Comparative Analysis of Coulomb Counting and Extended Kalman Filter for State of Charge Estimation in Battery Management Systems

Presented by:

Christopher Chibuzor Francis

Prepared for:

Joyce B. Mwangama

September 12, 2023

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for the degree of MSc(Eng) Electrical Engineering (Coursewrk & Diss).

University of Cape Town
Cape Town, South Africa
September, 2023

Abstract

State of Charge (SOC) is simply a measure of the amount of available charge in a battery cell. It is not possible to directly measure SOC because it is a function of the stoichiometric concentration of ions in the cell, hence current and voltage measurements were used to obtain the required accurate and precise estimation. Various authors have proposed methods for estimating SOC, however most authors have presented only high level reports. In this research, a comparative investigation of the traditional Coulomb Counting (CC) method, and the state-of-the-art Extended Kalman Filter method for SOC estimation was undertaken using a model based approach, involving simulation using Simulink and Simscape. Besides a current integration model, a cell model was developed and parameterized using a Lithium based Nickel Cobalt Aluminium (NCA) oxide battery's pulse discharge test data. The Extended Kalman Filter (EKF) was implemented to estimate the SOC of the cell model and the performance of the estimation models were evaluated on the metric of RMSE, and convergence time. It was concluded that the EKF method, outperformed the CC method as a state-of-the-art SOC estimation technique, employed in battery management system (BMS) by battery developers for the EV use case.

Acknowledgments

My most profound gratitude is to God Almighty in whom I live and move and have my being for by His Mercy and Grace have I walked the path of this research journey. I am also deeply grateful to the management of the University of Cape Town for maintaining a safe and serene learning environment.

The words 'THANK YOU', limits the magnitude of my gratitude to my amiable supervisor, Dr. Joyce B. Mwangama, for her relentless academic support, encouragement, patience, understanding, kind disposition and guidance which helped me to also be patient with myself and to walk with my head high, shoulders back and chest front.

I am profusely grateful to my industry mentor, Director Paul Rogerson of BBSTelematics in South Africa, who was always there for me with unflinching moral support, astute recommendations, insightful advisement, and most helpful disposition. You are indeed a destiny helper!

My appreciation goes out to my friend, brother and senior colleague, Amakan Elisha who held me by the hand and gave me an invaluable orientation about UCT, and the Nigerian community in Cape Town. Experiencing Cape Town with you was the best and I can not imagine Cape Town without you.

I will not fail to tender my sincere gratitude to my friends; Noncedo Sokana, Faith Okeke (Mgbemere), Nora Ehujuo, Sampson Nwachukwu, Chigozie Boniface, Iyanuoluwa Oyetunji, Owen Muawra, Mzamo Mcibi, Vukosi Mboweni, and the entire

MCF scholars family in UCT, your support, companionship, encouragements and knowledge share contributed immensely to achieving this great feet.

My sincere thanks goes out to my church family in Cape Town, Gospel Ramah Church. By the sound doctrine and undiluted word of truth which I learned while with you, and the relationships I found within the local assembly of the body of Christ, I was immensely blessed.

Many thanks also to the Nigerian Students Society AKA NaihaSoc, National Association of Nigerian Students South Africa (NANSSA) and also to the Nigerian Citizens Association South Africa (NICASA). Thank you for giving me an experience of Nigeria in the diaspora.

I am deeply grateful for the technical support that I received from Javier Gazzarri via the MathWorks community, and from Prof. Gregeory Plett via LinkedIn and OptiNum Solutions. This research would not have been successful without your kind support.

My heart rings out with special appreciation to my dear wife, Mrs Rita Francis who stood by me throughout this research journey and never failing to lend me her listening ears, caring heart and encouraging words, during the highs and the lows of my research.

I wish to also appreciate the center for excellence and communication research group, for the financial, moral and academic support which I received from them.

Finally, I will not fail to appreciate my sponsors, the Mastercard Foundation Scholars Program at UCT. Thank you for giving me a spring-board into a life of sound knowledge and prosperity.

This research is dedicated to my beloved mother Late Mrs Bridget Nkemakolam Onwukwe (of blessed memory), who went to be with the Lord on the 14th December 2022.

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

3. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

4. I have the IEEE convention for citation and referencing. Each contribution to, and quotation in, this masters dissertaion report from the work(s) of other people, has been attributed and has been cited and referenced.

5. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as there own work or part thereof.

Signed by candidate

.....

September 12, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Motivation

Lithium-ion cells are increasingly gaining usefulness across major vertical markets, such as the automobile, renewable energy, and telecommunication markets. While development in battery technology continues to advance, innovative ways must be developed enhance their performance and ensure their safe operation and durability. In the electric vehicle (EV) use case, information on EV range capability is dependent on the accuracy of the state of charge (SOC) estimation. Most modern engineering systems such as wireless communication devices, renewable energy systems and electric vehicles (EVs), features the use of lithium-ion cells, which supplies energy during operation and stores energy which is available but not used at production. In an EV use case, 30% to 40% of the EV cost valuation is determined by the batteries [1]. Batteries are non-linear and dynamic in their behavior; therefore, it is extremely important to know their dynamic response which is often referred to as the 'battery personality' [2]. The knowledge of this dynamic response, is required to control the battery efficiently using control algorithms. For instance, full hybrid electric vehicles (HEVs) are built with the batteries carrying a large percentage of

the drive train load [3] and due to the stochastic cycling of the batteries by the drive train, they become prone to poor performance and premature aging if not controlled using a sophisticated battery management system (BMS). One of the requirements for reliable EV performance is that it's BMS constantly receives updates on two fundamental battery quantities which are the battery's power delivery and SOC. State of charge estimation is the measure of the available energy in an EV's battery and is an important parameter in EV range calculations [4].

1.1.1 Importance of SOC Estimation in Battery Management System (BMS)

The SOC of a cell is generally defined as the ratio of a its residual (instantaneous) charge capacity to its nominal charge capacity. SOC is an important input in the balancing, energy and power calculations [5]. SOC estimation is an important BMS functionality which serves as an enabler for ensuring the safe operation, durability, reliability, and optimal performance of the battery system. State of charge (SOC) estimation which is the BMS functionality of interest in this research, is important as it supports these important aspects of battery systems [5];

Longevity: By avoiding transient currents, overcharging and overdischarging the battery cells.

Performance: A good SOC estimator can allow aggressive use of the battery pack capacity.

Reliability: A good SOC estimator is consistent and dependable for any driving profile.

Density: Because accurate SOC estimators allows for aggressive use of the power-pack, the pack does not need to be over-engineered.

Economy: Smaller battery systems with warranty services costs less.

1.1.2 BMS in the EV Batteries Supply Chain

BMS integration with battery packs constitutes one of the multiple complex stages in the EV battery supply chain [1] and contributes to the acceleration of global supply of electric vehicles, amidst rising demand. EVs require a fast-responding battery pack with an optimally designed battery management system (BMS). A BMS is a power electronic peripheral circuit which takes current, voltage and temperature as inputs to gives useful output signals including temperature, individual cell voltage and state estimates of the battery system, in order to provide supervisory control of the battery and ensure its safe and reliable operation. In a BMS, states and parameters are continuously updated in a program loop viz; Initialize - Measure voltage, current and temperature - Estimate state of charge - Estimate state of health - Balance cells - Compute power limits - Store data [5]. With a BMS model simulation, we can explore a wide range of operational and environmental conditions that would be difficult to reproduce with hardware testing [6]. In the EV use case, the battery management system (BMS) which is also called the battery controller, must know the dynamic response of the battery along with its temperature dependencies and SOC, so as to ensure that the EV battery is operated within the safe operation area (SOA).

1.1.3 EV Batteries and the Race to Net Zero

According to the International Energy Agency (IEA) report on the road-map for reaching global net zero emission by 2050, it is interesting to note that 75% of global green house gas is caused by the energy sector [7]. However the concern for reducing global carbon dioxide (CO_2) emission, on the pathway to Net Zero, is seen to be consistent with the rapid increase in global demand for EV batteries, with China dominating the entire downstream EV battery supply chain. [1]. This is seen in figure 1.1 which illustrates the geographical distribution of the global EV supply

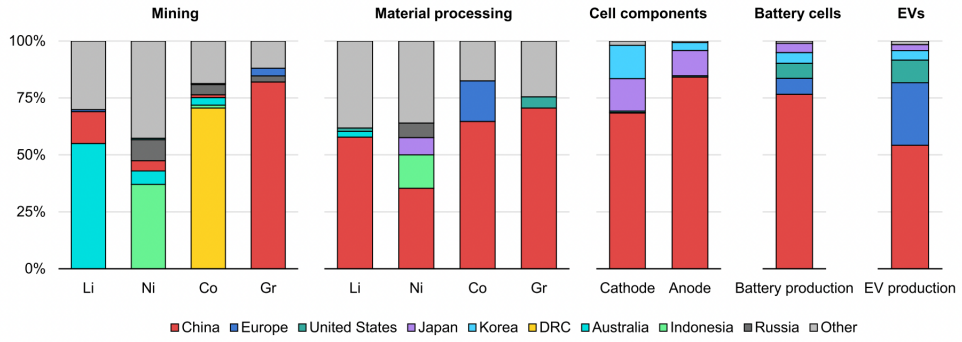

Figure 1.1: Geographical Distribution of the Global EV Supply Chain [1]

chain.

