Design and Implementation of a Robust State of Charge Estimation Approach for a Single Battery Cell, a Hardware-in-the-Loop Test Bench, and a Battery Disconnect Unit for an Electric Vehicle Battery Pack

by

Nguyen Truong Son Pham

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2025

© Nguyen Truong Son Pham 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

As transportation electrification accelerates, battery-powered vehicles, including cars, airplanes, and boats, are rapidly emerging. This thesis provides solutions and practical insights on two key topics: implementing robust machine learning algorithms on commercial Battery Management System (BMS), and building a high-performance Battery Disconnect Unit (BDU). The experience was gained during participation in the North American Battery Workforce Challenge.

First, two machine learning approaches for State of Charge (SoC) estimation are introduced. The first approach is an adaptive algorithm using SoC-OCV-T (State of Charge-Open Circuit Voltage-Temperature) lookup table and Extreme Learning Machine (ELM). The experiment began at 100% SoC, with temperature ranging from $-20\,^{\circ}\mathrm{C}$ to $60\,^{\circ}\mathrm{C}$. From $-20\,^{\circ}\mathrm{C}$ to $0\,^{\circ}\mathrm{C}$, the maximum absolute error (MAE) ranged from 0.030 to 0.025. In the mid-range from $5\,^{\circ}\mathrm{C}$ to $40\,^{\circ}\mathrm{C}$, the MAE decreased to within 0.015 to 0.020 range. Lastly, at higher temperature range of $45\,^{\circ}\mathrm{C}$ to $60\,^{\circ}\mathrm{C}$, the MAE was below 0.013.

In the second approach, advanced differential features are added to improve the accuracy of the ELM model, particularly below $0\,^{\circ}\mathrm{C}$. Under noisy condition, both the maximum absolute error (MAE) and the root mean square error (RMSE) were reduced to below 1.5% at -20, 20, and $60\,^{\circ}\mathrm{C}$. Both algorithms were validated on a customized Hardware-in-the-loop (HIL) test bench. The HIL platform was developed to streamline validation of algorithms such as SoC estimation.

Finally, the thesis details the design and testing process for the BDU, highlighting key design considerations, test results, and engineering challenges.

Acknowledgements

I would not have made it this far if it hadn't been for the help of all the people around me.

The BDU was assembled neatly at Lambton College with the help of Prof. McMillin and Robin. I also want to thank Ravi, David, Ajay, Hannah, Battery Workforce Challenge, AVL, Stellantis, Batemo GmbH, MATLAB, ADI, Eaton and OpenECU staff who helped me overcome technical issues.

The Battery Workforce Challenge is not merely a competition but also a hub for collaboration among students and engineers from diverse disciplines.

Thanks to the amazing students and professors that I have met at the University of Waterloo. Thanks Profs. Kazerani and Rangom for selecting me to lead the team and giving feedback for my thesis. Thanks to Grace for splitting the tasks with me while we participated in the BWC.

To all the students I have worked with during BWC, your contributions are valuable no matter how big or small.

Lastly, I am grateful for the support of my family and friends who encourage me to continue learning.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures viii

List of Tables x

List of Abbreviations xii

1 Introduction 1

1.1 Objectives 2

1.2 Organization of Thesis 2

2 Literature Review 3

2.1 State of Charge Estimation 3

2.1.1 Introduction to Battery Management Systems (BMS) 4

2.1.2 Traditional SoC Estimation Techniques 4

2.1.3 Advanced SoC Estimation Techniques 5

2.1.4 MCU Feasibility for ML/AI SoC Estimation Model 10

2.1.5 Conclusion . . . 10
2.2 Hardware-in-the-Loop . . . 11
2.2.1 Usage . . . 11
2.2.2 Software . . . 11
2.2.3 Hardware . . . 11
2.3 Battery Disconnect Unit . . . 13
2.3.1 Relay . . . 14
2.3.2 Conclusion . . . 15

3 ELM and SoC-OCV-T for State of Charge Estimation 16
3.1 Introduction . . . 16
3.1.1 Extreme Learning Machine (ELM) . . . 17
3.1.2 Adaptive Algorithm . . . 18
3.2 Dataset Overview . . . 18
3.3 ELM Model Setup . . . 22
3.4 Adaptive Algorithm Design . . . 24
3.5 Building the Model on MATLAB Simulink . . . 25
3.6 Building Hardware-in-the-Loop (HIL) Bench . . . 27
3.7 Implement Algorithm on BMS . . . 27
3.8 Results and Discussion . . . 28
3.8.1 Result from Simulation . . . 28
3.8.2 Results for Implementation of SoC Estimation Algorithm on BMS . 32
3.8.3 Discussion . . . 32

4 Robust State of Charge Estimation Algorithm with Differential Features 35
4.1 Introduction . . . 35
4.2 Survey on Differential Features for SoC . . . 36
4.3 Prepare New Features . . . 37
4.4 Implementation on BMS . . . 41
4.5 Result and Discussion . . . 43

5 Battery Disconnect Unit (BDU) 45

5.1 Introduction to BDU . . . 45

5.1.1 Material: copper vs aluminum . . . 45

5.1.2 Main Components . . . 46

5.1.3 BDU Topologies: Series vs Parallel . . . 48

5.2 Designing BDU . . . 49

5.2.1 Determine Pack Configuration . . . 49

5.2.2 Precharge Circuit . . . 51

5.2.3 Fuse Selection . . . 54

5.2.4 Creepage & Clearance . . . 54

5.3 Integration . . . 55

5.3.1 Physical Integration . . . 55

5.3.2 Electrical Integration . . . 56

5.3.3 Design Challenge and Solutions . . . 56

5.4 Validation . . . 57

5.5 Result and Discussion . . . 60

6 Conclusion 61

References 63

Glossary 74

List of Figures

2.1 n-RC Thevenin Models 6
2.2 Typical SoC prediction algorithm implementation using neural networks . . 9
2.3 Review of softwares being used in HIL 12
2.4 High-level BDU System Diagram 14

3.1 Histogram of Current Profiles 20
3.2 SoC-OCV Curves at Different Temperatures 22
3.3 ELM Model with Inputs and Output 23
3.4 SoC-OCV at Different Temperatures and Different Size 399x172 (Left), 99x86 (Right) 24
3.5 Adaptive Algorithm Flow Chart 26
3.6 ELM Simulink Block 27
3.7 Completed Adaptive Coulomb Counting Simulink Block: (1) ELM Simulink Block, (2) Adaptive Coulomb counting, (3) SoC Selector block 28
3.8 HIL bench: (1) Pack Monitor, (2) 12V Relay, (3) Precharge Relay, (4) Breaktor, (5) Negative and Positive busbar for motors, (6) Negative and positive busbar from Battery, (7) Loss of isolation, (8,10) CAN signal and CAN/LIN interface, (9) ADBMS6830B Cell Supervision Circuit, (0) DC2472A Cell Emulator 29
3.9 Comparison Between Small and Big SoC-OCV-T Table 30
3.10 Setup to Program M450 32
3.11 Monitoring Real-Time SoC_Pred and Adaptive_cc with OpenECU Calibrator 33

3.12 Result from BMS (Left) and Simulation (Right) 33

4.1 Initial SOC Prediction from SoC-OCV-T Lookup Table and ELM Model with three features of Terminal Voltage, Surface Temperature, Current . . 36
4.2 Simplified Correlation Matrix (SOC_lag_1 is prior SOC value at 1 time step. Terminal_voltage_rolling_3 is moving average with size of 3) 41
4.3 Initial prediction starting at 0.75 SoC. SOC_NEW was SoC Prediction with Coulomb Counting. SoC was the True SoC. SOC_Pred was SoC prediction from ELM Model. Without Coulomb counting, SOC_Pred is sensitive to suddent current change. SOC_NEW solved the issue. 42
4.4 ELM with Advanced Features Simulink Block 42
4.5 Monitoring Real-Time Robust Adaptive_cc with OpenECU Calibrator . . . 43
4.6 Result from Simulation (Left) and BMS (Right) 43

5.1 BDU Topology: Parallel Configuration (left), Series Configuration (right) . 49
5.2 Example of Parallel First Configuration 50
5.3 Process and Equation for Determining Cell Configuration 51
5.4 Calculation of Power for Precharge Resistor 52
5.5 Simulate Precharge Circuit on Simulink 53
5.6 Visualization Creepage and Clearance 54
5.7 Simplified Flow to Determine Creepage and Clearance 55
5.8 High Voltage Schematic for BDU 56
5.9 Development Workflow for BDU 57
5.10 BDU Design Timeline With Different Iterations 58
5.11 Locations of Thermistors and Flow of Current 59
5.12 Temperature at Different Locations During Testing 59
5.13 Elevated Temperature of Relay without Economizer in the BDU 60

List of Tables

3.1 Algorithm Benchmarking with Weighted Rank and Size Constraint 17
3.2 Drive Cycle Profiles 19
3.3 Cell's Limits 20
3.4 Cell Safety Limits 21
3.5 Duration for Current Profile Testing 21
3.6 Structure of ELM model 23
3.7 Comparison of Error Metrics Between Moving Average and Adaptive Coulomb Counting Methods on Small and Big Lookup Tables 30
3.8 Error Metrics for Three Methods: SoC predictor, Adaptive Coulomb Counting (adaptive_cc), and Moving Average (movingavg_w100). 31

4.1 ELM Model performance at three different temperatures and SoC of 0.75 . 35
4.2 Time-series features for SoC estimation algorithm 38
4.3 Benchmark Result for New Feature Sets (Ranked by MAE). Lag or prior value features were highlighted as green. New discovered feature Power over Surface Temperature was highlighted as blue. 39
4.4 Benchmark Result with suddent shift in SoC ($\pm 0.05, 0.1, 0.15, 0.2$) at different temperature (0, 20, 40) and initial SoC values (1, 0.75, 0.5) 40
4.5 Comparison between new algorithms with and without noise starting at 75% SoC 44

5.1 Comparison Table Among Four Type of Mechanical Contactor 47
5.2 Comparison Between Mechanical Contactor and Solid-State Relay 47

5.3 Trade of Between I/O Pins and Components Required for BDU 48
5.4 Cell's Specification 50
5.5 Specification of Designed Battery Pack 52
5.6 Precharge Calculation Results 53
5.7 Parameter for Creepage and Clearance Distances 55

List of Abbreviations

BDU Battery disconnect unit 4, 13

BMS Battery Management System 3, 4, 17

BSF battery size factor 51

DCFC DC fast charging 57

ECM Equivalent circuit models 3

ELM Extreme Learning Machine 8

GRU Gate Recurrent Unit 10

HWFET Highway Fuel Economy Test 19

LSTM Long short-term memory 8

MAE Mean absolute error 7, 8, 28

MaxAE Maximum absolute error 28

NYCC New York City Cycle 19, 21

OCV Open-circuit voltage 21

RMSE Root mean squared error 7, 28

SoC State of Charge 3, 6, 21

SSR Solid-state relay 14, 46

UDDS Urban Dynamometer Driving Schedule 19

Chapter 1

Introduction

Battery systems serve a critical role in transportation, mobile devices, household appliances, and space exploration. Given the broad scope of battery systems, this thesis focuses on implementing Machine Learning methods for State of Charge (SoC) estimation in commercial Battery Management Systems (BMSs). Additionally, it includes the process of building a Battery Disconnect Unit (BDU), and a Hardware-in-the-loop (HIL) system to allow validation of the developed algorithm.

The literature review in Chapter 2 identifies gaps in the field, including design of machine learning-based State of Charge (SoC) estimation algorithms and their implementation on an actual BMS hardware, the design of Hardware-in-the-loop (HIL) systems, and the development of battery disconnect units (BDUs).

To address the SoC gaps, two simple and effective methods for SoC estimation are proposed and implemented on a commercial BMS. The first method combines Coulomb counting, SoC-OCV-T lookup tables, and a Single Feed Forward Network (SFFN) or Extreme Learning Machine (ELM). The second method adds differential features to improve the accuracy of ELM's in extreme temperature ($-20\,^{\circ}\mathrm{C}$ and $60\,^{\circ}\mathrm{C}$) and noisy environment. Along with the development of the two proposed algorithms, the feasibility of deploying such a machine learning model on commercial BMS hardware is assessed.

On the hardware side, a complete design a BDU prototype, and a HIL test bench are presented, supported by test results and analysis of practical challenges.

1.1 Objectives

The two objectives of the study reported in this thesis are as follows.

- To develop a high-performance SoC estimation algorithm for electric vehicle battery packs, assess the feasibility of implementing machine learning-based methods for battery state estimation on commercial BMS, and test the robustness of the developed algorithm in noisy and extreme temperature ($-20\,^{\circ}\mathrm{C}$ and $60\,^{\circ}\mathrm{C}$) environments.
- To provide guidelines on building a BDU and HIL test bench.

1.2 Organization of Thesis

The organization of this thesis is as follows. Chapter 2 provides a review of recent developments in SoC estimation algorithms, battery disconnect unit design, and hardware-in-the-loop platforms. Chapter 3 describes the proposed machine learnning-based SoC algorithm and its implementation on a commercial BMS. Chapter 4 presents an improved SoC estimation algorithm based on ELM model and that is robust under extreme temperature ($-20\,^{\circ}\mathrm{C}$ and $60\,^{\circ}\mathrm{C}$) and noise environment. Chapter 5 offers the design process for the battery disconnect unit. Chapter 6 concludes the thesis with a summary of key findings and recommendations for future research.

Chapter 2

Literature Review

2.1 State of Charge Estimation

State of Charge (SoC) estimation is a crucial function of Battery Management System (BMS). Among all estimation methods, Coulomb counting has been the simplest approach[61]. However, it requires an initial SoC value, which may drift over many cycles [58]. To address this issue, Kalman filter or the non-linear variant Extended Kalman Filter are combined with Equivalent circuit models (ECM) to reduce noise and sensor measurement errors [71].

Besides, advanced methods are categorized into model-based and data-driven. Model-based method include electrochemical models and ECM. Domenico et al. proposed a method to incorporate extended Kalman filter into the electrochemical model [21].

Data-driven methods are based on machine learning algorithms such as feedforward neural networks (FFNN), recurrent neural networks (RNN), and generative models. Recent studies have adapted RNNs and generative models for time series prediction. The details are presented in the Data-Driven Methods in subsection 2.1.3. Although these techniques emerged from natural language processing, they have been extended to battery state estimation.

Although many machine learning methods are available for SoC, commercial BMS platforms often have computational limitations, primarily memory and processing power. These limitations make the selection of algorithm critical. The details of these SoC estimation methods will be discussed in subsection 2.1.2 and subsection 2.1.3.

2.1.1 Introduction to Battery Management Systems (BMS)

Battery management systems are essential components of battery energy storage systems (BESS). They serve two main functions: monitoring and control. The BMS monitors the voltage, current, and temperature of individual cells of the pack. It also measures the voltage at critical locations to detect abnormal events. The main objectives of the BMS are to protect the battery and prevent thermal runaway.

In addition to monitoring, the BMS controls the contactors (or relays) to connect or disconnect the battery from the vehicle. Those components are commonly located within the Battery disconnect unit (BDU). The BMS also communicates critical data, such as SoC and State of Health (SoH), to the Electronic Control Unit (ECU) or other controllers via Controller Area Network (CAN) bus or other interfaces. In this thesis, lithium-based battery is the main focus because of their high cell potential and low weight contribute to high energy density [44].

Currently, many companies are developing wireless BMS systems (wBMS), in which data are transmitted via radio frequency within the vehicle [9]. Among the leading technologies, Analog Devices uses the ISM radio band to transmit data [2]. NXP employs UWB (Ultra-Wideband) [63], and Texas Instruments utilizes Bluetooth technology [86]. LG Innotek also showcased their wBMS at CES 2025; however, details regarding frequency and range were not provided [48].

At the application layer, researchers and industry have proposed solutions to monitor or process battery data on a remote server, rather than processing it directly in the BMS [88, 51]. In the next section, both traditional and advanced SoC estimation techniques will be explored.

2.1.2 Traditional SoC Estimation Techniques

Coulomb Counting

Coulomb Counting is a simple method for estimating the SoC of lithium-ion batteries. In this technique, charge transfer is calculated by integrating the measured current over each sampling period. The ratio of the present value of remaining charge in the battery to the present capacity of the battery is then used to estimate the state of charge. Ng et al. proposed an improvement to this method by calculating the Depth of Discharge during charging and discharging cycles [62].

Another approach, presented in [40], involved periodically resetting the SoC and assigning a new value based on an open circuit voltage (OCV) table. This adaptive algorithm resulted in a 2.07% accuracy improvement with respect to the standard Coulomb counting.

Open-Circuit Voltage Methods

Open-Circuit Voltage Methods are based on the observation that OCV has a one-to-one relationship with SoC [67]. Due to relaxation effect, lookup value for SoC should be obtained at a long rest time [68, 70]. To improve accuracy, the OCV method is often combined with an equivalent circuit model (ECM). The ECM represents the cell's internal resistance and relaxation characteristics. In this method, SoC-OCV data are often collected through slow discharge test. Then, least mean squares estimation is used to determine the parameters of the ECM model.

Pattipati et al. also showed that SoC-OCV values did not deviate significantly within the SoC range of 0.1 to 0.8 [67]. The findings suggests that in applications with less stringent accuracy requirements, a single temperature condition (such as 25°C) may be used to set the initial SoC. However, our experiments showed SoC-OCV fluctuated at low temperatures. Therefore, more tests should be performed if the battery is frequently used under low-temperature conditions.

Other advanced SoC estimation methods, including model-based and machine learning approaches, will be discussed in the following sections.

2.1.3 Advanced SoC Estimation Techniques

Model-Based Methods

Thevenin models include a pair of Resistor and Capacitor (RC) or multiple RC circuits, as shown in Figure 2.1. R_{int} represents the internal resistance of the battery, and the RC pairs represent the dynamic voltage response during charging or discharging.

Electrochemical models involve the Butler-Volmer current equation, in which voltage is a function of battery current and average solid concentration [22]. Electrochemical model shows high accuracy when paired with a Kalman filter. However, it does not account for changes when the battery undergoes many cycles. It is also unclear how to update the model parameters over time.

Kalman filters belong to state-space model, and have been widely used in various applications for robust estimation since 1995 [93]. The filter adjusts predicted values based

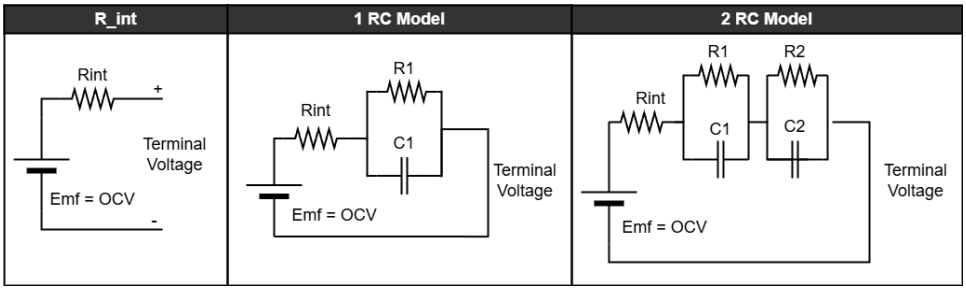


Figure 2.1: n-RC Thevenin Models

on measurements and it is commonly used to estimate SoC [47]. Researchers have further improved the Kalman filter by introducing derivatives to predict nonlinear behaviors. This has led to the development of the Extended Kalman Filter (EKF) [75]. In addition, there are Unscented Kalman Filters (UKF), Central-Difference Kalman Filters (CDKF), and Sigma Point Kalman Filters (SPKF) [72].

He et al. [34] observed a divergence issue in SoC estimation, where the estimation error increased proportionally with the amount of data. To address this issue, the Sage–Husa adaptive filter was employed alongside the Extended Kalman filter and a Thevenin circuit model. As a result, the maximum error was reduced to 4.15

Additionally, Xing et al. [96] combined the Unscented Kalman Filter with a dynamic SoC-OCV-T table. They set random initial guesses for SoC to test the algorithm's robustness. It was difficult to determine how long the algorithm took to correct the SoC from the initial guess, but the results suggested it was less than 100 seconds.

Data-Driven Methods

Data-driven methods are classified into two groups: conventional methods (such as decision trees and support vector machines) and neural network methods (including deep learning models). While both groups are important, this work focuses on neural network methods.

With the emergence of large language models (LLM) and generative AI, many time series and generative models have been adapted for SoC estimation [12, 99, 10]. For example, Generative Adversarial Networks (GANs), originally used for image generation, have been explored for this purpose. Zhang et al. proposed a methods that used GAN-Conditional Latent Space (CLS) to extract data from images. Then, Bidirectional long short-term memory (BLSTM) was trained with the data to predict SoC [99]. GANs consist of two main components: the discriminator and the generator. Each block tries to minimize

its own loss function. During training, the generator may produce identical data, a phenomenon known as mode collapse. To address this, Wasserstein GANs were developed. In [31], a time series Wasserstein GAN (TS-WGAN) was proposed for SoC prediction and tested under four temperature conditions: $-10°C$, $0°C$, $10°C$, and $25°C$. At $25°C$, the model achieved a Mean absolute error (MAE) of 0.0077 and a Root mean squared error (RMSE) of 0.0123, demonstrating high accuracy. However, the storage size and number of model parameters present challenges for implementation on actual BMS hardware.

Another generative model is the Autoencoder (AE), which consists of two main components: an encoder and a decoder [46]. Unlike GANs, AEs have a streamlined structure. The encoder compresses input data into a lower-dimensional latent space, while the decoder reconstructs the output from this latent representation. Model weights are updated through backpropagation. Junxiong et al. combined a Denoising Autoencoder (DEA) with a Gated Recurrent Unit–Recurrent Neural Network (GRU-RNN) for SoC prediction [11]. The DEA helped reconstruct noisy data, while the GRU-RNN reduced the overfitting often seen with LSTM networks. The model achieved errors of less than 2.8% for UDDS, HWFET, and NEDC drive cycles. However, the code size or RAM usage were not reported.

Transformer-based neural networks are also being investigated for SoC estimation. Shi et al. proposed a Spatial-Temporal Self-Attention Transformer Network for SoC estimation [79]. To address the limitations of current on-board microcontroller units, they suggested performing offline training on cloud services. However, the model was not implemented directly on commercial BMS for validation.

