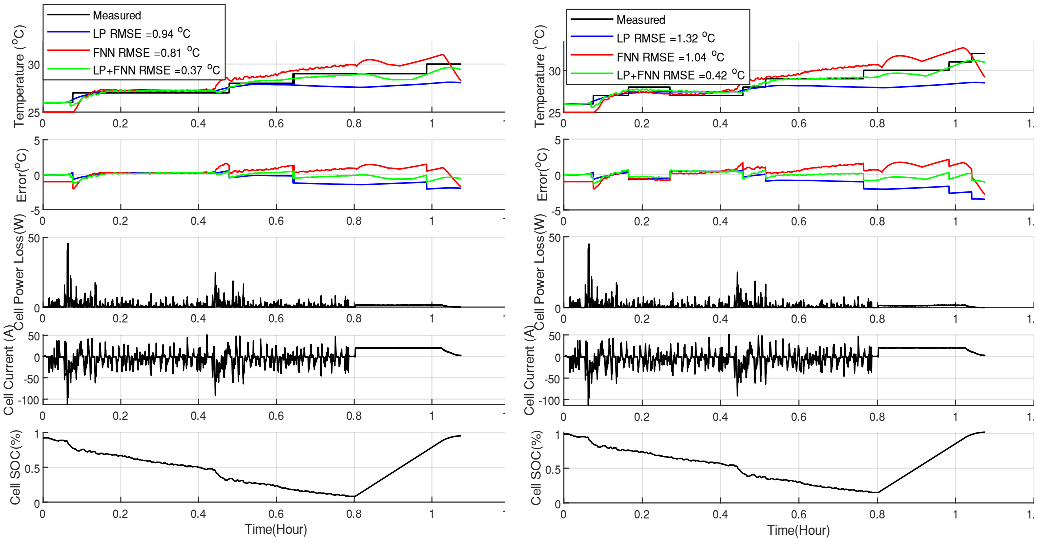


Figure 5-21: Models' temperature estimation RMS error for all testing cases

The average RMSE at 15 ºC ambient temperatures is 1.46 and 1.72, 0.49 for LP, FNN, and LP+FNN, respectively. On the other hand, the LP, FNN, and LP+FNN models show an average RMSE of 1.14, 1.95 and 0.41 ºC, respectively for 25 ºC test cases. The LP+FNN shows better and more stable performance with ambient temperature change than the LP and the FNN models. Overall, the LP+FNN achieved almost 65% and 75% less total average RMSE than the LP model and the FNN models, respectively. The RMSE is an important factor when developing a fault detection algorithm since every 1% RMSE impacts the prognosis decision. Overall, the results show how the FNN and the LP models can model the thermal behavior of the cells by incorporating information from the measured pack parameters. However, in fault diagnosis applications where the accuracy of the model contributes to the robustness of the fault detection, LP+FNN can be useful and potentially can enhance the BMS capability of thermal fault detection and decision time reduction.

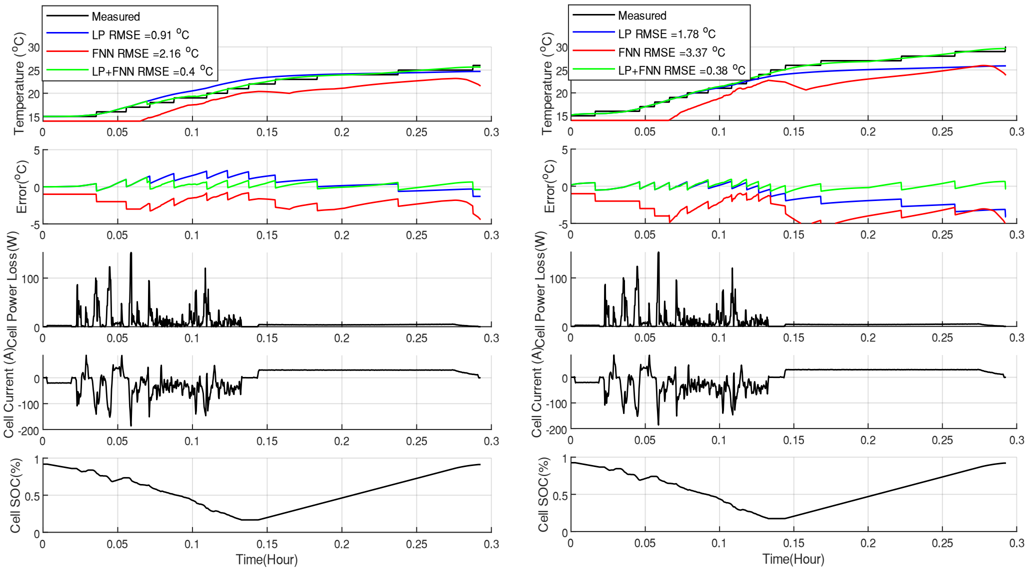
Figure 5-22 shows the time-domain models estimated temperatures to the UDDS&4C at 25 ºC, the lowest temperature rise test case, for minimum and maximum temperature rise cells, namely cell#1 and cell#25. The figure also shows the cell current, SOC, and power loss. The figure shows that the minimum and maximum cells' temperature rise are 5 and 7 ºC, respectively. The LP and FNN models show close performance for the two cells, with the FNN models showing a slightly lower MAXE of 3 ºC. The FNN model also shows the highest error near the end of the charge, which might highlight the inability of the FNN to mimic the temperature relaxation of the battery. The LP+FNN model shows significantly better performance than the LP and FNN models, with less than 0.5 and 1 ºC RMS and MAX errors, respectively. However, the performance of both LP and FNN models is still acceptable for cells that experience a mild temperature rise of less than 10 ºC and less aggressive thermal conditions with less than 1.5 ºC RMSE.



(a) Minimum temperature rise, cell#1 (b) Maximum temperature rise, cell#25

Figure 5-22: Performance of the models for minimum and maximum temperature rise cells for UDDS&4C testing profile at 25 °C

Figure 5-23 shows the models' temperature estimation for the minimum and maximum temperature rise cells, namely cell#1 and cell#30, for US06&6C at 15 °C profile, which is one of the highest temperature rise cases. The temperature rise of the cells for this drive cycle is significantly higher than UDDS&4C test and ranges from 11 to 15 °C. The LP model shows a significantly better temperature estimation than the FNN, especially in the cell where the temperature rise is 15 °C, namely cell#30 with a MAXE of 3 °C. The LP model also shows almost double the RMSE for cell#30 as compared to cell#1. The FNN struggles to estimate the temperatures of the studied cells with a RMS error of more than 2 °C and a MAX error of 6 °C during the test and near the end of the profile. Both LP and FNN are challenged by this test case where the temperature rise reaches up to 15 °C leading to significantly higher error than the previous test case with UDDS&4C. On the other hand, the LP+FNN shows a stable and better accuracy for both cells as compared to both LP and FNN models, with less than half the RMS and 2 ºC MAX errors. Hence, intergang the LP to the FNN in an integrated model as in LP+FNN achieves the optimum performance, especially in thermally aggressive and higher temperature rise cases.



(a) Minimum temperature rise, cell#1 (b) Maximum temperature rise, cell#30

Figure 5-23: Performance of the models for minimum and maximum temperature rise cells for US06&6C testing profile at 15 ºC

5.5 Temperature estimation models robustness testing

In all previous training and testing profiles, the cells' initial temperatures always match the inlet air temperature (ambient). In this case, the pack is tested at two different conditions where the air temperature and the initial cells' temperatures do not match. The first test case is LA92&8C, but the inlet air temperature is set to 20 °C while the initial cells' temperatures are adjusted to 15 °C. The second test is UDDS&4C, with the inlet air temperature are set to 15 °C while the initial cells' temperatures are set to 20 °C. These test cases are developed to mimic actual scenarios where the driver may drive the vehicle while the cells are warmer or colder than set coolant or ambient temperatures.

The three models are tested using these robustness cases, and a summary of the RMSE for all cells is presented in Figure 5-24. The FNN, and LP+FNN models show 98%, and 91% higher total average RMSE, respectively as compared to their performance when tested using standard cycles where the cells' initial temperature matches the inlet air temperature discussed in the previous subsection. The LP model shows almost no degradation in its performance when tested using the robustness test cases, demonstrating its ability to encode different thermal scenarios than the FNN and the LP+FNN models. However, the LP+FNN model still shows the least RMS for these robustness test cases with less than 1 °C RMSE.

The error standard deviation is another important indicator that is utilized in assessing how each model can estimate the temperature of all cells with similar performance. The maximum standard deviation of the LP, FNN, and LP+FNN models' maximum standard deviations are 0.49 °C, 1.02 °C, and 0.29 °C, respectively. The FNN shows the highest standard deviation of all cells meaning that the FNN model struggles to estimate the temperature of all cells with similar or close performance. The LP+FNN model shows the lowest standard deviation in all cells and cases, while the LP model shows a slightly higher standard deviation of the error. This demonstrates that the LP+FNN model is also able to estimate the temperature accurately and encode the variation of the cells' temperatures through the pack using the LP model.

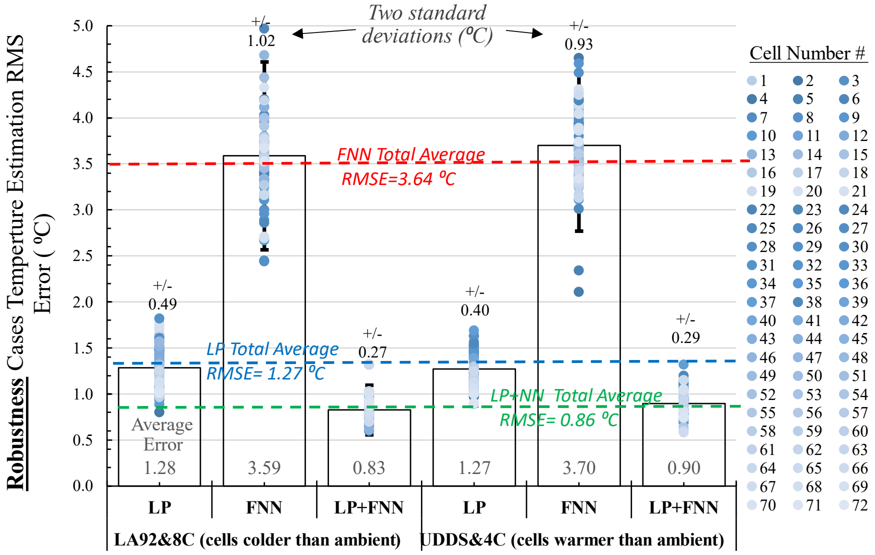
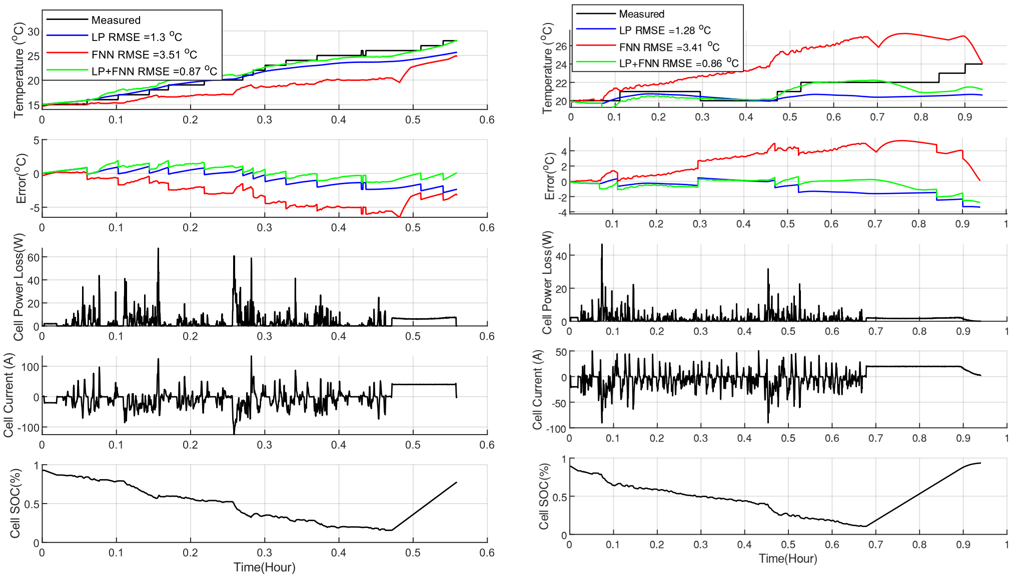


Figure 5-24: Models' temperature estimation RMS error for the robustness test cases

Figure 5-25 shows the models' performance for both robustness test cases for one average cell, i.e., cell#52. The figure shows how the LP+FNN could outperform the LP and the FNN models in real scenarios which were not included in the training dataset with a MAXE of less than 3 ºC as compared to 4 and 6 ºC for the LP and FNN models, respectively.



(a) LA92&8C with inlet air temperature of 20 °C and initial cell temperatures of 15 °C

(b) UDDS&4C with inlet air temperature of 15 °C and initial cell temperatures of 20 °C

Figure 5-25: Models' temperature estimation of cell#52 for the robustness test cases

Finally, **Error! Not a valid bookmark self-reference.** shows the summary of the error obtained for each model for all studied test cases. The presented maximum errors are for the most challenging test case and worst cell temperature estimation. Overall, the results demonstrate the capability of the proposed LP+FNN to model the temperature of a multi-cell battery pack under different dynamic and thermal conditions with 1 and 3 °C RMS and MAX errors, respectively.

Table 5.8: Summary of LP, FNN, and LP+FNN models error for studied cases

Test Case	Error	LP	FNN	LP+FNN
Standard test cases at 15 ºC	Average RMSE (ºC)	1.46	1.72	0.49
	MAXE (ºC)	5	8	2
Standard test cases at 25 ºC	Average RMSE (ºC)	1.14	1.95	0.41
	MAXE (ºC)	5	8	2
Robustness test cases	Average RMSE (ºC)	1.27	3.64	0.86
	MAXE (ºC)	6	9	3

5.6 Summary

In this chapter, an accurate integrated physics -neural network-based thermal model (LP+FNN) was introduced to estimate the temperature of the cells of an air-cooled lithium-ion battery pack (LIBP). Besides the proposed model is benchmarked against two temperature estimation modeling approaches, including a physics-based thermal lumped parameters (LP) model and a feedforward neural network (FNN). The parametrization and training of the models were discussed, then the models were benchmarked using test cases under various thermal conditions. Overall, the LP, FNN, and LP+FNN models showed an average root mean square error (RMSE) of 1.3, 1.8, and 0.4 ºC, respectively, when tested with standard drive cycles and different C-rate charges at different ambient conditions. The LP and FNN models achieved reasonable accuracy and close performance to the LP+FNN for lower temperature rise and less thermally aggressive test cases. However, for aggressive thermal conditions, the LP+FNN model showed significantly better temperature estimation accuracy, with a maximum error (MAXE) of less than 2 ºC for these cases. The models also were tested using realistic test cases while the cells were warmer or colder than the inlet air and ambient temperatures. The LP+FNN showed promising accuracy with less than 1 ºC average RMSE, typically 33%, 76% less than the error of the LP and FNN, respectively, and 3 ºC MAXE for these realistic test cases. Based on the studied test cases, the accuracy of the LP+FNN model can potentially enhance the BMS's thermal fault detection capability and reduce the time of fault detection.