Summarily, among the factors affecting the global adoption of EVs is the fact that the EV technology, is still in its infancy and people are not used to driving EVs, and considering the intensity by which EVs are driven in terms of its range capability. These causes what is generally referred to as **'range anxiety'**. However, by optimizing the accuracy of SOC estimation in BMS on the EV supply chain, user confidence and reliability on the battery system is increased while improving the battery pack performance. For battery system engineers and EV users, accurate SOC estimation promotes acceptable durability and safe operation of the battery system which is expected to justify their corresponding high costs of procurement [8] [9]. Battery and automakers can leverage the improved SOC estimation techniques presented in this research, to drive higher market penetration for EVs while optimizing the battery performance and minimizing the concern of both range anxiety and carbon dioxide (CO_2) emission. Therefore, the techno-economic importance of SOC estimation in battery management systems, along with the ecological benefits of battery systems, contributes to the motivation for this research.

1.2 Background on Battery System

The electric battery has evolved from the ancient Parthian battery invented 2000 years ago [10], to the recent lithium ion battery used across the energy vertical. The smallest individual unit of an electrochemical energy source which converts chemical energy to electrical energy, is called a 'cell' [10]. These cells are then grouped together and electrically connected either in series, parallel or both, to form the battery.

From an electro-chemical perspective, the reaction occurring inside a cell is a diffusion (oxidation-reduction) reaction which involves the delocalization and flow of electrons, transferred from one active material to another [11]. Three major components of the cell include the anode (reducing material), the cathode (oxidizing agent) and the electrolyte which provides the necessary internal ionic conductivity. For instance, lithium-ion chemistries, electrons move from the negative (anode) electrode where oxidation occurs to the positive (cathode) where reduction occurs during discharge. This results in the depletion of electrons at the negative (anode) electrode while the positive (cathode) electrode becomes saturated with electrons until neither the positive electrode can accept electrons, nor the negative electrode can supply electrons and hence current stops flowing. Charging the battery then reverses the reaction. Having described the basic topology of a battery and its electro-chemistry, we will attempt to define some battery terminologies.

1.2.1 Preliminary Terminologies and Definitions on State of Charge Estimation

Primary Cell: Primary cells are those which cannot be recharged and are not designed to be recharged. Examples include zinc carbon (Leclanche´) and alka- line cells.

Secondary Cell: These cells can be recharged several times (usually >200). Examples include lithium - ion and lead acid cells.

Fully Charged Cell: A fully charged cell is one whose open circuit voltage (OCV) reaches $v_h(T)$, which is a temperature dependent high nominal voltage as specified by the manufacturer. For instance, the high nominal voltage specification $v_h(25°C)$ = 3.6V for a lithium iron phosphate (LFP) battery, means that the battery is fully charged at 3.6V. Cells are fully charged by first supplying a constant-current charge profile until its terminal voltage is equal to the high nominal voltage $v_h(T)$ then followed by a constant-voltage profile until the charging current becomes infinitesimal. SOC of a cell at full charge is 100%.

Fully Discharged Cell: A fully discharged cell is conversely one whose OCV reaches $v_l(T)$, which is a temperature dependent low nominal voltage as specified by the manufacturer. For instance, the low nominal voltage specification $v_l(25°C)$ = 2.0V for an LFP battery, means that the battery is fully discharged at 2.0V. Cells are fully discharged by first loading a constant-current discharge profile until its terminal voltage equals $v_l(T)$ then followed by a constant-voltage profile until the discharge current becomes infinitesimal. SOC of a fully discharged cell is 0%.

C-Rate: This is a specification of the speed at which a cell is being charged or discharged. Mathematically, it may be computed as the ratio of charge or discharge current to the cell's nominal capacity.

Nominal Voltage: The nominal voltage of a cell is the average discharge voltage which is evaluated and specified by the manufacturer. Cells may be operated above or below its nominal voltage. Nominal voltage for Lead acid cells is 2.0V while Nickel Cadmium is 1.2V and Lithium based cells are 3.0V [10].

The Cell Capacity: The capacity of the cell may be considered in the following contexts;

Total Capacity (Q): A cell's total capacity is defined as the quantity of extractable

charge from the cell as it is being discharged from 100% SOC to 0% SOC. In practice, the total capacity is typically dimensioned in ampere hours (Ah) or milliampere hours (mAh) however, the SI unit for charge is coulombs (C). The total capacity of a cell is a slowly decaying *'parameter'*, just as the cell's state of health degrades.

Discharge Capacity (Q_{rate}): A cell's discharge capacity is defined as the quantity of charge extracted from the cell while discharging with a load, at a constant rate from 100% SOC to 0% SOC. Hence the discharge capacity captures the discharge rate and is determined in terms of the cell's loaded terminal voltage. The loaded terminal voltage, is dependent on the cell's internal resistance which is a function of rate and temperature which therefore implies that the discharge capacity is rate and temperature dependent. Where i(t) is the discharge current and R_0 is internal resistance, it is noteworthy to state that the resistive drop i(t)$*$$R_0$, reduces the discharge capacity such that it is less than the total capacity except when i(t) is negligible. However the discharge capacity decays slowly just like the total capacity as the cell continues to age.

Nominal Capacity (Q_{nom}): A cell's nominal capacity is a manufacturer specified quantity that indicates the amount of charge that the cell is designed (rated) to hold. This parameter is largely influenced by the cell's anticipated application. Typically, the nominal capacity is considered to represent the discharge capacity at 1C-rate. However, while the discharge capacity is specified at the cell level for an individual cell, the nominal capacity is specified at the battery level for a lot of cells.

Residual Capacity: The quantity of extractable charge, depleted from a cell as it is discharged from it's state at a present point in time to a fully discharged state, is called the residual capacity of the cell.

NB: It is important to highlight that the cell's charge capacity is different from its energy capacity. While the charge capacity defines the maximum amount of

charge delivered by the cell in Ampere-Hours (Ah), the energy capacity defines the maximum amount of extractable energy in the cell in Watts-Hour (Wh).

Open Circuit Voltage (OCV): The OCV of a cell block is a temperature and SOC dependent output voltage, measured when the cell is approximated as an ideal DC source and the terminals are not connected to any load. The OCV may also be denoted as 'E_m'.

Terminal Voltage (V_t): The terminal voltage is a more realistic approximation of cell's output voltage, which is not only dependent on SOC and temperature, but also on the internal resistance and dynamics of the cell which better captures the instantaneous response (polarization) and dynamic response (diffusion) of the cell block.

Cell Efficiency Because cells are not perfectly efficient as charge is practically lost as a result of unwanted side reactions, an efficiency factor $\eta(t)$, is therefore defined for a cell. $\eta(t)$ is called the coulombic efficiency and is defined as the ratio of the output charge to the input charge. The coulombic efficiency $\eta(t)$ is usually modelled as $\eta(t) \leq 1$ during charging and as $\eta(t) = 1$ during discharge. Furthermore, energy is lost in a cell through resistive heating, hence the energy efficency is defined as the ratio of the output energy to the input energy. The coulombic (charge) efficiency is not to be confused with the energy efficiency. While the coulombic efficiency for a typical lithium-ion cell is about 99%, its energy efficiency is about 95%. The coulumbic efficiency is defined for the charge capacity while the energy efficiency id defined for the energy capacity which were described the preceding section.

Depth of Discharge (DOD): A cell's depth of discharge is simply defined as the inverse of the SOC. DOD = 1− SOC.

Requirements for SOC Estimation

The high-level requirements for SOC estimation are; estimation accuracy, real-time processing and minimal computational complexity [12]. As a real-time monitoring task, SOC estimation in EV application, requires a tight time window of about $3 * 10^{-9}$, to sample input data, perform mathematical computation to obtain output for updating the BMS [13]. The inputs available for the estimation of SOC include cell voltages, pack current and cell temperatures. SOC is a function of the particle average concentration, however, phenomenons such as temperature changes, cell resting, and usage history does not affect the average concentration of the electrode. Instead, SOC is only affected by the passage of either charge/discharge current or a self-discharge of the cell itself. Because SOC is dependent on change in current, hence the SOC z*(t)* of a cell can be mathematically modelled as;

$$z(t) \ = z(0) - \frac{1}{Q} \int_0^t \eta i(t)\, dt$$

.

Where cell current is positive during discharge and negative during charge., η is the cell's coulombic efficiency and Q is the cell's total capacity in ampere seconds. Using the above equation for the estimation of SOC is called coulomb counting which is one of the methods, considered in this research study. It is important to note that SOC is undefined on the pack level due to the inherent tendency of cell SOC divergence in a battery pack. However the concept of cell balancing can provide the average SOC as a reference for gauging the battery pack energy. However, 'cell balancing' falls outside the scope of this research.

1.3 Research Aim and Objectives

This research study is aimed at investigating the integrity of the proposed methods. The performance of the extended Kalman filter is critically analysed in contrast to the Coulomb counting method and compared on the metric of their accuracy, for SOC estimation in BMS as applied in the EV use case. Accuracy is simply an indication of how close a measurement is to the true value.
In view of the above aim, the objectives of this research study are as follows;

- To describe the state of the art on the methods for cell SOC estimation from literature.
- To simulate a dynamic equivalent circuit model of a 2.9Ah Panasonic 18650PF NMC cell with realistic response to voltage and current inputs for parameterization and SOC estimation.
- To characterize the equivalent circuit model by parameterization.
- To implement the SOC estimation using Coulomb counting and Kalman filtering.
- To critically analyse the accuracy of the proposed estimation methods as measured by the estimation error percentage.
- To propose the optimal method for the EV usage scenario.