Beyond Transformer-based models, the Mamba State-Space Model Neural Network has also been applied to SoC estimation, as described in recent research. Jang et al. developed an integrated framework by combining several advanced components: an autoencoder for feature extraction and noise reduction, a Mamba network for modeling temporal dependencies, an Unscented Kalman Filter for robust state estimation, and a physics-based model to incorporate domain knowledge [41]. This hybrid approach enabled the joint estimation of both state-of-health (SoH) and SoC. The error in estimation was less than 1% indicating high accuracy and strong potential for practical battery management applications.

Although the novel development in Deep Learning, or Neural Network or Decision Tree has brought significant improvement on the accuracy of the SoC esitmation. The code size and computational resource are still the main constrains to implement those advanced methods on microcontroller for BMS. In the next section, we will look at Extreme Learning Machine method, that gives both accuracy and ease in implementation.

Extreme Learning Machine for SoC Estimation

Extreme Learning Machine (ELM) is a Single Layer Feed Forward Network (SLFFN) that was introduced in 2006 [37].

In many research papers, the first step was to select an AI algorithm, such as Long short-term memory (LSTM), Support Vector Machine (SVM) or Extreme Learning Machine (ELM). Sometimes, multiple models were combined to improve prediction accuracy. The common implementation pattern for SoC algorithms involving ELM is illustrated in Figure 2.2.

After model selection, an optimization algorithm was often used to tune the model's hyperparameter. In case of ELM, optimization algorithms were used to set weights and biases instead of using random values. These algorithms were crucial for achieving higher accuracy. Metaheuristic optimizations were commonly used. For example, Ren et al. proposed Genetic Algorithms (GA) combined with Kalman Filter and ELM to reduce mean absolute error (MAE) to less than 1.2% [73]. Gu et al. employed Beetle Antennae Search for recurrent ELMs [30]. Ge et al. proposed an improved bat algorithm IBA-ELM model for SoH and SoC that reduced MAE to 23.79% compared to BA-ELM [28]. Xiao et at. proposed an ELM with Improved Sand Cat Swarm optimization to predict SoC level for vanadium redox flow batteries. The inputs were voltage, current, and cycle which were similar to li-ion battery [95]. Not only optimizing initial weights and biases, Pan et al. added a new feature such as incremental capacity analysis (dQ/dV) with Butterfly Optimization Algorithm (BOA). The results of remaining of life prediction were less than 2% for MAE and RMSE. The other optimization algorithms that were implemented with ELM were Salp Swarm Algorithm [24], Evolutionary Mating Algorithm [80], Gravitational Search Algorithm (GSA) [55], Seagull Optimization Algorithm [94], and Particle Swarm Optimization (PSO) [101, 13].

Besides using optimization algorithms, core of the ELM was also improved. Wang et al. used Kernel Mean P-Power Error as loss function instead of tradition mean square error (MSE) function. Two temperature conditions were tested. At 25 °C, the RMSE and MAE were 0.188 and 0.0148 respectively [92]. Chen et al. added regularization coefficients to calculate output weights and added Radial Basis Kernel to initialize the weight. Additionally, PSO was used to optimize the regularize coefficient and S or "spread" parameter of the kernel function [13]. Moving further, Li et al. proposed a Stacked Noising Autoencoder-Extreme Learning Machine combined with robust data cleaning process on Cloud BMS. The maximum relative error was limited to 3%.

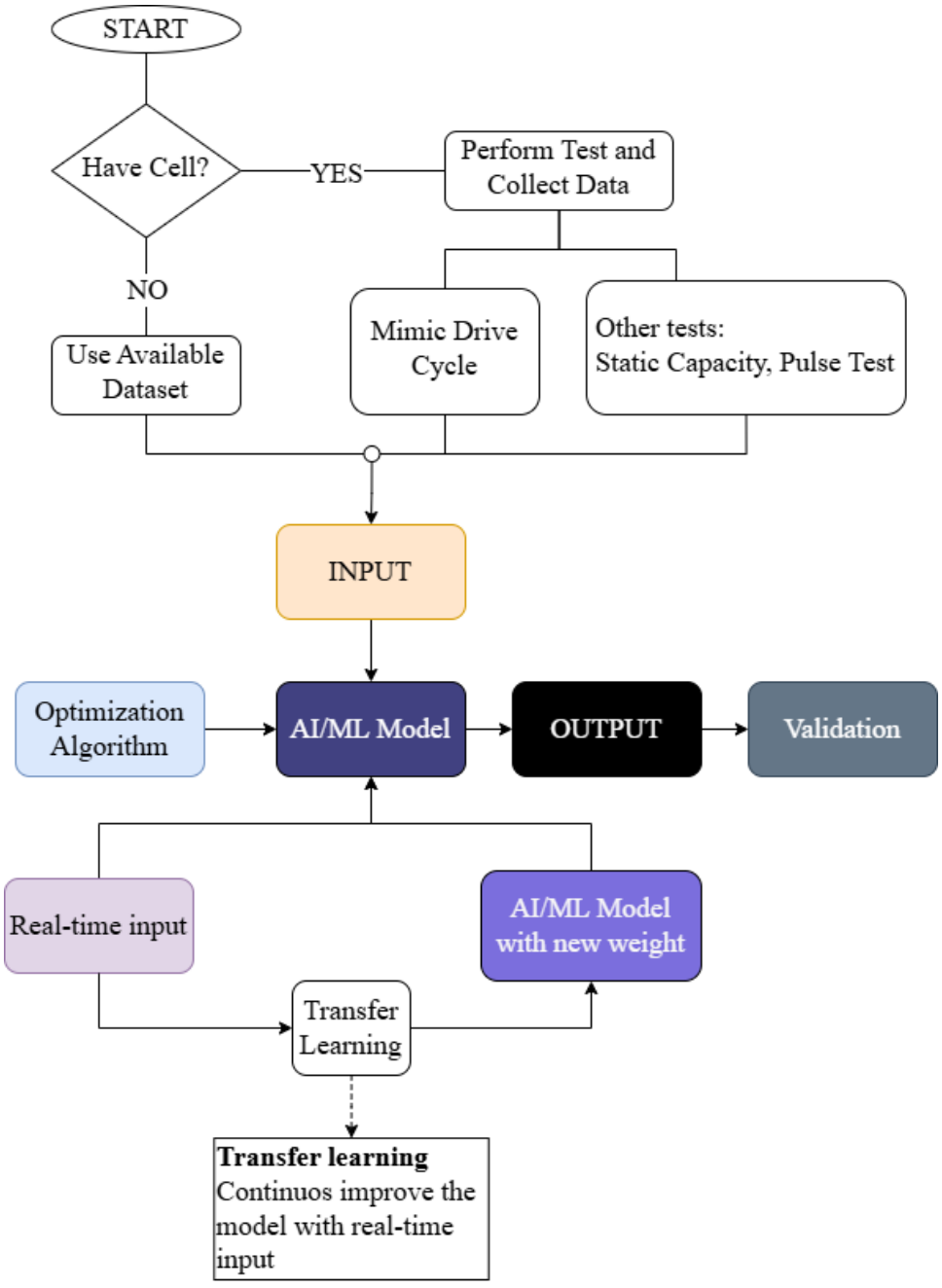


Figure 2.2: Typical SoC prediction algorithm implementation using neural networks

2.1.4 MCU Feasibility for ML/AI SoC Estimation Model

Most studies tested state of charge (SoC) algorithms through simulations. Few evaluated these algorithms on real battery management system hardware. Practical challenges such as limited computing power or unpredictable operation are rarely addressed. This section examines whether microcontrollers (MCU) used in automotive Battery Management Systems are suitable for machine learning SoC algorithms. The focus is on MCUs with limited RAM, which is a key constraint for implementing advanced SoC estimation.

We evaluated automotive-grade MCUs from manufacturers such as NXP (formerly Freescale) and Infineon. Some high-end options offer large RAM capacities, but their costs are often prohibitively high for automotive BMS applications.

As a member of the Battery Workforce Challenge, I worked with the OpenECU M450. Our research focused on the M450 ECU from Dana's OpenECU line, which uses the MPCS746C PowerPC chip. The initial hypothesis was that the M450's 384 KiB RAM might not be sufficient for complex deep neural network models. These models require many layers and intermediate calculations.

Although advanced time-series algorithms such as LSTM or Gate Recurrent Unit (GRU) are feasible in theory, a simpler and more efficient approach are preferred. Extreme Learning Machines (ELM) provide greater computational efficiency and require less memory than LSTM. This makes ELM better suited to resource-constrained MCUs. Safaei et al and Frances et al implemented OS-ELM on FPGA platform [77, 27].

Based on calculations adapted from [97], our model uses 1,801 parameters. Using 64-bit precision, each inference needs 14.4 KB of RAM, which is only 7 percent of the M450's available memory. In comparison, LSTM models described in [12] require 211.6 KB, which is about 55 percent of the total RAM capacity, per inference. Although implementation of LSTM is theoretically possible on the M450, the remaining 172.4 KB of RAM must support other critical BMS tasks. These tasks include cell voltage monitoring, contactor control, and communications. Implementing LSTM will pose a significant challenge for resource allocation.

2.1.5 Conclusion

Recent literature shows major advancements in algorithms for state of charge (SoC) estimation, from classic methods like enhanced Coulomb counting to advanced techniques such as generative neural networks. Hybrid methods have also led to notable reductions in prediction error.

However, after reviewing many papers on SoC estimation methods and experimental setups, we identified a gap: most studies do not address the real-world implementation of these algorithms on commercial BMS hardware. Since we have access to a commercial BMS, OpenECU M450, we will implement adaptive algorithms using Extreme Learning Machines (ELM) to show that AI and machine learning models can work on existing BMS platforms (see Chapter 3). This work aims to help bridge the gap between academic research and industry practice.

In the next section, we review hardware-in-the-loop platforms used to test different BMS functions.

2.2 Hardware-in-the-Loop

Twenty-six papers related to the implementation of Hardware-in-the-Loop (HIL) for electric vehicles were reviewed. Based on this, HIL setups were classified according to three main attributes: usage, hardware, and software. We examined custom HIL setups to identify potential gaps. The findings are summarized below.

2.2.1 Usage

Usage is first analyzed to narrow the scope, since our goal was to design an HIL test bench for an EV battery system. Of the 26 papers, 15 focused on estimation topics such as SoC estimation. The rest addressed areas including powertrain, cell balancing, electromagnetic interference, EV cybersecurity, grid integration, and fast charging.

2.2.2 Software

In the software category, 70% of the experiments used MATLAB. Other approaches included FPGA-based, HDL-based, and custom code solutions. Alternative platforms accounted for about 8% of the studies as shown in Figure 2.3.

2.2.3 Hardware

HIL setups for SoC estimation algorithms and battery pack functions fall under two categories: commercial test platforms and custom hardware-in-the-loop systems (Custom HIL).

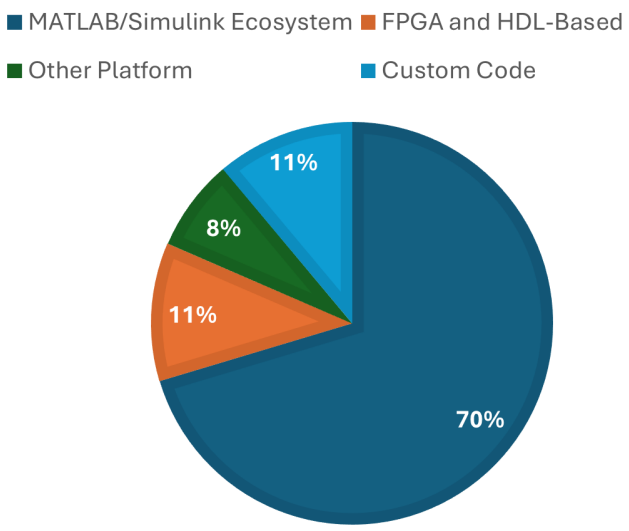


Figure 2.3: Review of softwares being used in HIL

Commercial Test Platform

Commercial test platforms include dSPACE, OPAL, and Typhoon systems. Each of these platforms offers distinct features for hardware-in-the-loop testing.

In [69], the dSPACE hardware could handle up to 400 V, reflecting the real conditions of an EV battery pack; however, the investment cost was high (33,000 euros, not including the testing rack, as of 2021). OPAL also offered real-time control solutions [56]. Typhoon provided another option, featuring compact hardware with built-in Simulink blocks for real-time applications [1]. All three platforms supported Simulink control blocks, making algorithm validation easier for users. According to the survey, commercial test platforms were used more frequently than custom HIL setups.

Custom Hardware-in-the-Loop

Despite the convenience and compactness of commercial test platforms, custom hardware-in-the-loop (Custom HIL) setups represent a step closer to Vehicle-in-the-Loop testing. These systems included custom PCBs, FPGAs, and MCUs.

Out of the 26 papers reviewed, 15 mentioned State of Charge (SoC) estimation. Among

these 15, five employed custom setups similar to our proposed approach. The following examples highlight different custom HIL configurations reported in the literature.

In [66], Park et al. designed a custom HIL setup that included a PCB with an Arduino as the main controller. The Arduino monitored voltage and temperature and sent data to a SoC estimation algorithm. However, the algorithm was not loaded onto the Arduino. It was trained and ran on a desktop computer.

Another approach, described in [82], used a real-time controller (xPC) instead of an Arduino. A programmable load applied current profiles, allowing charge and discharge operations on five Panasonic 18650 lithium-ion cells. Similarly, Zhang et al. designed an H filter in Simulink and used xPC to control the electrical load and collect battery data. The data collected by xPC was then sent to a host computer for processing [100].

In [57], Morello et al. implemented a mixed algorithm that combined a dual Kalman filter with a moving window least squares method on an Altera MAX 10 FPGA. This approach was cost-effective, but the FPGA was used only for SoC validation. The setup might not be modular or easily expandable if more sensors or validation functions are needed.

Lastly, Araujo et al. embedded mathematical equations in TMS320F2869 microcontrollers. Each MCU represented a block in the EV, such as the battery pack, DC-DC converter, and motor drive/load. However, this setup was a numerical simulation and did not use sensors or physical devices [4] .

In summary, custom HIL setups fall into two categories: algorithms running directly on MCUs and algorithms running on computational models. In our HIL setup, the algorithm was run on a computational model, and then was embedded directly into the MCU.

2.3 Battery Disconnect Unit

The primary function of the BDU is to connect or disconnect the high-voltage battery from the load. A BDU functional diagram is shown in Figure 2.4. Since the BDU is an implemented design rather than a standalone research topic, published literature on the BDU is limited.

In 2017, Svensson [8] proposed an optimized packaging solution for BDU with a selection matrix; however, electrical components selection and the topology of the BDU were not discussed. In this literature review section, we discuss new relay technologies that can be used in BDUs, as well as BDU's topology.

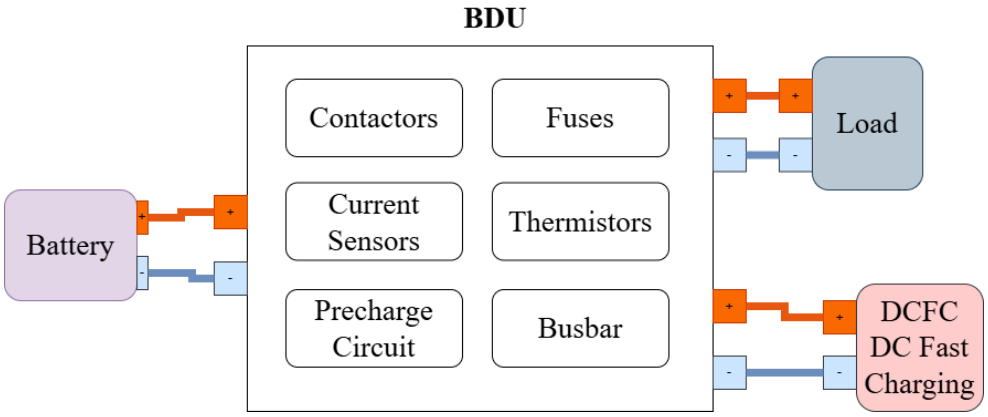


Figure 2.4: High-level BDU System Diagram

2.3.1 Relay

The main component in a BDU is the switching unit. Mechanical contactors use electromagnetic actuation. Due to their low cost, high current ratings, and reliability, they are widely used in commercial EVs.

Another candidate is the Solid-state relay (SSR). SSRs have no moving parts and provide rapid switching at microsecond scale. They do not form arcs and have a longer service life compared to mechanical contactors. In high-voltage BDU applications, SSRs can eliminate problems related to arcing and mechanical wear. However, SSR failures often occur in the closed state [64], which poses a safety risk unless it is combined with a mechanical contactor. SSRs outperform mechanical contactors in most categories except rated current and cooling capability. At present, mechanical contactors support higher current ratings. Most SSRs can only handle up to 200 A and they are generally more expensive, according to Digikey's catalogue (May 2025).

Recently, Eaton introduced Breaktor, a double-pole single-throw contactor. Breaktors simplify BDU configuration [45]. Instead of using four contactors and one relay for precharge, two Breaktors and two relays are now sufficient. According to Eaton, a Breaktor can replace a pyro fuse because it provides overload protection up to 25 kA and switches off at a threshold of about 1.7 kA ($\pm 20\%$).

2.3.2 Conclusion

This literature review examined the current state on Battery Disconnect Unit (BDU) design, focusing on relay and contactor technology. While existing studies and application notes provide a foundation for BDU design, a complete design guide for BDUs is lacking. Our proposal and design for the BDU are presented in Chapter 5.

Chapter 3

ELM and SoC-OCV-T for State of Charge Estimation

3.1 Introduction

Many machine learning algorithms have been applied to state-of-charge (SoC) estimation. To determine the most suitable algorithm under the MCU's constraints, benchmarking was performed with several candidates, including LSTM, Random Forest, k-nearest neighbors (k-NN), Decision Tree, Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), CatBoost, Extreme Learning Machine (ELM) with Tanh, Sigmoid, and ReLU activation functions, as well as Online Sequential-ELM (OS-ELM).

KNN regression is a dynamic lookup table and remains one of the simplest methods among the candidates. LSTM is a time-series neural network for sequence prediction, as discussed in the literature review. Besides time-series models, tree-based algorithms are widely used in SoC estimation because they are robust and explainable through node-level scores. Their accuracy is often enhanced by ensemble learning, which can be divided into bagging and boosting. Random Forest is a bagging method that averages the outputs of multiple decision trees to form a final prediction. Boosting methods such as CatBoost, XGBoost, and LightGBM iteratively refine predictions by correcting errors from previous iterations. Finally, Extreme Learning Machines (ELM) with activation functions such as Tanh, Sigmoid, and ReLU provide a lightweight alternative, while Online Sequential-ELM (OS-ELM) further improves training speed for real-time applications.

Three main criteria were used to select the algorithm: mean absolute error (MAE), training time, and model size (KB). Model size was given the highest weight of 0.45 due

to the constraints of the embedded system. The next most important criterion was Test MAE with a weight of 0.35. Training time was assigned a weight of 0.20, since the model is not retrained during operation and the training process can instead be prepared on the cloud. In addition, because the MCU's code space is limited to 2302 KiB, a hard threshold was applied: any model exceeding this size was excluded.

It was concluded that tree-based algorithms, including Random Forest, Decision Tree, XGBoost, and LightGBM, achieved low MAE. KNN regression also performed well with low test MAE, followed by ELM-Sigmoid and ELM-Tanh. Model sizes were recorded for comparison, with ELM producing the smallest size among all algorithms. When a hard threshold was applied, ELM-Sigmoid ranked first due to its minimal model size, followed by CatBoost. In particular, three of the top five models were ELM variants. The results are summarized in Table 3.1. Based on these findings, ELM was selected as the final choice.

Table 3.1: Algorithm Benchmarking with Weighted Rank and Size Constraint

Algorithm	MAE	Training Time	Model Size (KB)	Rank Weighted	Rank Size Constraint
ELM-Sigmoid	0.014	6.069	7	6	1
CatBoost	0.014	3.895	117	5	2
OS-ELM	0.016	1.271	21	8	3
ELM-Tanh	0.014	13.159	13	7	4
LightGBM	0.013	0.892	312	4	5
ELM-ReLU	0.017	7.986	9	9	6
XGBoost	0.012	2.198	481	3	7
Decision Tree	0.006	6.577	90,600	1	Dropped
KNN	0.008	0.768	47,713	2	Dropped
Random Forest	0.006	41.108	5,981,401	10	Dropped

3.1.1 Extreme Learning Machine (ELM)

ELM is a single-layer feedforward neural network. Compared to other neural networks, it has a simple structure. Simplicity makes ELM suitable for BMS use with limited computational resources.

The first step in training the ELM was the initialization of random weights and biases. Subsequently, the outputs of the activation functions were computed. Several activation

functions were tested, including Sigmoid, Tanh, Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU), Gaussian Error Linear Unit (GELU), and Softmax. Among these, the Tanh activation function yielded the highest accuracy.

Equations (3.1)-(3.3) represent the steps for training and predicting when implementing ELM.

$$\begin{cases} \boldsymbol{\beta} = \mathbf{H_{train}}^{\dagger}\mathbf{T}, & (3.1)(Train) \\ H_{ij} = g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i), & (3.2)(Activation) \\ \hat{\mathbf{T}} = \mathbf{H_{pred}} \cdot \boldsymbol{\beta}, & (3.3)(Predict) \end{cases}$$

$\mathbf{H}$ is the hidden layer output after applying $g(.)$, the activation function. (β) (output weights) is a product of Moore–Penrose generalized inverse $\mathbf{H}^{\dagger}$ and Target matrix ($\mathbf{T}$). Weights, input features, and biases are denoted by $\mathbf{w}, \mathbf{x}$ and $\mathbf{b}$ respectively [38].

The process is as follows: first, weights $\mathbf{w}$, biases $\mathbf{b}$ are randomly initialized. Then, the inputs are passed through the activation function, yielding $H_{ij} = g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i)$. Next, the pseudo-inverse of $\mathbf{H_{train}}$ is multiplied by the target matrix $\mathbf{T}$ to compute the output weights $\boldsymbol{\beta} = \mathbf{H_{train}}^{\dagger}\mathbf{T}$. For prediction, new inputs go through the activation function to produce $\mathbf{H_{pred}}$, which is then multiplied by $\boldsymbol{\beta}$ to estimate the output: $\hat{\mathbf{T}} = \mathbf{H_{pred}} \cdot \boldsymbol{\beta}$.