Chapter 6

Thermal Fault Detection of Lithium-Ion Battery Packs Through an Integrated Physics and Deep Neural Network Based Model

A variety of faults and failures may occur in a battery pack over time, many of which are hard to determine based on conventional measurements. For example, a blockage in the cooling system would result in higher temperatures, but a fault would typically only be triggered if temperatures exceed a protection limit [159]. Failure to detect and identify these faults leads to accelerated battery degradation and, in the worst case, triggers pack fire and explosion [159]. These faults pose reliability and financial challenges for electric vehicle (EV) manufacturers [160]. Most of these faults can be detected early by capturing anomalies in the measured voltage, current, and temperature. Therefore, a robust battery management system (BMS) that is able to detect and identify these faults is essential. Besides, a BMS that can early detect the fault can avoid severe consequences of faults and thermal runway. As there are numerous fault types that can happen in LIBPs such as internal and external short circuits, onboard chargers, cooling systems and sensor reading faults. Internal short circuit fault, one of the main causes of triggering thermal runway in the pack, occurs due to the damage of the insulating separator and the defection of the electrodes. The external short circuit is also another common fault caused by the deformation of a battery pack after an EV collision or failure in the connection cables. Several studies have addressed the internal [161]-[163] and external short circuit [164]-[165] faults detection and identification. In addition, faults can also happen at onboard chargers, and robust fault detection should be able to take preventive measures to detect and isolate these faults [166]. The LIBPs utilize hundreds of sensors to monitor the cells' currents, voltages, and temperatures. The possibility of faults occurring in any sensor increases with such large number of sensors. As the BMS main functions are dependent on the accuracy of the collected measurements from the sensors, neglecting any of the sensor reading faults leads to incorrectly reported data to the driver and several safety hazards. Several studies proposed a methodology for the sensors’ fault detection [167]-[170].

Battery fault detection methods are classified as data-driven and model-based methods. In data-driven methods, the fault features are extracted from the battery measurements and correlated to different faults [165],[170]-[171]. Model-based methods are coupled with battery electrical or thermal models and are used in different fault detection applications [163],[164],[167]-[169]. In these model-based methods, the faults are evaluated by analyzing the residuals between the measured and model battery key parameters.

In the existing literature, most studies rely on measured voltage anomalies to detect different faults. The advantage of using measured voltages is the availability of voltage sensors installed on each cell in the pack [163]-[171]. The measured cells and pack voltages demonstrated their correlation with several electrical faults; however, voltage is less sensitive to the thermal characteristics of the cell, especially for thermal faults where the temperature change is quite small. Few studies developed a thermal fault detection system based on the measured temperature. The studies in [172] and [173] used the temperature sensors reading and were able to detect higher than normal heat generation [172] and thermal runway [173]. However, these studies addressed severe and easy-to-distinguish thermal conditions where the cell and the pack experience a high-temperature rise or thermal runway. There are various thermal faults that can occur in the pack, including cooling system failure, coolant lower or higher than a setpoint, coolant flow blockage and temperature sensor reading faults. These thermal faults involve moderate temperature rise, which could make it hard to be detected using the existing methods. In addition, to the extent of the author’s knowledge, none of the existing work has developed multi-fault detection and identification method specifically for thermal faults. Hence, the contributions of this chapter are as follows:

1) Development a novel physics-based combined with a neural network model to precisely mimic the temperature of cells in an air-cooled LIBP
2) Development of a cumulative log probability algorithm to detect different thermal faults
3) Development of a fault identification scheme to differentiate between different thermal faults
4) Validation of the proposed fault detection and identification method using various thermal faults, including cooling fan system failure, airflow lower or higher than setpoint, airflow blockage and temperature sensor reading faults

6.1 Test Setup and method overview

An air-cooled LIBP extracted from a Plug-in Hybrid vehicle (PHEV) is utilized in this study. The battery pack consists of 72 series 72s SB Limotive cells, each 5.2 Ah nominal capacity. The pack is divided into two modules; each is 36 series cells, as shown in Figure 5-4. Eight temperature sensors are selected on the two modules at cells# 1, 13, 24, 36, 37, 49, 60, and 72, dividing the pack into six submodules (SMs) as shown in Figure 6-1. The eight sensors are selected to mimic actual packs where only eight to ten temperature sensors are installed for every 100 cells.

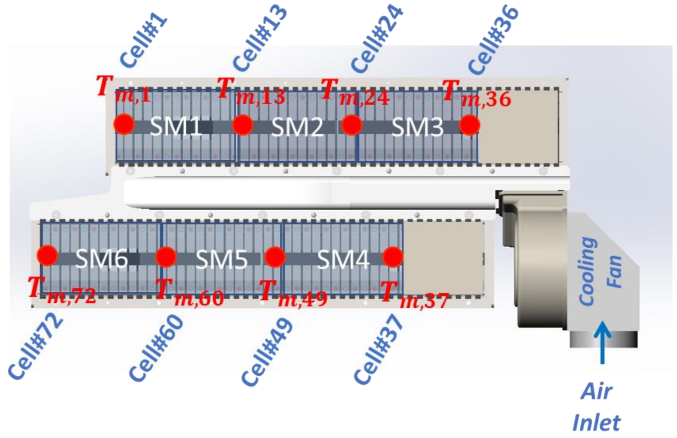


Figure 6-1: Tested pack and the locations of the eight temperatures sensors installed to monitor the whole pack

An integrating physics and NN-based thermal model (LP+FNN) is utilized to mimic the estimate the temperature of the eight cells. The idea of integrating a lumped thermal parameters (LP) model and feedforward neural network (FNN) models, shown in Figure 5-1, is proposed to develop an accurate thermal model of multicell LIBP with the best possible accuracy. The development stages of the LP+FNN model were discussed in Section 5.3, except only eight cells are selected and eight installed temperature sensors. Figure 6-2 illustrates the theory of operation of the proposed fault detection and identification method. The proposed method works by comparing the measured and the modeled temperatures of the eight sensors. Then residuals are generated and evaluated. The fault decision and type are determined based on the behavior of the generated residuals. Detailed illustrations of the residual calculation, evaluations, and fault detection and identification are presented in Section 6.2.

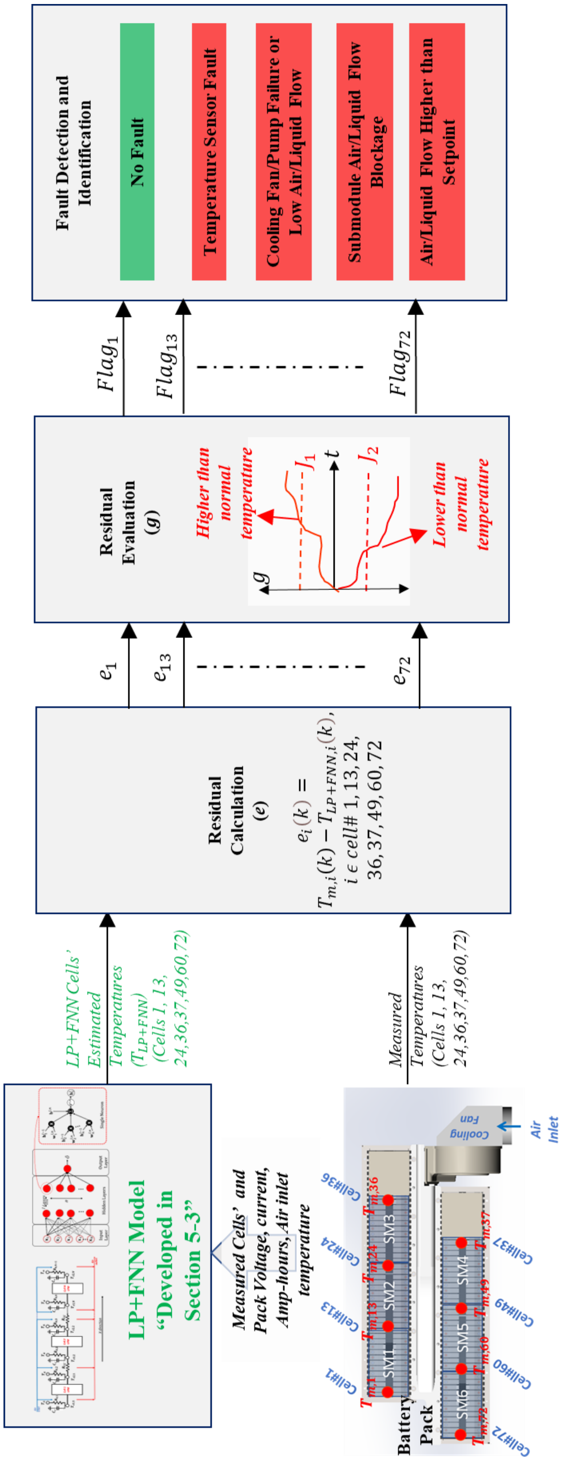


Figure 6-2: The proposed fault detection and identification method overview

6.2 Multi-fault detection and identification method

The proposed fault detection method is built based on the residuals between the measured and modeled temperatures. The residuals are generated and evaluated first, then the fault detection decision and type are determined, as will be discussed in the following subsections.

6.2.1 Residual calculation and evaluation

In this work, residual (e_j) between the sensor reading and model temperatures is calculated as in equation (6.1). A model cannot be 100% accurate; hence zero residuals even in healthy conditions are hard to achieve. Therefore, comparing the residuals directly with a fixed threshold can lead to false alarms and worsen the methods' reliability. Hence, a cumulative probability model is used to evaluate the faults [174],[175]. The results collected from fault-free test cases are fit to a normal probability density function, and a normal (μ) and standard deviation (σ) of the residuals are obtained. Then a cumulative summation of the log of the probability of the residuals (g) is calculated using equations (6.2)-(6.3). The log of the probability has the advantage of providing higher weights to the residuals with low probabilities, i.e., outside the $\mu \pm 3\sigma$ residual thresholds, as shown in Figure 6-3. The fault can be differentiated based on the g values where the measured temperature is higher than the modeled temperature and vice versa. The value of the g function accumulates every time the residual exceeds the three standard deviations threshold ($\mu \pm 3\sigma$) and resets to zero when the residual value goes back inside the thresholds. A fault flag is used to represent faults where the measured temperature is higher than the modeled temperature and vice versa. The fault is determined when the g value exceeds predetermined limits (J) as in equation (6.4).

$$e_i(k) = T_{m,i}(k) - \hat{T}_{LP+FNN,i}(k) \tag{6.1}$$

$$i \in \{cell\#1,13,24,36,37,49,60,72\}$$

$$g_i(k) = \begin{cases} g_i(k-1) - \log\big(PDF\big(e_i(k)\big)\big) & if\ e_i(k) > \mu + 3\sigma \\ g_i(k-1) + \log\big(PDF\big(e_i(k)\big)\big) & if\ e_i(k) < \mu - 3\sigma \\ 0 & if\ \mu - 3\sigma \le e_i(k) \le \mu + 3\sigma \end{cases} \tag{6.2}$$

$$PDF(x) = \frac{1}{\sigma \times \sqrt{2\pi}} \mathrm{e}^{-0.5\left(\frac{x-\mu}{\sigma}\right)^2} \tag{6.3}$$

$$F_i(k) = \begin{cases} High & if\ g_i(k) > J_1 \\ Low & if\ g_i(k) < J_2 \\ Null & if\ J_1 > g_i(k) > J_2 \end{cases} \tag{6.4}$$

Where F is the fault flag declaring faults that the reading is higher and lower than the modeled temperature respectively, and J_1 and J_2 are fault thresholds that are tuned using fault free test cases.

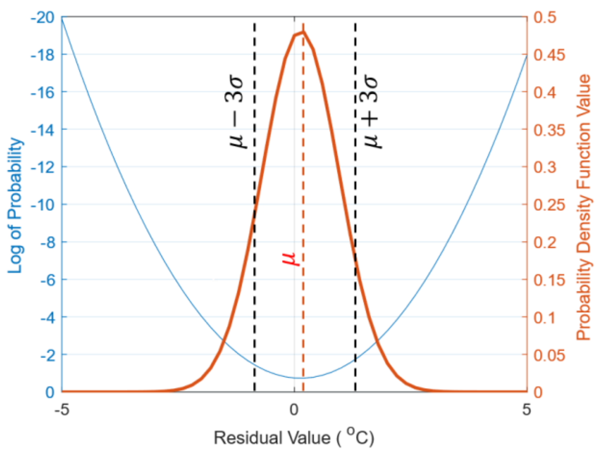


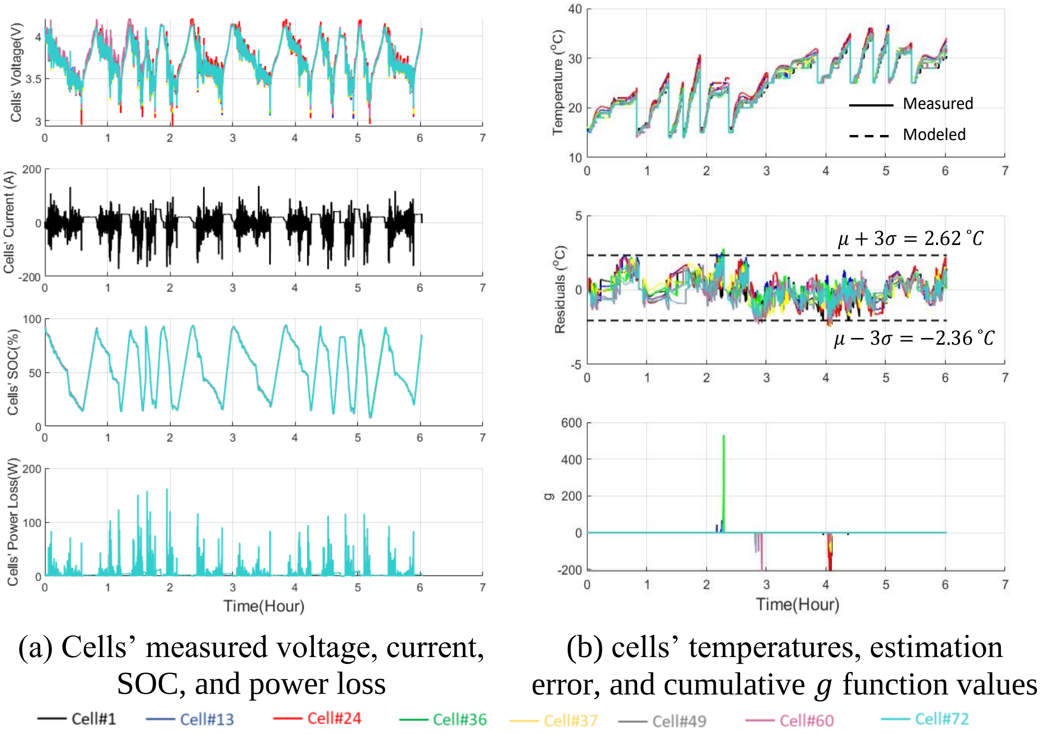
Figure 6-3: The impact of log probability on the residual value

In this study the residuals are generated using twelve healthy mix cycles at two different ambient and inlet air temperatures, namely 15 ^oC and 25 ^oC as summarized in Table 6.1 and shown in Figure 6-4 (a). The Mix cycles are used to develop the LP+FNN model, as discussed in Chapter 5. Figure 6-5 shows the histogram of the concatenated residuals of the eight sensors and models for the fault-free test cases. The figure also shows the fit of the residuals to a normal distribution function with a mean of 0.13 ^oC and a standard deviation of 0.83 ^oC.

Figure 6-4 (b) shows the residuals and log probability function values (g) for the fault free mix cycles. The figure shows that the maximum g function values obtained for these healthy cycles are 520 and -210, respectively. Hence J_1 and J_2 are adjusted slightly higher the obtained value, is 600, and -250 respectively for all sensors to avoid any false alarms.