The two state-of-the-art SOC estimation methods are therefore compared, to ascertain which scales well for the EV use case. The battery of interest is the lithium based nickel cobalt alluminium (NCA) battery. This is due to its wide range of application, high energy and power density, high efficiency and cycling capability, low self-discharge, and less maintenance [10].

1.4 Problem Statement

SOC is analogous to the dashboard fuel gauge which measures petrol level. While there exists sensors to measure petrol levels directly in internal combustion engine vehicels (ICEVs) tanks, there are presently no sensors to measure SOC in battery electric vehicles (BEVs). Because SOC is dependent on the average stoichiometric concentration in the electrode, there is presently no direct way to measure the concentration stoichiometry which will allow for the SOC calculation. Therefore SOC is a hidden state which must be estimated using measurements of cell terminal voltage and cell current. Although cell SOC is closely related to OCV, however, the terminal voltage is a poor predictor of OCV unless the cell is in electrochemical equilibrium, which is not practical for most online battery use cases as cells must be disconnected and allowed for period of time to reach electrochemical equilibrium.

A major challenge in the management of lithium ion batteries including lithium nickel cobalt aluminium oxide (NCA), lithium nickel manganese cobalt oxide (NMC) and lithium iron phosphate (LFP), is to obtain accurate estimation of the SOC in the presence of stochastic charge-discharge cycles and across varying temperature conditions.

1.5 Hypothesis and Key Questions

SOC is a hidden variable of a nonlinear dynamic cell, hence it must be estimated accurately. Because SOC is a hidden variable with a nonlinear continuous value relationship with battery dependencies such as open circuit voltage (OCV), charge/discharge current *(I)* and ambient temperature *(T)*, it is difficult to estimate directly [14] especially at run-time or even when the battery terminal voltage is assumed to approximate the value of the OCV after a long period of rest, which

however is impractical in real-time scenarios [15]. This research maintains that the conventional methods for SOC estimation are not suitable for online SOC estimation required in the EV use case. Hypothetically, it posits that adaptive methods offer a more accurate SOC estimation by investigating and comparing both Coulomb counting and Extended Kalman filter methods. It further proposes the method which outperforms the other. These techniques will involve modelling and parameterization of a cell. Data is adopted from a controlled pulse discharge experiment on an NCA battery cell. Performance of the methods implemented are measured based on the root mean square error (RMSE) and convergence time. The models are validated, using drive cycle data.

In an attempt to prove the conceived hypothesis, some key questions to be investigated are viz;

- How can we know the true SOC to evaluate our estimators?
- How can experimental data be acquired for SOC estimation?
- What computing capabilities and tools are required for performing an SOC estimation task?
- What performance metrics are relevant for validating SOC estimation accuracy?
- How does the Coulomb Counting estimation results compare to the Extended Kalman filter results?

1.6 Scope of Research

Battery management is a broad subject as there are many management functions performed by a BMS as highlighted earlier. Hence, it is important to establish

that the scope of this research is confined around two state of the art SOC estimation techniques employed in BMS. The methods understudied thereof, are based on system identification which focuses on the use of a model and data for analysis of the battery system. In this research, our battery chemistry of preference is the lithium-ion chemistry. This is due to their wide range of applications, high energy and power density, high efficiency and cycling capability, low self-discharge and low maintenance. However they require careful management and are highly temperature dependent.

1.7 Thesis Contributions

The main contributions of this dissertation are as follows:

- A table based equivalent circuit battery model which takes advantage of experimental measurements including voltage, current and temperature.
- Proposal of optimal state of the art methods for SOC estimation.
- Minimization of SOC estimation error related to the dependence on the traditional ***voltage method*** for SOC estimation.
- Addition of clarity to the knowledge economy around the problem of SOC estimation.
- Proposes a practical means for battery system developers and engineers, to investigate and identify SOC.

1.8 Publications

The following paper have been written, presented and published out of this research study. [16]

- C. Chibuzor Francis and J. B. Mwangama, "Accurate State of Charge Estimation for Lithium Iron Phosphate Battery Cell Using Equivalent Circuit Model, Parameter Tuning and Unscented Kalman Filter," *2022 5th International Conference on Energy, Electrical and Power Engineering (CEEPE)*, 2022, pp. 199-204, doi: 10.1109/CEEPE55110.2022.9783371.

1.9 Plan of Development

Sequel to this introductory chapter, further parts of this thesis is structured thus:

Chapter 2, Literature Review: Theoretical Background and Review of Related Works

This chapter reviews the existing literature and related works done by various authors and researchers on the subject area, to provide substantial intuition on the subject of interest and appreciate the research gap.

Chapter 3, Methodology: Description of Research Implementation.

Chapter 3 provides a high-level description of the of the main steps in this research study. Detail is provided on the research data, and the motivation is given for the chosen methods. The SOC estimation workflow for both methods are describe with an illustration of the implementation block diagram. Battery modelling tools and components are also discussed.

Chapter 4, Design: Implementation of the Research Design

This chapter presents a detailed description of the equivalent circuit model implementation, characterization and SOC estimation, featuring system block interconnection for Coulomb counting and Extended Kalman filter methods. The methods are also validated.

Chapter 5, Results: Comparative Analysis of Results

In this chapter the research results are elaborately discussed, and an analysis of the performance is investigated.

Chapter 6, Conclusions: Conclusion and Future Work

This chapter concludes the current study by emphasizing on the major aspects and key lessons learned through the research study/journey. A proposition of the optimal SOC estimation method is put forward along with future works that aligns with the current study.

Chapter 2

Review on SOC Estimation in BMS

2.1 Chapter Introduction

This chapter presents an in-depth discussion on various theoretical concepts relating to BMS and SOC estimation methods, and reviews various relevant related works on SOC estimation methods. The chapter starts by briefly reviewing the BMS in terms of its purpose and design. The review then progresses to the main research focus which is the SOC estimation methods. A critical review of related works is then synthesized in terms of their approach, strength, weakness, application and contribution to knowledge, while highlighting the gaps in the literature.

2.2 Battery Management System

The Institute of Electrical and Electronics Engineering (IEEE) standard 1491, defines a BMS as an embedded system which is installed permanently with the battery to measure the battery parameters and conditions. This means that the

BMS must measure, store and report the battery operating parameters, as well as deploy supervisory control. The BMS design complexity is dependent on the built-in processing power and application, which accounts for the cost price and features of the BMS [17]. As earlier established, the purpose of the BMS is to protect the battery's individual cells for safe operation, optimal performance and acceptable durability of the battery pack. The design goals of a BMS is to develop feedback and supervisory control in the aspects of;

- Monitoring cell control parameters.
- Battery state estimation
- Power limitation for thermal and overcharge protection.
- Charge profile controls
- Cell SOC balancing
- Pack isolation

2.2.1 BMS Sensing and Design

The BMS achieves its design objectives by measuring control parameters including voltage, temperature and the battery current with the aid of appropriate sensors. From a design perspective, the BMS senses the cell voltages using analog digital converters (ADC) which are electronic chipsets produced by various vendors whose chip topology influence the measurement accuracy and performance. These converters are based on various topologies such as the successive approximation register (SAR) converter shown in figure 2.1, which is used for sequential measurement. There is also the sigma-delta converter which may be deployed for permanent monitoring of cell voltages [17].

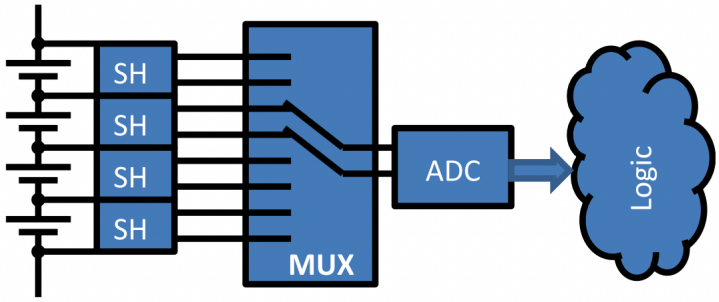

Figure 2.1: SAR Converter Topology [17]

Excessive heat and cold temperature affects batteries adversely and usually results in reduced battery life. Hence, temperature regulation is an essential function of every BMS. The measurement of temperature input required by the BMS for proper thermal management, is achieved using sensors strategically placed on one or more cells. These sensors must produce a voltage signal that indicates the temperature of the battery. The thermocouple and thermistor are examples of such sensors. The battery current measurements required as input to the BMS for SOC estimation and supervisory control, may be obtained using a shunt resistors or Hall effect sensors [4]. Because cells have varying coulombic efficiencies, they are seen to possess an inherent tendency to degrade at different rates. This diminishes the performance of the battery pack by restricting maximum charge delivery and acceptance capabilities of the pack. The BMS deals with this problem by scheduling and implementing cell balancing tasks. For coupling and decoupling the battery pack from loads as a safety requirement, the BMS uses current controlled relays (contactors) to achieve electrical isolation. Limiting power delivery and acceptance for thermal and overcharge protection is achieved by computing the voltage, current, power and energy limits for the battery pack. This design limits, defines the safe and optimal operation area for the battery system which must not be violated.