3.1.2 Adaptive Algorithm

Data-driven methods exhibit prediction fluctuations, while Coulomb counting method face issue with initial capacity estimation. To mitigate these errors, we incorporated an adaptive method to adjust the SoC. The algorithm's workflow is detailed in the Adaptive Algorithm Design section. In the next sections, seven steps from processing data to implementing algorithm on BMS are reviewed.

3.2 Dataset Overview

In the study, three data sources were used: the LG dataset from McMaster University [43], the current profile from the SLIDE project [36], and the data collected from accurate FMU model from Batemo and AVL. The ELM model was first built in Python. Drive cycle data from the LG dataset was used to verify whether the prediction error remained within an acceptable range of 3%. During this stage, model parameters and loss functions were optimized.

After the algorithm was validated, current profiles from the SLIDE project [36] were used as input for the experiments. The current profiles included Highway Fuel Economy Test (HWFET), New York City Cycle (NYCC), Urban Dynamometer Driving Schedule (UDDS), and US06, as shown in Table 3.2.

Each current profile lasted between 10 and 20 minutes and was insufficient to cover the full cell capacity, depleting only about 10% of state of charge (SoC) from a fully charged cell (100%). To obtain data spanning the entire capacity range, the current profiles were repeated until the SoC reached approximately 2%. Because the SLIDE project cell had lower capacity compared to the experimental cell (Samsung INR21700G), the maximum charge and discharge currents were scaled up accordingly. The specifications and limits for the Samsung INR21700-50G cell are shown in the Table 3.3.

Table 3.2: Drive Cycle Profiles

Original Current profiles	Min (Ampere)	Max (Ampere)	Total Steps (s)	Distance (mile)	Average Speed (mph)
HWFET	-7.8668	8.1	766	10.26	48.3
NYCC	-4.4136	8.1	599	1.18	7.1
UDDS	-4.4929	8.1	1370	7.45	19.59
US06	-4.2071	8.1	601	8.01	48.37

After the current profiles were obtained, the cell was cycled at temperatures ranging from -20°C to 60°C. Lithium-ion cells are known to exhibit different behaviors under extreme conditions; therefore, additional tests at low temperatures were conducted [90].

The histogram illustrated trends and outliers in the data. As shown in Figure 3.1, HWFET and US06 exhibited the highest discharge currents due to highway driving conditions. In contrast, UDDS and NYCC showed lower discharge currents, which are characteristic of city driving conditions.

The discharge and charge currents were limited to 15.5 A and 7 A, respectively, to cover full capability of the cell. The experiment was conducted across various temperature ranges. To safeguard the cell, stop limits were implemented as outlined in Table 3.4, ensuring that the cycler halted whenever any condition was met. The sampling rate was set at 10 Hz (0.1 seconds). If the cell exceeded any limit specified in Table 3.4 during testing, the last State of Charge (SoC) was recorded, the cell was allowed to rest, and cycling resumed from that point until the final SoC was reached.

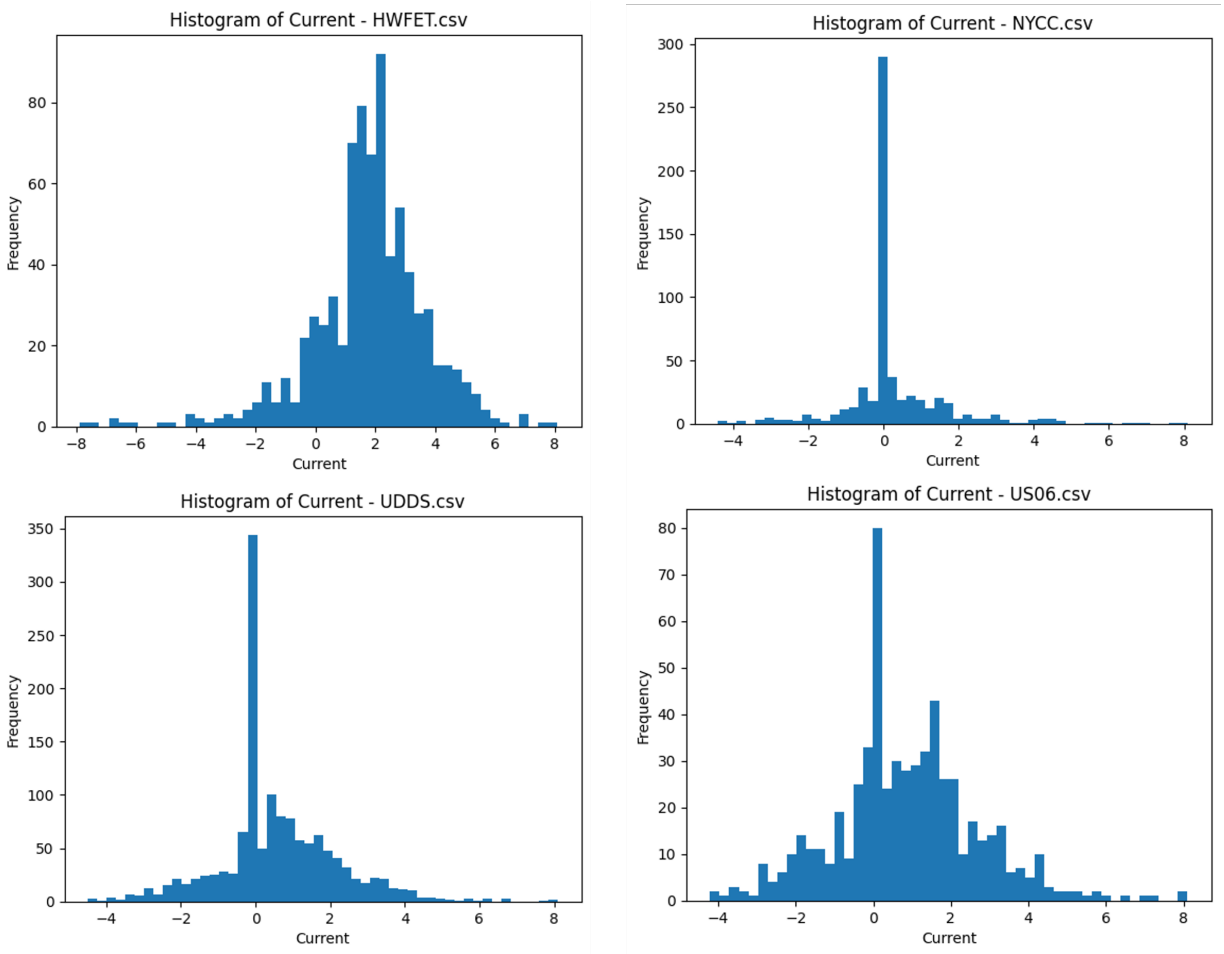


Figure 3.1: Histogram of Current Profiles

Table 3.3: Cell's Limits

	Range	Unit
Discharge Current	14.7 (Max – 2s), 9.8 (continuous)	A
Charging Current	7A (Max – 2s), 2.5 (continuous)	A
Capacity	5	Ah
Terminal Voltage	2.5 – 4.2	V
Maximum Surface Temperature	353	K

Although the maximum charge and discharge currents can be higher according to the dataset, such values are not practical for experimental operation because lower discharge currents are typically used. High currents cause rapid cell depletion, resulting in insufficient data for model training. In addition, limits were set to ensure cell safety. When these safeguards were implemented, virtual cycling was performed using the Functional Mockup Unit (FMU) provided by AVL and Batemo GmbH.

Table 3.4: Cell Safety Limits

	Range	Unit
Terminal Voltage	2.5 – 4.2	Volt
Maximum Surface Temperature	353	Kelvin
Charge	0 - 17538	Coulomb

The time required to complete all current profiles is shown in Table 3.5. The NYCC profile required the longest duration because its discharge current was lower than those of the other profiles, as shown in Figure 3.1. In total, the testing process required approximately 24 days to collect the necessary data, assuming all other conditions were met. To reduce the testing duration, three parameters could be optimized: (1) increasing the number of cyclers, (2) reducing the sampling rate, or (3) narrowing the temperature range (for example, from -10°C to 45°C).

Table 3.5: Duration for Current Profile Testing

Current Profile	Time without Rest (hours)
HWFET	0d:19h:50m
NYCC	4d:16h:05m
UDDS	2d:09h:02m
US06	1d:18h:23m
Slow discharge C/20	12d:12h:03m

After the drive cycle tests were completed, a slow discharge test at C/20 was performed to determine the relationship between open-circuit voltage (Open-circuit voltage (OCV)) and State of Charge (SoC) under various temperature conditions. In this implementation, only the discharge curve was used instead of the average discharge and charging curve. The data collected from these tests was used for developing and validating the Extreme Learning Machine (ELM) model described in the following section.

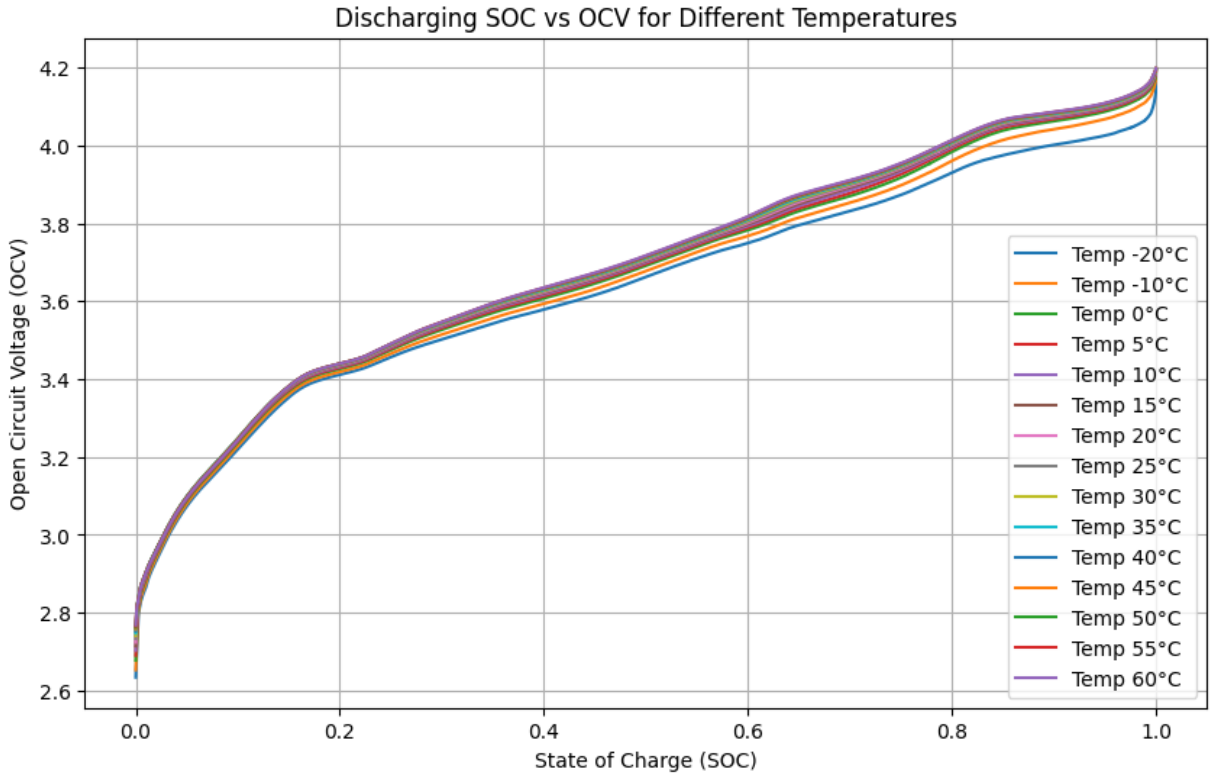


Figure 3.2: SoC-OCV Curves at Different Temperatures

3.3 ELM Model Setup

The Extreme Learning Machine (ELM) model was built in Python and evaluated using the LG dataset from McMaster University [43]. Because the model would later be deployed on a Battery Management System (BMS) using MATLAB Embedded Coder, it was also implemented in MATLAB to minimize future integration errors.

The Python-based Extreme Learning Machine (ELM) model was used to optimize hyperparameters, including the number of hidden layer nodes, the loss function, and the activation function. The optimal hyperparameters are presented in Table 3.6. Subsequently, data from all four drive cycles were combined. The training features (X_{train}) consisted of terminal voltage, surface temperature and current, while the label (Y_{train}) corresponded to the state of charge (SoC). To prevent data leakage, the dataset was split into 70% training and 30% testing after feature selection [42]. Figure 3.3 illustrates the ELM model with

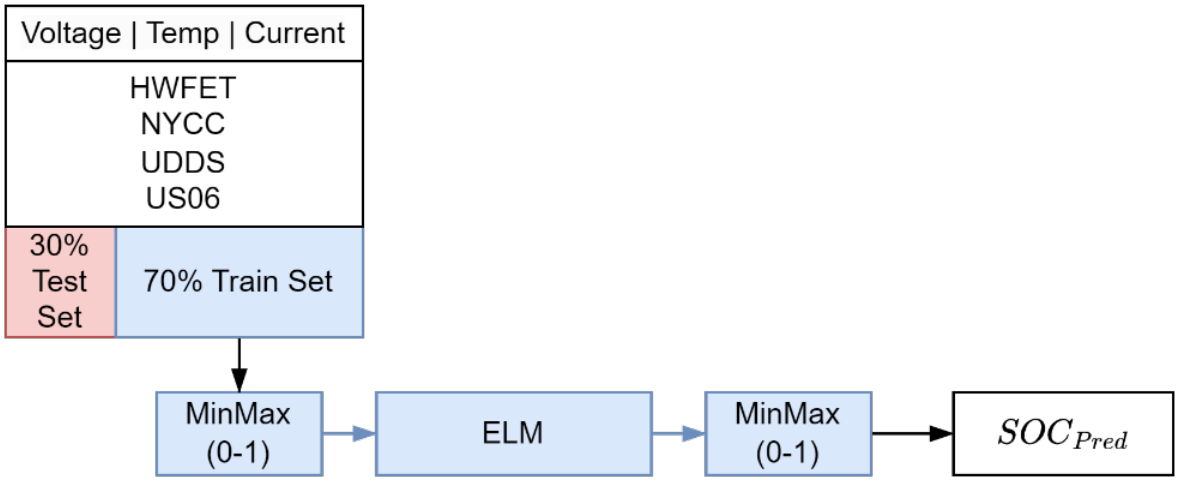


Figure 3.3: ELM Model with Inputs and Output

its inputs and output. The training set was exported to a '.mat' file for direct model training in MATLAB. Finally, data normalization was performed using min–max scaling in MATLAB after splitting, also to prevent data leakage [42].

Next, X_{train} and the selected hyperparameter configuration were used to train the model in MATLAB. After training, the weights, biases, and model structure were stored in a '.mat' file. A MATLAB function block was then added to perform inference based on the stored weights, biases, Beta, and activation function. At this stage, the model was ready to predict SoC using Terminal Voltage, Surface Temperature, and Current as inputs.

Table 3.6: Structure of ELM model

Layer	Size
Input Nodes	3
Hidden Nodes	300
Output Node	1
Activation Function	Tanh

For the State of Charge–Open Circuit Voltage–Temperature (SoC-OCV-T) lookup table, discharge curves collected at various temperatures were used. The data contained approximately one million samples, which exceeded the available code space in the Battery Management System (BMS). To address this limitation, Terminal Voltage and Temperature

were binned and arranged in a two-dimensional grid, with each grid intersection representing a SoC value. Terminal Voltage was used instead of open-circuit voltage (OCV) because, under slow discharge conditions, terminal voltage could be assumed to be approximately equal to OCV. Linear regression was applied to fill in missing SoC values in the dataset. Additionally, two sets of lookup tables with different resolutions were created as shown in Figure 3.4 to compare their effectiveness in SoC estimation.

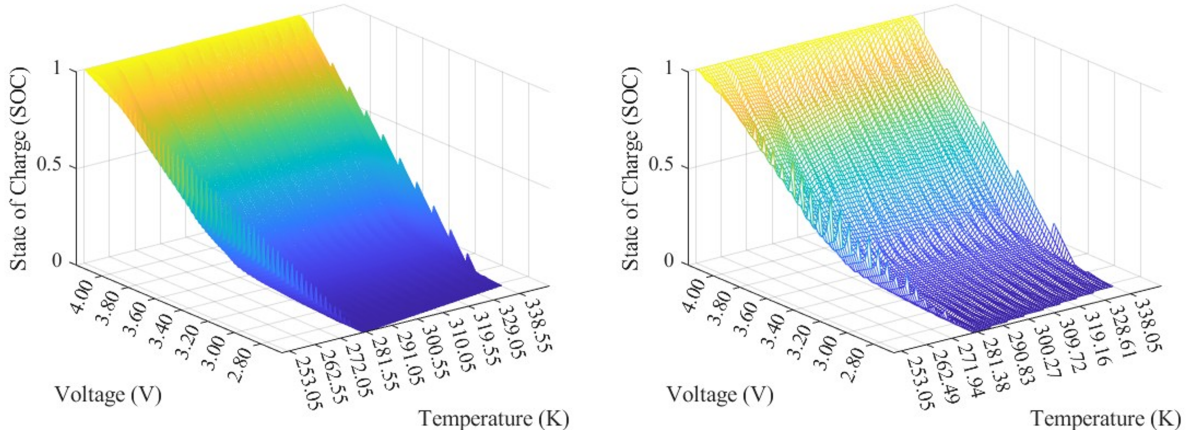


Figure 3.4: SoC-OCV at Different Temperatures and Different Size 399x172 (Left), 99x86 (Right)

Although the SoC-OCV-T table covered a wide range of temperatures, some parts of the table still had missing data, which reduced its overall accuracy. To address this issue, the method described in the next section was used.

3.4 Adaptive Algorithm Design

In this algorithm, Coulomb counting was primarily used for real-time State of Charge (SoC) estimation; however, accurate initial SoC was required. To set the initial SoC value, Terminal Voltage, Surface Temperature, and Current were monitored. These values were then used with the SoC-OCV-T table and the Extreme Learning Machine (ELM) model to adjust the SoC value obtained from Coulomb counting.

The algorithm workflow is illustrated in Figure 3.5 and proceeds as follows. When the vehicle is switched on, the last SoC from previous run is set as the current SoC assuming

there is no self-discharge. If the SoC value cannot be recalled from the vehicle's memory, the algorithm monitors discharge current during the startup phase, which is set to 250 seconds. During this phase, either a passive or an active method can be applied. In the passive method, while the vehicle completes its startup procedure, the algorithm monitors Terminal Voltage, Surface Temperature, and Current. Alternatively, because the Battery Management System (BMS) supports discharge balancing, the algorithm can actively discharge the battery with a small current over a short period using Pulse Width Modulation (PWM). These input values (Surface Temperature, Terminal Voltage, and Current) are provided to both the Extreme Learning Machine (ELM) model and the SoC-OCV-T table. The initial SoC is then set by averaging the SoC values obtained from the SoC-OCV-T lookup table and the ELM model prediction.

The Ignition Phase ends when the startup time expires. Coulomb counting is then initialized with the recently determined State of Charge (SoC), and the Cruise Phase begins. During this phase, the algorithm monitors resting events, which are defined as intervals of 80 seconds or longer with measured current less than 0.015 A.

When a resting event occurs, the algorithm obtains State of Charge (SoC) predictions from both the Extreme Learning Machine (ELM) model and the SoC-OCV-T table. It then compares SoC_Lookup, SoC_Prediction, and the average of these two predictions with the SoC value from the previous time step (SoC_{CC}). The SoC value closest to the previous value is selected.

Because the initial SoC is updated only during battery discharge, the algorithm applies the SoC-OCV-T lookup table only during discharge events. The next section illustrates the implementation in MATLAB.

3.5 Building the Model on MATLAB Simulink

MATLAB/Simulink was used to implement the algorithm. The Extreme Learning Machine (ELM) model was trained separately. The weights and Beta layers obtained from the training process were saved in a '.mat' file for use during prediction. The prediction code was integrated into a MATLAB Function Block.

Figure 3.6 shows the ELM block with three main features along with other components of the Simulink model.

The complete Simulink model is shown in Figure 3.7. In which, the Adaptive Coulomb counting module allowed the algorithm to update the initial State of Charge (SoC) at any time.