Table 6.1: Summary of fault-free and faulty tests performed on air-cooled pack

Test Cases	Status	Purpose
Mix#1-6& 4C-10C charges at 15 °C and 25 °C ambient temperatures	Fault-free	Residuals generations and limits tuning
HWCUST&10C at 15 °C and 25 °C ambient temperatures	Fault-free	Test the system under fault-free conditions
US06&6C, UDDS&4C and LA92,8C at 15 °C ambient temperature	Fan is off	Fan failure
	Fan speed is 200% of the nominal value	Airflow higher than setpoint
UDDS&4C at 25 °C ambient temperature	Fan speed is 50% lower than the nominal value	Airflow lower than setpoint
HWFET&10C at 15 °C ambient temperature	Submodule (SM3) is blocked	Submodule airflow blockage
UDDS&4C, LA92&8C, HWFET&10C, US06&6C at 15 °C and 25 °C ambient temperatures	Offset, scale, and incorrect reading faults are added to these healthy cycles	Sensor reading fault



(a) Cells' measured voltage, current, SOC, and power loss

(b) cells' temperatures, estimation error, and cumulative g function values

Figure 6-4: Cells' measured parameters and model estimated temperatures and the corresponding g values for Mix#1-6 cycles at 15 and 25 °C ambient temperatures

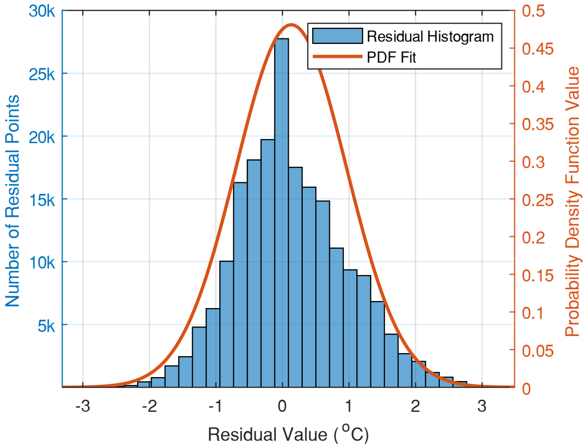


Figure 6-5: Residuals and normal distribution fit under healthy drive cycles

6.2.2 Fault identification method

Faults can occur in any part of the thermal system, including fan, cooling path, and data acquisition, i.e., temperature sensor reading. Several thermal faults are discussed in this study, including fan/pump failure, liquid/air flow higher/lower than setpoint, submodule air/liquid flow blockage and temperature sensor reading faults. One of the most severe faults that happen in the LIBP is the cooling fan or pump operation failure, as this triggers thermal runway [176]. This failure of the cooling system fan or pump is directly associated with a rapid temperature rise of the cells due to poor heat dissipation. Besides, air/liquid flow blockage can also lead to poor heat dissipation in some cells of the pack. Air/liquid flow blockage, although less severe, should be detected to avoid triggering the temperature rise of some cells in the pack, which could lead to a thermal runway if not detected and resolved [177]. In addition, lower or higher air/liquid flow than setpoint is also another important fault that should be monitored to ensure correct fan/pump operation. Temperature sensors may have unexpected readings while operating due to various reasons. This reason varies from manufacturing, such as manufacturing defects or operational reasons, such as overtime exposure at high temperatures, vibration, shock, and signal interference [178]. Different temperature sensor reading faults detection are presented in this study, including offset, scale, and incorrect readings. Figure 6-6 shows the flow chart of the proposed fault detection and identification method. The proposed method is applied on an air-cooled pack in this study; however, it works the same way if applied on a liquid-cooled LIBP.

The method starts by collecting the residuals and evaluating the recorded fault flags (F) in each sensor based on the g function values. Then based on the number and nature of the recorded flags, the fault type is determined. The fault is declared once one or more flags are recorded during a waiting window (WT). Then the fault type is determined based on the type and number of the recorded flags. If only F is recorded in only one sensor, then a sensor fault is declared. If high F is declared in two to four subsequent sensors, then airflow blockage fault is declared in the corresponding submodule(s). A fan failure or low airflow faults are declared if more than four high F flags are recorded within the window time. The fan off and low airflow faults can be separated by adjusting a certain level of the g value for each fault. Finally, if more than one low F flags are recorded, then an airflow higher than the setpoint fault is declared. The faults in this study are assumed to sustain a long time until being detected by the proposed method. The WT of fault declaration is tuned to ten minutes in this study. Also, the faults are assumed to happen one at a time.

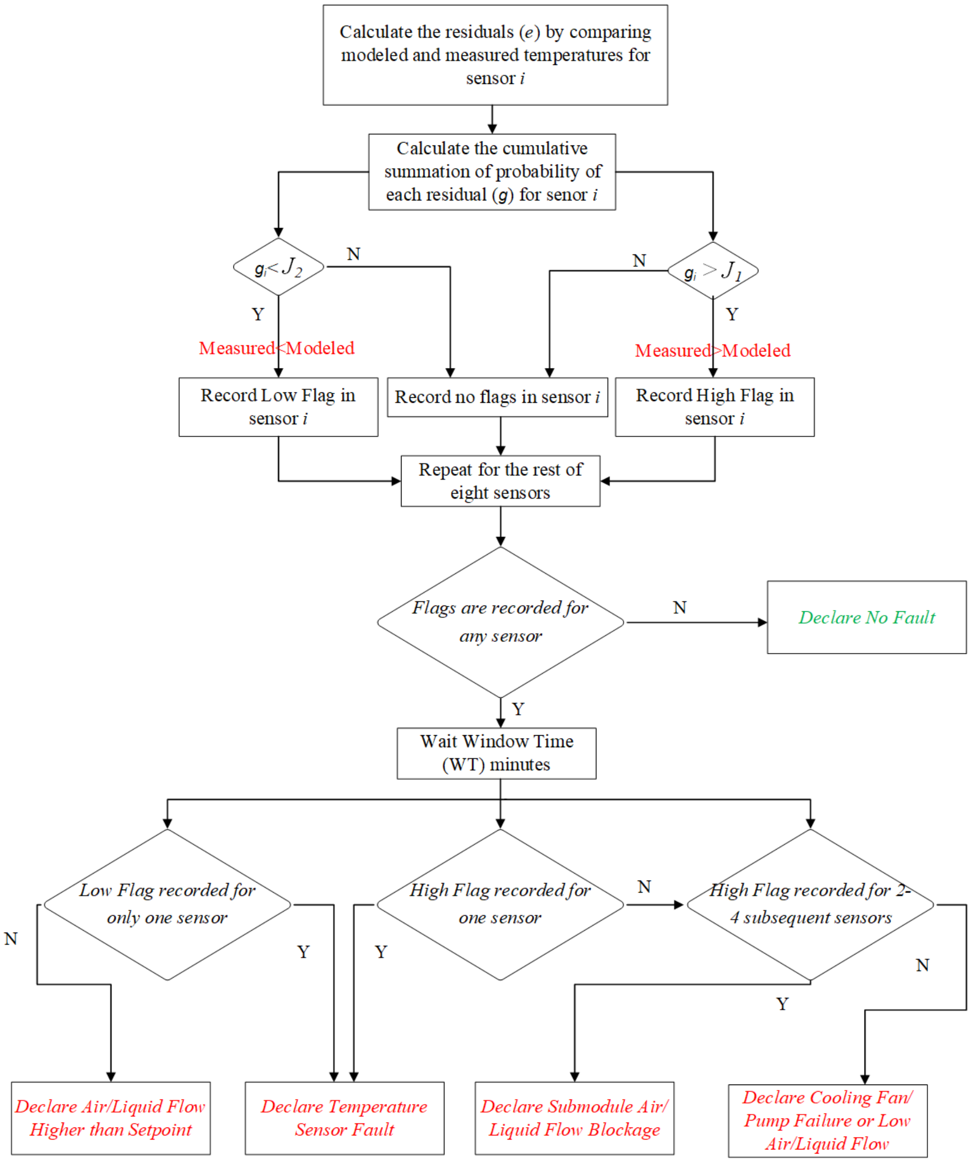


Figure 6-6: Flow chart of the proposed fault detection and identification method

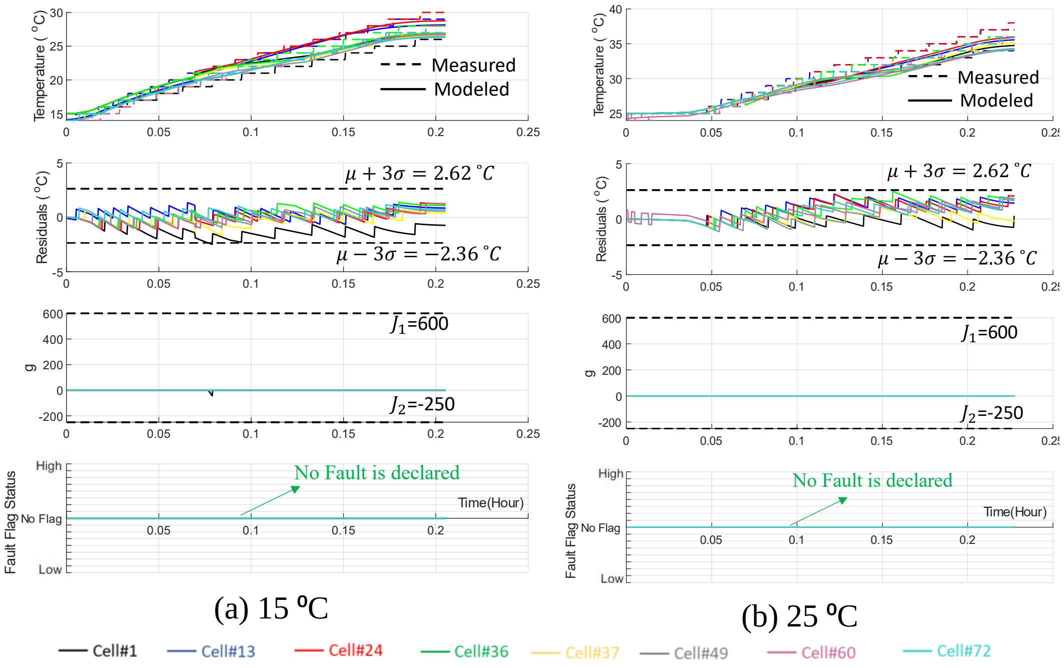
6.3 Experimental validation

The proposed fault detection method is tested using different experimental test cases, including fault-free and thermal faults. Firstly, the system is tested using fault-free test cases summarized in Table 6.1. Then, the proposed method is tested using tests containing different faults, including fan off case simulating cooling system failure, fan speed higher and lower than nominal simulating airflow higher or lower than setpoint and submodule#3 air path blocked simulating airflow blockage, as presented in Table 6.1. Finally, the proposed method is tested by adding different sensor reading faults to the eight studied sensors, including offset, scale, and incorrect reading. The sensor fault is added to fault-free testing standard drive cycles at two ambient and air inlet temperatures, namely 15 ºC and 25 ºC presented in Table 6.1.

6.3.1 Fault-free system analysis

The purpose of this section is to validate the performance of the proposed method at fault-free conditions. Inaccurate fault threshold adjustment can lead to a false alarm which is always not desired. The proposed method is tested using two cycles that were not included in developing the model and tuning the proposed fault detection method. HWCUST, a customized highway drive cycle with speeds varying between 100 and 130 km/h followed by 10C charge, is used to test the proposed method. The cycle is repeated at two different ambient (inlet air) temperatures, as mentioned in Table 6.1. This drive cycle mimics an aggressive usage of the pack followed by a high C-rate fast charge.

Figure 6-7 shows the proposed method performance for the HWCUST cycle at 15 ºC and 25 ºC ambient and inlet air temperatures. It can be seen that residuals of both cycles reside within the μ±3σ except for cell#1 at minute#8 for 15 ºC test. However, the g value does not exceed the J limits. This consistency suggests the reasonability of the proposed limits used and demonstrates that the LP+FNN temperature estimation model can learn the pack's thermal behavior with reasonable accuracy even for cycles were not included in the development stages.



(a) 15 ºC

(b) 25 ºC

Figure 6-7: Proposed method performance for HWCUST&10C fault-free test cases

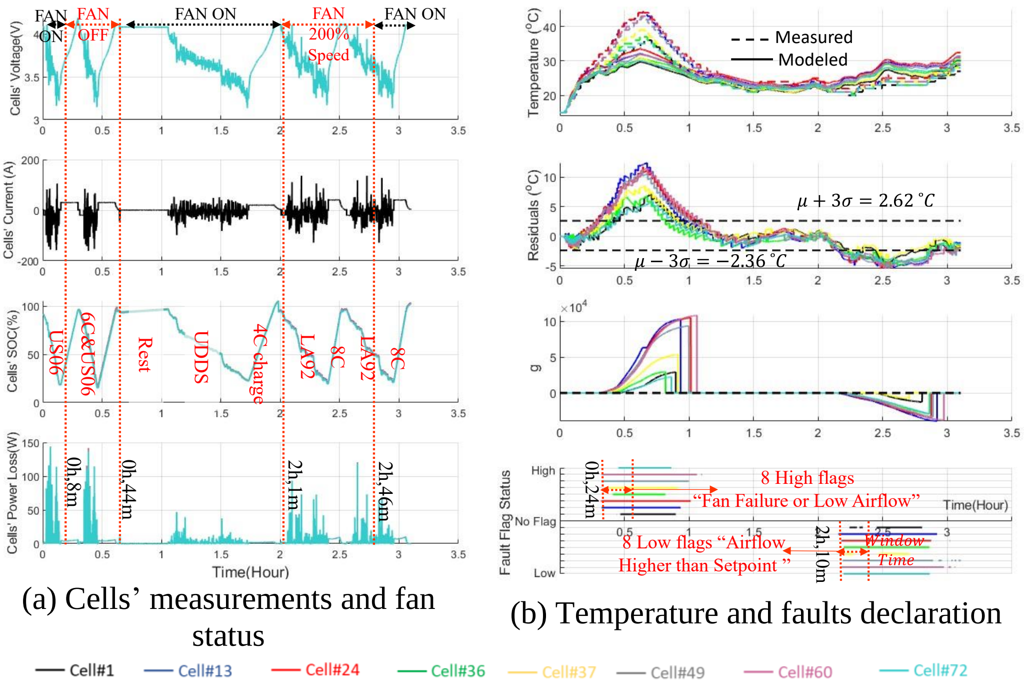
6.3.2 Cooling system faults analysis

In this section, the proposed method is tested using tests containing different faults in the cooling system, including fan-off simulating fan failure fault. In addition, airflow higher and lower than nominal fan speeds simulating flow higher or lower than setpoint and airflow submodule blockage faults are discussed.

A long driving scenario containing a series of US06&6C, UDDS&4C and LA92&8C at 15 ºC is used, as shown in Figure 6-8 (a). At the beginning of the US06, the system is fault-free. Then, after eight minutes, the fan is switched off for 36 minutes. Then, the pack is cycled using UDDS and 4C charge with the fan switched on again for an hour and 17 minutes which is sufficient for the cells to return to their non-fault status. Then, the fan speed is increased to 200% of its nominal value at the beginning of the LA92 cycle, which lasts for 45 minutes and is returned to its nominal speed.

Figure 6-8 (b) shows the system measured, estimated temperatures and the corresponding residuals and g values. In addition, the type, and the number of recorded flags during the waiting window of each fault. Firstly, the system shows no flags during the first 24 minutes of the beginning of the cycles and 16 minutes after the fan is switched off. Then the first two high flags are recorded at minute#24 from cell#24 and cell#60. After 10 minutes of a waiting window, eight high flags are recorded in the system. Since five or more high flags are recorded in the system, a fan failure or low airflow fault is declared. The proposed method is able to detect the fan failure fault even before the tempeture reaches the 45 ºC.

The fault is then cleared after one hour from the beginning of the test, and the residuals go back to the $\mu \pm 3\sigma$ threshold. This period ensures the ability of the proposed method to work on nonfaulty conditions with no false alarm. Then the fan speed is increased after approximately one hour from the fan off fault clearance. Figure 6-8 (b) shows that it takes nine minutes for the system to record the first low flag from cell#13. After ten minutes of the waiting time window, eight low flags are recorded, declaring airflow lower than the setpoint fault. The method shows less time in detecting this fault than the previous fault since the limits for J_2 is tuned lower than J_1. However, this fault detection time is highly dependent on the temperature rise of the cells, driving behavior and nature of the fault. It should be noted that in both faults, the middle cells report the faults faster than the side cells showing larger sensitivity of these middle cells to the heat dissipated by the cooling system.