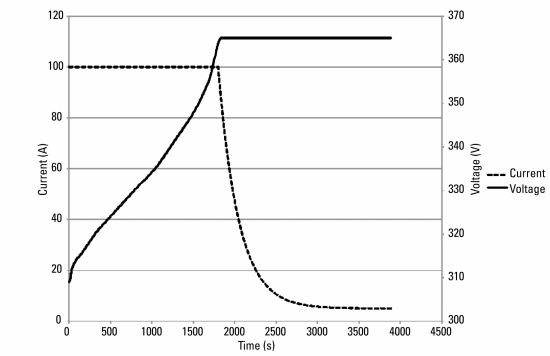

Figure 2.2: CC-CV Charging Profile [10]

BMS algorithms such as the charging logic, fault monitoring logic and the supervisory control logic, are designed for the different BMS functions, using stateflow, control system toolbox and Simulink Test toolboxes. These algorithms are combined to form a whole BMS unit.

Two design schemes implemented for the BMS are the centralized and the modular designs. In the centralized BMS design, a single control unit is used to measure, estimate and forecast the operation conditions of the battery over time. On the other hand, the modular design, utilizes separate control units to handle certain groups of cells within the pack. [18]

In summary, the BMS consists of two major aspects, which are the measurement monitoring (i.e Sensing) and Control. The measurement monitoring aspect further consists of the voltage limit calculations, current limit calculations, fault monitoring and SOC estimation tasks. On the other hand, the control aspect consists of the supervisory control, balancing control, charging control, pack isolation and output/display. [17]

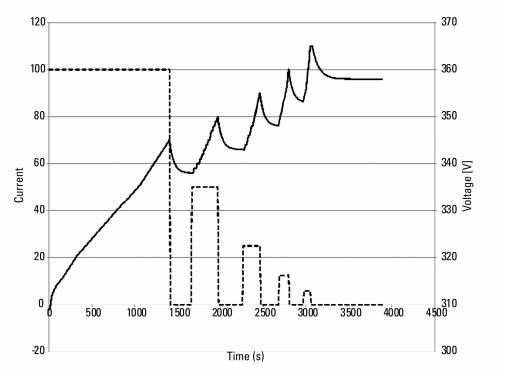

Figure 2.3: Target Voltage Charging Profile [10]

BMS Charging Strategies:

The BMS uses controlled charging profiles such as the constant current constant voltage (CC-CV), as illustrated in figure 2.2, to bring the cells to a full state of charge (i.e SOC = 100%)

CC-CV Charging: The constant-current constant-voltage charging scheme is conceived as the best BMS charging method for lithium ion batteries. However, the CC-CV charging scheme cannot be deployed on a large scale battery with many stacks of individual cells because individual cell voltage or current cannot be controlled in this charging scheme. [10]

Target Voltage Charging: This charging scheme may be deployed with large format battery packs. The target voltage is set slightly below $v_h(t)$ as the initial. Maximum charging current is applied until the highest cell voltage in the pack reaches the target. Depending on the cell configuration in the pack, the target voltage is measured at the battery terminals. The charging current is then scaled down while the target voltage is scaled up. The process is repeated until the target voltage reaches $v_h(t)$. This is illustrated in figure 2.3. [19]

Constant Current Charging: This charging scheme is typically applied in most simple battery systems, and involves charging the battery pack at a fixed current.

It offers a safe charging option for lithium ion battery when current regulation is not specified. Similar to the target voltage charging, a target voltage is established, however the current applied is completely cut off while the cells are allowed to equilibrate until the maximum cell voltage falls below a threshold and the current is applied again. This is illustrated in figure 2.4 [19]

2.2.2 Battery System and Modelling Approaches

The Battery

A battery is a device that converts the chemical energy contained in its active materials, directly into electrical energy by means of an oxidation-reduction reaction [20]. The oxidation-reduction reaction involves the production and transfer of electrons from one active material to another [21]. By juxtaposition, this transfer of electrons occurs directly in non-electrochemical reactions by exposure to heat and/or light. The basic unit of the battery is the cell as illustrated in figure 2.5. A battery pack consists of one or more cells which are connected in series or parallel depending on the desired output voltage and charge capacity. Three major components of the battery cell include the anode (reducing material), the cathode (oxidizing agent) and

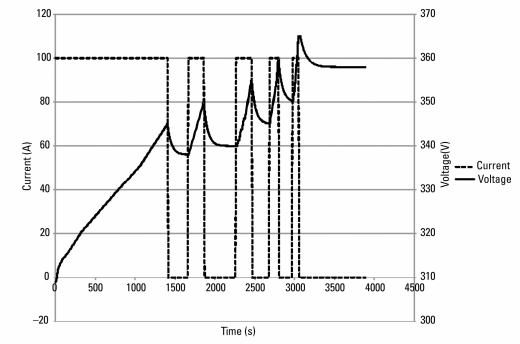

Figure 2.4: Constsnt Current Charging Profile [10]

the electrolyte which provides the necessary internal ionic conductivity. The electrolyte is usually liquid but may also be solid depending on the battery chemistry. The solid electrolytes are ionic conductors at their operating temperatures [20]. A practical, individual, electrochemical cell is formed by pairing two suitable electrodes and placing them in a suitable electrolyte with the necessary separator which gives the optimum conversion of chemically stored energy in the battery cell into electrical energy. The separator material serves to mechanically separate the anode and the cathode electrodes.

Primary batteries are used in civilian, military, and industrial applications with relative low cost, light weight and suitability for portable devices and they are not designed to be recharged. Secondary batteries are widely used in many applications including energy storage devices and are characterized by their ability to be recharged, high power density, high discharge rates, flat discharge curves and good low temperature performance. As a recent development in battery technology, lithium batteries have several advantages over other primary and secondary battery chemistries. Lithium is considered an attractive anode because of its relative light weight and high voltage (2v - 3.6v). Lithium batteries are also preferred for their high energy density, high power density, flat discharge curve and excellent service life over a wide range of temperature [20].

Battery(Cell) Modelling

Battery state of charge estimation methods often require a good and realistic battery model. A model is generally a simplified representation of the real system. Models are used to mimic the essential features of the system which are of interest, while leaving out other features which does not affect the system behavior being observed and identified. Models are beneficial to battery system developers and

engineers in the following regards;

- Helps in the design of a suitable controller for the battery system and find optimal solutions for it.
- Help to improve state estimation by combination with sensor measurement.
- Enable the performance of formal analysis in order to guarantee performance or safety.
- Useful as replacement for physical battery systems in testing by simulation.

It should be noted that in modelling a battery system for SOC estimation, focus is given to the essential features that affects the dynamics (SOC) being observed, and these features include, current, voltage and temperature. These features are the inputs and outputs and forms part of the model structure along with the other relevant battery parameters.

There are various modelling approaches used in control system design and analysis, for obtaining accurate plant models such as a battery model. Three approaches found in literature include;

First Principle (White Box) Approach: This approach requires a proper domain knowledge of the system of interest and the ability to build a system model

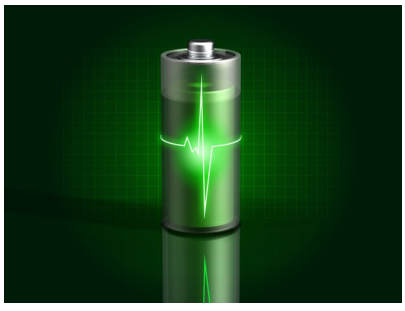

Figure 2.5: A Battery Cell

from scratch, using mathematical equations e.g differential equations and physical modelling. Differential equations closely represent the dynamics of the physical battery system. Physical modelling software such as Simscape may also be used to directly model the battery system components (e.g resistor and capacitor).

Data-Driven Approach (Black Box) Approach: In this modelling approach, sufficient domain knowledge of the system of interest is not available, however, measured data is available, with which the system model can be obtained using system identification. The data-driven SOC estimation technique can determine SOC accurately by measuring battery parameters like voltage, temperature and current. Machine Learning helps obtain relationships and rules between these parametric data. Also, data driven methods have shorter development time along with good accuracy and less cost. The data driven method is developed by analyzing data from the charge-discharge process.