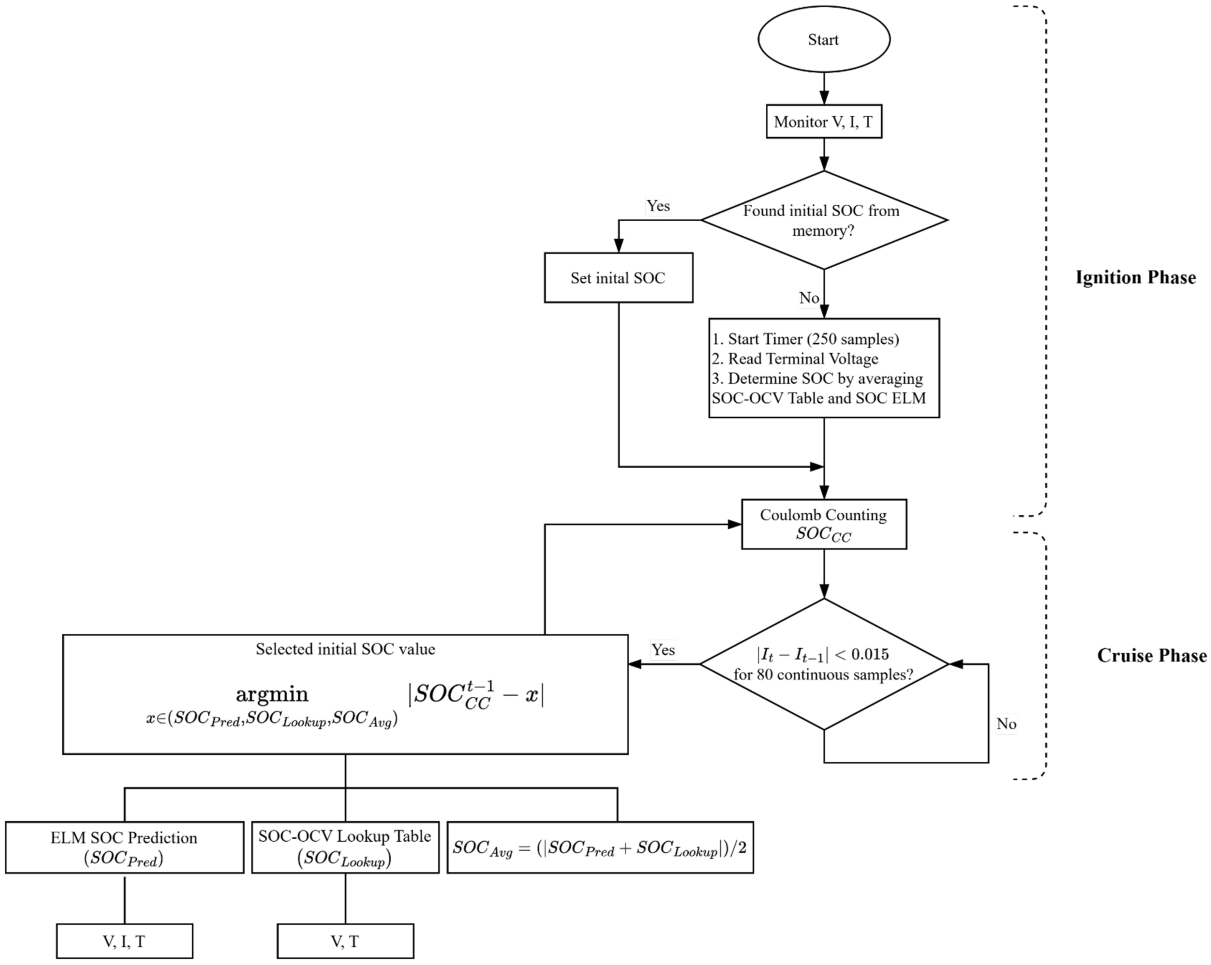


Figure 3.5: Adaptive Algorithm Flow Chart

The SoC Selector block compared the two predictions and selected the value with the smallest difference from the current SoC. The SoC-OCV lookup table was embedded in an n-dimensional lookup block using linear Lagrange interpolation.

To validate the model, a new current profile, LA92, was used. This profile was extracted and scaled up from the LG dataset. This data were not included during the training process.

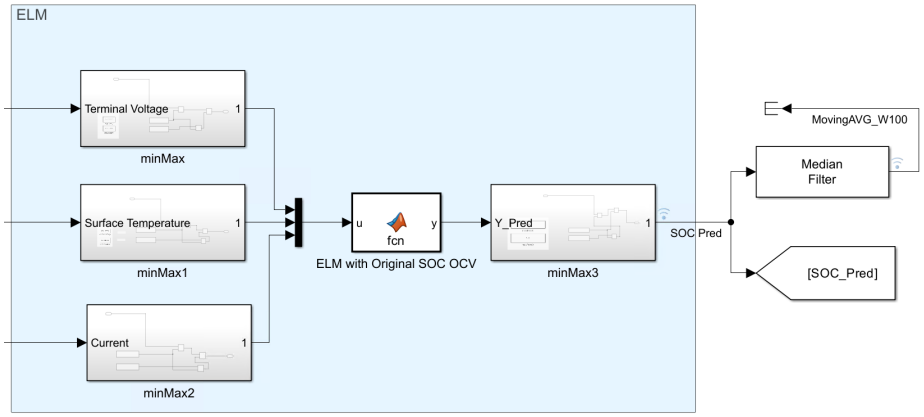


Figure 3.6: ELM Simulink Block

3.6 Building Hardware-in-the-Loop (HIL) Bench

To provide a secure system for testing the algorithm and sensors for the battery pack, a custom hardware-in-the-loop (HIL) bench was constructed with a human-centered design approach. The test bench was assembled on an acrylic sheet with separate areas designated for each component. The HIL system used the M450 as a real-time controller.

Although breadboards and jumper wires provide a fast environment for prototyping, they are not robust, and issues often arise from loose wires and unreliable connections. In the HIL design, terminal blocks and ring connectors were used instead of breadboards to ensure secure connections. The complete design is shown in Figure 3.8. The HIL system was modular and easily extendable to full vehicle-in-the-loop applications, as well as simplify future troubleshooting.

3.7 Implement Algorithm on BMS

The OpenECU M450 and the Kvaser Leaf Light v2 CAN adapter were used to communicate with the OpenECU Calibrator on a desktop computer. OpenECU provides Simulink blocks that allow code to be generated and embedded directly into the Battery Management System (BMS).

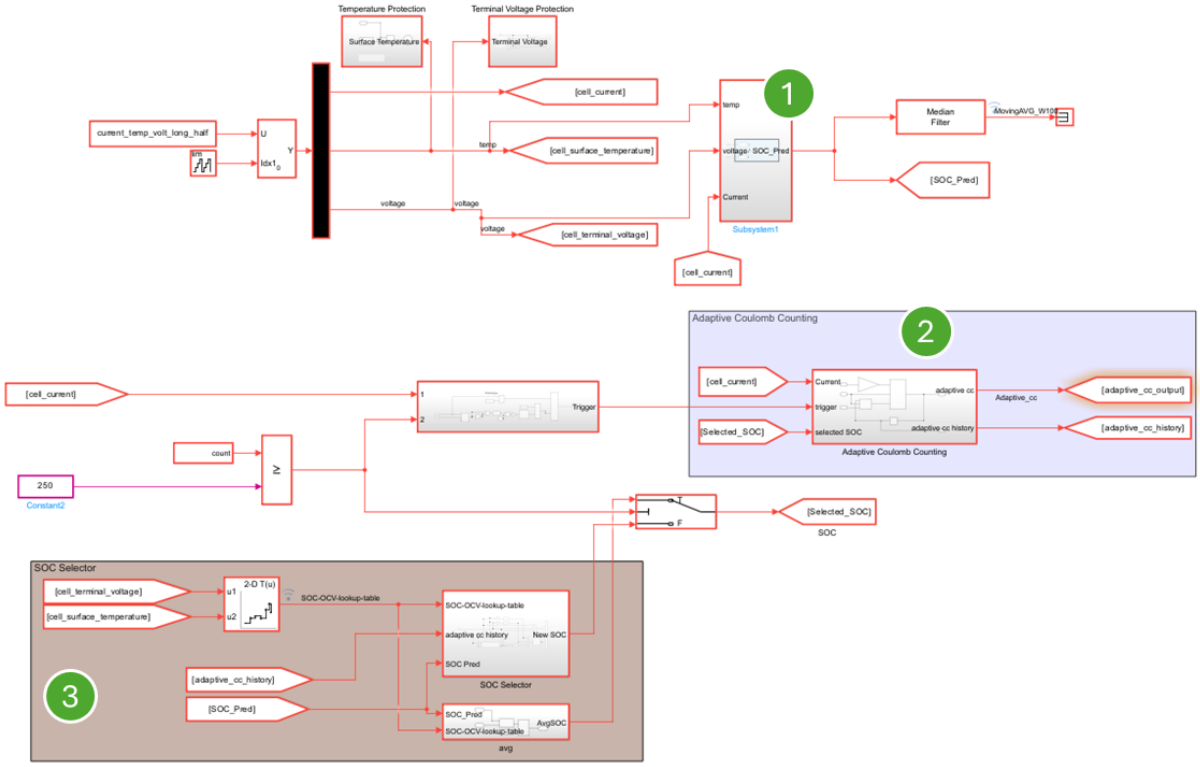


Figure 3.7: Completed Adaptive Coulomb Counting Simulink Block: (1) ELM Simulink Block, (2) Adaptive Coulomb counting, (3) SoC Selector block

3.8 Results and Discussion

3.8.1 Result from Simulation

Three metrics were selected to evaluate the algorithm: MAE, RMSE, and Maximum absolute error (MaxAE). The model was validated on fifteen temperature conditions. At high temperatures, the test was paused when the cells' Surface temperature and Terminal Voltage exceeded the thresholds. The cells were then allowed to rest before resuming testing. The test ended when the State of Charge (SoC) reached 0.02. Functional Mockup Unit (FMU) was used.

For further validation, two different lookup tables were tested: one with a size of 99 x 86, and another with a size of 399 x 172 as shown in Figure 3.4. The results in Figure 3.9 did not show significant differences between the two, indicating that the increase in resolution

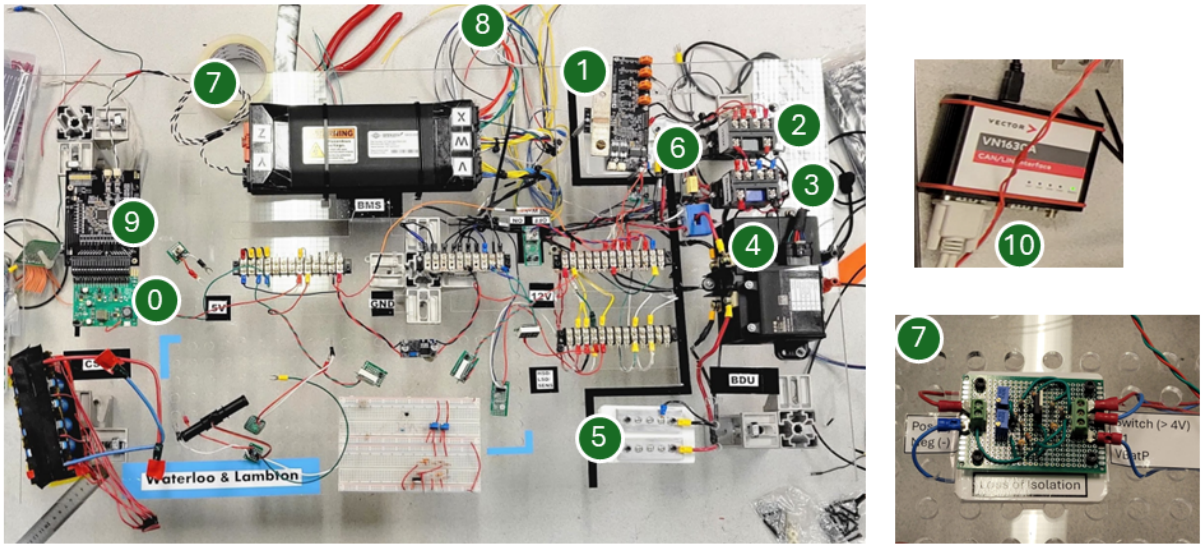


Figure 3.8: HIL bench: (1) Pack Monitor, (2) 12V Relay, (3) Precharge Relay, (4) Breaktor, (5) Negative and Positive busbar for motors, (6) Negative and positive busbar from Battery, (7) Loss of isolation, (8,10) CAN signal and CAN/LIN interface, (9) ADBMS6830B Cell Supervision Circuit, (0) DC2472A Cell Emulator

did not improve performance.

Metrics were also compared between Adaptive Coulomb counting (adaptive_cc) and State of Charge (SoC) prediction using the Extreme Learning Machine (ELM) model with a median filter employing a window size of 100 (movingavg_w100). The movingavg_w100 method relies solely on SoC predictions from the ELM model and does not utilize the SoC-OCV-T table.

Table 3.7 presents the average error across fifteen temperature conditions ranging from -20°C to 60°C, and stared at 100% SoC. Both mean absolute error (MAE) and maximum absolute error (MaxAE) of Adaptive Coulomb counting were low when using the smaller lookup table. This result demonstrates that Adaptive Coulomb counting improves the accuracy of State of Charge (SoC) estimation. Additionally, a lookup table with accurate data can outperform one with higher resolution.

Adaptive CC with the small SoC-OCV-T table achieved a mean absolute error (MAE) of 1.8%. This result indicates that the minimal data-driven approach is effective for State of Charge (SoC) estimation.

To further analyze the algorithm, each ambient temperature condition was evaluated

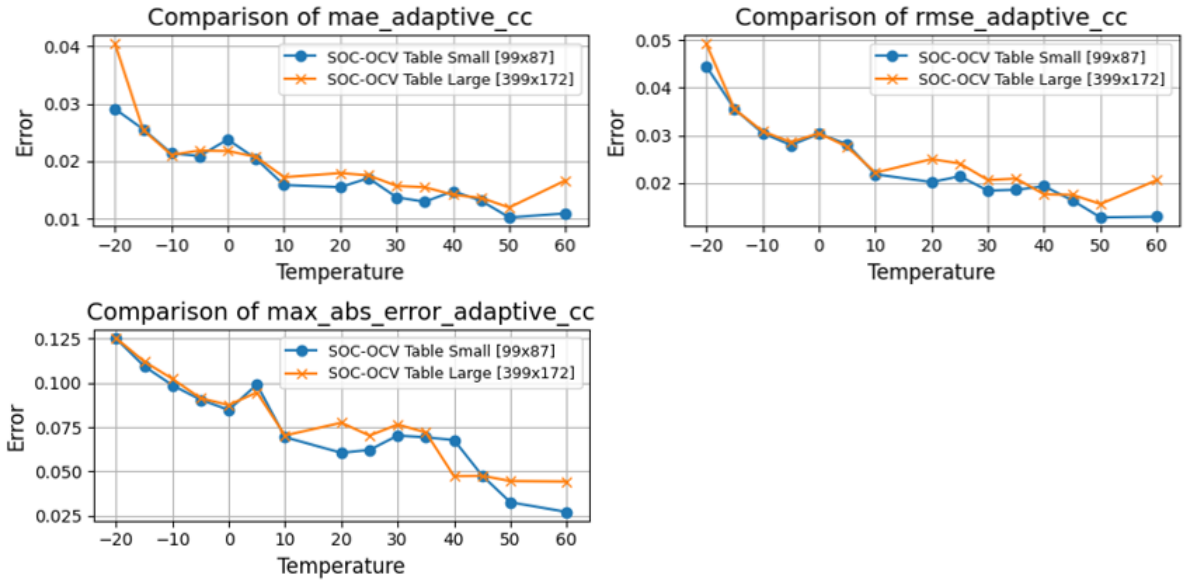


Figure 3.9: Comparison Between Small and Big SoC-OCV-T Table

Table 3.7: Comparison of Error Metrics Between Moving Average and Adaptive Coulomb Counting Methods on Small and Big Lookup Tables

Metric	MovingAvg Small Table [99x86]	MovingAvg Big Table [399x172]	Adaptive CC Small Table [99x86]	Adaptive CC Big Table [399x172]
MAE	0.017759173	0.018200976	0.017689804	0.019450565
RMSE	0.022414046	0.022718813	0.023958472	0.025797389
MaxAE	0.103272003	0.100245299	0.074269086	0.077566062

using the smaller SoC-OCV-T lookup table.

Table 3.8: Error Metrics for Three Methods: SoC predictor, Adaptive Coulomb Counting (adaptive_cc), and Moving Average (movingavg_w100).

Temp	SoC			Adaptive CC			MovingAvg		
	MAE	RMSE	MaxAE	MAE	RMSE	MaxAE	MAE	RMSE	MaxAE
-20	0.025	0.035	0.171	0.029	0.045	0.125	0.024	0.031	0.113
-15	0.025	0.032	0.129	0.026	0.036	0.109	0.024	0.030	0.101
-10	0.022	0.029	0.124	0.021	0.030	0.099	0.020	0.026	0.098
-5	0.022	0.028	0.115	0.021	0.028	0.091	0.021	0.026	0.095
0	0.025	0.030	0.107	0.024	0.030	0.085	0.024	0.028	0.091
5	0.020	0.025	0.095	0.020	0.028	0.099	0.019	0.023	0.086
10	0.017	0.022	0.086	0.016	0.022	0.069	0.015	0.019	0.080
20	0.017	0.021	0.074	0.015	0.020	0.061	0.016	0.019	0.068
25	0.019	0.023	0.068	0.017	0.021	0.062	0.019	0.022	0.062
30	0.015	0.019	0.064	0.014	0.018	0.070	0.014	0.018	0.066
35	0.013	0.018	0.072	0.013	0.019	0.069	0.013	0.017	0.074
40	0.014	0.018	0.118	0.015	0.019	0.068	0.014	0.018	0.081
45	0.015	0.019	0.131	0.013	0.016	0.048	0.015	0.019	0.092
50	0.013	0.017	0.114	0.010	0.013	0.033	0.013	0.018	0.158
60	0.013	0.017	0.322	0.011	0.013	0.027	0.014	0.022	0.282

Table 3.8 shows that when the temperature was below 20°C, the Extreme Learning Machine (ELM) model with a moving average window achieved a lower maximum absolute error (MaxAE). However, as temperature increased, Adaptive Coulomb counting (Adaptive CC) produced a lower MaxAE compared to the moving average method. This improvement can be attributed to the increased accuracy of the SoC-OCV-T discharge curve at higher temperatures. The results also informed further enhancements to the adaptive algorithm, such that the ELM model with moving average is used when the temperature is below 20°C, and Adaptive CC is selected when the temperature exceeds 20°C. The next step involves implementing the model and algorithm on an actual Battery Management System (BMS) using the custom hardware-in-the-loop (HIL) test bench.

3.8.2 Results for Implementation of SoC Estimation Algorithm on BMS

After simulation was completed, the compiled binary was embedded in the BMS, and signals were visualized using OpenECU Calibrator. The setup to program the M450 is shown in Figure 3.10.

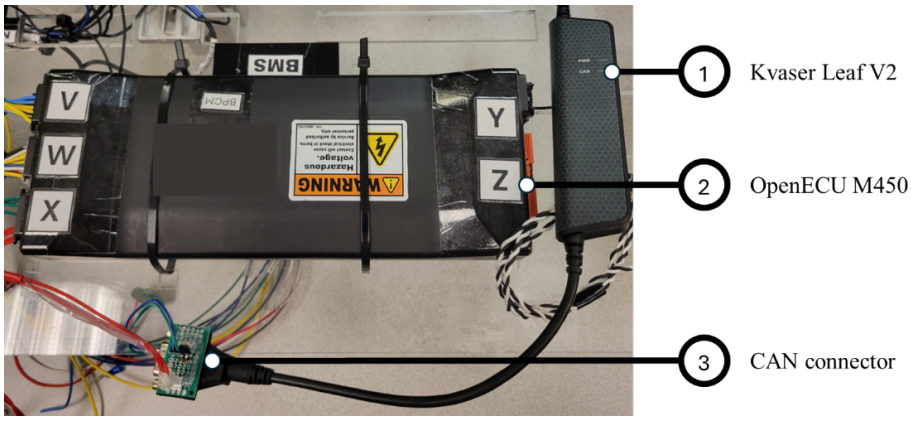


Figure 3.10: Setup to Program M450

To test real cells, a dump load and discharge switch were required. Due to time constraints and hardware limitations, 5000 samples of the LA92 current profile were used. The Surface Temperature, Terminal Voltage, Current, and true State of Charge (SoC) from the LA92 cycle at 25°C were provided as inputs to the Battery Management System (BMS).

OpenECU Calibrator was used to monitor and record measurements directly during BMS operation as shown in Figure 3.11.

In Figure 3.12, the results from the BMS were collected and compared with the actual SoC.

According to Figure 3.11, the stack size while operating (cpu_usedstacksize_bytes) was about 3912 bytes that takes up about 1% RAM space of the OpenECU M450 (384 KiB) [19]; therefore, advanced machine learning algorithm could be implemented on this BMS.

3.8.3 Discussion

Potential enhancements to this algorithm can be made through improvements in three areas, i.e., the SoC-OCV-T table, the ELM model, and the adaptive algorithm. Further

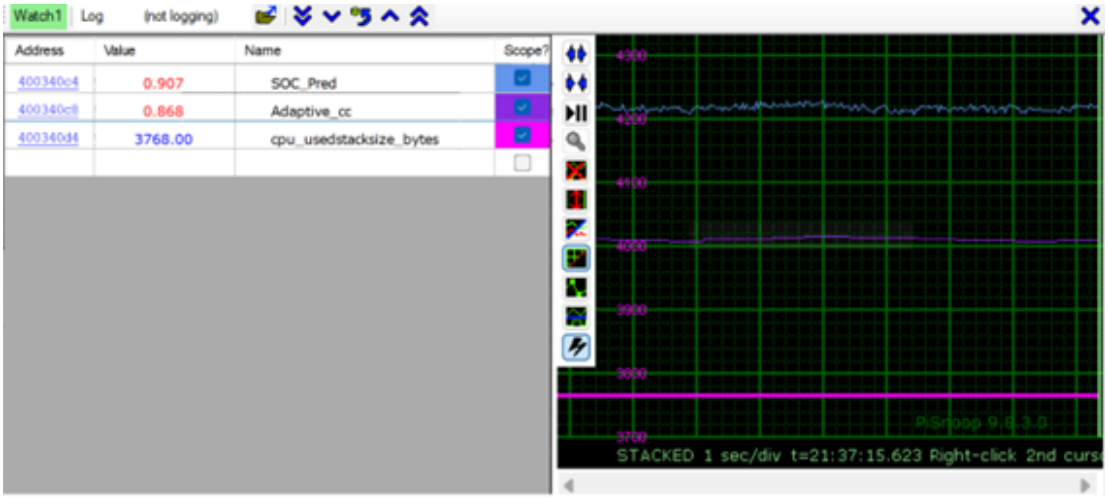


Figure 3.11: Monitoring Real-Time SoC_Pred and Adaptive_cc with OpenECU Calibrator

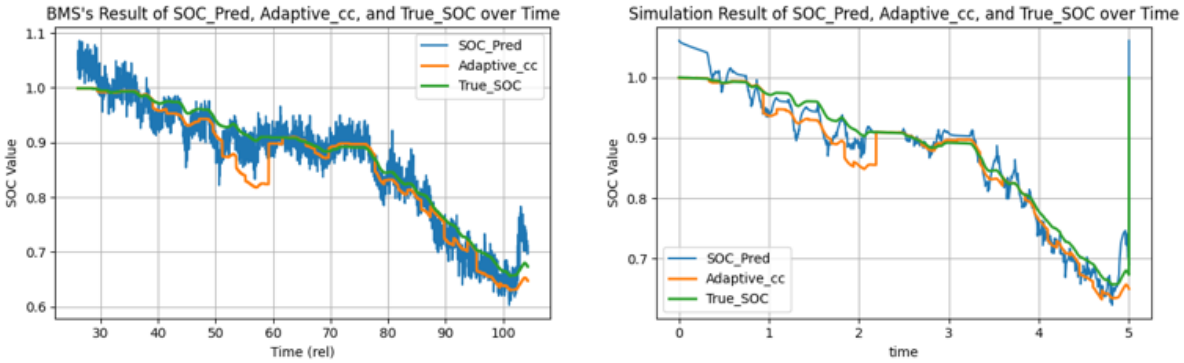


Figure 3.12: Result from BMS (Left) and Simulation (Right)

improvements are considered as items for next steps.