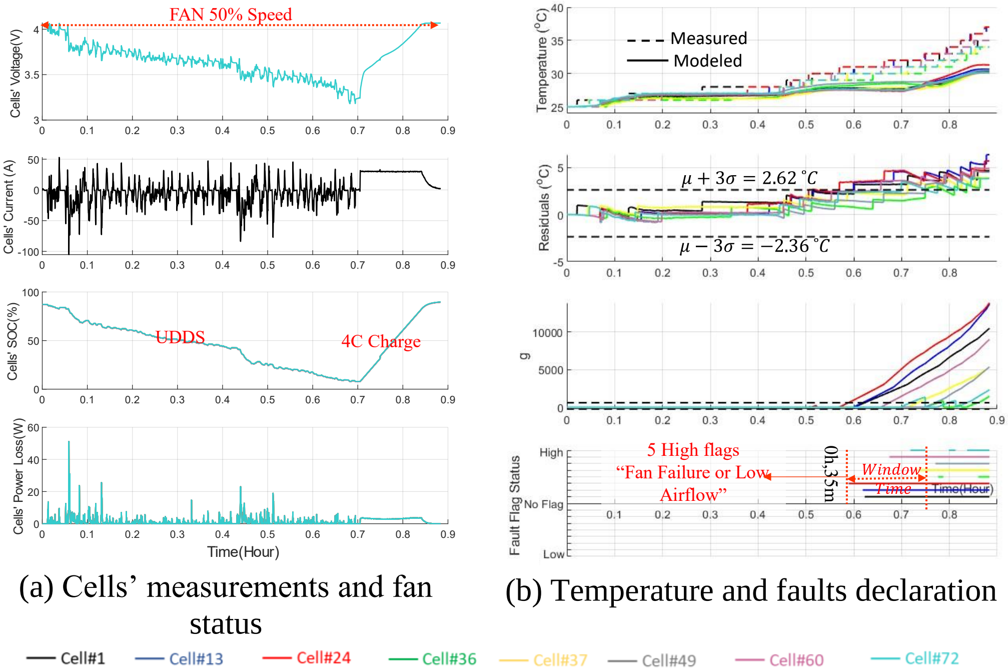


(a) Cells' measurements and fan status

(b) Temperature and faults declaration

Figure 6-8: Proposed method performance for a long driving scenario at 15 ºC with the fan off and higher airflow than setpoint faults

Another testing case with UDDS and 4C charge while the fan is set to 50% of its nominal speed to test the proposed method's sensitivity to low airflow faults. The fan speed is halved at the beginning of the UDDS cycle, as shown in Figure 6-9 (a). Figure 6-9 (b) shows the system measured, estimated temperatures and the corresponding residuals and g values of the eight sensors for this test case. The figure shows that no flags are recorded in the first 35 minutes. Then five high flags are recorded during the ten minutes waiting window, declaring fan or low airflow fault. The results shown in Figure 6-9 (b) demonstrate that the sensitivity of the proposed method is sufficient to detect less severe faults such as airflow lower than the setpoint.



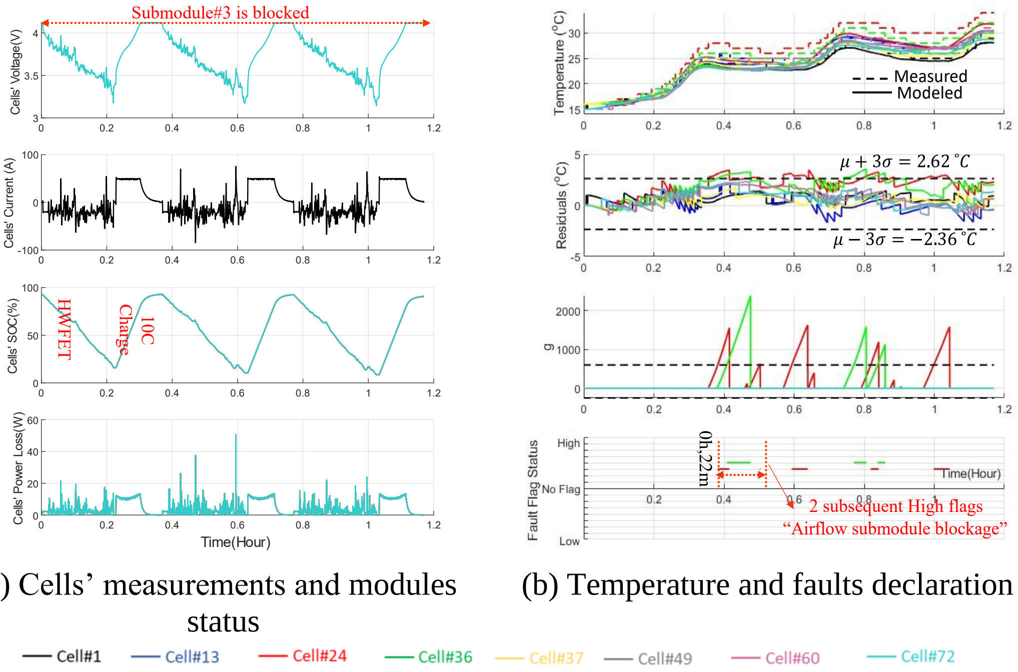
(a) Cells' measurements and fan status

(b) Temperature and faults declaration

Figure 6-9: Proposed method performance for UDDS&4C at 15 ºC with lower airflow than setpoint fault

However, the proposed method shows 220% higher time in detecting the fault than the fan off fault, which might be due to the low power loss and temperature rise experienced with UDDS and 4C charge.

Another testing case with repeated HWFET and 10C charge at 15 ºC ambient temperature simulating highway driving followed by a fast charge with the airflow blocked on submodule#3 containing cells#24 to 36 is also studied and shown in Figure 6-10 (a). Figure 6-10 (b) shows that cell#24 shows a high flag after 22 minutes from the beginning of the test.



(a) Cells' measurements and modules status

(b) Temperature and faults declaration

Figure 6-10: Proposed method performance for HWFET&10C at 15 ºC with submodule airflow blockage fault

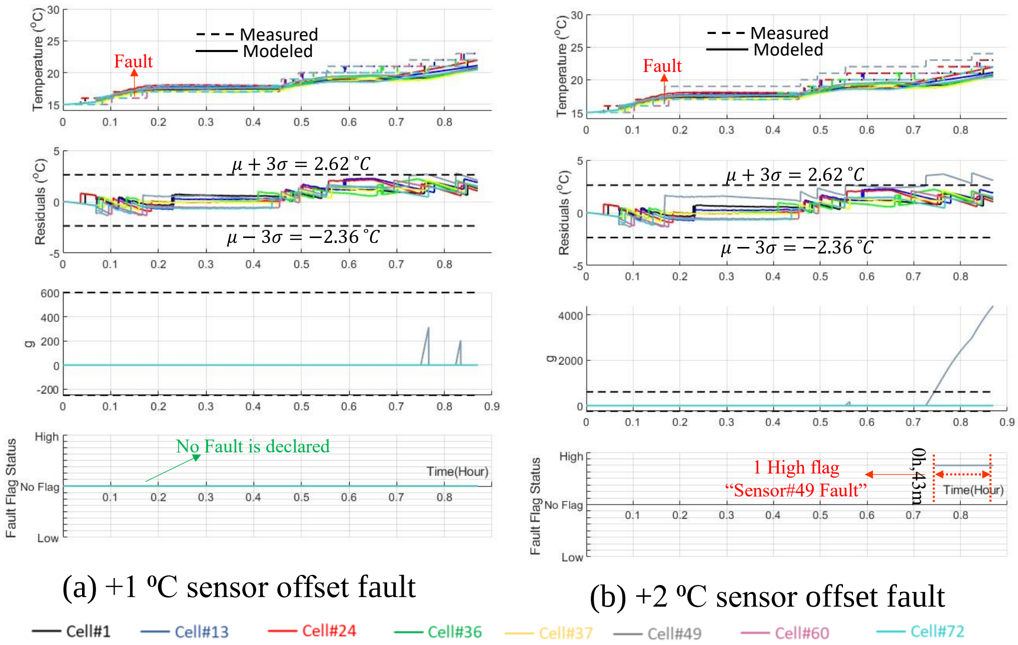
Two high flags are recorded during the ten minutes waiting window from cell#24, and cell#36 declaring airflow blockage fault in submodule#3. The flags also frequently appear as the fault is not cleared till the end of the cycle due to the safety of the test procedure. During the test, the residuals exceed μ±3σ thresholds by only 0.3 ºC or less. However, the proposed method's sensitivity is sufficient to detect the fault after a reasonable time.

6.3.3 Temperature sensor fault analysis

The proposed fault method is tested using different temperature sensor reading fault types, including offset, scale, and incorrect reading. The sensor faults are added to eight fault-free test cases at two different ambient temperatures, as described in Table 6.1. The temperature rise of these test cases is different, ranging from 7 to 18 ºC, which impacts the LP+FNN model performance. The temperature sensor reading faults are assumed to occur one at a time.

An offset error ranging from -4 to 4 ºC in 0.5 ºC steps is added to the eight sensors' readings during the studied cases. Figure 6-11 shows the proposed method performance for an example test case, namely UDDS&4C at 15 ºC ambient temperature with +1 ºC and +2 ºC offset faults added to sensor#49 after ten minutes from the beginning of the test. The figure shows that the model is not able to distinguish the +1 ºC offset fault despite showing some spikes in the g function, which is less than the J_1 threshold. However, the method is able to detect the +2 ºC offset fault after 33 minutes from the moment of fault occurrence. This detection time is highly dependent on the model accuracy for the test case and the fault offset value.

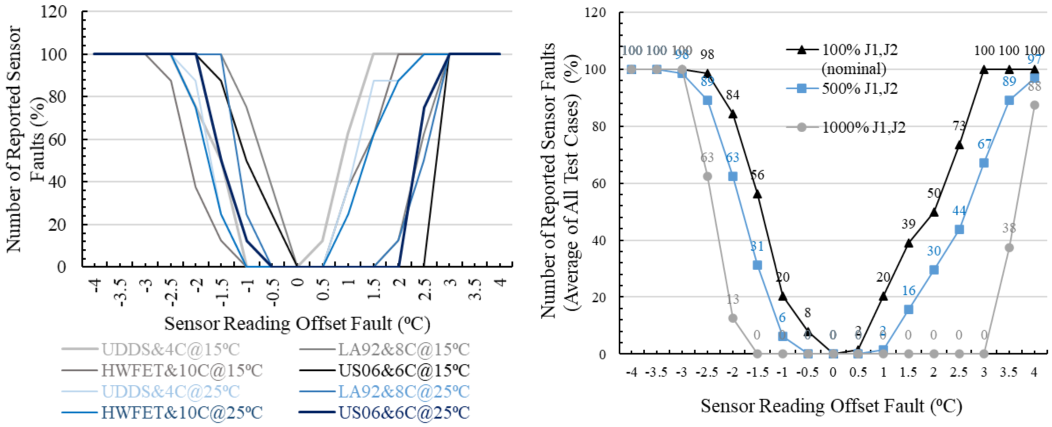
Figure 6-11 (a) shows the number of reported sensor faults for each studied test case versus the different offset values. The figure shows that the proposed method shows no false reported sensor faults in all zero offset cases. The figure also shows that the offset faults of the higher temperature rise cases, namely US06&6C and HWFET&10C, are hard to distinguish due to the challenging performance of the LP+FNN for these cases.



(a) +1 ºC sensor offset fault

(b) +2 ºC sensor offset fault

Figure 6-11: UDDS&4C at 15 ºC ambient temperature test case with +1 ºC and +2 ºC offset reading faults added to sensor#49



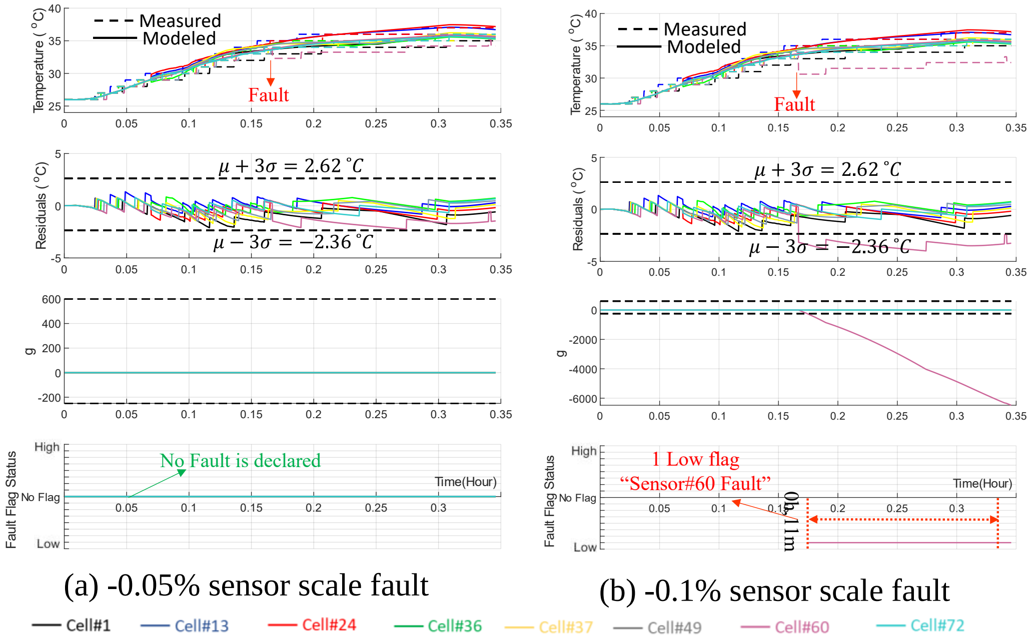
(a) Offset fault for different test cases at nominal *J* threshold values

(b) Average offset fault for different *J* threshold values

Figure 6-12: Proposed method performance for a range of temperature sensor reading offset faults and different test cases

Overall, for all test cases and ambient temperatures, the proposed method is able to detect offset faults with 100% accuracy, meaning the eight sensors' faults are reported, for temperature sensor reading offsets higher than ±3 ºC at the nominal J threshold values. Figure 6-11 (b) shows the impact of the adjustment of the J threshold on the proposed method's performance in detecting offset sensor faults. The figure shows that even with J threshold increases to ten times its nominal value, the proposed method is able to detect offsets faults of ±4 ºC with an accuracy of 88% or higher.

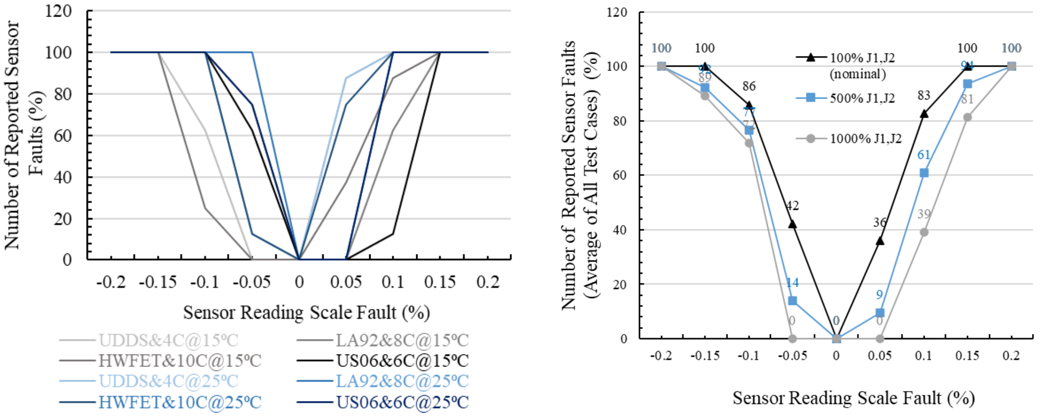
A temperature sensor reading scale fault is also investigated in the study. A scale reading faults ranging from -0.2% to 0.2% in 0.05 % steps is added to the eight sensors' readings during the studied test cases. Figure 6-13 shows the proposed method performance for another test case, namely US06&6C at 25 ºC ambient temperature with -0.05% and -0.1% scale faults added to sensor#60. The figure depicts that for the -0.05% scale fault, the g function increases to some extent during the test; but this rise is not sufficient to hit the J_1 threshold and report the fault accordingly. However, the proposed method is able to detect the -0.1% scale fault after a few minutes from the occurrence of the fault.