Model Based (Gray Box) Approach: In this modelling approach, an idea of the system's behaviour is known, hence it can be approached from first principles, however the system parameters are not known. This means that the system model does not match available measured data from the real system. The model is therefore tuned based on measurement data to find the system parameters. The measured data was cleaned using the MATLAB data cleaner function in the MATLAB tool strip, by removing outliers and normalizing the 'test time' data. The tuning is achieved by running an optimization algorithm (Simulink Design Optimization) on the model, in correlation with the measured data until parameters are found at which the simulation converges to the measured data. Model based approach yields a relatively good real-time performance compared to other methods however it is time consuming due to numerous experiments that must be conducted before obtaining a

relationship between parameters and is also comparatively expensive. This approach sits between the first principle and data driven approaches. This research scope is established under this approach and consists of the following;

- Data acquisition and preparation
- Model Design
- Model Parameter Estimation
- Extended Kalman Filter Estimation
- Estimation Validation

Equivalent Circuit Model (ECM)

Having described the internal chemistry and modelling approach of the battery (cell), it is important to describe the battery (cell) analogously using the cell's simplification to an electrical equivalent circuit. An equivalent circuit model is electrically analogous to the electrochemical battery (cell). By performing model correlation with experimental data on an equivalent circuit model, the 'realistic response' of the battery (cell) model is achieved. With a good (realistic) battery model, an adequate controller can be designed, using modelling and simulation. To achieve simplified and less algorithm-intensive models, practical battery equivalent circuit models for engineering applications as shown in figure 2.6 consists of;

- A DC voltage source
- Internal resistor
- Parallel resistor-capacitor (RC) sub circuits

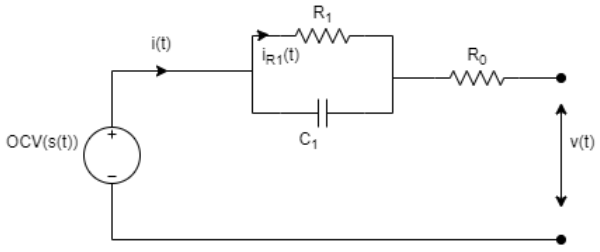

Figure 2.6: Battery cell equivalent circuit model (ECM)

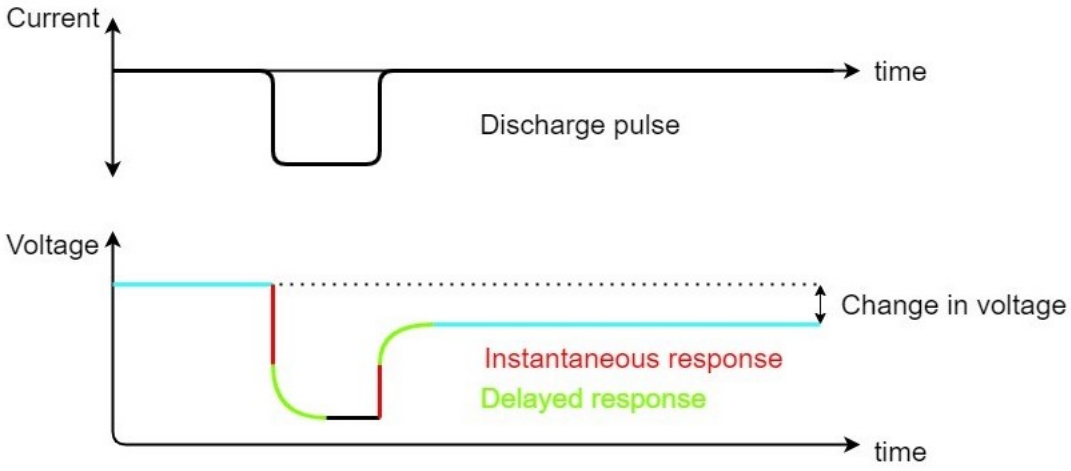

Figure 2.7: Pulse Discharge Characteristics

The OCV is an SOC and temperature dependent voltage source. The internal resistance is also dependent on SOC and temperature and represents the behavior of the separator inside the battery which describes the instantaneous response of the battery to charge and discharge current. The parallel resistor-capacitor sub circuit(s), describes the dynamic response of the battery. The voltage response of the battery cell to current pulses as shown in figure 2.7, provides a means to evaluate the circuit model parameters and the change in voltage is used to estimate the SOC.

2.3 State of Charge Definition

The oxidation-reduction reaction results in the migration of electrons from the anode, through the separator to the cathode, thus defining the state of charge dynamics of the cell, as illustrated in figure 2.8, and can be carefully defined based on the available lithium ion concentration stoichiometry in equation (2.1) [4]

$$\theta = \frac{c_{s,avg}}{c_{s,max}} \tag{2.1}$$

Where θ is the lithium concentration stoichiometry, $c_{s,avg}$ is the average concentration of lithium in the solid (s) electrode and $c_{s,max}$ is the maximum concentration of lithium in the solid (s) electrode.

This stoichiometric operating point is intended to remain between $\theta_{0\%}$ and $\theta_{100\%}$ The cell SOC is then computed as in equation (2.2);

$$z_k = \frac{\theta_{k\%} - \theta_{100\%}}{\theta_{100\%} - \theta_{0\%}} \tag{2.2}$$

Where z_k is the state of the cell and k is a unitless number between zero and 1.

However, it is not yet possible to measure the cell ion average concentration [4], therefore equation (2.2) does not suffice as a sufficient means for SOC estimation.

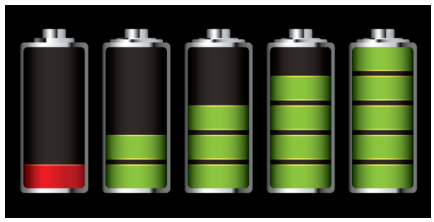

Figure 2.8: Battery (Cell) state of Charge.

Considering the relationship between SOC and current which is referred to as 'current integration'.

Let the total charge capacity of the cell be defined by equation (2.3);

$$Q = Q(i, t) \tag{2.3}$$

Where the total charge capacity Q is dependent on current 'i' and time 't'.

Assuming the cell to be fully charged at time $\boldsymbol{t_0} = 0$, the extracted charge Q_e is defined by equation (2.4) as;

$$Q_e = \int_{t_0}^{t} i(t)\, dt \tag{2.4}$$

The state of charge is defined as the ratio of the available charge (Q - Q_e) to the total charge capacity Q as in equation (2.5)

$$\boldsymbol{s(t)} = \boldsymbol{1} - \frac{Q_e}{Q} \tag{2.5}$$

Where Q is the total capacity of the cell at the temperature of consideration and Q_e is the extracted charge. SOC definitions must consider the conditions under which the cell is discharged by referring to the temperature under which the SOC is being estimated.

Let the initial state of charge (SOC) at time $\boldsymbol{t_0}$ be defined by equation (2.6);

$$\boldsymbol{s(t_0)} = 1 \tag{2.6}$$

Having defined the total capacity $\boldsymbol{Q}$ of the cell as the amount of charge removed when discharging from $\boldsymbol{z} = 1$ to $\boldsymbol{z} = 0$. $\boldsymbol{Q}$ is the charge capacity in Ah or mAh.

Therefore the state of charge of the cell at time $\boldsymbol{t}$ is defined by equation (2.7).

$$\boldsymbol{s(t) = s(t_0) - \frac{1}{Q}} \int_{t_0}^{t} i(t)\, dt \tag{2.7}$$

Where $\boldsymbol{i(t)}$ is positive discharge current. Assuming constant current pulse is applied over the interval $\Delta\ t$ with the aim of discretizing the function, resulting in equation (2.8)

$$\boldsymbol{s[t+1] = s[t] - \frac{i[t]\Delta t}{Q}} \tag{2.8}$$

To capture the inefficiency of cells, the coulombic efficiency is introduced to describe the discharge efficiency $\boldsymbol{\eta(t)}$. Coulombic efficiency is <= 1 for charging and = 1 for discharging. It is important to note that coulombic efficiency is different from and is not equal to energy efficiency. Coulombic efficiency in a typical lithium cell is 99% while energy efficiency is 95% [4]. Coulombic efficiency is higher than energy efficiency because energy is lost due to resistive heating, but charge is not lost through resistive heating.

Therefore equation (2.9), takes cognizance of the coulombic efficiency;

$$\boldsymbol{s(t) = \frac{i(t)\eta\ (t)}{Q}} \tag{2.9}$$

Depth of discharge $\boldsymbol{DOD}$ is the inverse of state of charge and is given in equation (2.10);

$$\boldsymbol{DOD = 1 - SOC} \tag{2.10}$$

$\boldsymbol{OCV}$ is plotted as a function of $\boldsymbol{SOC}$. However, $\boldsymbol{OCV}$ is also a function of temperature $\boldsymbol{T(t)}$, hence $\boldsymbol{OCV(s(t),\ T(t))}$ is introduced in the voltage model.

The terminal voltage drop $\boldsymbol{v(t)}$ of a cell under load can be modelled as in equation (2.11);

$$\boldsymbol{v(t) = OCV(s(t),\ T(t)) - i(t)R_0} \tag{2.11}$$

Note that $\boldsymbol{v(t) > OCV}$ during charge and vice versa during discharge.

$$\boldsymbol{v(t) = OCV(s(t),\ T(t)) - V_{C1}\ (t) - i(t)R_0} \tag{2.12}$$

When using experimental data to identify the equivalent circuit model parameters, the equation may be simplified in terms of the element current, which captures the current through the resistors and voltage across the capacitor, as shown as in figure 2.11 equation (2.13);

$$\boldsymbol{v(t) = OCV(s(t),\ T(t)) - R_1 i_{R1}(t) - i(t)R_0} \tag{2.13}$$

Where $\boldsymbol{R_1 i_{R1}(t)}$ is the voltage across the capacitor $\boldsymbol{C_1}$, $\boldsymbol{i_{R1}}$ is the branch current flowing through R_1, $\boldsymbol{i}$ is the total current flowing in the circuit shown in figure 2.9.