SoC-OCV-T Table

Instead of relying solely on Coulomb counting with a small discharge current, alternative methods such as the voltage relaxation technique can be explored for estimating the State of Charge–Open Circuit Voltage (SoC-OCV) relationship.

To improve SoC-OCV estimation, especially at low temperatures where higher errors are observed, additional temperature ranges can be included. Current testing is performed

at discrete intervals such as [-20, -10, 0, . . .]. Greater resolution can be achieved by adding intermediate temperatures, for example, [-20, -15, -10, -5, 0].

ELM Model

For the Extreme Learning Machine (ELM) model, techniques used in Long Short-Term Memory (LSTM) networks can be adapted. Incorporating a moving average window as an input may reduce outliers and improve algorithm performance.

Adaptive Algorithm

The proposed adaptive algorithm tends to favor the dominant prediction curve. To address this, weighing strategies can be applied. Another improvement is the use of a timer or counter; after a specified duration, the algorithm can alternate between SoC values from the lookup table or ELM prediction, rather than continuously comparing differences.

Simulation results indicated that the ELM model performs better at high temperatures, while Adaptive Coulomb counting with an average window is more effective below 20°C. A switching mechanism can be implemented to select the optimal method based on temperature.

Other parameters, such as duration and trigger for resting events, can also be refined. Currently, a current less than 0.015 A for 80 time steps or 80 seconds is used, based on the assumption that vehicles experience periods of no discharge, such as waiting at traffic lights. Further investigation into the relationship between the SoC-OCV-T table and charge/discharge current is recommended.

Next Steps

Given the fast learning rate of the ELM model, the Battery Management System (BMS) can be allowed to collect data during initial operation and train the model while charging the vehicle. Although the fusion approach using SoC-OCV results in low errors, the state of health of the battery should also be considered.

Additionally, noise measurements are not included in the current simulation. Future work can introduce noise to evaluate model robustness in noisy environments.

Chapter 4

Robust State of Charge Estimation Algorithm with Differential Features

4.1 Introduction

In the previous adaptive algorithm, if current reading was fluctuating due to noise, the algorithm might not detect small changes in current, which led to inaccurate SoC adjustment. Additionally, the SoC output from the lookup table would be shifted if the input had noise. Based on these observations, SoC-OCV-T was deemed unsuitable in the presence of noise unless filters were added. Besides, in Table 4.1, starting at 0.75 SoC (75%), the RMSE and MAE of the ELM model was 0.30 (or 30%) at $-20\,^{\circ}\mathrm{C}$ indicating that the model did not perform well at low temperature.

Table 4.1: ELM Model performance at three different temperatures and SoC of 0.75

Temperature	RMSE	MAE
-20	0.3014	0.3007
20	0.0443	0.0424
60	0.0155	0.0131

Next, a test was conducted at SoC of 0.75 and temperature of $20\,^{\circ}\mathrm{C}$ to see if the model could adapt to different initial SoC values. The results in Figure 4.1 showed that ELM model was better than SoC-OCV-T lookup table at predicting initial SoC with the difference up to 0.02.

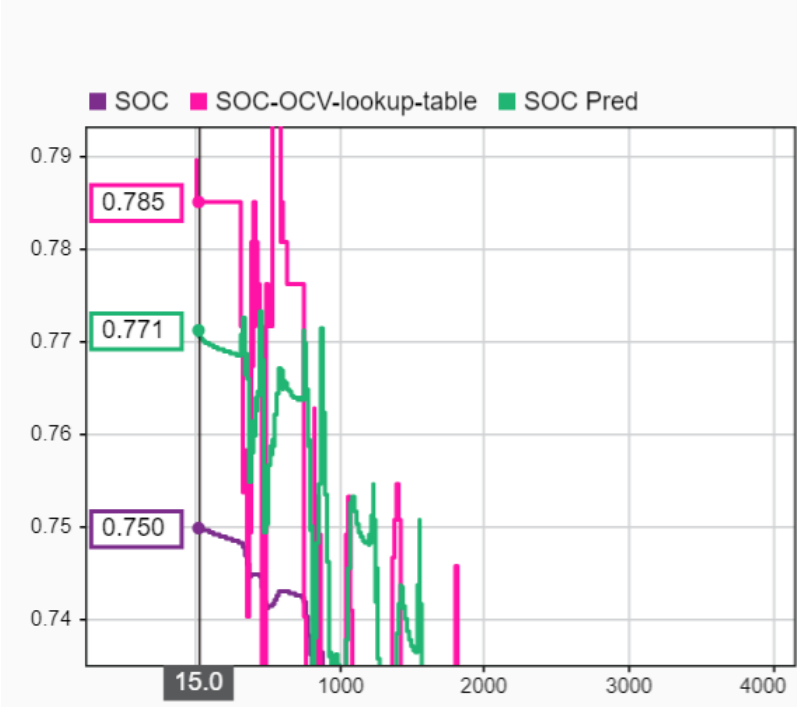


Figure 4.1: Initial SOC Prediction from SoC-OCV-T Lookup Table and ELM Model with three features of Terminal Voltage, Surface Temperature, Current

In summary, the previous ELM model did not work well at low temperatures and further improvement was required. Additionally, SoC-OCV-T lookup table was only accurate at starting or during a rest and ELM could perform that task well. Therefore, SoC-OCV-T lookup table would be excluded.

To improve the model, a simple but effective approach was developed. Instead of modifying kernel or using other methods as described in the literature review, the new approach involved selecting the best features from feature engineering process. Among other features, differential features performed robustly at extreme temperatures ($-20\,^{\circ}\mathrm{C}$, or $60\,^{\circ}\mathrm{C}$) and in the presence of noise.

4.2 Survey on Differential Features for SoC

Augmented features for SoC are often selected based on Pearson correlation coefficients (PCCs) between features and SoC value [52]. It is common to see differential features in

State of Health (SoH) estimation. Wang et al. proposed a capacity difference features (CD) based on dQ/dV curves over different cycles [91]. In addition to CD, Gaun et al. proposed dT/dV [32]. For SoC, differential features have been less common.

4.3 Prepare New Features

In order to compare and select the best features, many new features were created from the three original features (Terminal Voltage, Surface Temperature, Current). These were power, rate of change, lag (prior values), accumulated current, accumulated power, average temperature, average current, and average voltage. The new features could be categorized into four different groups, i.e., accumulated, average, prior, and rate of change. Equations for new features are shown in Table 4.2.

All features were evaluated with the dataset from the previous chapter. The first noticeable feature was the prior values of SoC. It was found that models that included prior SoC values as features achieved a lower error score compared to the cases where other features were considered. The benchmark results in Table 4.3 showed that features that included prior values were in the top five without SoC-OCV-T table.

Prior SoC feature improved ELM model; however, since the training data and evaluation both started at 100% SoC, the model might be overfitting to the initial value. To test the previous hypothesis, another evaluation was conducted at 25 °C with different initial SoC levels 0.50, 0.75, and 1.00. In this experiment, a new test set was used. Besides different initial SoC values, a sudden shift in SoC was also added to assess if the models were able to recover. The results showed that the SoC_lag feature exhibited instability, whereas the differential features emerged and adapted to the change in initial SoC values. Table 4.4 showed top five feature sets that included differential features. Therefore, SOC_lag features were excluded from further evaluation.

Apart from robustness under noise and extreme temperature (−20 °C and 60 °C), an SoC algorithm was required to meet an average error of less than 3%. The existing differential feature models did not meet the requirement; so improvement was needed. As observed in the previous assessment, adding new features might improve overall accuracy. Pearson correlation coefficients analysis was performed, and a simplified version was shown in Figure 4.2.

It was found that Surface Temperature had close correlation with Terminal Voltage. Also, Current has close correlation with Terminal Voltage. Therefore, Current (I), Terminal

Table 4.2: Time-series features for SoC estimation algorithm

Feature	**Equation / Definition**	**Description**
Original	$V(t),\ I(t),\ T(t)$	Terminal Voltage, Current, and Surface Temperature
Power	$P(t) = V(t)\, I(t)$	
Lag (by k samples)	$L^k\{x\}(t) = x(t-k)$	Past value of a feature shifted by k time steps
Rate of change	Let $x \in \{V, T, I, P\}$. $\dfrac{\Delta x}{\Delta t}(t) = \dfrac{x(t) - x(t-k)}{\mathrm{Time}(t) - \mathrm{Time}(t-k)}$	Discrete derivative over a k-step interval
Moving average (window w)	$\bar{x}_i = \dfrac{1}{\min(i, w)} \displaystyle\sum_{j=\max(1,\, i-w+1)}^{i} x_j$	Mean of the last w samples
Rolling maximum (window w)	$\max_i = \max\{\, x_j \mid j \in [\max(1,\, i-w+1),\, i]\,\}$	Maximum over the last w samples
Rolling minimum (window w)	$\min_i = \min\{\, x_j \mid j \in [\max(1,\, i-w+1),\, i]\,\}$	Minimum over the last w samples
Exponential moving average (EMA)	$\mathrm{EMA}_t = \alpha\, x_t + (1-\alpha)\, \mathrm{EMA}_{t-1}$ $\alpha \in (0, 1]$	Exponentially weighted average of x_t
Accumulation of Current	$S_i = \displaystyle\sum_{j=1}^{i} I(t_j)$	Discrete integral of current

Voltage (V), and Surface Temperature (T) were combined into a new differential feature and referred to as Power_Temp_Rate feature as showed in equation (4.1).

$$Power\_Temp\_Rate = \frac{V * I}{T} \tag{4.1}$$

Additionally, to let the Power_Temp_Rate feature capture the dynamic of the data, rate of change were then implemented over different time periods as shown in equation 4.2.

Table 4.3: Benchmark Result for New Feature Sets (Ranked by MAE). Lag or prior value features were highlighted as green. New discovered feature Power over Surface Temperature was highlighted as blue.

Features	Test MSE	Test MAE	Test R2	Feature Components
Lag_Rate_Feature_2	0.000058	0.004491	0.999226	Terminal Voltage, Current, Surface Temperature, SOC_lag_1, Voltage_lag_3, SOC_Rate_1, Voltage_EMA_0.1
Ultra_Base_No_ST_2	0.000087	0.00454	0.998848	Terminal Voltage, Current, SOC_lag_1, SOC_lag_3, Voltage_EMA_0.1
Customized_Feature_11	0.000058	0.004704	0.999228	Terminal Voltage, Current, Charge_Accumulation, SOC_lag_3, Voltage_rolling_max_10, Terminal_Voltage_rolling_20
Customized_Feature_6	0.00006	0.004761	0.999204	Terminal Voltage, Current, Surface Temperature, Charge_Accumulation, Voltage_EMA_0.1
Ultra_Optimized_4_2	0.000094	0.004868	0.998746	Terminal Voltage, Current, SOC_lag_1, SOC_lag_3, SOC_lag_5, Voltage_EMA_0.1

Another assessment was conducted with all new features.

$$Power\_Temp\_Rate\_\{k\} = \frac{P^t - P^{(t-k)}}{T^t - T^{(t-k)}} \tag{4.2}$$

The results indicated that one, three, or five time steps, often yielded better results compared to longer time steps. Therefore, a final feature set included six features: power over temperature in one, three, five time steps (Eq. 4.3 with k=1, 3, 5), Current over three time steps (refer to Table 4.2 with k=3), Terminal Voltage and Current. The features were as follows: Power_Temp_Rate_1, Power_Temp_Rate_3, Power_Temp_Rate_5, Current_Rate_3, Terminal Voltage, and Current.

Table 4.4: Benchmark Result with suddent shift in SoC ($\pm 0.05, 0.1, 0.15, 0.2$) at different temperature (0, 20, 40) and initial SoC values (1, 0.75, 0.5)

Feature Set	MAE Compared to Baseline (0.02)	Features
Customized_Feature_9	0.0008	Terminal Voltage, Current, Surface Temperature, Voltage_rolling_max_10
Multi_Rate_Feature_2	0.0093	Current_Rate_3, Terminal_Voltage_Rate_3, SOC_Rate_3, Temp_Rate
Ultra_Minimal_No_ST_2	0.0077	Terminal Voltage, Current, Power_Rate_1, Terminal_Voltage_Rate_1
Ultra_Base_No_ST_7	0.0056	Terminal Voltage, Current, Terminal_Voltage_Rate_1, Current_Rate_1, Power_Rate_1, Temp_Rate
Feature_7_Plus_Power_Rates	0.009	Terminal Voltage, Current, Surface Temperature, Voltage_EMA_0.1, Power_Rate_1, Power_Rate_3, Power_Rate_5

After having all features, a benchmark was run with different SoC levels and a temperature ranging from $-20\,^{\circ}\mathrm{C}$ to $60\,^{\circ}\mathrm{C}$ again. The top 10 features were selected.

Finally, Terminal Voltage, Surface Temperature and Current were shifted to evaluate if top 10 features could recover with all changes. The offset ranges for current were ± 0.5A, ± 1.0A. The offsets for voltage were ± 0.05, ± 0.02, $\pm 0.1\,$V. And the temperature offsets were ± 1.11, ± 2.77, ± 5.55, $\pm 8.33\,^{\circ}\mathrm{C}$. As a result, the differential feature set proposed above ranked first for Surface Temperature and Terminal Voltage and ranked second for Current shifts.

Although having low MAE and RMSE, the new prediction model was sensitive to sudden increase in discharge current during dynamic driving cycle as shown as a green line in Figure 4.3. To resolve the issue, in the final algorithm, ELM's model was used to set initial SoC for Coulomb counting and used Coulomb counting as a baseline to update SoC

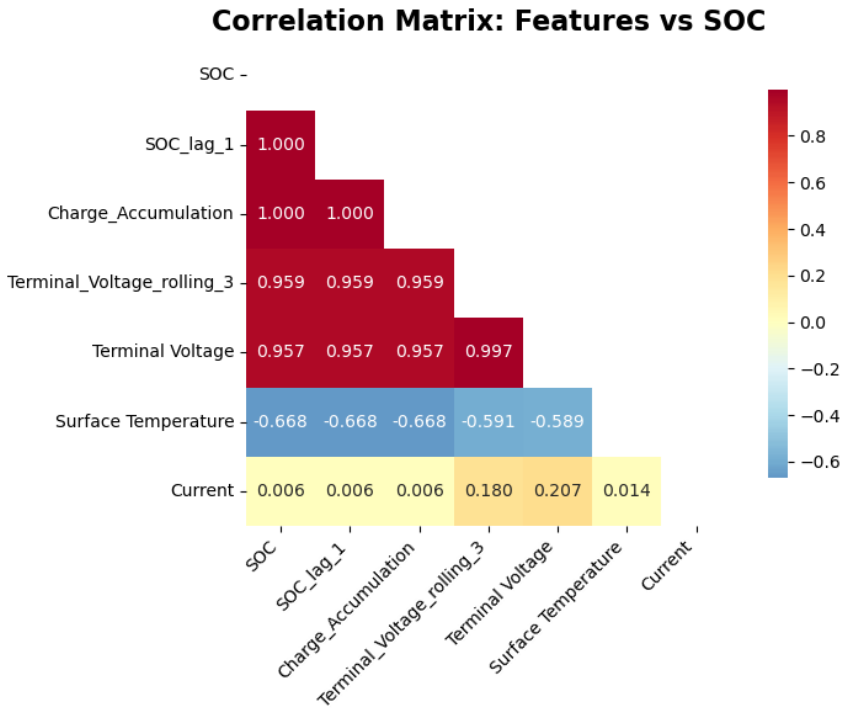


Figure 4.2: Simplified Correlation Matrix (SOC_lag_1 is prior SOC value at 1 time step. Terminal_voltage_rolling_3 is moving average with size of 3)

value.

The final SoC estimation value was ELM's output if the differences between the model and Coulomb counting results was 0.01. Otherwise, Coulomb counting was used. Implementing Coulomb counting reduced the prediction error from 0.06 to 0.02 at the second cursor as shown in Figure 4.3.

4.4 Implementation on BMS

The new algorithm was implemented on the BMS. Figure 4.4 showed Simulink blocks for new ELM model with differential features.

Figure 4.5 presents the real-time values obtained in OpenECU. The results show clear improvements over the previous algorithm based on SoC-OCV-T, as shown in Figure 4.6.

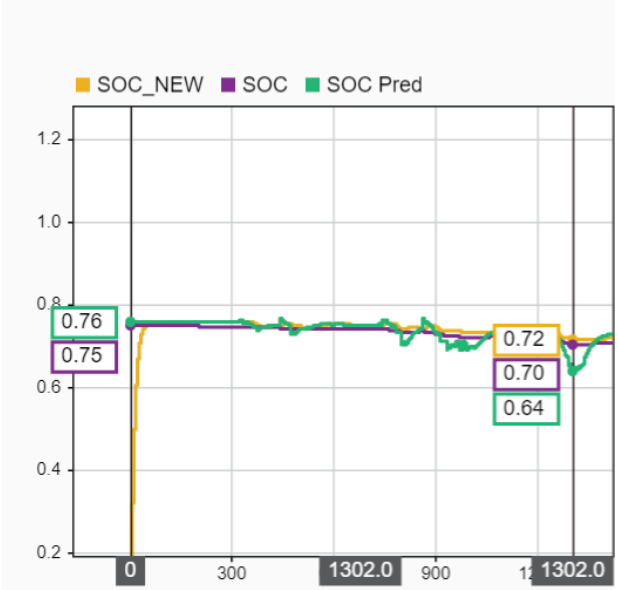


Figure 4.3: Initial prediction starting at 0.75 SoC. SOC_NEW was SoC Prediction with Coulomb Counting. SoC was the True SoC. SOC_Pred was SoC prediction from ELM Model. Without Coulomb counting, SOC_Pred is sensitive to suddent current change. SOC_NEW solved the issue.

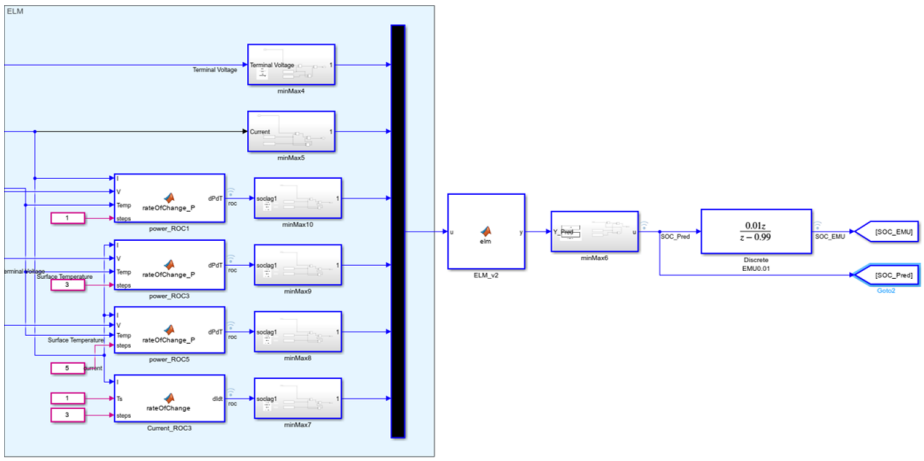


Figure 4.4: ELM with Advanced Features Simulink Block

Furthermore, reducing the number of blocks would ease the deployment process when a new fine-tuned model is added.

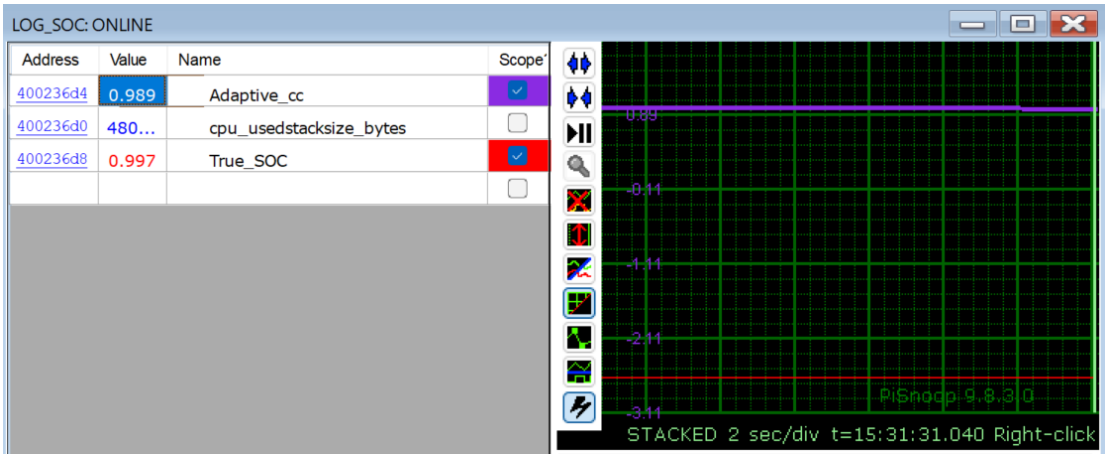


Figure 4.5: Monitoring Real-Time Robust Adaptive_cc with OpenECU Calibrator

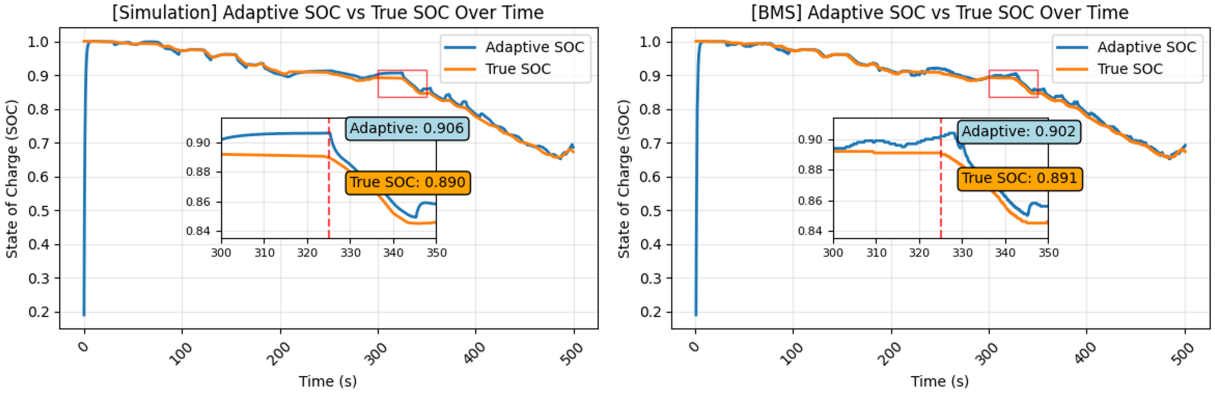


Figure 4.6: Result from Simulation (Left) and BMS (Right)

4.5 Result and Discussion

Differential features worked well with noise and low temperature. Exponential filter was also added to flat out the prediction value. The final results showed in Table 4.5 included Gaussian noise. The new model performed better with noise. RMSE and MAE were 0.0113 and 0.0095 respectively at $-20\,^{\circ}\mathrm{C}$ indicating the model was robust at low temperature and noise environment.