(a) -0.05% sensor scale fault

(b) -0.1% sensor scale fault

Figure 6-13: US06&6C at 25 ºC ambient temperature test case with -0.05% and -0.1% scale reading faults added to sensor#60



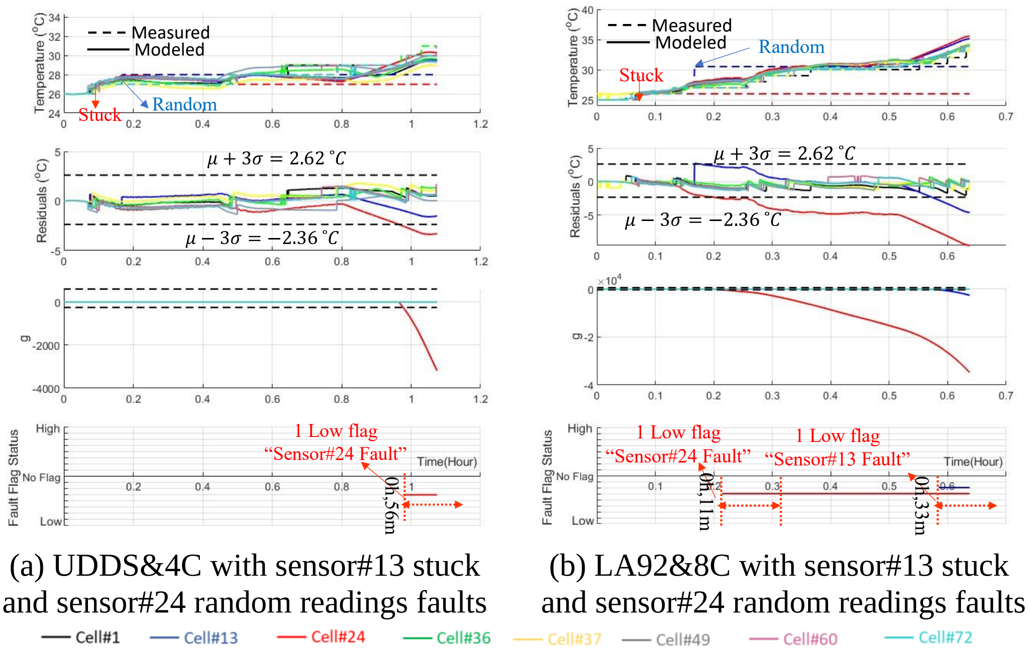
(a) Scale fault for different test cases at nominal J threshold values

(b) Scale fault for different J threshold values

Figure 6-14: Proposed method performance for a range of temperature sensor reading scale faults for different test cases

Figure 6-14 (a) shows the number of reported sensor faults for each of the studied test cases versus the different scale fault values. Similar to the offset fault, the proposed method's capability to detect the scale fault varies for each case, and for all zero scale fault cases, the proposed method does not show any false reported sensor faults. For all studied test cases and ambient temperatures, the proposed method is able to detect offset faults with 100% accuracy for temperature sensor scale faults equal to or higher than ±0.15% at the nominal J threshold value. Similar to Figure 6-12 (b), Figure 6-14 (b) shows the impact of the adjustment of the J threshold on the proposed method's performance for detecting the sensor reading scale faults. The proposed method shows 100% detection accuracy for the scale reading faults of ±0.2% or higher, with the J threshold being ten times higher than its nominal value.

Finally, the detection capability of the proposed method is validated using senor stuck reading faults. Two incorrect sensor reading scenarios are investigated, including sensor stuck and random readings faults. Figure 6-15 shows the performance of the proposed method for UDDS&4C and LA92&8C test cases at 25 ºC ambient temperature after adding incorrect temperature reading to sensors #13 and 24. The maximum temperature rise of the UDDS&4C test is 5 ºC, while the LA92&8C is 11 ºC. A stuck fault of 28, and a random reading of 30.5 ºC, are applied to sensor#24 sensor#13 five and ten minutes after the beginning of the test, respectively. The random sensor reading value is selected based on the average temperature rise of each test. This random value makes the residuals close to or inside the $\mu\pm3\sigma$ thresholds. The UDDS&4C, being the lowest temperature rise test, makes detecting any stuck or random reading error challenging to the proposed method. The proposed method is not able to detect the random reading error of 28 ºC and takes around 51 minutes to detect the stuck reading fault for the UDDS&4C test. However, the proposed method is able to detect both faults for the LA92&8C test due to the dynamic change in the temperature of this LA92&8C. In addition, the proposed takes fewer minutes than the UDDS&4C case to detect the same stuck sensor reading fault. Based on the analysis above, the proposed method is able to efficiently detect and declare temperature sensor stuck reading faults when error exceeds the $\mu\pm3\sigma$ thresholds. The method also struggles, and in some cases could not detect, random sensor readings equal to or close to the average of the temperature rise of the test case. The studied test cases here are very specific and challenging to show the limitations of the proposed method. However, the any random or sensor reading fault can be detected with longer cycle time and the accumulation of the residuals.



(a) UDDS&4C with sensor#13 stuck and sensor#24 random readings faults

(b) LA92&8C with sensor#13 stuck and sensor#24 random readings faults

Figure 6-15: Proposed method performance for UDDS&4C and LA92&8C at 25 ºC with incorrect sensor reading faults

6.4 Summary

In this chapter, a method to detect and identify different thermal faults of lithium-ion battery packs (LIBPs) was proposed. The method works by comparing the measured and the estimated temperatures from an accurate physics-based combined with a neural network model. The proposed fault detection and identification method was validated using various thermal faults, including fan failure, airflow lower higher than setpoint, airflow blockage and temperature sensor reading faults. The proposed method showed no false faults when tested by a customized highway driving test which was not included in developing and tuning the parameters of the proposed method. The proposed method also was able to detect and declare different cooling system faults within 10 to 35 minutes after the fault occurrence and before even hitting the maximum temperature of 45 ºC. The fault detection time appeared to be dependent on the fault's nature and driving behavior. The fan off fault, being the most severe fault, is shown to be fast detected, while lower than normal airflow and submodule airflow blockage faults are shown to be more challenging to be detected. Hence, it is always important to test any thermal fault detection methods using these scenarios to ensure the robustness of the method in detecting similar less severe faults. Finally, the proposed method was tested using different sensor reading faults, including offset, scale, and incorrect reading. The method demonstrated being capable of detecting sensor reading offset and scale faults of ±3 ºC and ±0.15% or more, respectively with 100% accuracy.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The main contribution of this thesis is proposing an accurate and computationally feasible state of charge (SOC) and surface temperature estimation models using deep neural networks (DNN). Two DNNs are utilized in this thesis, including a feedforward neural network (FNN) enhanced with external filters and a recurrent neural network with a long short-term memory layer (LSTM). The computational load of the DNN SOC estimation models is benchmarked against the conventional Kaman filters using a processor in the loop (PIL) platform. The capability of the DNNs to estimate the surface temperature of different batteries is also tested at realistic and challenging driving, charge, and aging conditions. Finally, an integrated physics and neural network-based (LP+FNN) model is developed to model the temperature for a multi-cell lithium-ion battery pack (LIBP). The proposed LP+FNN model also demonstrates its capability to help in detecting and identifying different thermal faults.

After presenting a comprehensive overview of the challenges of the developed SOC estimation algorithms for LIBPs, the necessity of providing a real-time platform to assess the computational load for different state estimation algorithms is clarified. It is concluded that the existing SOC algorithms lack real-time testing to ensure their feasibility when run on battery management system (BMS) processors. Besides, DNN are practical to run on the BMS microprocessor without violating the execution time and memory limits. It is also concluded that developing accurate temperature estimation models for lithium-ion batteries is beneficial to improve the reliability and safety of electric vehicles (EVs). Accurate temperature estimation models can work with temperature sensors to detect different types of thermal faults, reducing the risk of the thermal runway and EV fire.

Two DNN SOC estimation algorithms are benchmarked against two conventional Kalman filters when tested on two different microprocessors using the PIL platform. The DNN SOC estimation algorithms include FNN and LSTM neural networks SOC estimation algorithms. An external Butterworth first-order filter is applied to the inputs to the FNN algorithm to add a memory feature to the algorithm. The other two SOC estimation algorithms include an extended Kalman filter (EKF) and EKF with a recursive least squares filter (EKF-RLS). The four SOC estimation algorithms are shown to be capable of accurately estimating the SOC of a drive cycle with less than 5% error. The LSTM algorithm shows the most time to execute, 0.71 ms on the higher power S32K344 processor, the FNN only had a lower execution time than the two EKF algorithms, 0.14 ms versus around 0.25 ms, showing that machine learning algorithms can have lower computational complexity to Kalman filters. The FNN and LSTM are also found to have less than 5% of the RAM use of the EKF algorithms. None of the tested algorithms exceed the Flash memory limits showing that the flash memory is not an actual limitation.

The DNN capability to efficiently learn the relationship between battery surface temperature and measured terminal parameters is also discussed. A FNN enhanced with filters, and LSTM models are proposed to estimate the surface temperature of different lithium-ion batteries. These models are trained and tested at a range of driving, charging and health conditions, with up to 35 ^oC of temperature rise and 450 A of current for the fast charge tests. The models are also deployed to an NXP S32K344 BMS microprocessor to assess their computational load. A comprehensive analysis of the low pass Butter-worth corner frequencies applied on the FNN inputs is investigated. Both FNN enhanced with filters, and LSTM models show a reasonable temperature estimation accuracy when benchmarked against other temperature estimation models in the literature. The proposed FNN with 1 mHz filter model shows a maximum error of less than 4.5 ^oC when tested using challenging, low-temperature drive cycles. In addition, the FNN with 1 mHz filter shows a maximum error of 0.3 ^oC when tested using 4C rate fast charge rate and

0.8 ºC error when trained on new cell condition and tested on an aged cell. The FNN and LSTM models with around 3,000 learnable parameters are deployed to the BMS microprocessor. The FNN shows only about 1/3 the execution time and much RAM and Flash memory use than of the LSTM, showing that the FNN is less computationally complex than the LSTM. Overall, the results show that machine learning algorithms can efficiently learn the relationship between battery surface temperature and measured terminal parameters, making it easier for non-experts to build less complex temperature estimation models. In addition, both DNN algorithms show a feasible performance when deployed to a BMS microprocessor and do not require excessive execution time or memory.

Finally, an integrated physics and machine learning-based (LP+FNN) model is proposed to model the temperature of an air-cooled LIBP. The model combines a thermal lumped parameters (LP) model and a feedforward neural network (FNN). Comprehensive details of the parameter determination of the thermal lumped parameters and training of the FNN are introduced. The model is tested under different drive cycles and charge conditions. Overall, the LP+FNN model shows an average root mean squares error (RMSE) of 0.4 ºC which is 70% and 77% less error than the LP and FNN models, respectively. The model is also tested using realistic test cases while the cells are warmer or colder than the coolant inlet and ambient temperatures. Overall, the LP+FNN shows less than 1 ºC average RMSE and MAXE of 3 ºC for all studied test cases. The LP+FNN model is then utilized to detect and identify different thermal faults of air-cooled LIBPs by comparing the measured temperatures and the estimated temperature from the LP+FNN model. The proposed fault detection and identification method is validated using various thermal faults, including fan system failure, airflow lower higher than setpoint, airflow blockage and temperature sensor reading faults. The proposed method shows no false faults when tested by a customized highway driving test which is not included in developing and tuning the parameters of the method. The proposed method also is able to detect different cooling system faults within 10 to 35 minutes after the fault occurrence before reaching the protection maximum temperature limit. Finally, the proposed method is tested using different sensor reading faults, including offset, scale, and incorrect reading. The proposed method demonstrated being capable of detecting sensor reading offset and scale faults of ±3 ºC and ±0.15% or more, respectively with 100% accuracy.

7.2 Future work

This thesis includes a number of improvements to battery SOC and temperature estimation modeling using DNNs. In addition, the developed DNN models' computational feasibility to run for multi-cell LIBPs is investigated. Also, the proposed DNN temperature estimation models are tested using batteries at different dynamic and fast charge, aging conditions. Finally, a LP+FNN temperature estimation model is applied to detect and identify different thermal faults.

With the fast-growing EVs connected to the cloud, the amount of generated data significantly increases than in the past. The first recommendation for the work presented in this thesis is to test the proposed DNN SOC and temperature estimation models using data collected from on-road electric vehicles. The performance of machine learning models, in general, improves by incorporating more information extracted from on-road driving scenarios such as driving routes, air conditioning and heating loads. However, these data are not always suitable to be fed to DNN, mostly due to the quality of the sensors. Hence, there is room for further investigation and improvement of the usage of cloud-collected data to enhance the performance of the DNN SOC and temperature estimation models.

Another recommendation for the modeling part is to add more than one fan speed setting to the proposed LP+FNN temperature estimation models in this thesis. The proposed temperature estimation models are developed on cells and air-cooled LIBP working on nominal fan speed, meaning one heat convection rate. This assumption is justified by operating the cells and the pack at a normal temperature operating margin, .i.e, 15 to 40 ºC. Adding more than one convection rate to the thermal model would enhance the capability of the models to mimic scenarios where an active thermal management system works when the temperature goes outside the operating margin.

A digital twin with a virtual model of a battery system mirroring the electro-thermal behavior of the LIBPs and providing a parallel fault thermal monitoring system to the battery is another substantial recommendation for future use work. Instead of running the proposed LP+FNN and fault detection method onboard, it runs on the cloud. This also will allow for updating the LP+FNN model online by adjusting the FNN parameters using the collected data. This can be enhanced by the continuous improvement of newly emerging technologies such as IoT devices and cloud computing.

7.3 Publications

7.3.1 Journal papers

1. **M. Naguib**, P. Kollmeyer, and A. Emadi, “Lithium-ion battery pack robust state of charge estimation, cell inconsistency, and balancing: review,” *IEEE Access*, vol. 9, pp. 50570-50582, 2021, doi: 10.1109/ACCESS.2021.3068776.
2. **M. Naguib**, P. Kollmeyer, and A. Emadi, "Application of deep neural networks for lithium-ion battery surface temperature estimation under driving and fast charge conditions," in *IEEE Transactions on Transportation Electrification*, 2022, doi: 10.1109/TTE.2022.3200225.
3. C. Vidal, P. Malysz, **M. Naguib**, A. Emadi, and P. J. Kollmeyer, “Estimating battery state of charge using recurrent and non-recurrent Neural Networks,” *Journal of Energy Storage*, p. 103660, 2021.

4. C.Vidal, P. Kollmeyer, **M., Naguib**, P. Malysz, et al., “Robust xev battery state-of-charge estimator design using a feedforward deep neural network," *SAE Int. J. Adv. & Curr. Prac. in Mobility*, vol 2, p.2872-2880, 2020, doi: 10.4271/2020-01-1181.

7.3.2 Conference papers

1. **M. Naguib,** P.J. Kollmeyer, O. Gross, and A. Emadi, “Microprocessor execution time and memory use of battery state of charge estimation algorithms”, *SAE World Congress*, Detroit, MI, April 2022.
2. **M. Naguib**, L. Bruck, and A. Emadi, "Neural network-based online energy management for multi-mode power split hybrid vehicles," *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2022, pp. 237-242, doi: 10.1109/ITEC53557.2022.9813876.
3. **M. Naguib**, C.Vidal, P. Kollmeyer, P. Malysz, et al., “Comparative study between equivalent circuit and recurrent neural network battery voltage models,” *2021 SAE WCX Technical Paper*, 2021, https://doi.org/10.4271/2021-01-0759.
4. **M. Naguib**, P. Kollmeyer, C. Vidal, and A. Emadi, “Accurate surface temperature estimation of lithium-ion batteries using feedforward and recurrent artificial neural networks,” *2021 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2021, pp. 52-57, doi: 10.1109/ITEC51675.2021.9490043.

5. P. J. Kollmeyer, **M. Naguib**, F. Khanum, and A. Emadi, "A blind modeling tool for standardized evaluation of battery state of charge estimation algorithms," *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2022, pp. 243-248, doi: 10.1109/ITEC53557.2022.9813996.
6. J. Duque, P. J. Kollmeyer, **M. Naguib,** and A. Emadi, "Battery dual extended kalman filter state of charge and health estimation strategy for traction applications," *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2022, pp. 975-980, doi: 10.1109/ITEC53557.2022.9813961.
7. Y. Liang, A. Emadi, O. Gross, O., C. Vidal, M. Canova, S. Panchal, P. Kollmeyer, **M. Naguib**, and F. Khanum, "A comparative study between physics, electrical and data driven lithium-ion battery voltage modeling approaches," *SAE Technical Paper* 2022-01-0700, 2022, doi:10.4271/2022-01-0700.