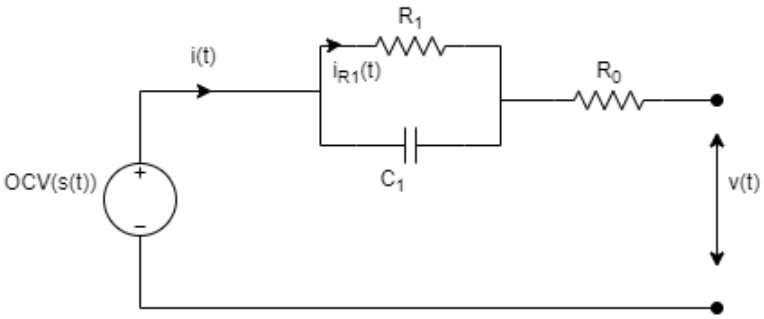

Figure 2.9: Adopted ECM in [22]

To find an expression given in equation 2.14 for the current $\boldsymbol{i_{R1}(t)}$, we take KCL at the parallel node of the ECM illustrated in figure 2.9

$$\boldsymbol{i_{R1}(t) + i_{C1}(t) = i(t)} \tag{2.14}$$

Equation 2.15 is given in terms of the capacitor current;

$$\boldsymbol{i_{C1}(t) = C_1 \frac{dV_{C1}(t)}{dt}} \tag{2.15}$$

Hence equation (2.14) may be modified to equation (2.16);

$$\boldsymbol{i_{R1}(t) + \frac{dV_{C1}(t)}{dt} = i(t)} \tag{2.16}$$

But from figure 2.9, equation (2.17) may be deduced;

$$\boldsymbol{V_{C1}\ (t) = R_1 i_{R1}(t)} \tag{2.17}$$

Therefore equation (2.16) may be modified to equation (2.18) ;

$$\boldsymbol{i_{R1}(t) + R_1 C_1 \frac{di_{R1}(t)}{dt} = i(t)} \tag{2.18}$$

$$\boldsymbol{\frac{di_{R1}(t)}{dt} = -\frac{1}{R_1 C_1} i_{R1}(t) + \frac{1}{R_1 C_1} i(t)} \tag{2.19}$$

This differential equation (2.19) may be simulated directly to determine $\boldsymbol{i_{R1}(t)}$. Note that $\boldsymbol{R_1 C_1}$ represents the time constant τ of the circuit.

Parameter estimation is a technique used for obtaining the parameter values of the components of an electrical equivalent circuit model which describes the behaviour of the battery it represents. A parameterized equivalent circuit model behaves in

such a way that when the battery block is simulated, it behaves as the real battery. This is called realistic response of the battery model. When a single parallel resistor-capacitor branch is used in the battery (cell) model, it is easy to get an idea of the parameter values. During a pulse discharge test, as the immediate change (instantaneous response) in voltage at the application of current is given in equation (2.20)

$$\Delta V_0 = \boldsymbol{R}_0 \Delta \ \mathbf{i} \tag{2.20}$$

Where from $\boldsymbol{R}_0$ is deduced.

The dynamic response (steady-state voltage) is given as in equation (2.21):

$$\Delta V_\infty = \boldsymbol{(R_0 \ + \ R_1)} \ \Delta \ \boldsymbol{i} \tag{2.21}$$

Where from $\boldsymbol{R}_1$ is deduced.

The number of R-C pairs, shows the number 'n' of the pulse response convergence time constant that characterize the battery's transient given as equation (2.22)

$$\Delta \ \boldsymbol{t} = \eta \ \boldsymbol{R}_1 \ \boldsymbol{C}_1 \tag{2.22}$$

Where from C1 is deduced.

2.4 State of Charge Estimation Methods

The term ***'estimation'*** is simply defined as the evaluation of the hidden states of a system. SOC is an intrinsic property of a battery system; it therefore must

be estimated. Primarily, every estimate has the attributes of accuracy and precision, therefore, 'accuracy' is an important performance metric for every estimation technique. When we speak of the accuracy of an estimate, we intend to know how close the estimated value of the parameter of interest, is to the true value. SOC is the amount of charge in an electrochemical cell, relative to its capacity and which describes the instantaneous state of the battery in use. SOC is a unitless quantity, expressed in percentage with 0% meaning empty and 100% representing a full state of charge.

2.4.1 Open Circuit Voltage Look-Up

According to [23], this is a white box SOC estimation technique which involves the direct mapping between SOC and other battery cell parameters including open circuit voltage (OCV), and internal resistance. The values relating the model based parameters are used to populate a table. This is achieved because SOC is proportional to the average lithium concentration, therefore, one can directly measure the output cell voltage and compare it with a look-up table OCV versus SOC. However the estimation results are poor because the terminal voltage is not a good indicator of SOC. Also, it is not practical for online application as the battery must be isolated and rested before measuring the OCV. The look-up table is the easiest method which involves a direct mapping of the SOC and the external characteristics of the battery. However, the voltage method is impractical for online SOC estimation.

2.4.2 Coulomb Counting

This method falls under the white box approach and suggests that if the initial SOC is known, then the current SOC can be calculated by counting or integrating the current. Coulomb counting method is simple to implement, however the initial

value of SOC cannot be accurately determined. Coulomb counting (current integration) is the most common technique for determining SOC at run-time. This is the integration of the output current from the cell over time. However one shortfall of this method is that it does not account for current measurement errors hence periodic error compensation is needed.

The limitations of the coulomb counting (current integration) are as follows;

1. The initial SOC must be known i.e SOC(0). This can be obtained from a fully charged and rested cell.

2. Noise in the current sensor can cause SOC drift. There is no feedback in the equation, to correct the drift until a know SOC is reached and SOC(0) can be reset.

2.4.3 Kalman Filter Algorithm

Under the gray box approach, one of the methods for evaluating the hidden state of a battery system is by using an observer [24]. However, since batteries especially lithium-ion based chemistry, are non-linear in their SOC relationship to OCV (open circuit voltage), a non-linear observer is most suitable. Therefore, an advanced estimation and tracking algorithm such as Kalman Filter can be used to improve the accuracy of a battery state estimation task. For instance, the unscented Kalman filter algorithm and unscented Kalman filter block, uses the unscented transformation to capture the propagation of the statistical properties of state estimates, through non-linear functions. The algorithm first generates a set of state values called sigma points. These sigma points capture the mean and covariance of the state estimates. The algorithm uses each of the sigma points as an input to the state transition and measurement functions to get a new set of transformed state points. The mean and covariance of the transformed points is then used to obtain the state estimates and

state estimation error covariance.

The authors in [25] provided a comprehensive review on the various methods for SOC estimation. They also provided a categorization of the methods to help researchers develop a systematic understanding of the SOC estimation techniques. Having highlighted some of the SOC estimation methods, Table 2.1, adapted from [25] shows an extensive spectrum of various SOC estimation methods.

2.5 Related Works

Because of the impact of temperature on battery systems, [22] reported how to develop a simulated lithium cell model with thermal dependence. The cell model is composed of a voltage source, one series resistance and a single R-C block. This model suitably accounted for the dynamics discovered during the charge experiment. The method adopted was effective for developing a high fidelity model, capable of predicting electrical (current/voltage) performance of the battery and estimating online SOC. The proposed method was validated by comparison with an independent set of experimental data called the New European Driving Cycle. Plotting the simulation result and experimental measurement, reveals the error percentage to be below 2%. In their model formulation, [22] chose the equivalent circuit model (ECM) because comparative electrochemical models which are able to simulate the internal dynamics of the lithium cell are computationally intensive, time consuming, inflexible and do not scale well for system level modelling and real-time applications. ECMs are capable of capturing the non-linearity of the cell's electrochemical properties and are particularly suitable for system-level analysis. The authors in [22] included the effects of temperature in the definition of the equivalent circuit elements and this is an improvement not captured in many related literature. Figure 2.10 shows the

Table 2.1: Various Methods for SOC Estimation in Literature. [26] [27]

Categories	SOC Estimation Methods	Advantages/Disadvantages
Direct Measurement	(i) Open circuit voltage method (ii) Terminal voltage method (iii) Impedance method (iv) Impedance spectroscopy method	(i) Less complexity/Impractical for batteries in service. (ii) Practical for batteries in service/Large estimation errors. (iii) Good accuracy of SOC estimation/Poor accuracy of estimated resistance. (iv) Sensitive to SOC variations/Difficult for online measurement.
Book-keeping Estimation	(i) Coulomb counting method (ii) Modified Coulomb counting method	(i) Less computational complexity/Accumulation of current sensor error. (ii) Corrected initial current/Battery must be discharged at constant current and constant temperature for accurate SOC estimation
Adaptive Systems	(i) BP neural network (ii) Support vector machine (iii) Fuzzy neural network (iv) Kalman filter	(i) Minimizes SOC estimation error value/Requires humongous training data for superior accuracy. (ii) Insensitive to small changes/Computationally complex. (iii) (iv) High accuracy and low sensitivity to measurement noise and initial SOC/Estimation accuracy depends largely on model accuracy

generalized equivalent circuit model adopted by the authors.