Table 4.5: Comparison between new algorithms with and without noise starting at 75% SoC

	Noise		Without Noise	
Temperature	RMSE	MAE	RMSE	MAE
-20	0.0113	0.0095	0.0171	0.0161
20	0.0149	0.0134	0.0211	0.0201
60	0.0159	0.0151	0.0237	0.0229

Chapter 5

Battery Disconnect Unit (BDU)

5.1 Introduction to BDU

The Battery Disconnect Unit (BDU) is an essential element associated with the battery pack in electrified vehicles. It consists of three main components: busbar, contactor, and fuse. In addition, thermistors and Hall sensors are included to monitor temperature and current, respectively. To ensure reliability, the arrangement of components inside the BDU must satisfy creepage and clearance requirements for both high-voltage and low-voltage. The following sections will analyze materials used in BDU.

5.1.1 Material: copper vs aluminum

Copper has long been the material of choice in high-voltage Busbar due to its excellent electrical conductivity and thermal properties. However, copper's high density and cost make it less ideal for lightweight designs.

Aluminum provides a lighter and more cost-effective alternative, but its electrical conductivity is lower [5]. Aluminum may also deform under prolonged exposure to elevated temperatures [18]. For BDUs with small size and high current demands, the weight savings of aluminum must be balanced against its reduced conductivity. Therefore, copper busbars are recommended for a compact design.

The use of different materials within the same design can lead to galvanic corrosion. Dexter [20] proposed strategies to mitigate this effect. The best approach is to use the same material throughout the design. If different materials are required, insulation should

be added between them. In addition, anodic materials should not be used as fasteners for cathodic materials. For example, copper busbars should not be combined with aluminum fasteners.

5.1.2 Main Components

In addition to busbar, both passive and active protection components are required in BDU. Active protection components include contactors and pyro fuses, while passive protection components include conventional fuses. These components are described in the following subsections.

Mechanical Contactors vs Solid-State Relays

Contactors and relays are classified as active components. Their primary function is to connect or disconnect the high voltage battery during faults, service, or normal operation. In addition to providing short switching times, contactors must withstand high current surge during direct current (DC) fast charging, or during current spikes associated with acceleration. Selecting the appropriate contactor, whether mechanical or solid-state, requires a careful evaluation of trade-offs to ensure robustness and safe operation.

Mechanical contactors operate by electromagnetic actuation and are widely used because of their low cost and high current ratings. The two common coil types are those with an economizer coil and those without. In an economizer coil, a Pulse Width Modulation (PWM) circuit is used to reduce energy loss in the closed state [23]. The PWM mechanism reduces heat generation and energy consumption compared to the coil without economizer. Table 5.1 presents a comparison among four types of mechanical contactors.

According to TDK's whitepaper [98], electrical arcs are generated when a contactor opens or closes. This phenomenon occurs because of ionized air formed in the presence of a large electrical potential difference. Arcing can melt the contact points, resulting in malfunction or a welded contactor. This disadvantage of mechanical contactors can be mitigated by implementing arc suppression mechanisms. Another drawback of the mechanical contactor is the relatively long response time (5–20 ms), which may not be suitable for ultra-fast fault detection systems.

The solid-state relay (SSR) addresses many of the limitations of mechanical contactors. The SSR contains no moving parts, provides rapid switching at the microsecond scale, eliminates arc formation, and offers a longer service life compared to mechanical contactors.

Table 5.1: Comparison Table Among Four Type of Mechanical Contactor

	PWM – Voltage Feedback	PWM – Current Feedback	Two Coil	Coil Without Economizer
Working Principles	Apply PWM voltage based on calibration table.	Apply PWM voltage based on current sense.	When switched ON pickup coil will activate then transfer to holding coil	Single coil energized continuously
Advantage	Moderate Accuracy	Accuracy	Less Accuracy	Less Accuracy
Cost	Moderate	High	Low	Low
Heat	Lower than coil without economizer	Lowest	Lower than coil without economizer	Highest

In high-voltage BDU applications, SSRs can eliminate issues associated with arcing and mechanical wear. However, SSRs typically fail in a closed state [64], which poses a safety risk unless a mechanical contactor is used in conjunction. Table 5.2 compares the characteristics of mechanical contactors and solid-state relays. In the table, "H" indicates high performance or an advantage, while "L" indicates low performance.

Table 5.2: Comparison Between Mechanical Contactor and Solid-State Relay

Feature	Mechanical Contactors	Solid-State Relays
Light Weight	L	H
Cost	L	H
Rated Current	H	L
Fast Acting	L	H
Cooling Capability	H	L
Arc Quenching	L	H
Lifetime	L	H

SSR stands out in most categories except rated current and cooling capability. At the time of writing this thesis, mechanical contactors were available with higher current ratings

for DC fast charging, whereas most solid-state relays could handle only up to 200 A and were generally more expensive, according to Digikey's May 2025 catalogue.

Breaktor

Breaktor is a double poles single throw contactor from Eaton. Two Breaktors and two relays can replace four contactors and one relay for precharge circuit. The configuration of Breaktors simplify BDU configuration [45]. According to a paper from Eaton, a Breaktor can replace pyro fuses because Breaktor has overload protection up to 25kA and switches off at the threshold of approximately 1.7 kA (+/-20%).

Additionally, four contactors and one relays may require five High Side Drive pins and one common ground; however, two Breaktors and two relays may require six high side drive pins and two common ground and additional two high side drive pins signaling open or closed state of the Breaktors. The comparison shows a tradeoff between compact packaging, and number of I/O pins required.

Table 5.3: Trade of Between I/O Pins and Components Required for BDU

	Breaktor	Contactor
Main components	4	5
I/O pins	10	6

In developing the BDU for the battery pack in the Battery Workforce Challenge (BWC), Breaktor was selected due to its accessibility, straightforward wiring, and suitability for battery's current and voltage range.

Pyro Fuse

Pyro fuse is a normal fuse with pyro trigger mechanism [78]. This mechanism reduces clearing time when the fuse is melted. Pyro fuse was found in Tesla's design along with contactors [84].

5.1.3 BDU Topologies: Series vs Parallel

This subsection explores different topologies for BDU.

According to David Dong, BDU can be arranged in series or parallel configuration which are illustrated in figure 5.1. In series topology, DC charger contactors are in series with the main contactors [23]. On the other contrary, in parallel topology, the DC charger contactor is connected directly to the battery. To optimize packaging size the main contactors, precharge contactor and resistor can be replaced by a solid-state relay; however, at least one mechanical contactor is required to be placed on the opposite terminal of the BDU to accommodate the SSR's issue that was mentioned above. A precharge circuit is included in both topologies to avoid inrush current caused by DC link capacitor. This circuit also avoids contactor welding during startup [60].

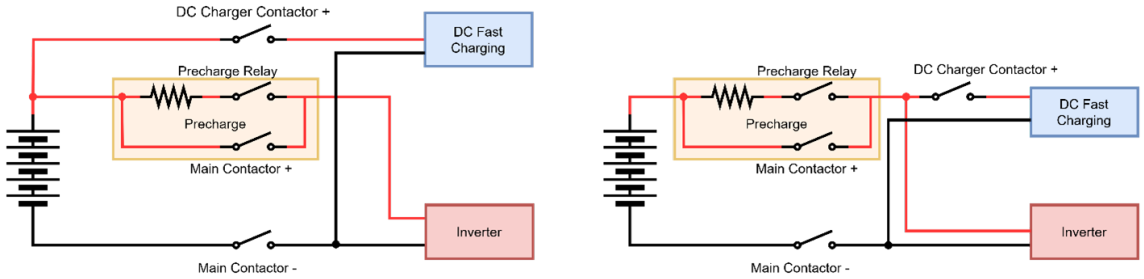


Figure 5.1: BDU Topology: Parallel Configuration (left), Series Configuration (right)

5.2 Designing BDU

In this section, steps to design and build a BDU are presented. Voltage and current inputs of the BDU are defined by the total number of cells and the pack configuration. Those values should be determined first because they will affect component selection and busbar design.

5.2.1 Determine Pack Configuration

Voltage and Capacity Requirements

Three important metrics to determine the total number of cells and pack configuration are the requirements for the battery pack including maximum, minimum operating voltage and battery capacity. For example, for the BWC project, the requirements were battery pack capacity of 60 kW. The minimum voltage was 300 V, and maximum voltage was 450

V. Additionally, it was needed to know the specification of one cell. The specifications of the selected lithium-ion battery cell are given in the Table 5.4.

Table 5.4: Cell's Specification

Item	Specification
Charging Voltage	4.2V
Nominal Voltage	3.63V
Discharge Cut-off Voltage	2.5V
Discharge Capacity	4900 mAh

Determining Total Number of Cells

The battery pack was composed of 11 modules, and each module consisted of hundreds of cells. In the module design, it was needed to determine how the cells were connected. There were two main configurations: series-first, and parallel-first [3]. The parallel-first configuration was preferred because it required fewer tap wires and was able to balance weak cells. Figure 5.2 shows an example of parallel-first configuration.

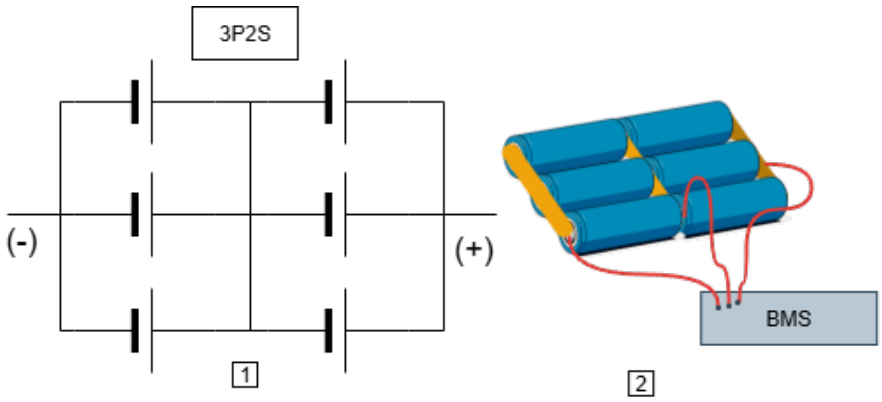


Figure 5.2: Example of Parallel First Configuration

Before deciding on the number of series and parallel cells, the constraints of the casing and the weights needed to be studied. Then, the energy of the pack was determined by

cell's capacity. The equation for calculating pack's energy and the process of selecting the number of cells are presented in Figure 5.3.

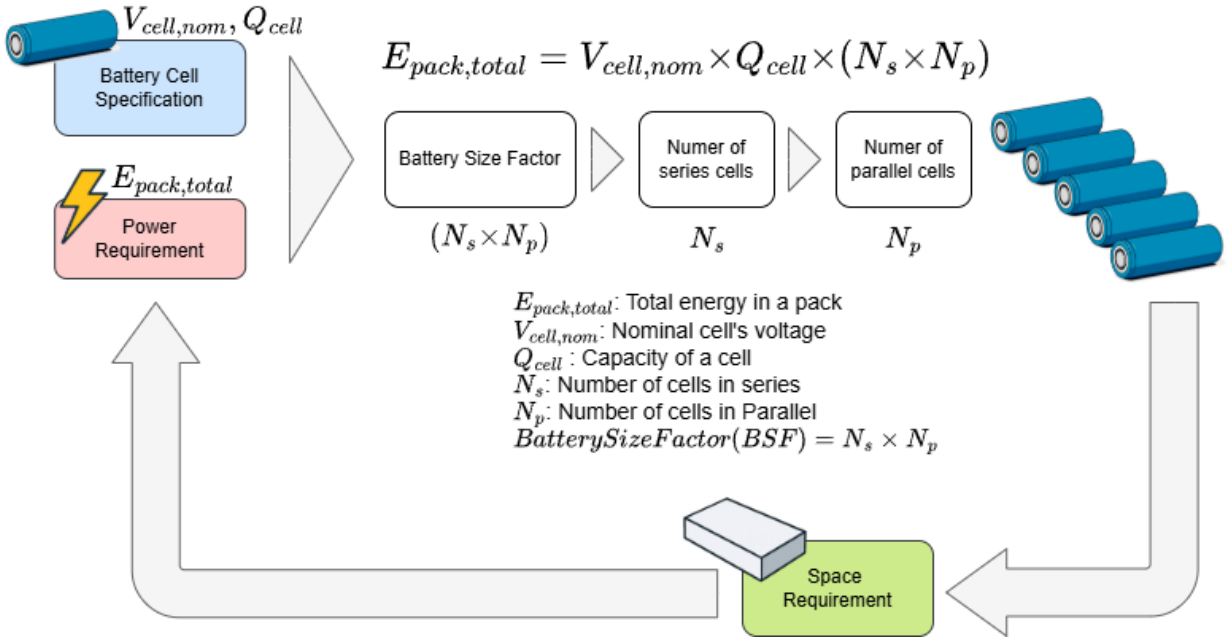


Figure 5.3: Process and Equation for Determining Cell Configuration

When a combination of cells met the energy requirement, packaging constrains needed to be checked as well. The cells would be verified through calculation and modeled in SolidWorks. If the size of all cells combined exceeded the module or pack's size, battery size factor (BSF) or total number of cells needed to be reduced or the cells needed to be rearranged.

In the design process, after evaluating all constraints, it was decided to have 11 modules of 26P10S configuration. Table 5.5 details the specifications of the battery pack.

5.2.2 Precharge Circuit

As mentioned above, the precharge circuit prevents inrush current from the battery pack to the DC-link capacitor or the filter capacitor in the inverter of traction motor. The main components of precharge circuit are precharge resistor and relay.

Table 5.5: Specification of Designed Battery Pack

Parameter	Values
Number of Module	11 modules
Number of Series cells in module	10 cells
Number of Parallel cells in module	26 cells
Battery Size Factor	2860 cells
Configuration	11 x 26P10S
Usable Battery Energy	51.909 kWh
Pack Nominal Voltage	399.3 V
Pack Nominal Current (1C ≡ 4.9A per cell)	127.4 A
Total weight	197.34 kg
Energy density	263.04 Wh/kg

There are two requirements for precharge circuit: time for precharge and capacitance of the DC-link capacitor. Time for precharge is a multiple of time constant τ, specified by the product of capacitance of DC-link capacitor and resistance of precharge resistor. It is often 3τ for 95% fully charged capacitor or 5τ for best practice.

The goal was to calculate the power and determine the resistance value for the precharge resistor. The process for determining the resistance of the precharge resistor is based on the Sensata white paper and shown in Figure 5.4 [60].

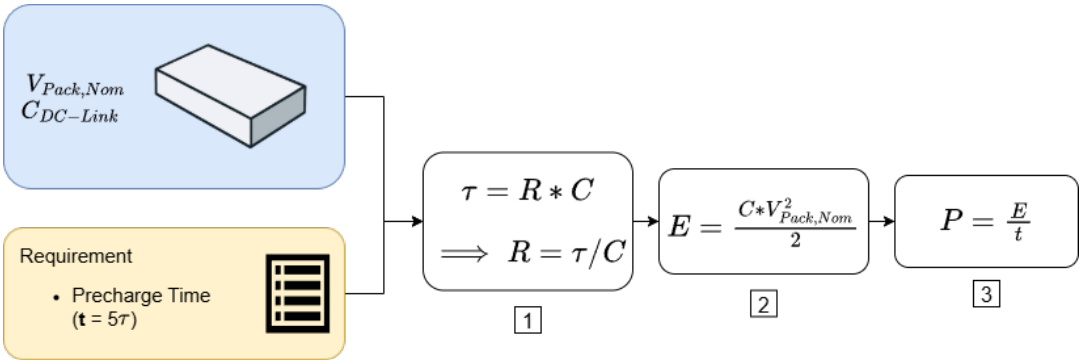


Figure 5.4: Calculation of Power for Precharge Resistor

As in box (1) of Figure 5.4, if a fixed capacitance value was given, then the resistor value would be proportional to time (t) or time constant (τ). The resistance value that met the precharge time requirement of 3τ in our case was 71 Ω. The resistance value

could be reduced to 50 or 25 Ω to shorten the precharge time and reduce resistive heating; however, lower resistance value increases the resistor's rated power (3). In the design, a 25 Ω resistor with high wattage rating was selected. Table 5.6 shows the power dissipated on 25 Ω resistor.

Table 5.6: Precharge Calculation Results

Result	Value
R	25 Ω
Energy	59.79 J
Power	637.76 W

The power rating from calculations and resistor's specification do not need to match precisely. For example, if our resistor was 25 Ω and the calculated power was 350 W, a 25 Ω resistor with 50 W or 75 W rating could be selected. The reason is that some resistors can handle short pulses of up to 25 times their rated power (see figure Power Overload on page 3 of [83]). However, it is recommended to discuss this with a vendor before finalizing the selection.

After calculation, validation was performed on Simulink as shown in Figure 5.5 to verify the effects of inrush current on our precharge circuit. Without precharge circuit, current through the capacitor was found to be huge compared to 15 A with the precharge circuit.

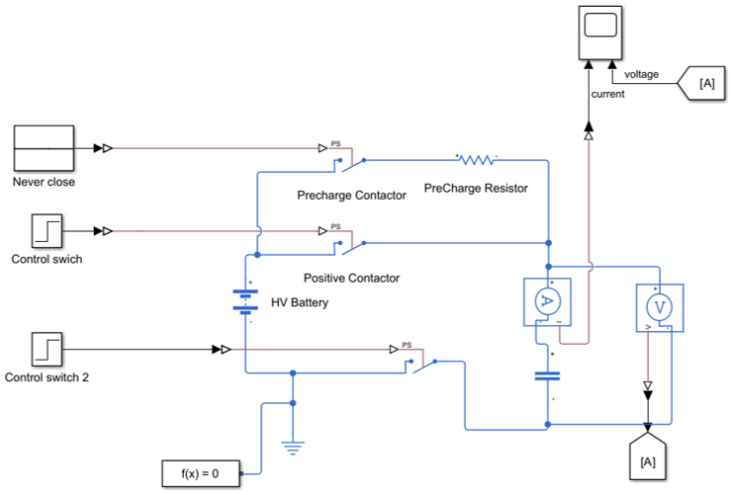


Figure 5.5: Simulate Precharge Circuit on Simulink

5.2.3 Fuse Selection

Fuse selection is an important aspect of BDU design. The procedure for calculating the fuse specification is well documented, with examples, in Eaton application note [26].

5.2.4 Creepage & Clearance

BDU is a high voltage device that includes many components; therefore, it needs to follow creepage and clearance requirements to operate safely.

In BDU, conductive materials should be kept away at a certain distance to avoid arc and short circuits as well as carbon build-up, all of which can create unwanted conductive paths. According to the IEC60664-1 standard, clearance is the shortest distance in air between two conductive parts; creepage is also the shortest distance between two conductive parts, but along the surface of a solid insulating material as shown in Figure 5.6.

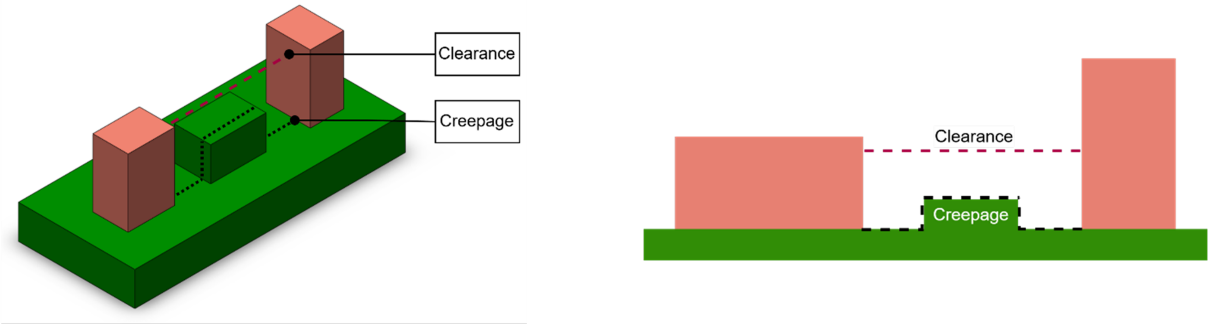


Figure 5.6: Visualization Creepage and Clearance

The process of determining creepage and clearance is illustrated in the flow chart of Figure 5.7, following the instructions from Texas Instrument's Design Seminar note in [85].

To begin, the pollution degree, nominal pack voltage, overvoltage category, and material group were defined. The pollution degree was set to 3, the highest relevant for creepage distance determination. The overvoltage category was set to III, and the material group to II. According to Table F.4 of IEC 60664-1, the minimum creepage distance is 5.6 mm. For clearance, the rated impulse voltage is taken from Table F.1. Since 400 V is not listed, 600 V with overvoltage category III was selected, corresponding to 6.0 kV. From Table F.2, the minimum clearance distance is 5.5 mm for an inhomogeneous field. Minimum creepage and clearance distance are shown in table 5.7.