References

[1] M. Baumann, L. Wildfeuer, S. Rohr, and M. Lienkamp, “Parameter variations within li-ion battery packs – theoretical investigations and experimental quantification,” *J. Energy Storage*, vol. 18, no. February, pp. 295–307, 2018, doi: 10.1016/j.est.2018.04.031.

[2] T. Bruen and J. Marco, “Modelling and experimental evaluation of parallel connected lithium ion cells for an electric vehicle battery system,” *J. Power Sources*, vol. 310, pp. 91–101, 2016, doi: 10.1016/j.jpowsour.2016.01.001.

[3] S. Rothgang, T. Baumhöfer, H. van Hoek, T. Lange, R. W. De Doncker, and D. U. Sauer, “Modular battery design for reliable, flexible and multi-technology energy storage systems,” *Appl. Energy*, vol. 137, pp. 931–937, 2015, doi: 10.1016/j.apenergy.2014.06.069.

[4] A. Bavand, S. Ali Khajehoddin, M. Ardakani, and A. Tabesh, "Online Estimations of Li-ion Battery SOC and SOH Applicable to Partial Charge/Discharge," in *IEEE Transactions on Transportation Electrification*, doi: 10.1109/TTE.2022.3162164.

[5] K. S. Ng, Y. F. Huang, C. S. Moo, and Y. C. Hsieh, “An enhanced coulomb counting method for estimating state-of-charge and state-of-health of lead-acid batteries,” *INTELEC, Int. Telecommun. Energy Conf.*, pp. 2–6, 2009, doi: 10.1109/INTLEC.2009.5351796.

[6] Y. Zhang, W. Song, S. Lin, and Z. Feng, “A novel model of the initial state of charge estimation for LiFePO 4 batteries,” *J. Power Sources*, vol. 248, pp. 1028–1033, 2014, doi: 10.1016/j.jpowsour.2013.09.135.

[7] S. Tong, M. P. Klein, and J. W. Park, “On-line optimization of battery open circuit voltage for improved state-of-charge and state-of-health estimation,” *J. Power Sources*, vol. 293, pp. 416–428, 2015, doi: 10.1016/j.jpowsour.2015.03.157.

[8] C. Truchot, M. Dubarry, and B. Y. Liaw, “State-of-charge estimation and uncertainty for lithium-ion battery strings,” *Appl. Energy*, vol. 119, pp. 218–227, 2014, doi: 10.1016/j.apenergy.2013.12.046.

[9] U. Westerhoff, T. Kroker, K. Kurbach, and M. Kurrat, “Electrochemical impedance spectroscopy based estimation of the state of charge of lithium-ion batteries,” *J. Energy Storage*, vol. 8, pp. 244–256, 2016, doi: 10.1016/j.est.2016.09.001.

[10] C. Zhang, Y. Jiang, J. Jiang, G. Cheng, W. Diao, and W. Zhang, “Study on battery pack consistency evolutions and equilibrium diagnosis for serial-connected lithium-ion batteries,” *Appl. Energy*, vol. 207, pp. 510–519, 2017, doi: 10.1016/j.apenergy.2017.05.176.

[11] S. Sepasi, L. R. Roose, and M. M. Matsuura, “Extended Kalman filter with a fuzzy method for accurate battery pack state of charge estimation,” *Energies*, vol. 8, no. 6, pp. 5217–5233, 2015, doi: 10.3390/en8065217.

[12] Z. Zhang, X. Cheng, Z. Y. Lu, and D. J. Gu, “SOC Estimation of Lithium-Ion Battery Pack Considering Balancing Current,” *IEEE Trans. Power Electron.*, vol. 33, no. 3, pp. 2216–2226, 2018, doi: 10.1109/TPEL.2017.2700324.

[13] C. N. Van, and T. N. Vinh, “Soc estimation of the lithium-ion battery pack using a sigma point Kalman filter based on a cell’s second order dynamic model,” *Appl. Sci.*, vol. 10, no. 5, 2020, doi: 10.3390/app10051896.

[14] S. L. Wang, C. Fernandez, C. Y. Zou, C. M. Yu, L. Chen, and L. Zhang, “A comprehensive working state monitoring method for power battery packs considering state of balance and aging correction,” *Energy*, vol. 171, pp. 444–455, 2019, doi: 10.1016/j.energy.2019.01.020.

[15] Q. Du, Q. Han, Y. Zhang, Z. Liu, S. Tian, and Z. Zhang, “Adopting combined strategies to make state of charge (SOC) estimation for practical use,” *J. Renew. Sustain. Energy*, vol. 10, no. 3, 2018, doi: 10.1063/1.5024031.

[16] J. Li, J. Klee Barillas, C. Guenther, and M. A. Danzer, “Multicell state estimation using variation based sequential Monte Carlo filter for automotive battery packs,” *J. Power Sources*, vol. 277, pp. 95–103, 2015, doi: 10.1016/j.jpowsour.2014.12.010.

[17] C. Zhang, W. Allafi, Q. Dinh, P. Ascencio, and J. Marco, “Online estimation of battery equivalent circuit model parameters and state of

charge using decoupled least squares technique," *Energy*, vol. 142, pp. 678–688, 2018, doi: 10.1016/j.energy.2017.10.043.

[18] Y. Li, J. Chen, and F. Lan, "Enhanced online model identification and state of charge estimation for lithium-ion battery under noise corrupted measurements by bias compensation recursive least squares," *J. Power Sources*, vol. 456, no. 381, p. 227984, 2020, doi: 10.1016/j.jpowsour.2020.227984.

[19] X. Hu, F. Sun, and Y. Zou, "Estimation of state of charge of a Lithium-Ion battery pack for electric vehicles using an adaptive luenberger observer," *Energies*, vol. 3, no. 9, pp. 1586–1603, 2010, doi: 10.3390/en3091586.

[20] C. Vidal *et al.*, "Hybrid energy storage system state-of-charge estimation using artificial neural network for micro-hybrid applications," *2018 IEEE Transp. Electrif. Conf. Expo, ITEC 2018*, pp. 868–873, 2018, doi: 10.1109/ITEC.2018.8450251.

[21] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6730–6739, 2018, doi: 10.1109/TIE.2017.2787586.

[22] J. C. Alvarez Anton, P. J. Garcia Nieto, C. Blanco Viejo, and J. A. Vilan Vilan, "Support vector machines used to estimate the battery state of charge," *IEEE Trans. Power Electron.*, vol. 28, no. 12, pp. 5919–5926, 2013, doi: 10.1109/TPEL.2013.2243918.

[23] J. Wu, Y. Wang, X. Zhang, and Z. Chen, “A novel state of health estimation method of Li-ion battery using group method of data handling,” *J. Power Sources*, vol. 327, pp. 457–464, 2016, doi: 10.1016/j.jpowsour.2016.07.065.

[24] K. Rumpf, M. Naumann, and A. Jossen, “Experimental investigation of parametric cell-to-cell variation and correlation based on 1100 commercial lithium-ion cells,” *J. Energy Storage*, vol. 14, pp. 224–243, 2017, doi: 10.1016/j.est.2017.09.010.

[25] T. Baumhöfer, M. Brühl, S. Rothgang, and D. U. Sauer, “Production caused variation in capacity aging trend and correlation to initial cell performance,” *J. Power Sources*, vol. 247, pp. 332–338, 2014, doi: 10.1016/j.jpowsour.2013.08.108.

[26] R. Xiong, Y. Zhang, H. He, X. Zhou, and M. G. Pecht, “A double-scale, particle-filtering, energy state prediction algorithm for lithium-ion batteries,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1526–1538, 2017, doi: 10.1109/TIE.2017.2733475.

[27] F. Sun, and R. Xiong, “A novel dual-scale cell state-of-charge estimation approach for series-connected battery pack used in electric vehicles,” *J. Power Sources*, vol. 274, pp. 582–594, 2015, doi: 10.1016/j.jpowsour.2014.10.119.

[28] M. J. Brand, E. I. Kolp, P. Berg, T. Bach, P. Schmidt, and A. Jossen, "Electrical resistances of soldered battery cell connections," *J. Energy Storage*, vol. 12, pp. 45–54, 2017, doi: 10.1016/j.est.2017.03.019.

[29] G. J. Offer, V. Yufit, D. A. Howey, B. Wu, and N. P. Brandon, "Module design and fault diagnosis in electric vehicle batteries," *J. Power Sources*, vol. 206, pp. 383–392, 2012, doi: 10.1016/j.jpowsour.2012.01.087.

[30] B. Wu, V. Yufit, M. Marinescu, G. J. Offer, R. F. Martinez-Botas, and N. P. Brandon, "Coupled thermal-electrochemical modelling of uneven heat generation in lithium-ion battery packs," *J. Power Sources*, vol. 243, pp. 544–554, 2013, doi: 10.1016/j.jpowsour.2013.05.164.

[31] J. Kim, "Discrete Wavelet transform-based feature extraction of experimental voltage signal for li-ion cell consistency," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1150–1161, 2016, doi: 10.1109/TVT.2015.2414936.

[32] A. Cordoba-Arenas, S. Onori, and G. Rizzoni, "A control-oriented lithium-ion battery pack model for plug-in hybrid electric vehicle cycle-life studies and system design with consideration of health management," *J. Power Sources*, vol. 279, pp. 791–808, 2015, doi: 10.1016/j.jpowsour.2014.12.048.

[33] R. Gogoana, M. B. Pinson, M. Z. Bazant, and S. E. Sarma, "Internal resistance matching for parallel-connected lithium-ion cells and impacts on

battery pack cycle life," *J. Power Sources*, vol. 252, pp. 8–13, 2014, doi: 10.1016/j.jpowsour.2013.11.101.

[34] N. Yang, X. Zhang, B. Shang, and G. Li, "Unbalanced discharging and aging due to temperature differences among the cells in a lithium-ion battery pack with parallel combination," *J. Power Sources*, vol. 306, pp. 733–741, 2016, doi: 10.1016/j.jpowsour.2015.12.079.

[35] M. S. Wu, C. Y. Lin, Y. Y. Wang, C. C. Wan, and C. R. Yang, "Numerical simulation for the discharge behaviors of batteries in series and/or parallel-connected battery pack," *Electrochim. Acta*, vol. 52, no. 3, pp. 1349–1357, 2006, doi: 10.1016/j.electacta.2006.07.036.

[36] L. Wang, Y. Cheng, and X. Zhao, "A LiFePO4 battery pack capacity estimation approach considering in-parallel cell safety in electric vehicles," *Appl. Energy*, vol. 142, pp. 293–302, 2015, doi: 10.1016/j.apenergy.2014.12.081.

[37] Y. Zhang, R. Zhao, J. Dubie, T. Jahns, and L. Juang, "Investigation of current sharing and heat dissipation in parallel-connected lithium-ion battery packs," *ECCE 2016 - IEEE Energy Convers. Congr. Expo. Proc.*, 2016, doi: 10.1109/ECCE.2016.7855217.

[38] M. Fleckenstein, O. Bohlen, M. A. Roscher, and B. Bäker, "Current density and state of charge inhomogeneities in Li-ion battery cells with LiFePO4 as cathode material due to temperature gradients," *J. Power Sources*, vol. 196, no. 10, pp. 4769–4778, 2011, doi: 10.1016/j.jpowsour.2011.01.043.

[39] [N. Ganesan *et al.*, “Physics based modeling of a series parallel battery pack for asymmetry analysis, predictive control and life extension,” *J. Power Sources*, vol. 322, pp. 57–67, 2016, doi: 10.1016/j.jpowsour.2016.05.005.

[40] F. Wen, C. Lin, J. Jiang, and Z. Wang, “A new evaluation method to the consistency of lithium-ion batteries in electric vehicles,” *Asia-Pacific Power Energy Eng. Conf. APPEEC*, pp. 0–3, 2012, doi: 10.1109/APPEEC.2012.6307481.

[41] X. Zhang, Y. Wang, J. Wu, and Z. Chen, “A novel method for lithium-ion battery state of energy and state of power estimation based on multi-time-scale filter,” *Appl. Energy*, vol. 216, no. September 2017, pp. 442–451, 2018, doi: 10.1016/j.apenergy.2018.02.117.

[42] Y. Zheng *et al.*, “Cell state-of-charge inconsistency estimation for LiFePO4 battery pack in hybrid electric vehicles using mean-difference model,” *Appl. Energy*, vol. 111, pp. 571–580, 2013, doi: 10.1016/j.apenergy.2013.05.048.

[43] F. Feng *et al.*, “A practical and comprehensive evaluation method for series-connected battery pack models,” *IEEE Trans. Transp. Electrif.*, vol. 6, no. 2, pp. 391–416, 2020, doi: 10.1109/TTE.2020.2983846.

[44] B. Duan, Z. Li, P. Gu, Z. Zhou, and C. Zhang, “Evaluation of battery inconsistency based on information entropy,” *J. Energy Storage*, vol. 16, pp. 160–166, 2018, doi: 10.1016/j.est.2018.01.010.

[45] L. Wang, L. Wang, C. Liao, and W. Zhang, "Research on multi-parameter evaluation of electric vehicle power battery consistency based on principal component analysis," *J. Shanghai Jiaotong Univ.*, vol. 23, no. 5, pp. 711–720, 2018, doi: 10.1007/s12204-018-1987-9.

[46] Y. Jiang, J. Jiang, C. Zhang, W. Zhang, Y. Gao, and C. Mi, "A Copula-based battery pack consistency modeling method and its application on the energy utilization efficiency estimation," *Energy*, vol. 189, p. 116219, 2019, doi: 10.1016/j.energy.2019.116219.

[47] C. Y. Chun, S. H. Yoon, B. H. Cho, and J. Kim, "A state-of-charge and capacity estimation algorithm for lithium-ion battery pack utilizing filtered terminal voltage," *World Electr. Veh. J.*, vol. 7, no. 1, pp. 71–75, 2015, doi: 10.3390/wevj7010071.

[48] Y. Hua, A. Cordoba-Arenas, N. Warner, and G. Rizzoni, "A multi time-scale state-of-charge and state-of-health estimation framework using nonlinear predictive filter for lithium-ion battery pack with passive balance control," *J. Power Sources*, vol. 280, pp. 293–312, 2015, doi: 10.1016/j.jpowsour.2015.01.112.

[49] D. Huang, Z. Chen, C. Zheng, and H. Li, "A model-based state-of-charge estimation method for series-connected lithium-ion battery pack considering fast-varying cell temperature," *Energy*, vol. 185, pp. 847–861, 2019, doi: 10.1016/j.energy.2019.07.063.

[50] J. Wei, G. Dong, Z. Chen, and Y. Kang, “System state estimation and optimal energy control framework for multicell lithium-ion battery system,” *Appl. Energy*, vol. 187, pp. 37–49, 2017, doi: 10.1016/j.apenergy.2016.11.057.

[51] H. Dai, X. Wei, Z. Sun, J. Wang, and W. Gu, “Online cell soc estimation of li-ion battery packs using a dual time-scale kalman filtering for ev applications,” *Appl. Energy*, vol. 95, pp. 227–237, 2012, doi: 10.1016/j.apenergy.2012.02.044.

[52] Y. Zheng, W. Gao, M. Ouyang, L. Lu, L. Zhou, and X. Han, “State-of-charge inconsistency estimation of lithium-ion battery pack using mean-difference model and extended Kalman filter,” *J. Power Sources*, vol. 383, no. February, pp. 50–58, 2018, doi: 10.1016/j.jpowsour.2018.02.058.

[53] G. L. Plett, “Efficient battery pack state estimation using bar-delta filtering,” *24th Int. Batter. Hybrid Fuel Cell Electr. Veh. Symp. Exhib. 2009, EVS 24*, vol. 1, pp. 163–170, 2009.