The parasitic branch along $\overline{BC}$ is neglected due to the negligible self discharge of the lithium cell chemistry. Furthermore, the authors [22] used the single block model with n = 1, as shown in figure 2.11. The number of circuit elements used determines the trade-off between model fidelity and complexity. The authors computed their thermal model, using the second order non-linear partial differential heat (diffusion) equation given in (2.23) and (2.24), describing the exchange of heat between a homogeneous body and its environment.

$$C_T \frac{dT}{dt} = -\frac{T - T_a}{R_T} + Ps \tag{2.23}$$

Applying Laplace transform,

$$T(s) = \frac{P_s R_T + T_a}{1 + R_T C_t s} \tag{2.24}$$

Where T is the inner cell temperature in (°C), P_s is the power dissipated inside the cell in (W), R_T is the convection resistance in ($\mathrm{Wm^{-2}K^{-1}}$). T_a is the ambient temperature in (°C), C_T is the heat capacitance in ($\mathrm{Jm^{-3}K^{-1}}$) and s is the Laplace transform variable.

The thermal model reported in [22] is considered at the cell level, however practical applications involves a combination of cells which must be modelled differently

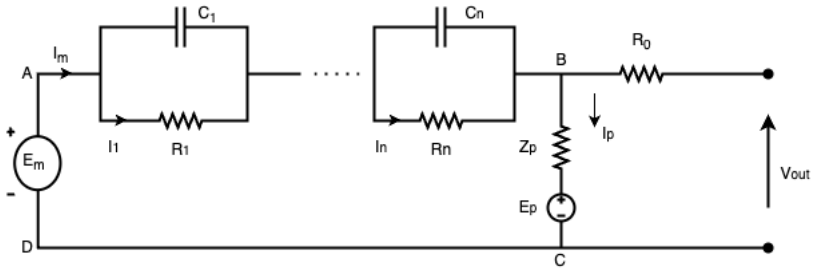

Figure 2.10: Generalized ECM of an Electrochemical Cell [22]

as thermal parameters in cell packs are different from those of isolated cells.

[28] investigated the combination of the multi-innovation least squares (MILS) and extended Kalman filter (EKF) algorithms as a composite model for the optimization of SOC estimation accuracy. The model and algorithm were verified using the urban dynamometer driving schedule (UDDS) and a charge-discharge loop test, where the error percentage was determined from the experimental curve. The result of their proposed composite electrochemistry-dual circuit polarization (E-DCP) model, showed a maximum SOC error of 2.2%. Although the joint algorithms employed by the authors, enhanced the accuracy of both the model parameter estimation and the SOC estimation, the combined algorithms are computationally complex for online SOC estimation. However, the multi-innovation least squares algorithm, proposed by the authors improved the parameter estimation accuracy while avoiding data saturation problem experienced when using the sub-optimal recursive least squares (RLS) algorithm.

The authors [29] in their paper presented a comparative analysis of two Kalman filter variations for SOC estimation. The authors adapted an ECM of which its parameters are estimated using Simulink Design Optimization and based on measurement data obtained from a controlled testing of a 2000 mAh INR 18650-20R

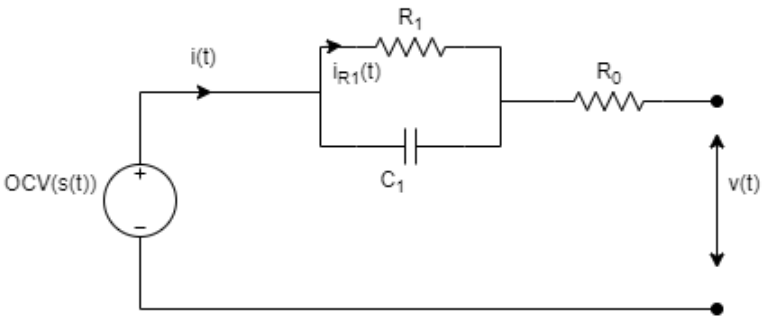

Figure 2.11: Adopted ECM in [22]

lithium-ion cell. The ECM featured three RC combination elements. In comparison with the two adaptive methods, the authors reported a faster convergence with the unscented Kalman filter method. The authors performed an elaborate comparison by considering various charge/discharge scenarios. The comparative analysis was considered only at 20 degree celcius which does not reflect the non-trivial thermal dependency of lithium-ion cells. In another article by [30], describing the design of SOC estimation using extended Kalman filter and artificial intelligence. In their approach, EKF was implemented with a Thevenin battery model with an SOC estimation error target of 3% and realised an estimation error of up to 2.8% which although is within the standard limit reported by [2] and [31], however, indicates a sub-optimal performance in comparison with the 2% realised by [22]. The authors reported a combination of EKF and Artificial Neural Network (ANN) to obtain an improved SOC estimation performance. In obtaining their battery model parameters, only parallel resistor and capacitor where computed using MATLAB parameter estimation toolbox, while the OCV (Em) and series internal resistance (R0) where computed using curve fitting which is sub-optimal and adds a layer of complexity to the parameterization task.

The authors in [32], investigated the combination of the EKF algorithm with a two-RC-block equivalent circuit and the legacy current integration method for an improved SOC estimation at runtime. In their model validation, the authors imposed errors of 40% to the initial SOC, and error of 24% to the current measurement while the voltage measurement was imposed with an error of 6%. Inspite of the contamination of the validation data with pseudo-random noise and offset in the initial condition, the novel combinatorial model of the authors, rapidly converged to an error percentage of 4% of the true SOC. In their model combination, the Coulomb counting was corrected with the SOC predictions from the EKF algorithm at ten

minutes interval unlike the traditional Coulomb counting method where the correction is done after every measurement and using the SOC-OCV correlation curve which does not suit lithium-ion chemistries. This modification reduced the computational complexity of the estimation allowing it to scale well for real-time application in EVs. However, with the SOC estimation error reaching 4%, the method employed by the authors still needs for further improvement on the SOC estimation accuracy.

The authors [33] recognised the voltage method as a common SOC estimation method, they highlighted the disadvantage of the voltage method, stating that with a poorly simulated cell model, a +0.001V modelling error in the voltage method can result in a 10% error in SOC estimation. The authors also acknowledged the Coulomb Counting as the most common method for SOC estimation. The authors stated that large errors can be produced from the EKF estimation algorithm if the battery model is highly nonlinear and this is because the EKF is only a first or second order approximation of a nonlinear model. To improve upon the short-coming of the EKF estimation method, the authors employed the UKF estimation method with the equivalent circuit cell model. The UKF is a method for calculating the statistics of a random variable propagating through a nonlinear system. The authors reported that the UKF outperforms the EKF in terms of accuracy and robustness for nonlinear estimatioin as it is a third order approximation for nonlinear models. In their identification of the model parameters including R and C, the authors used the least square algorithm which is reported in [34] to be sub-optimal due to saturation problems. The authors tested their developed method using data from various batteries and at various loading conditions. In testing their developed SOC estimation method, the result showed that the SOC estimation RMS error was less than 4% while handling unit-to-unit variation and changes in loading condition.

Authors in [35] estimated the state of charge of lithium-ion batteries using a

gray extended Kalman filter and a novel open-circuit voltage method which is based on piece-wise cubic-Hermite interpolation. In their experimentation, [35] obtained data for their battery modelling by performing a discharge test on three lithium-nickel-manganese-cobalt (LiNMC) batteries using various dynamic loading profiles to mimic the realistic EV usage scenario. Their combined gray prediction model, OCV model and Extended Kalman filter, abbreviated as GMEKF was used to estimate the SOC under driving conditions including the Urban Dynamometer Driving Schedule (UDDS) and the New European Driving Cycle (NEDC). Based on the RMSE, results from the SOC estimation evaluation, showed that the GMEKF (with maximum RMSE of 0.277) outperformed the simple EKF (with maximum RMSE of 0.0323) in estimation accuracy in both driving conditions. The only shortcoming of the method is in the employment of the forgetting factor recursive least square method was used in the identification of the model parameters, however the error caused during OCV-SOC fitting was avoided by combining a new method based on piece-wise cubic-Hermite interpolation.

[36] employed coulomb counting, equivalent circuit model and a mathematical model based on Peukert Law in their attempt to estimate state of charge. Although the authors did not specify the battery chemistry in consideration, they reported to have modified the equivalent circuit model and coulomb counting by applying the coefficient of discharge efficiency determined from the Peukert mathematical model with the aim of mitigating the inaccuracy issue of the coulomb counting method. The SOC was estimated from the open circuit voltage (OCV) which was also estimated from the internal resistance, rendering the estimation task in double layer which is not well suited for online state of charge estimation.

The look-up table is a model based SOC estimation technique which involves the direct mapping between SOC and other battery cell parameters including open

circuit voltage (OCV), and internal resistance. The values relating the model-based parameters are used to populate a table [37]. The impedance method is related to the equivalent circuit method; however, it is tedious as it requires a signal generator in the process which is impractical for online application [38]. For a battery system, input signals include current and ambient temperature while terminal voltage, battery temperature, state of charge, state of health, internal resistance and open circuit voltage are considered output signals.

2.6 Major Research Gaps

Most literature in the field of battery management, provide a high level conceptualization on various approaches to SOC estimation without a clear and repeatable road map that shows the implementation steps. This work will contribute towards filling the knowledge gap by providing a coherent and repeatable method for SOC estimation and to provide a performance comparison of two state of the art techniques for SOC estimation. Furthermore, the extended Kalman filter was employed in closing the knowledge gap and also improving on the SOC estimation accuracy in comparison to the conventional Coulomb counting method.