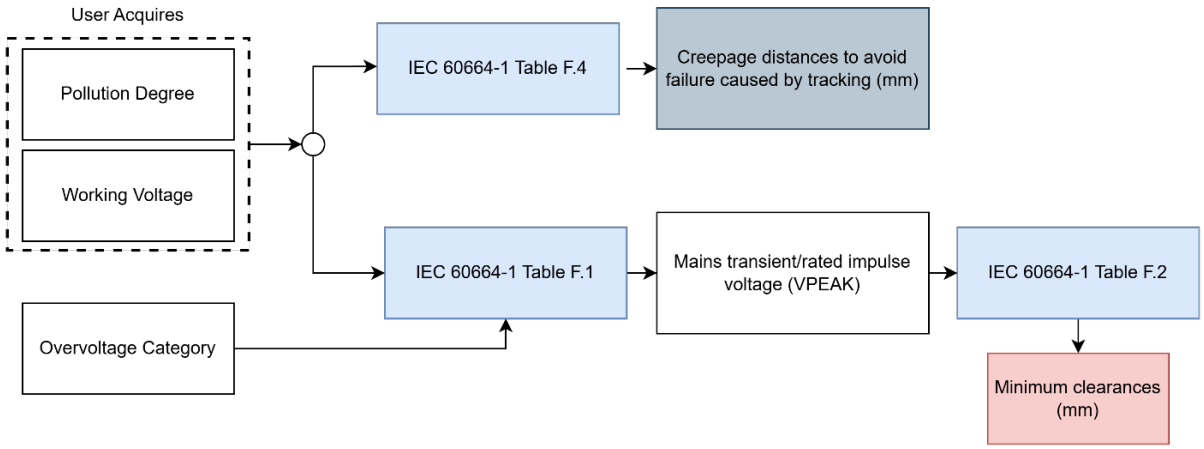


Figure 5.7: Simplified Flow to Determine Creepage and Clearance

Table 5.7: Parameter for Creepage and Clearance Distances

Parameters	Value
Pollution Degree	3
Material Group	II
Working Voltage	400 V
Overvoltage Category	III
Creepage	$\geq$ 5.6 mm
Clearance	$\geq$ 5.5 mm

5.3 Integration

5.3.1 Physical Integration

The BDU was designed to be near the high voltage outlet of the battery pack. The size of the battery was optimized so that it could fit inside the battery tray with other components. The actual location is not discussed in this thesis.

In the current design, the BDU was mounted in 3D-printed casing. Because the Breaktors were heavy, the design needed to be more secure and robust to withstand the vibration conditions of electric vehicles.

For serviceability, fuses were connected directly to the busbar. Additionally, low-voltage

(LV) and high-voltage (HV) cables were separated to minimize electromagnetic interference. LV wiring was routed at the back of the BDU and HV wiring was routed in the front.

5.3.2 Electrical Integration

The high voltage schematic for the designed BDU is shown in figure 5.8.

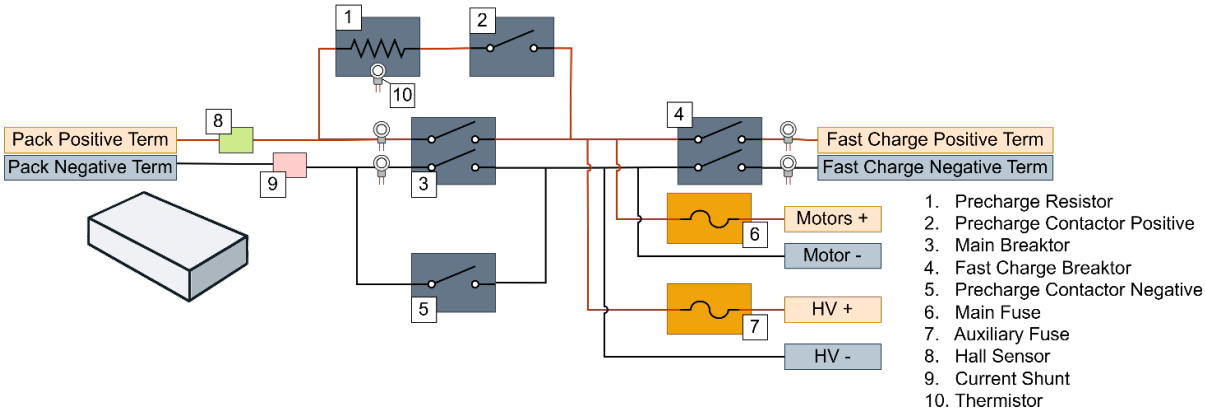


Figure 5.8: High Voltage Schematic for BDU

For low voltage, high side drive pins were used to control the relays, and analog pin to read values from (8) Hall sensor, (9) current shunt, and (10) thermistors.

5.3.3 Design Challenge and Solutions

This BDU design was designed for Battery Workforce Challenge (BWC) competition, in which, short design time and lack of adequate resources were significant challenges. To address these issues, cardboard prototypes were used along with SolidWorks modeling. The interleaving process allowed to produce six iterations in a short period of time. The research and development steps are shown in Figure 5.9.

The HV schematic was drawn first. Components were then selected based on the working voltage, current, and the capability of the BMS (e.g., number of I/O pins). In Step 4 of Figure 5.9, mockup parts were used to determine the optimal placement of each component. In this step, the exact dimension was not used. After having a clear layout,

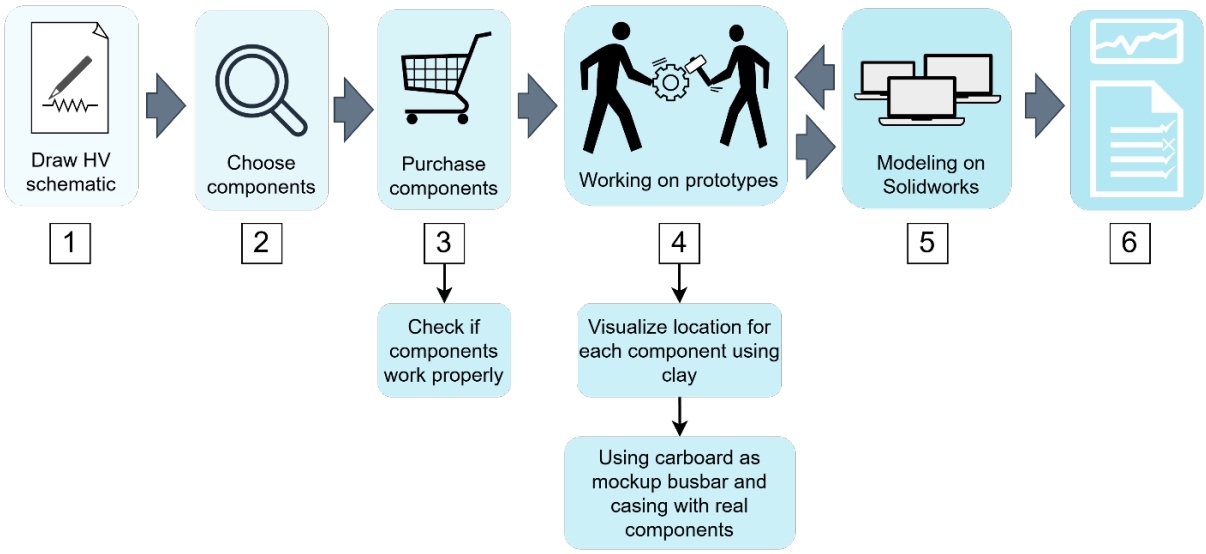


Figure 5.9: Development Workflow for BDU

prototypes were built using actual components or cardboard with 1:1 scale that matched the final design. Once the configuration met the requirements, a model was created in SolidWorks.

Figure 5.10 shows the iterations completed during the design process.

5.4 Validation

A continuity test was performed to ensure there was no short circuit between low-voltage and high-voltage wires. The creepage and clearance distances among components were measured and validated. Additional resistance among different components were documented. After these preliminary checks, both Breaktors were closed for DC fast charging (DCFC) testing. A current of 300 A was applied to the DCFC busbar for a specific duration. Temperatures at various locations were monitored using thermistors. The locations of thermistors and the corresponding results were presented in Figure 5.11.

The data showed that temperatures near the DCFC terminals were lower compared to the battery terminals, likely due to the difference in busbar size. In Figure 5.11 near point (1), the busbar was narrowed to fit a Hall sensor; near point (2), it was bent to fit a current sensor. Overall, thermistors' readings stayed below 33 °C for both Breaktors. The

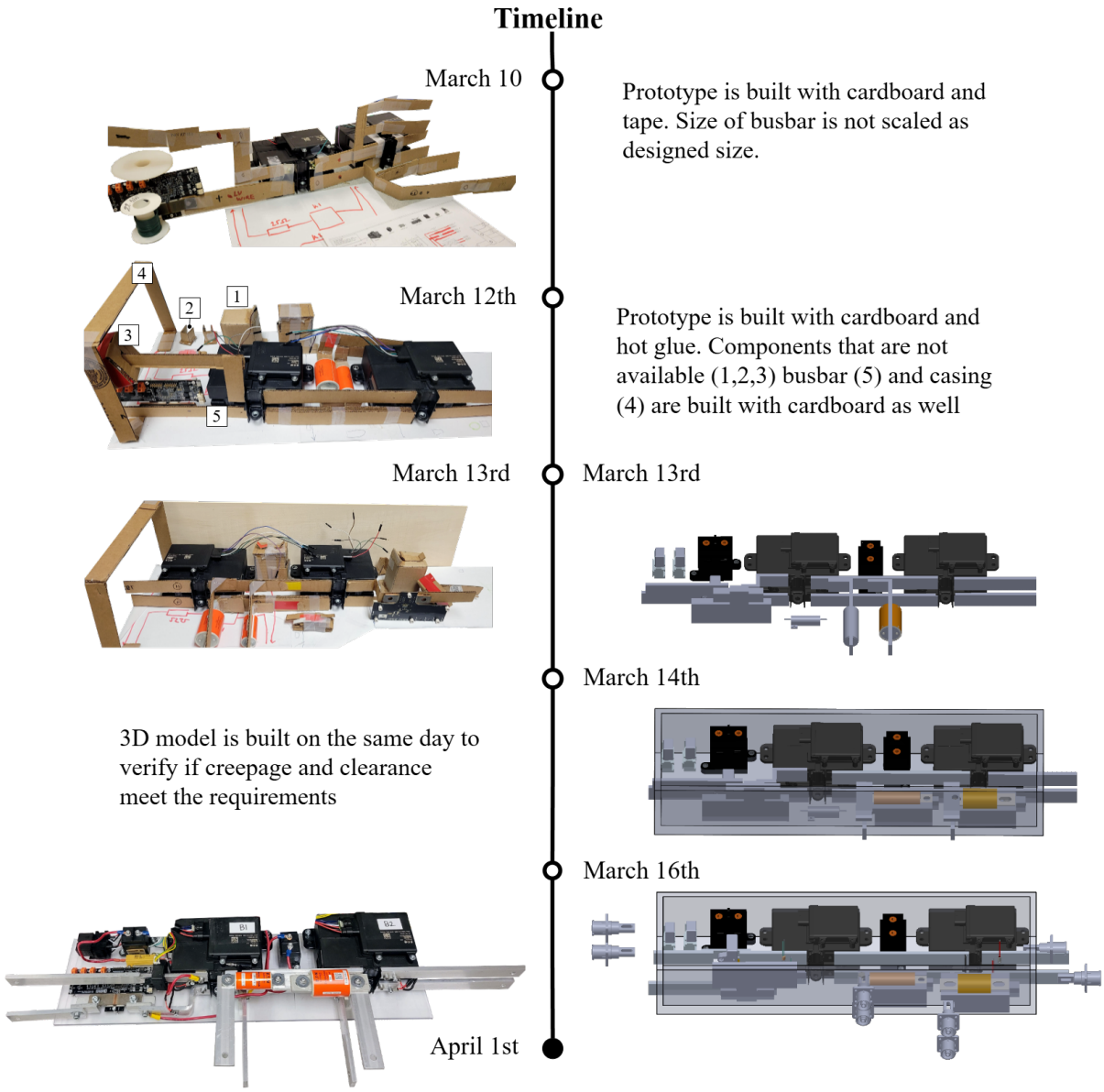


Figure 5.10: BDU Design Timeline With Different Iterations

precharge circuit temperature remained constant since the current did not run through it. Thermistors' temperatures are shown in Figure 5.12.

It was noticed that the temperature of non-economizer relays was significantly higher

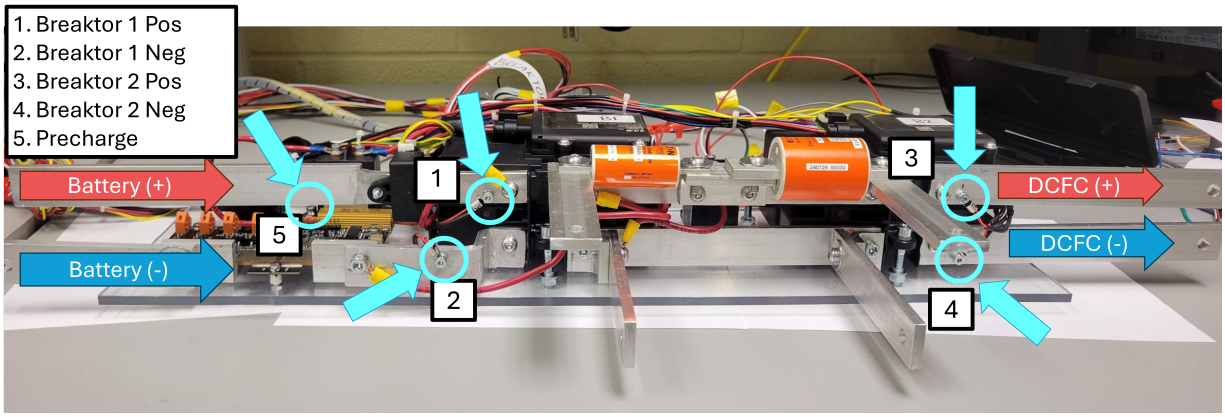


Figure 5.11: Locations of Thermistors and Flow of Current

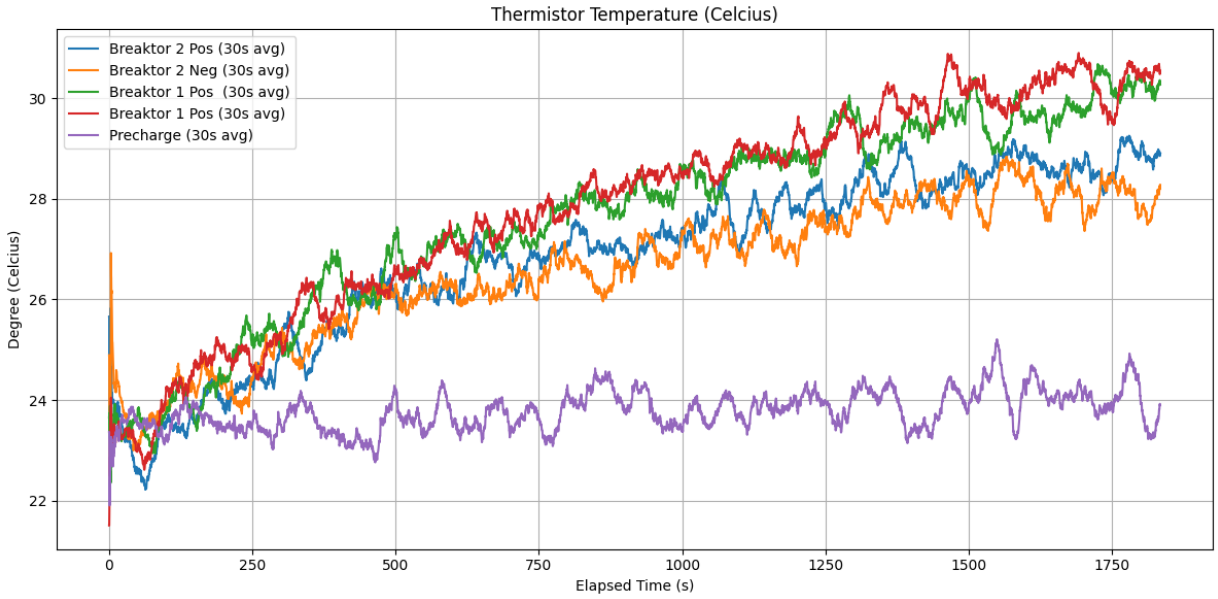


Figure 5.12: Temperature at Different Locations During Testing

compared to the rest of the system as shown in Figure 5.13. These relays were used to cut off voltage from the power supply to the Breaktors to prevent energy leakage.

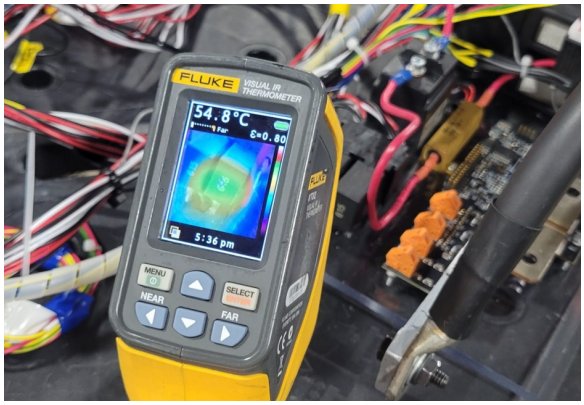


Figure 5.13: Elevated Temperature of Relay without Economizer in the BDU

5.5 Result and Discussion

The BDU met most of the creepage and clearance requirements and passed the DC fast charging test with an average temperature below 33 °C. The design and building process took less than 2 months, showing the effectiveness of the simple building process mentioned above. The key was to build physical prototypes in parallel with modeling. This process is also used in designing Formula 1 cars [89]. The documentation of building this BDU made a streamlined process for other researchers or engineers to improve the current state of BDU design and manufacturing.

Although the BDU has passed preliminary and DC fast charging tests, improvements was needed for long-term reliability. Thermal interface material (TIM) can be added to reduce heat [39]. As previously discussed, aluminum busbar can be replaced with copper to improve conductivity and avoid galvanic corrosion. Additionally, non-economizer relays need to be swapped with economizer types to reduce heat generated.

Chapter 6

Conclusion

In this thesis, State of Charge, Battery Disconnect Unit, and Hardware-in-the-Loop for battery pack development in electric vehicles were presented. State of Charge (SoC) prediction, achieved using both a machine learning approach based on the Extreme Learning Machine (ELM) and a SoC-OCV-T lookup table. Furthermore, differential features improved the SoC prediction in noisy environment and under low-temperature conditions. As a result, the RMSE and MAE were reduced to the low levels of 0.0113 and 0.0095 respectively. Hardware-in-the-loop (HIL) test bench, developed to validate algorithms such as SoC estimation and to verify the functionality of sensors, Controller Area Network (CAN) messages, and cell monitoring within the Battery Management System (BMS). Battery Disconnect Unit (BDU), designed to control the connection between the high-voltage battery and the load.

For SoC estimation, the procedure for data cycling and the implementation of an adaptive Coulomb counting algorithm, which integrated both the machine learning method and the SoC-OCV-T lookup table, were detailed. The algorithm was implemented in Simulink and achieved a maximum absolute error ranging from 0.011 to 0.030. Its deployment on a rapid-prototype BMS (M450) was also outlined.

Differential features such as $d(V * I)/dT$ computed over a time window significantly improved the SoC prediction under noisy and extreme temperature conditions. At $-20\,^{\circ}\mathrm{C}$, RMSE and MAE were 0.0113 and 0.0095, respectively.

The hardware-in-the-loop test bench was constructed using low-voltage components with a user-centric approach. This design enables rapid modification and expansion for future testing requirements.

In the BDU section, the design process for the Battery Disconnect Unit was presented, beginning with the determination of cell count. The selection of materials and components was discussed, and the risk of galvanic corrosion was highlighted when different materials were used within a single prototype. A schematic for constructing a BDU was provided, along with a simplified chart for determining the minimum distance required for creepage and clearance.

The methodologies and results presented in this thesis are intended to assist the Battery Workforce Challenge team and other researchers in starting battery development and in gaining a fundamental understanding of SoC estimation algorithm design, hardware validation, and BDU construction.

Future Work Current SoC prediction has limitations, only evaluated on an actual cell instead of its virtual digital twin. To validate on a real cell, a dump load can be setup to draw current from the cell. As shown in Figure 3.8, cell supervision circuit (ADBMS6830B evaluation board) has 16 channels. Therefore, it can measure up to 16 parallel blocks in a battery module. Besides, general purpose input output pins (GPIOs) from ADBMS6830B evaluation board can be used to monitor temperatures of cell blocks.

To expand to module and pack level, the simplest method is to scale up the ELM model blocks. According to Table 5.5, 110 ELM model blocks will be needed for a proposed battery pack. Since parameters for ELM models are saved as constant in C++, code space's size will not be affected. However, inferring 110 models in real time will greatly affect the CPU and RAM.

For the battery disconnect unit, the busbar material should be changed to avoid galvanic corrosion. BDU contains fuses that may need to be replaced; therefore, serviceable should be one of the main targets when designing a BDU.

Further research should address the implementation of these algorithms at module and pack levels, long-term reliability testing of BDU components, and the integration of additional machine learning techniques to improve SoC and state of health (SoH) prediction. Improvements in hardware-in-the-loop platforms and advanced diagnostic methods for battery systems are also important areas for continued study.

References

[1] Ahmed S. Abdelrahman, Khalil S. Algarny, and Mohamed Z. Youssef. A Novel Platform for Powertrain Modeling of Electric Cars With Experimental Validation Using Real-Time Hardware in the Loop (HIL): A Case Study of GM Second Generation Chevrolet Volt. *IEEE Transactions on Power Electronics*, 33(11):9762–9771, November 2018.

[2] Analog Devices. Wireless Battery Management Systems (wBMS) | Analog Devices. https://www.analog.com/en/product-category/wireless-battery-management-systems.html#category-detail.