[54] X. Chen, H. Lei, and R. Xiong, “A bias correction based state-of-charge estimation method for multi-cell battery pack under different working conditions,” *IEEE Access*, vol. 6, pp. 78184–78192, 2018, doi: 10.1109/ACCESS.2018.2884844.

[55] F. Sun, R. Xiong, and H. He, “A systematic state-of-charge estimation framework for multi-cell battery pack in electric vehicles using bias

correction technique," *Appl. Energy*, vol. 162, pp. 1399–1409, 2016, doi: 10.1016/j.apenergy.2014.12.021.

[56] Z. Zhang, H. Gui, D. J. Gu, Y. Yang, and X. Ren, "A hierarchical active balancing architecture for lithium-ion batteries," *IEEE Trans. Power Electron.*, vol. 32, no. 4, pp. 2757–2768, 2017, doi: 10.1109/TPEL.2016.2575844.

[57] W. Luo, J. Lv, W. Song, and Z. Feng, "Study on passive balancing characteristics of serially connected lithium-ion battery string," *ICEMI 2017 - Proc. IEEE 13th Int. Conf. Electron. Meas. Instruments*, vol. 2018-Janua, pp. 489–495, 2017, doi: 10.1109/ICEMI.2017.8265862.

[58] A. Probstl, S. Park, S. Narayanaswamy, S. Steinhorst, and S. Chakraborty, "Soh-Aware active cell balancing strategy for high power battery packs," *Proc. 2018 Des. Autom. Test Eur. Conf. Exhib. DATE 2018*, vol. 2018-Janua, no. 291763, pp. 431–436, 2018, doi: 10.23919/DATE.2018.8342048.

[59] A. C. Baughman and M. Ferdowsi, "Double-tiered switched-capacitor battery charge equalization technique," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2277–2285, 2008, doi: 10.1109/TIE.2008.918401.

[60] M. Daowd, M. Antoine, N. Omar, P. van den Bossche, and J. van Mierlo, "Single switched capacitor battery balancing system enhancements," *Energies*, vol. 6, no. 4, pp. 2149–2179, 2013, doi: 10.3390/en6042149.

[61] R. K. Vardhan, T. Selvathai, R. Reginald, P. Sivakumar, and S. Sundaresh, "Balancing Circuit for Hybrid Electric Vehicles," pp. 2293–2298, 2017.

[62] X. Cui, W. Shen, Y. Zhang, and C. Hu, "A fast multi-switched inductor balancing system based on a fuzzy logic controller for lithium-ion battery packs in electric vehicles," *Energies*, vol. 10, no. 7, 2017, doi:

[63] D. V. Cadar, D. M. Petreus, and T. M. Patarau, "An energy converter method for battery cell balancing," *ISSE 2010 - 33rd Int. Spring Semin. Electron. Technol. Polym. Electron. Nanotechnologies Towar. Syst. Integr. - Conf. Proc.*, pp. 290–293, 2010, doi: 10.1109/ISSE.2010.5547305.

[64] 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, doi: 10.1109/TVT.2011.2153886.

[65] A. M. Imtiaz, F. H. Khan, and H. Kamath, "A low-cost time shared cell balancing technique for future lithium-ion battery storage system featuring regenerative energy distribution," *Conf. Proc. - IEEE Appl. Power Electron. Conf. Expo. - APEC*, pp. 792–799, 2011, doi: 10.1109/APEC.2011.5744686.

[66] P. Connected *et al.*, "Active cell balancing control strategy for parallel connected LiFePO4 batteries," *CSEE J. Power Energy Syst.*, vol. 7, no. 1, pp. 86–92, 2020, doi: 10.17775/cseejpes.2020.00740.

[67] J. Yan, Z. Cheng, G. Xu, H. Qian, and Y. Xu, “Fuzzy control for battery equalization based on state of charge,” *IEEE Veh. Technol. Conf.*, 2010, doi: 10.1109/VETECF.2010.5594380.

[68] X. Cui, W. Shen, Y. Zhang, and C. Hu, “A novel active online state of charge based balancing approach for Lithium-ion battery packs during fast charging process in electric vehicles,” *Energies*, vol. 10, no. 11, pp. 1–19, 2017, doi: 10.3390/en10111766.

[69] E. Chatzinikolaou and D. J. Rogers, “Cell soc balancing using a cascaded full-bridge multilevel converter in battery energy storage systems,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5394–5402, 2016, doi: 10.1109/TIE.2016.2565463.

[70] Y. S. Lee and G. T. Cheng, “Quasi-resonant zero-current-switching bidirectional converter for battery equalization applications,” *IEEE Trans. Power Electron.*, vol. 21, no. 5, pp. 1213–1224, 2006, doi: 10.1109/TPEL.2006.880349.

[71] Y. Chen, X. Liu, H. K. Fathy, J. Zou, and S. Yang, “A graph-theoretic framework for analyzing the speeds and efficiencies of battery pack equalization circuits,” *Int. J. Electr. Power Energy Syst.*, vol. 98, no. 92, pp. 85–99, 2018, doi: 10.1016/j.ijepes.2017.11.039.

[72] A. Turksoy, A. Teke, and A. Alkaya, “A comprehensive overview of the dc-dc converter-based battery charge balancing methods in electric

vehicles," *Renew. Sustain. Energy Rev.*, vol. 133, no. October 2019, p. 110274, 2020, doi: 10.1016/j.rser.2020.110274.

[73] M. Daowd, N. Omar, P. van den Bossche, and J. van Mierlo, "A review of passive and active battery balancing based on MATLAB/Simulink," *Int. Rev. Electr. Eng.*, vol. 6, no. 7, pp. 2974–2989, 2011.

[74] M. M. Hoque, M. A. Hannan, and A. Mohamed, "Voltage equalization control algorithm for monitoring and balancing of series connected lithium-ion battery," *J. Renew. Sustain. Energy*, vol. 8, no. 2, 2016, doi: 10.1063/1.4944961.

[75] Z. B. Omariba, L. Zhang, and D. Sun, "Review of Battery Cell Balancing Methodologies for Optimizing Battery Pack Performance in Electric Vehicles," *IEEE Access*, vol. 7, pp. 129335–129352, 2019, doi: 10.1109/ACCESS.2019.2940090.

[76] P. Kollmeyer, H. Andreas, and A. Emadi, "Li-ion battery model performance for automotive drive cycles with current pulse and eis parameterization," in 2*017 IEEE Transportation Electrification Conference and Expo (ITEC)*, 2017, doi:10.1109/ITEC.2017.7993319.

[77] M. Naguib, P. Kollmeyer, and A. Emadi, "Lithium-ion battery pack robust state of charge estimation, cell inconsistency, and balancing: review," *IEEE Access*, vol. 9, pp. 50570-50582, 2021, doi: 10.1109/ACCESS.2021.3068776.

[78] Z. Chen, H. Sun, G. Dong, J. Wei, et al., "Particle filter-based state-of-charge estimation and remaining-dischargeable-time prediction method for lithium-ion batteries," *J. Power Sources*, vol. 414, pp. 158-166, 2019, doi: 10.1016/j.jpowsour.2019.01.012.

[79] S. Rajakumar, and U. Sinha, "Estimation of lithium-ion battery state of charge for electric vehicles using a nonlinear state observer," *Energy Storage*, vol. 414, pp. 158-166, 2021, doi: 10.1002/est2.290.

[80] C. Vidal, P. Malysz, P. Kollmeyer, and A. Emadi, "Machine learning applied to electrified vehicle battery state of charge and state of health estimation: state-of-the-art," in *IEEE Access*, vol. 8, pp. 52796-52814, 2020, doi: 10.1109/ACCESS.2020.2980961.

[81] H. Dai, G. Zhao, M. Lin, J. Wu, and G. Zheng, "A novel estimation method for the state of health of lithium-ion battery using prior knowledge-based neural network and markov chain," in *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 7706-7716, Oct. 2019, doi: 10.1109/TIE.2018.2880703.

[82] M., Naguib, C.Vidal, P. Kollmeyer, P. Malysz, et al., "Comparative Study between Equivalent Circuit and Recurrent Neural Network Battery Voltage Models," *2021 SAE WCX Technical Paper*, 2021, https://doi.org/10.4271/2021-01-0759.

[83] M. Naguib, P. Kollmeyer, C. Vidal, and A. Emadi, "Accurate surface temperature estimation of lithium-ion batteries using feedforward and

recurrent artificial neural networks," *2021 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2021, pp. 52-57, doi: 10.1109/ITEC51675.2021.9490043.

[84] Q. Wang, H. Gu, M. Ye, M. Wei, and X. Xu, "State of charge estimation for lithium-ion battery based on narx recurrent neural network and moving window method," in *IEEE Access*, vol. 9, pp. 83364-83375, 2021, doi: 10.1109/ACCESS.2021.3086507.

[85] P. Shrivastava, T. Kok Soon, M. Y. I. Bin Idris, S. Mekhilef, and S. B. R. S. Adnan, "Combined state of charge and state of energy estimation of lithium-ion battery using dual forgetting factor-based adaptive extended kalman filter for electric vehicle applications," in *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1200-1215, Feb. 2021, doi: 10.1109/TVT.2021.3051655.

[86] J. Barillas, J. Li, C.Günther, and M. Danzer, "A comparative study and validation of state estimation algorithms for Li-ion batteries in battery management systems," *Applied Energy*, vol. 155, pp. 455-462, 2015, doi:10.1016/j.apenergy.2015.05.102.

[87] S32K Family, "S32K Family – 32-Bit Arm Cortex Automotive MCUs," https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/s32k-automotive-mcus:S32K-MCUS, accessed Oct. 2021.

[88] PEmicro development and production tools, “Open SDA Support,” https://www.pemicro.com/opensda/, accessed Oct. 2021.

[89] NXP Model-Based Design Toolbox, “NXP Model-Based Design Toolbox (MBDT),” https://www.nxp.com/design/automotive-software-and-tools/nxp-model-based-design-toolbox-mbdt:MBDT, accessed Oct. 2021.

[90] NXP Model-Based Design Toolbox Team , “NXP Support Package,” https://community.nxp.com/t5/Model-Based-Design-Toolbox-MBDT/S32K-How-To/m-p/719984, accessed Oct. 2021.

[91] K. Yang, Y. Tang, and Z. Zhang, “Parameter identification and state-of-charge estimation for lithium-ion batteries using separated time scales and extended kalman filter,” *Energies*, vol. 14, pp. 1054, 2021, doi: 10.3390/en14041054

[92] M. U. Usman, J. Ospina, and M. O. Faruque, "Fault classification and location identification in a smart network using ANN," *2018 IEEE Power & Energy Society General Meeting (PESGM)*, 2018, pp. 1-6.

[93] C. Vidal, P. Malysz, M. Naguib, A. Emadi, and P. J. Kollmeyer, “Estimating battery state of charge using recurrent and non-recurrent Neural Networks,” *Journal of Energy Storage*, p. 103660, 2021.

[94] C. Vidal, et al., “Robust XEV Battery State-of-Charge Estimator Design Using a Feedforward Deep Neural Network,” *2020 SAE WCX Technical Paper Series*, 2020, doi: 10.4271/2020-01-1181.

[95] K. A. Althelaya, E. S. M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional LSTM for short and long-term stock market prediction," 2*018 9th Int. Conf. Inf. Commun. Syst. ICICS 2018*, vol. 2018-Janua, pp. 151–156, 2018.

[96] ARM Developer, “Cortex-M4 Technical Reference Manual,” https://developer.arm.com/ip-products/processors/cortex-m/cortex-m4, accessed Oct. 2021.

[97] ARM Developer, “Cortex-M7 Technical Reference Manual,” https://developer.arm.com/ip-products/processors/cortex-m/cortex-m7, accessed Oct. 2021.

[98] Y. Xie et al., "An enhanced online temperature estimation for lithium-ion batteries," *IEEE Transactions on Transportation Electrification*, vol. 6, no. 2, pp. 375-390, June 2020.

[99] O. Ojo, H. Lang, Y. Kim, X. Hu, B. Mu, and X. Lin, "A neural network based method for thermal fault detection in lithium-ion batteries," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4068-4078, May 2021.

[100] K. Zhang, X. Hu, Y. Liu, X. Lin, and W. Liu, "multi-fault detection and isolation for lithium-ion battery systems," in *IEEE Transactions on Power Electronics*, vol. 37, no. 1, pp. 971-989, Jan. 2022.

[101] R. R. Richardson, and D. A. Howey, "Sensorless battery internal temperature estimation using a Kalman filter with impedance

measurement," *IEEE Transact. on Sustain. Energy*, vol. 6, no. 4, pp. 1190–1199, Oct. 2015.

[102] L. H. J Raijmakers, D.L. Danilov, J. P. M. Van Lammeren, M. J. G. Lammers, and P. H. L. Notten, "Sensorless battery temperature measurements based on electrochemical impedance spectroscopy," *J. Power Sources*, vol. 247, pp. 539–544, 2014.

[103] J. Zhu, Z. Sun, X. Wei, and H. Dai, "A new lithium-ion battery internal temperature on-line estimate method based on electrochemical impedance spectroscopy measurement," J. Power Sources, vol. 274, pp. 990–1004, Jan. 2015.

[104] Y. Li, B. Xiong, D. M. Vilathgamuwa, Z. Wei, C. Xie, and C. Zou, "Constrained ensemble kalman filter for distributed electrochemical state estimation of lithium-ion batteries," I*EEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 240-250, Jan. 2021.

[105] X. Cheng, N. Shi, Y. Li, and S. Wang, "Engineering-oriented modeling for thermal behaviors of 18650 li-ion batteries", *Energy Procedia*, vol. 105, pp. 4757-4762, 2017.

[106] M. Mastali, E. Foreman, and A. Modjtahed, "Electrochemical-thermal modeling and experimental validation of commercial graphite/LiFePO4 pouch lithium-ion batteries", *International Journal of Thermal Sciences*, pp. 129, 2018.

[107] Z. Wang, J. Ma, and L. Zhang, "Finite element thermal model and simulation for a cylindrical li-ion battery," IEEE Access, vol. 5, pp. 15372-15379, 2017.

[108] L. W.-J. Juang et al., "Electric vehicle battery monitoring system," *U.S. Patent* 10,183,590. 22, Jan. 2019.

[109] A. A. Hussein, and A. A. Chehade, "Robust artificial neural network-based models for accurate surface temperature estimation of batteries," *IEEE Transactions on Industry Applications*, vol. 56, no. 5, pp. 5269-5278, Sept.-Oct. 2020.

[110] K. Liu, K. Li, Q. Peng, Y. Guo, and L. Zhang, "Data-driven hybrid internal temperature estimation approach for battery thermal management", *Complexity*, vol. 2018, pp. 1-15, 2018.

[111] B. Oh, K. Toh, A. B. J. Teoh, and Z. Lin, "An analytic gabor feedforward network for single-sample and pose-invariant face recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2791-2805, June 2018.

[112] LeCun Yann, Yoshua Bengio, and Geoffrey Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[113] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*.

[114] Kollmeyer, Phillip (2018), "*Panasonic 18650PF Li-ion Battery Data*", Mendeley Data, V1, doi: 10.17632/wykht8y7tg.1.

[115] Kokam 31 Ah Battery, "*Kokam 31Ah SLPB78216216H battery specifications*,”. Accessed on: Oct 2021. [online]. available: https://kokam.com/skin/pc_en/img/cell_down/TDS_SLPB78216216H%20(31Ah)_EN.nx.

[116] R. Geqay, and T. Liu, "Nonlinear modeling and prediction with feedforward and recurrent networks", Physica D, vol. 108, pp. 119-134, 1997.

[117] M. Naguib, P.J. Kollmeyer, O. Gross, and A. Emadi, “Microprocessor execution time and memory use of battery state of charge estimation algorithms”, *SAE World Congress*, Detroit, MI, April 2022.

[118] X. Tang, K. Yao, B. Liu, W. Hu, and F. Gao, "Long-term battery voltage power and surface temperature prediction using a model-based extreme learning machine", Energies, vol. 11, no. 1, pp. 86, 2018.