[3] Davide Andrea. *Lithium-Ion Batteries and Applications: A Practical and Comprehensive Guide to Lithium-Ion Batteries and Arrays, from Toys to Towns*. Artech House, Boston, MA, 2020.

[4] J. F. B. Araújo, H. M. Pedrosa, M. C. B. P. Rodrigues, and P. G. Barbosa. Real-time "hardware-in-the-loop" simulation of components of an electric vehicle powertrain: Modeling and implementation. In *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1–7, November 2016.

[5] Asm material data sheet. https://asm.matweb.com/search/specificmaterial.asp?bassnum=ma6061t6.

[6] Thomas P. Barrera, editor. *Spacecraft Lithium-Ion Battery Power Systems*. Wiley-IEEE Press, Hoboken, NJ, USA, first edition edition, 2023.

[7] Manuel Bataller-Mompeán, José Miguel Martínez-Villena, Alfredo Rosado-Muñoz, José V Francés-Víllora, Juan F Guerrero-Martínez, Marek Wegrzyn, and Marian Adamski. Support tool for the combined software/hardware design of on-chip elm training for slff neural networks. *IEEE Transactions on Industrial Informatics*, 12(3):1114–1123, 2016.

[8] Magnus Bratt Svensson and Anna Engelbrektsson. An Optimised Packaging Solution of a Battery Disconnect Unit. Master's thesis, Chalmers University of Technology, 2017.

[9] Zhi Cao, Wei Gao, Yuhong Fu, and Chris Mi. Wireless Battery Management Systems: Innovations, Challenges, and Future Perspectives. *Energies*, 17(13):3277, January 2024. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute.

[10] Ephrem Chemali, Phillip J Kollmeyer, Matthias Preindl, Ryan Ahmed, and Ali Emadi. Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries. *IEEE Transactions on Industrial Electronics*, 65(8):6730–6739, 2017.

[11] Junxiong Chen, Xiong Feng, Lin Jiang, and Qiao Zhu. State of charge estimation of lithium-ion battery using denoising autoencoder and gated recurrent unit recurrent neural network. *Energy*, 227:120451, July 2021.

[12] Junxiong Chen, Yu Zhang, Ji Wu, Weisong Cheng, and Qiao Zhu. SOC estimation for lithium-ion battery using the LSTM-RNN with extended input and constrained output. *Energy*, 262:125375, January 2023.

[13] Mengyuan Chen, Hongyan Ma, and Wei He. The state of charge estimation of lithium-ion battery pack based on pso-kelm. In *2024 36th Chinese Control and Decision Conference (CCDC)*, pages 1671–1675. IEEE, 2024.

[14] Xiaowu Chen and Zhen Liu. Wiener process and extreme learning machine swarm based transfer learning for remaining useful life prediction of lithium battery. In *2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI)*, pages 1–10. IEEE, 2023.

[15] Cheng Siong Chin and Zuchang Gao. State-of-charge estimation of battery pack under varying ambient temperature using an adaptive sequential extreme learning machine. *Energies*, 11(4):711, 2018.

[16] Martin Coleman, Chi Kwan Lee, Chunbo Zhu, and William Gerard Hurley. State-of-charge determination from emf voltage estimation: Using impedance, terminal voltage, and current for lead-acid and lithium-ion batteries. *IEEE Transactions on industrial electronics*, 54(5):2550–2557, 2007.

[17] Zhihan Cui, Boyu Huang, Haowen Dou, Guanru Tan, Shiqiang Zheng, and Teng Zhou. GSA-ELM: A hybrid learning model for short-term traffic flow forecasting. *IET Intelligent Transport Systems*, 16(1):41–52, 2022.

[18] Frank Czerwinski. Thermal Stability of Aluminum Alloys. *Materials*, 13(15):3441, January 2020.

[19] Dana OpenECU. M450. https://openecu.com/product/m450/.

[20] Stephen C. Dexter. Galvanic Corrosion. Technical report, University of Delaware Sea Grant College Program, November 1999.

[21] Domenico Di Domenico, Giovanni Fiengo, and Anna Stefanopoulou. Lithium-ion battery state of charge estimation with a kalman filter based on a electrochemical model. In *2008 IEEE International Conference on Control Applications*, pages 702–707. Ieee, 2008.

[22] Domenico Di Domenico, Giovanni Fiengo, and Anna Stefanopoulou. Lithium-ion battery state of charge estimation with a Kalman Filter based on a electrochemical model. In *2008 IEEE International Conference on Control Applications*, pages 702–707, September 2008.

[23] David Dong. High Voltage Contactor Control. https://www.ti.com/lit/ml/slyp693/slyp693.pdf.

[24] Jiaming Dou, Hongyan Ma, Yingda Zhang, Shuai Wang, Yongxue Ye, Shengyan Li, and Lujin Hu. Extreme learning machine model for state-of-charge estimation of lithium-ion battery using salp swarm algorithm. *Journal of Energy Storage*, 52:104996, 2022.

[25] Jiani Du, Zhitao Liu, and Youyi Wang. State of charge estimation for li-ion battery based on model from extreme learning machine. *Control Engineering Practice*, 26:11–19, 2014.

[26] Eaton. Bussmannseries Hybrid and Electric Vehicle application guide. https://www.eaton.com/content/dam/eaton/products/electrical-circuit-protection/bussmann-iec-high-speed-semi-conductors-fuses/bussmann-iec-electric-vehicles-high-speed-fuses/eaton-bussmann-series-electric-vehicle-application-guide-br132016en-en-gb.pdf. see page 8.

[27] Jose V Frances-Villora, Alfredo Rosado-Muñoz, Manuel Bataller-Mompean, Juan Barrios-Aviles, and Juan F Guerrero-Martinez. Moving learning machine towards fast real-time applications: A high-speed fpga-based implementation of the os-elm training algorithm. *Electronics*, 7(11):308, 2018.

[28] Dongdong Ge, Guiyang Jin, Jianqiang Wang, and Zhendong Zhang. A novel data-driven iba-elm model for soh/soc estimation of lithium-ion batteries. *Energy*, 305:132395, 2024.

[29] Alexey Grigorev. *Machine learning bookcamp: build a portfolio of real-life projects*. Simon and Schuster, 2021.

[30] Tianyu Gu and Dongqing Wang. A beetle antennae search optimized recurrent extreme learning machine for battery state of charge estimation. *International Journal of Energy Research*, 46(13):19190–19205, 2022.

[31] Xinyu Gu, K. W. See, Yanbin Liu, Bilal Arshad, Liang Zhao, and Yunpeng Wang. A time-series Wasserstein GAN method for state-of-charge estimation of lithium-ion batteries. *Journal of Power Sources*, 581:233472, October 2023.

[32] Siwei Guan, Zhiwei He, Shenhui Ma, Zhekang Dong, and Mingyu Gao. Early life prediction of fast-charging battery based on feature engineering and state space models. *Journal of Energy Storage*, 127:116969, 2025.

[33] Yin Hang and Hussameddine Kabban. Thermal management in mobile devices: Challenges and solutions. In *2015 31st Thermal Measurement, Modeling & Management Symposium (SEMI-THERM)*, pages 46–49, March 2015.

[34] Hongwen He, Rui Xiong, Xiaowei Zhang, Fengchun Sun, and JinXin Fan. State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model. *IEEE Transactions on Vehicular Technology*, 60(4):1461–1469, May 2011.

[35] Molla S. Hossain Lipu, Mahammad A. Hannan, Aini Hussain, Mohamad H. Saad, Afida Ayob, and Mohammad Nasir Uddin. Extreme Learning Machine Model for State-of-Charge Estimation of Lithium-Ion Battery Using Gravitational Search Algorithm. *IEEE Transactions on Industry Applications*, 55(4):4225–4234, July 2019.

[36] David Howey. Davidhowey/SLIDE: Linux-compatible initial release. Zenodo, May 2019.

[37] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[38] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[39] Sumitomo Electric Industries. Connection Parts in BEV Battery Packs. Technical Report 98, Industries, Sumitomo Electric, 2024.

[40] Yong-Min Jeong, Yong-Ki Cho, Jung-Hoon Ahn, Seung-Hee Ryu, and Byoung-Kuk Lee. Enhanced Coulomb counting method with adaptive SOC reset time for estimating OCV. In *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1313–1318, September 2014.

[41] Fusheng Jiang, Yi Ren, Ting Tang, Zeyu Wu, Quan Xia, Cheng Qian, Bo Sun, and Dezhen Yang. A physics-enhanced online joint estimation method for SOH and SOC of lithium-ion batteries in eVTOL aircraft applications. *Journal of Energy Storage*, 112:115567, March 2025.

[42] Christian Kastner. *Machine learning in production: from models to products*. MIT Press, 2025.

[43] Philip Kollmeyer, Carlos Vidal, Mina Naguib, and Michael Skells. LG 18650HG2 Li-ion Battery Data and Example Deep Neural Network xEV SOC Estimator Script. Mendeley Data, Dataset, March 2020.

[44] R. Ranjith Kumar, C. Bharatiraja, K. Udhayakumar, S. Devakirubakaran, K. Sathiya Sekar, and Lucian Mihet-Popa. Advances in Batteries, Battery Modeling, Battery Management System, Battery Thermal Management, SOC, SOH, and Charge/Discharge Characteristics in EV Applications. *IEEE Access*, 11:105761–105809, 2023.

[45] Mike Lau and Kevin Calzada. Breaktor™ battery disconnect unit. In Michael Bargende, Hans-Christian Reuss, and Andreas Wagner, editors, *22. Internationales Stuttgarter Symposium*, pages 199–206, Wiesbaden, 2022. Springer Fachmedien Wiesbaden.

[46] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[47] Jaemoon Lee, Oanyong Nam, and B. H. Cho. Li-ion battery SOC estimation method based on the reduced order extended Kalman filtering. *Journal of Power Sources*, 174(1):9–15, November 2007.

[48] LG. LG Innotek Starts to Occupy the Wireless BMS Market Drastically. https://www.lginnotek.com/news/pressView.do?idx=6410, February 2023.

[49] Liang Li, Guoren Wang, Gang Wu, and Qi Zhang. An experimental evaluation of extreme learning machines on several hardware devices. *Neural Computing and Applications*, 32(18):14385–14397, 2020.

[50] Shuangqi Li, Hongwen He, and Jianwei Li. Big data driven lithium-ion battery modeling method based on sdae-elm algorithm and data pre-processing technology. *Applied energy*, 242:1259–1273, 2019.

[51] Weihan Li, Monika Rentemeister, Julia Badeda, Dominik Jöst, Dominik Schulte, and Dirk Uwe Sauer. Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation. *Journal of energy storage*, 30:101557, 2020.

[52] Xiaoyu Li, Jianhua Xu, Xuejing Ding, and Hongqiang Lyu. State of charge estimation for batteries based on common feature extraction and transfer learning. *Batteries*, 9(5):266, 2023.

[53] Xu Li, Huilin Yu, Jianchun Wang, Yuhang Xia, Haotian Zheng, and Hongzheng Song. Prediction of state-of-health of lithium-ion battery based on ceemdan-sg-lstm combined model. *Materials Today Sustainability*, 28:100999, 2024.

[54] MS Hossain Lipu, MA Hannan, Aini Hussain, Shaheer Ansari, Afida Ayob, Mohamad HM Saad, and Kashem M Muttaqi. Differential search optimized random forest regression algorithm for state of charge estimation in electric vehicle batteries. In *2021 IEEE Industry Applications Society Annual Meeting (IAS)*, pages 1–8. IEEE, 2021.

[55] MS Hossain Lipu, MA Hannan, Aini Hussain, Mohamad Hanif Md Saad, Afida Ayob, and M Nasir Uddin. Extreme learning machine for soc estimation of lithium-ion battery using gravitational search algorithm. In *2018 IEEE Industry Applications Society Annual Meeting (IAS)*, pages 1–8. IEEE, 2018.

[56] Pascal Messier, Mathieu Blanchard, Félix-A LeBel, and João Pedro F. Trovão. Power-Hardware-In-The-Loop for Electric Vehicle Battery Validation. In *2024 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6, October 2024.

[57] R. Morello, F. Baronti, X. Tian, T. Chau, R. Di Rienzo, R. Roncella, B. Jeppesen, W. H. Lin, T. Ikushima, and R. Saletti. Hardware-in-the-loop simulation of FPGA-based state estimators for electric vehicle batteries. In *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pages 280–285, June 2016.

[58] Kiarash Movassagh, Arif Raihan, Balakumar Balasingam, and Krishna Pattipati. A critical look at coulomb counting approach for state of charge estimation in batteries. *Energies*, 14(14):4074, 2021.

[59] Kiarash Movassagh, Arif Raihan, Balakumar Balasingam, and Krishna Pattipati. A Critical Look at Coulomb Counting Approach for State of Charge Estimation in Batteries. *Energies*, 14(14):4074, January 2021.

[60] Brian Munari and Andrew Schneer. How to design a precharge circuit for hybrid and electric vehicle applications. *Tech. White Paper, Sensata Technol., Attleboro, MA, USA*, 2020.

[61] Kong Soon Ng, Chin-Sien Moo, Yi-Ping Chen, and Yao-Ching Hsieh. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied energy*, 86(9):1506–1511, 2009.

[62] Kong Soon Ng, Chin-Sien Moo, Yi-Ping Chen, and Yao-Ching Hsieh. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied Energy*, 86(9):1506–1511, September 2009.

[63] NXP. NXP Announces Industry-First Ultra-Wideband Wireless Battery Management System Solution. https://www.nxp.com/pages/:NW-NXP-ANNOUNCES-INDUSTRY-FIRST-ULTRA.

[64] Omron. Technical Explanation for Solid-state Relays. https://www.ia.omron.com/data_pdf/guide/18/ssr_tg_e_9_2.pdf.

[65] Yuelong Pan and Jialong Ji. Remaining useful life prediction of lithium battery based on charging ic curve and improved elm. In *2024 4th International Conference on Energy Engineering and Power Systems (EEPS)*, pages 871–875. IEEE, 2024.

[66] Saehong Park, Scott Moura, and Kyoungtae Lee. Integration of Hardware and Software for Battery Hardware-in-the-Loop Toward Battery Artificial Intelligence. *IEEE Transactions on Transportation Electrification*, 10(1):888–900, March 2024.

[67] B. Pattipati, B. Balasingam, G. V. Avvari, K. R. Pattipati, and Y. Bar-Shalom. Open circuit voltage characterization of lithium-ion batteries. *Journal of Power Sources*, 269:317–333, December 2014.

[68] Lei Pei, Rengui Lu, and Chunbo Zhu. Relaxation model of the open-circuit voltage for state-of-charge estimation in lithium-ion batteries. *IET Electrical Systems in Transportation*, 3(4):112–117, 2013.

[69] Christian Pernaci, Giovanni Muscato, and Antonio Scandurra. Test of Automotive Battery Management System Control Strategies on Hardware-in-the-Loop Systems: A Low-Voltage Approach. *SAE International Journal of Electrified Vehicles*, 10(14-10-02-0009):123–135, April 2021.

[70] Sabine Piller, Marion Perrin, and Andreas Jossen. Methods for state-of-charge determination and their applications. *Journal of power sources*, 96(1):113–120, 2001.

[71] Gregory L Plett. Extended kalman filtering for battery management systems of lipb-based hev battery packs: Part 3. state and parameter estimation. *Journal of Power sources*, 134(2):277–292, 2004.

[72] Gregory L. Plett. *Battery Management Systems, Volume II: Equivalent-Circuit Methods*. Artech House, December 2015.

[73] Yitong Ren, Jie Ding, and Tianyun Hao. Ga-elm-based adaptive kalman estimator for soc of lithium-ion batteries. *Ionics*, 28(12):5377–5388, December 2022.

[74] Yitong Ren, Jie Ding, and Tianyun Hao. Ga-elm-based adaptive kalman estimator for soc of lithium-ion batteries. *Ionics*, 28(12):5377–5388, 2022.

[75] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(46):3736–3741, 2004.

[76] Rohm. High Side and Low Side Switches | Electronics Basics | ROHM. https://www.rohm.com/electronics-basics/switch/switch_what2.

[77] Amin Safaei, QM Jonathan Wu, Thangarajah Akilan, and Yimin Yang. System-on-a-chip (soc)-based hardware acceleration for an online sequential extreme learning machine (os-elm). *IEEE transactions on computer-aided design of integrated circuits and systems*, 38(11):2127–2138, 2018.

[78] Tomokazu Sakuraba, Rémy Ouaida, Song Chen, and Thibaut Chailloux. Evaluation of Novel Hybrid Protection Based on Pyroswitch and Fuse Technologies. In *2018 International Power Electronics Conference (IPEC-Niigata 2018 -ECCE Asia)*, pages 2153–2157, May 2018.

[79] Dapai Shi, Jingyuan Zhao, Zhenghong Wang, Heng Zhao, Junbin Wang, Yubo Lian, and Andrew F. Burke. Spatial-Temporal Self-Attention Transformer Networks for Battery State of Charge Estimation. *Electronics*, 12(12):2598, January 2023.

[80] Mohd Herwan Sulaiman, Zuriani Mustaffa, Nor Farizan Zakaria, and Mohd Mawardi Saari. Using the evolutionary mating algorithm for optimizing deep learning parameters for battery state of charge estimation of electric vehicle. *Energy*, 279:128094, 2023.

[81] Shukai Sun, Huiming Zhang, Jiamin Ge, and Liang Che. State-of-health estimation for lithium-ion battery using model-based feature optimization and deep extreme learning machine. *Journal of Energy Storage*, 72:108732, 2023.

[82] Marco J. Tavernini, Benjamin A. Niemoeller, and Philip T. Krein. Real-time low-level simulation of hybrid vehicle systems for hardware-in-the-loop applications. In *2009 IEEE Vehicle Power and Propulsion Conference*, pages 890–895, September 2009.

[83] TE Connectivity. ALUMINIUM HOUSED POWER RESISTORS. https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=1773035&DocType=DS&DocLang=English. See figure Power Overload on page 3.

[84] Model S Service Manual. https://service.tesla.com/docs/ModelS/ServiceManual/en-us/GUID-219B553D-392B-4B3B-9735-A20132B03C92.html.

[85] Texas Instrument. Demystifying Clearance and Creepage Distance for High-Voltage End Equipment. https://www.ti.com/lit/ml/slup419/slup419.pdf?ts=1751430238547.

[86] Texas Instrument. Using wireless technologies to replace cables in automotive access and battery management systems. https://www.ti.com/lit/ml/slyp670/slyp670.pdf, September 2020.

[87] Jinpeng Tian, Rui Xiong, and Weixiang Shen. State-of-health estimation based on differential temperature for lithium ion batteries. *IEEE Transactions on Power Electronics*, 35(10):10363–10373, 2020.

[88] Manh-Kien Tran, Satyam Panchal, Tran Dinh Khang, Kirti Panchal, Roydon Fraser, and Michael Fowler. Concept Review of a Cloud-Based Smart Battery Management System for Lithium-Ion Batteries: Feasibility, Logistics, and Functionality. *Batteries*, 8(2):19, February 2022. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[89] David Tremayne. *The Science of Formula 1 Design: Expert Analysis of the Anatomy of the Modern Grand Prix Car*. Haynes North America, Newbury Park, Calif., 2004.

[90] OAR US EPA. Dynamometer Drive Schedules. https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules, September 2015.

[91] Rui Wang, Huizhong Lin, Jeongsub Choi, Abolfazl Hashemi, and Mengmeng Zhu. Novel differential voltage features based machine learning approach to lithium-ion batteries soh prediction at various c-rates. *Energy*, page 137651, 2025.

[92] Xiaofei Wang, Quan Sun, Liang Chen, Di Mu, and Rui Liu. Robust state of charge estimation via kmpe loss based extreme machine learning. In *2022 IEEE 5th International Conference on Electronic Information and Communication Technology (ICEICT)*, pages 707–711. IEEE, 2022.

[93] Greg Welch. An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, 1997.

[94] Meng Xiangdong, Zhang Haifeng, Li Dexin, Dong Yunchang, Zhang Jiajun, Cao Xinyu, and Li Gang. Rul prediction method for lithium-ion batteries based on the soa-elm algorithm. *Engineering Reports*, 7(3):e70073, 2025.

[95] Dong Xiao, Boyan Li, Jiawei Shan, Zelin Yan, and Jie Huang. Soc estimation of vanadium redox flow batteries based on the iscso-elm algorithm. *ACS omega*, 8(48):45708–45714, 2023.

[96] Yinjiao Xing, Wei He, Michael Pecht, and Kwok Leung Tsui. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. *Applied Energy*, 113:106–115, January 2014.

[97] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer, March 2022.

[98] Shun Yu, Claas Rosenkoetter, Robert Hoffmann, and Frank Werner. High current, high voltage dc switching. Technical report, TDK Corporation, 2022.

[99] Haoliang Zhang, Wei Tang, Woonki Na, Pyeong-Yeon Lee, and Jonghoon Kim. Implementation of generative adversarial network-cls combined with bidirectional long short-term memory for lithium-ion battery state prediction. *Journal of Energy Storage*, 31:101489, 2020.

[100] Yongzhi Zhang, Rui Xiong, Hongwen He, and Weixiang Shen. Lithium-Ion Battery Pack State of Charge and State of Energy Estimation Algorithms Using a Hardware-in-the-Loop Validation. *IEEE Transactions on Power Electronics*, 32(6):4421–4431, June 2017.

[101] Jun Zhou and Tianhong Zhang. Aero-engine thrust estimation based on ensemble of improved wavelet extreme learning machine. *Transactions of Nanjing University of Aeronautics and Astronautics*, 35(2):290–299, 2018.

Glossary

Busbar A metallic strip used as conductor in high-voltage application. 45

FMU Functional mockup unit or FMU is a black box model. In our case FMU represent digital twin of the cell being tested. The FMU is provided by AVL and Batemo via Battery Workforce Challenge. 21, 28

High Side Drive A switch in which one terminal is connected to the voltage source and the other terminal is connected to its load. In contrast, Low Side Drive is a switch with one terminal connected to the load and other terminal connects to ground [76]. 48

LA92 The LA-92 is for Class 3 Heavy-Duty vehicles that are subject to Tier 3 standards [90]. 26, 32

US06 The US06 is a high acceleration aggressive driving schedule [90]. 19