[119] G. Zubi, R. S. Adhikari, N. E. Sánchez, and W. Acuña-Bravo, "Lithium-ion battery-packs for solar home systems: Layout, cost and implementation perspectives," *J. Energy Storage*, vol. 32, no. October, 2020.

[120] K. A. Murashko, A. V. Mityakov, J. Pyrhönen, V. Y. Mityakov, and S. S. Sapozhnikov, "Thermal parameters determination of battery cells by local heat flux measurements," *J. Power Sources*, vol. 271, pp. 48–54, 2014.

[121] C. Forgez, D. Vinh Do, G. Friedrich, M. Morcrette, and C. Delacourt, "Thermal modeling of a cylindrical LiFePO4/graphite lithium-ion battery," *J. Power Sources*, vol. 195, no. 9, pp. 2961–2968, 2010.

[122] H. Maleki, S. Al, J. R. Selman, R. B. Dinwiddie and H. Wang, "Thermal properties of lithium-ion battery and components", *J. Electrochem*. Soc., vol. 146, no. 3, pp. 947-954, Feb. 1999.

[123] N. Wang et al., "Core temperature estimation method for lithium-ion battery based on long short-term memory model with transfer learning," in *IEEE Journal of Emerging and Selected Topics in Power Electronics*, doi: 10.1109/JESTPE.2021.3136906.

[124] C. Wang, C. Li, G. Wang, C. Zhang, and N. Cui, "Fast identification method for thermal model parameters of Lithium-ion battery based on discharge temperature rise," *J. Energy Storage*, vol. 44, no. PB, p. 103362, 2021.

[125] H. Ruan, J. Jiang, B. Sun, W. Gao, L. Wang, and W. Zhang, "Online estimation of thermal parameters based on a reduced wide-temperature-range electro-thermal coupled model for lithium-ion batteries," *J. Power Sources*, vol. 396, no. July, pp. 715–724, 2018.

[126] T. S. Bryden et al., "Methodology to determine the heat capacity of lithium-ion cells," *J. Power Sources*, vol. 395, no. May, pp. 369–378, 2018.

[127] M. Akbarzadeh et al., "Thermal modeling of a high-energy prismatic lithium-ion battery cell and module based on a new thermal characterization methodology," *J. Energy Storage*, vol. 32, no. April, p. 101707, 2020.

[128] L. Sheng, L. Su, H. Zhang, Y. Fang, H. Xu, and W. Ye, "An improved calorimetric method for characterizations of the specific heat and the heat generation rate in a prismatic lithium ion battery cell," *Energy Convers. Manag.*, vol. 180, no. November 2018, pp. 724–732, 2019.

[129] X. Lin, H. E. Perez, J. B. Siegel, and A. G. Stefanopoulou, "Robust Estimation of Battery System Temperature Distribution Under Sparse Sensing and Uncertainty," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 753–765, 2020.

[130] M. U. Usman, J. Ospina, and M. O. Faruque, "Fault Classification and Location Identification in a Smart Distribution Network Using ANN," in *2018 IEEE Power Energy Society General Meeting* (PESGM), 2018, pp. 1–6.

[131] M. J. JIANG YuHeng, YU YiFei, HUANG JianQing, CAI WeiWei, "Li-ion battery temperature estimation based on recurrent neural networks," S*ci. CHINA Technol*. Sci. June, vol. 64, no. 6, pp. 1335–1344, 2021.

[132] J. Kleiner, M. Stuckenberger, L. Komsiyska, and C. Endisch, "Real-time core temperature prediction of prismatic automotive lithium-ion battery cells based on artificial neural networks," *J. Energy Storage*, vol. 39, no. October 2020, p. 102588, 2021, doi: 10.1016/j.est.2021.102588.

[133] S. Zhu, C. He, N. Zhao, and J. Sha, "Data-driven analysis on thermal effects and temperature changes of lithium-ion battery," *J. Power Sources*, vol. 482, p. 228983, 2021, doi: https://doi.org/10.1016/j.jpowsour.2020.228983.

[134] X. Sun, B. Wan, G. Lei, X. Tian, Y. Guo, and J. Zhu, "Multiobjective and Multiphysics Design Optimization of a Switched Reluctance Motor for Electric Vehicle Applications," *IEEE Trans. Energy Convers.*, vol. 36, no. 4, pp. 3294–3304, 2021, doi: 10.1109/TEC.2021.3078547.

[135] P. N. Phuc, D. Bozalakov, H. Vansompel, K. Stockman, and G. Crevecoeur, "Rotor temperature virtual sensing for induction machines using a lumped-parameter thermal network and dual kalman filtering," *IEEE Trans. Energy Convers.*, vol. 36, no. 3, pp. 1688–1699, 2021, doi: 10.1109/TEC.2021.3060478.

[136] A. H. Mohamed, H. Vansompel, and P. Sergeant, "Electrothermal design of a discrete gan-based converter for integrated modular motor drives," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 9, no. 5, pp. 5390–5406, 2021, doi: 10.1109/JESTPE.2021.3058107.

[137] D. Li and A. Ukil, "Fault location estimation in voltage source converter based dc system: the l location," *IEEE Trans. Ind. Electron.*, p. 1, 2021, doi: 10.1109/TIE.2021.3125662.

[138] Z. Y. Jiang, Z. G. Qu, J. F. Zhang, and Z. H. Rao, "Rapid prediction method for thermal runaway propagation in battery pack based on lumped thermal resistance network and electric circuit analogy," *Appl. Energy*, vol. 268, p. 115007, 2020, doi: https://doi.org/10.1016/j.apenergy.2020.115007.

[139] O. BMS, "Orion BMS Utility Manual," 2022. [Online]. Available: http://www.orionbms.com/manuals/utility/.

[140] O. BMS, "Thermistor Expansion Module," 2022. [Online]. Available: https://www.orionbms.com/products/thermistor-expansion-module/.

[141] Y. Ye, L. H. Saw, Y. Shi, K. Somasundaram, and A. A. O. Tay, "Effect of thermal contact resistances on fast charging of large format lithium ion batteries," *Electrochim. Acta*, vol. 134, pp. 327–337, 2014, doi: https://doi.org/10.1016/j.electacta.2014.04.134.

[142] M. Steinhardt, E. I. Gillich, M. Stiegler, and A. Jossen, "Thermal conductivity inside prismatic lithium-ion cells with dependencies on temperature and external compression pressure," *J. Energy Storage*, vol. 32, p. 101680, 2020, doi: https://doi.org/10.1016/j.est.2020.101680.

[143] D. Werner, A. Loges, D. J. Becker, and T. Wetzel, "Thermal conductivity of Li-ion batteries and their electrode configurations – A novel combination of modelling and experimental approach," *J. Power Sources*, vol. 364, pp. 72–83, 2017, doi: https://doi.org/10.1016/j.jpowsour.2017.07.105.

[144] K. A. Murashko, J. Pyrhönen, and J. Jokiniemi, "Determination of the through-plane thermal conductivity and specific heat capacity of a Li-ion cylindrical cell," *Int. J. Heat Mass Transf.*, vol. 162, p. 120330, 2020, doi: https://doi.org/10.1016/j.ijheatmasstransfer.2020.120330.

[145] S. J. Bazinski and X. Wang, "Experimental study on the influence of temperature and state-of-charge on the thermophysical properties of an LFP pouch cell," *J. Power Sources*, vol. 293, pp. 283–291, 2015, doi: https://doi.org/10.1016/j.jpowsour.2015.05.084.

[146] T. Aree and H.-B. Bürgi, "Specific heat of molecular crystals from atomic mean square displacements with the einstein, debye, and nernst−lindemann models," *J. Phys. Chem. B*, vol. 110, no. 51, pp. 26129–26134, Dec. 2006, doi: 10.1021/jp0636322.

[147] C.-H. Doh, Y.-C. Ha, and S. Eom, "Entropy measurement of a large format lithium ion battery and its application to calculate heat generation," *Electrochim. Acta*, vol. 309, pp. 382–391, 2019, doi: https://doi.org/10.1016/j.electacta.2019.04.026.

[148] L. Sheng, L. Su, and H. Zhang, "Experimental determination on thermal parameters of prismatic lithium ion battery cells," *Int. J. Heat Mass Transf.*, vol. 139, pp. 231–239, 2019, doi: https://doi.org/10.1016/j.ijheatmasstransfer.2019.04.143.

[149] J. Jaguemont et al., "1D-Thermal Analysis and Electro-Thermal Modeling of Prismatic-Shape LTO and NMC Batteries," in *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2019, pp. 1–5, doi: 10.1109/VPPC46532.2019.8952187.

[150] P.-L. Lapointe, "Application of the thermal impedance spectroscopy method in three dimensions to a large prismatic Li-ion cell," in *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*, 2013, pp. 1–6, doi: 10.1109/EVS.2013.6915051.

[151] S. Balkur, N. Roy Chowdhury, J. Groot, and T. Thiringer, "A cost and time effective novel methodology to determine specific heat capacity of lithium-

ion cells," *J. Power Sources*, vol. 500, p. 229981, 2021, doi: https://doi.org/10.1016/j.jpowsour.2021.229981.

[152] X. Zhang et al., "Evaluation of convective heat transfer coefficient and specific heat capacity of a lithium-ion battery using infrared camera and lumped capacitance method," *J. Power Sources*, vol. 412, pp. 552–558, 2019, doi: https://doi.org/10.1016/j.jpowsour.2018.11.064.

[153] R. Rizk, H. Louahlia, H. Gualous, and P. Schaetzel, "Experimental analysis and transient thermal modelling of a high capacity prismatic lithium-ion battery," *Int. Commun. Heat Mass Transf.*, vol. 94, pp. 115–125, 2018, doi: https://doi.org/10.1016/j.icheatmasstransfer.2018.03.018.

[154] P. Bohn, G. Liebig, L. Komsiyska, and G. Wittstock, "Temperature propagation in prismatic lithium-ion-cells after short term thermal stress," *J. Power Sources*, vol. 313, pp. 30–36, 2016, doi: https://doi.org/10.1016/j.jpowsour.2016.02.055.

[155] X. Cheng, Y. Tang, and Z. Wang, "Thermal Property Measurements of a Large Prismatic Lithium-ion Battery for Electric Vehicles," *J. Therm. Sci.*, vol. 30, no. 2, pp. 477–492, 2021, doi: 10.1007/s11630-021-1398-3.

[156] J. Carvill, "3 - Thermodynamics and heat transfer," in Mechanical Engineer's Data Handbook, J. Carvill, Ed. Oxford: Butterworth-Heinemann, 1993, pp. 102–145.

[157] D. Li, Z. Zhang, P. Liu, Z. Wang, and L. Zhang, "Battery fault diagnosis for electric vehicles based on voltage abnormality by combining the long short-

term memory neural network and the equivalent circuit model," in *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 1303-1315, Feb. 2021, doi: 10.1109/TPEL.2020.3008194.

[158] P. Sprent and N. C. Smeeton, Applied Nonparametric Statistical Methods, USA, FL, Boca Raton:CRC Press, 2001.

[159] Lai, Xin, et al. “Mechanism, modeling, detection, and prevention of the internal short circuit in lithium-ion batteries: recent advances and perspectives.” *Energy Storage Materials*, vol. 35, 2021, pp. 470–499.

[160] J. Jiang, X. Cong, S. Li, C. Zhang, W. Zhang, and Y. Jiang, "A hybrid signal-based fault diagnosis method for lithium-ion batteries in electric vehicles," in *IEEE Access*, vol. 9, pp. 19175-19186, 2021.

[161] D. Finegan et al., "Characterising thermal runaway within lithium-ion cells by inducing and monitoring internal short circuits", *Energy Environ*. Sci., vol. 10, pp. 1377-1388, Apr. 2017.

[162] P. Ramadass, W. Fang and Z. Zhang, "Study of internal short in a Li-ion cell I. Test method development using infra-red imaging technique", *J. Power Source*s, vol. 248, pp. 769-776, Feb. 2014.

[163] J. Hu, H. He, Z. Wei, and Y. Li, "Disturbance-Immune and Aging-Robust Internal Short Circuit Diagnostic for Lithium-Ion Battery," in *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1988-1999, Feb. 2022, doi: 10.1109/TIE.2021.3063968.

[164] R. Yang, R. Xiong, W. Shen, and X. Lin, "Extreme learning machine-based thermal model for lithium-ion batteries of electric vehicles under external short circuit", *Engineering*, vol. 7, no. 3, pp. 395-405, Mar. 2021.

[165] Y. Xu, X. Ge, W. Shen, and R. Yang, "A soft short-circuit diagnosis method for lithium-ion battery packs in electric vehicles," in *IEEE Transactions on Power Electronics*, vol. 37, no. 7, pp. 8572-8581, July 2022, doi: 10.1109/TPEL.2022.3151620.

[166] Seyed Mohammad Mahdi Alavi, Sajjad Fekriasl, Seyed Nami Niyakan, and Mehrdad Saif, "Fault detection and isolation in batteries power electronics and chargers", *Journal of Energy Storage*, vol. 25, pp. 100807, 2019.

[167] J. Hu, X. Bian, Z. Wei, J. Li, and H. He, "Residual statistics-based current sensor fault diagnosis for smart battery management," in *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 10, no. 2, pp. 2435-2444, April 2022, doi: 10.1109/JESTPE.2021.3131696.

[168] S. Dey, S. Mohon, P. Pisu and B. Ayalew, "Sensor fault detection, isolation, and estimation in lithium-ion batteries," in *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2141-2149, Nov. 2016, doi: 10.1109/TCST.2016.2538200.

[169] R. Xiong, Q. Yu, W. Shen, C. Lin, and F. Sun, "A sensor fault diagnosis method for a lithium-ion battery pack in electric vehicles," in *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 9709-9718, Oct. 2019, doi: 10.1109/TPEL.2019.2893622.

[170] Y. Kang, B. Duan, Z. Zhou, Y. Shang, and C. Zhang, "A multi-fault diagnostic method based on an interleaved voltage measurement topology for series connected battery packs", J. Power Sources, vol. 417, pp. 132-144, 2019.

[171] Z. Sun et al., "An online data-driven fault diagnosis and thermal runaway early warning for electric vehicle batteries," in *IEEE Transactions on Power Electronics*, vol. 37, no. 10, pp. 12636-12646, Oct. 2022, doi: 10.1109/TPEL.2022.3173038.

[172] J. Wei, G. Dong, and Z. Chen, "Lyapunov-based thermal fault diagnosis of cylindrical lithium-ion batteries," in *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 4670-4679, June 2020, doi: 10.1109/TIE.2019.2931275.

[173] D. Li et al., "Battery thermal runaway fault prognosis in electric vehicles based on abnormal heat generation and deep learning algorithms," in *IEEE Transactions on Power Electronics*, vol. 37, no. 7, pp. 8513-8525, July 2022, doi: 10.1109/TPEL.2022.3150026.

[174] H. O. Omoregbee, and P. S. Heyns, "Fault detection in roller bearing operating at low speed and varying loads using Bayesian robust new hidden Markov model", *J. Mech. Sci. Technol.*, vol. 32, no. 9, pp. 4025-4036, Sep. 2018.

[175] S. Rahman Fahim, S. K. Sarker, S. Muyeen, M. Sheikh, R. Islam, and S. K. Das, "Microgrid fault detection and classification: machine learning based

approach comparison and reviews", *Energies*, vol. 13, no. 13, pp. 3460, 2020.

[176] G. Xia, L. Cao, and G. Bi, "A review on battery thermal management in electric vehicle application", *J. Power Sources*, vol. 367, pp. 90-105, Nov. 2017.

[177] X. Hu, K. Zhang, K. Liu, X. Lin, S. Dey and S. Onori, "Advanced fault diagnosis for lithium-ion battery systems: a review of fault mechanisms, fault features, and diagnosis procedures," in I*EEE Industrial Electronics Magazine*, vol. 14, no. 3, pp. 65-91, Sept. 2020, doi: 10.1109/MIE.2020.2964814.

[178] Ma, Guijun, et al. “Fault detection of lithium-ion battery packs with a graph-based method.” *Journal of Energy Storage*, vol. 43, 2021, p. 103209, doi: 10.1016/j.est.2021.103209.