Chapter 3

Research Methodology

3.1 Introduction

In this chapter, the methods adopted in accomplishing the SOC estimation task are described, which includes the Coulomb Counting (Current Integration) and the Extended Kalman Filter methods. Starting with a brief description of the experimental data which was used in the research, to the computational tools and resources, used in the implementation of the research project. A step by step description of the Coulomb Counting estimation method, is therefore provided. Similarly, the Extended Kalman Filter estimation method, is described according to the implementation steps.

3.2 Battery Data

Because battery cells in their service life undergo charge and discharge processes, therefore one of the ways to obtain experimental/measured data from a battery system is to perform a controlled charge/discharge experiment on the battery of

preference at relevant temperature conditions. The controlled full charging and discharging of a cell may be achieved in a laboratory setting, and used to calibrate a dataset at the end of a test. This measured dataset is further used as reference 'realistic' dataset in the modelling and simulation of a cell for the estimation of real SOC.

3.2.1 The Panasonic 18650PF Li-on Battery Data

A Panasonic 18650PF Li-ion battery was tested at the University of Wisconsin-Madison by [39] and the dataset is suitable for developing battery models for implementing the Kalman Filter SOC estimation. The battery data provided by [39] was generated by testing a brand new 2.9Ah Panasonic 18650PF NMC cell in a Universal Battery Tester channel with an 8 cu.ft thermal chamber and a 25 amps, 18 volt Digatron Firing Circuits. Several tests including discharge and charge test, pulse discharge (HPPC) test, electrochemical impedance spectroscopy (EIS) test and drive cycle tests were captured in the dataset. However, for the Coulomb Counting experiment in this research, the discharge data file '07-23-17-01.42 4020-Dis1C-Rp' (discharge and charge tests) was used. This data was obtained as an open-access battery dataset downloaded from Mendeley repository [39]. The data in this file contains variables including the discharge 'Time' and 'Current' of 1C-rate measured at 25deg. The data was sampled at a period of 0.2 seconds. The discharge 'Time' and 'Current' data will be used in the Coulomb counting model to determine the SOC estimation performance of the model.

3.2.2 Battery Intelligence Lab, University of Oxford

Battery degradation dataset was adopted from the Battery Intelligence lab at the University of Oxford. Commercial lithium cells of capacity 3 Ah of the 18650 form factor with NCA/graphite (cathode/anode) were tested. The generated test dataset

is constituted in 3 parts which features 28 cells categorized into 10 groups. More information about the data may be obtained in the report presented by authors in [40]

This research employed a section of the control experiments captured under group 5 of the 'Path Dependent Battery Degradation Dataset Part 1' which consist of cells exposed to continuous cycling at C/2-rate. The selected data section consists of pulse discharge/charge data which is contained in the file named 'TPG5.1-Cell5.mat'. This data-file was obtained simply by downloading the 'Path Dependent Battery Degradation Dataset Part 1' from [41]. The data is a reference performance tests (RPT), which was used to characterise the cells periodically.

3.3 Implementation using MATLAB, Simulink and Simscape

This research was implemented using MATLAB, Simulink and Simscape as design tool and this is because of the modelling and simulation capability of the software. Licence to the software was provided through the University of Cape Town's Campus wide licence for MATLAB. MATLAB is a high-level language with an interactive development environment and it is used for numerical computation, data analysis and visualizations, algorithm development and programming, application development and deployment. Simulink is a block diagram environment for multi-domain simulation and model-based design. It supports system-level design, simulation, automatic code generation and continuous test and verification of embedded systems including the BMS. Simulink provides a graphical editor, customisable block libraries and solvers for modelling and simulating dynamic systems such as a battery. Simulink is integrated with MATLAB, enabling the incorporation of MATLAB algorithms into Simulink models and exporting simulation results to MATLAB for further analysis.

Simulink is best known for signal-based modelling using transfer functions. However, the Simscape extension is known for modelling physical network which enables bidirectional flow of energy between model components. MATLAB alone was considered as the modelling tool in this research. This is not because there are no other modelling tools available but firstly because the Simulink/Simscape extensions of MATLAB provides a graphical user interface which minimizes the rigour of programming for modelling and simulation tasks. Secondly, MATLAB feature toolboxes such as the control system toolbox and parameter estimation toolbox which contain relevant blocks and algorithms required for the research. Lastly, MATLAB was chosen because of the student's proficiency and preference for MATLAB, for which access is provided through the University of Cape Town's Campus-wide licence for MATLAB.

3.3.1 Description of Model Blocks

The model building blocks to be implemented are provided in the Simulink/Simscape library. To help the reader appreciate the various model blocks that are implemented towards achieving the SOC estimation objectives, the description of the individual block components in all models are presented viz;

1. **Table-Based Battery**: The battery to be used in the system design, can be found either in the Electrical > Sources or Battery > Cells section of the Simscape library. This Battery (Table-Based) block illustrated in figure 3.1, represents a battery model with high fidelity and it calculates OCV in terms of SoC and temperature (optional) with the use of lookup tables. Various modelling options including self-discharge, battery fade, charge dynamics and calendar aging are captured by the battery model. The battery block was employed for the parameterization of a realistic battery. For a more detailed description of the battery (table-based), see [42]

Figure 3.1: Table-Based Battery

2. **Controlled Current Source**: This block, shown in figure 3.2, resides in the Electrical Sources section of the Simacape foundation library. It models an ideal current source which delivers a constant current regardless of the source voltage. The numerical value for current input at the physical signal port is equal to the constant current output maintained by the block. Refer to [43] for further information.

Figure 3.2: Controlled Current Source

3. **Voltage Sensor**: The Voltage Sensor block as illustrated in figure 3.3 is found in the Electrical Sensors section of the Simscape Foundation Library. As an ideal voltage sensor, it measures and converts the voltage across its terminals to a physical signal with equal magnitude as the measured voltage. The physical signal is outputted through the physical signal port 'V'. [44]

4. **Inport/Outport**: The inport and outport blocks as shown in figure 3.4 and figure 3.5, are commonly used Simulink blocks in model-based design and can be found in various libraries including the Ports and Subsystem section of the

Figure 3.3: Voltage Sensor

Simulink library. While the inport block enables the linking of signals from outside the system into the system, the outport block facilitates the linking of signals from a system to an extenal destination such as the Workspace. Both inport and outport blocks also serve to connect signals flowing to and from a subsystem to various parts of the model. [45] [46]

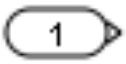

Figure 3.4: Inport

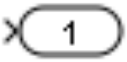

Figure 3.5: Outport

5. **Solver Configuration**: The solver block as illustrated in figure 3.6, is found

in the Utilities section of the Simscape library and is required by every physical network built of Simscape blocks. This block serves to specify the solver parameters to be used by the model before simulation. [47]

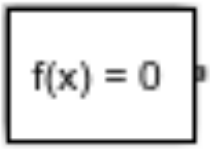

Figure 3.6: Solver Configuration

6. **Scope**: The scope as shown in figure 3.7, is also a commonly used block found across various libraries such as the Sinks section of the Simulink library. The block is relevant for displaying time-domain signals generated during simulation. More information about the scope can be obtained from the MATLAB documentation [48]

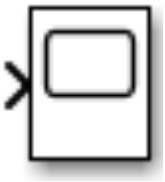

Figure 3.7: Scope

7. **Unit Delay**: The unit delay block shown in figure 4.9, is found in the 'Discrete' section of the Simulink library. The block holds and delays its inputs by a specified sample period and can accept one input for which it generates one corresponding output. The input/output signal may either be a scalar or vector. The first output sample of the block must be specified with the initial condition parameter. The block also allows for the specification of the sample time between samples. [49]

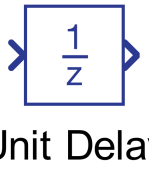

Figure 3.8: Unit Delay

8. **Add/Divide/Product**: These blocks are shown in figures 3.9, 3.10, 3.11 and are employed for implementing mathematical operations. They are found in the Math Operations section of the Simulink library. For more information regarding the dynamic implementation of the blocks, please refer to the MATLAB documentation [50] [51] [52]

Figure 3.9: Add

Figure 3.10: Divide

9. **Constant**: The constant block shown in figure 3.12 serves to generate a real or complex constant value signal which may be scalar, vector, or matrix output, with the same dimensions and elements as the specified constant value

Figure 3.11: Product

parameter. The block can be found in the commonly used blocks section of the Simulink library. [53]

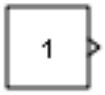

Figure 3.12: Constant

10. **Gain**: Another commonly used block is the Gain block, shown in figure 3.13 which multiplies an input by a specified constant value called the 'gain'. Both the input signal and the specified gain parameter may be a scalar, vector or matrix. [54]

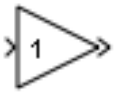

Figure 3.13: Gain

11. **GoTo/From**: The GoTo and From blocks, shown in figures 3.14,3.15, are Simulink signal routing blocks which allows the linking of signals from one block

to another without actually connecting them. The blocks are implemented in association, using the GoTo tag and therefore every GoTo block must have at least one coresponding From block associated to it while every From block can only receive singnal from one associated GoTo block. [55] [56]

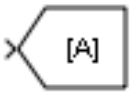

Figure 3.14: GoTo

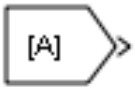

Figure 3.15: From

12. **Selector**: The selector block shown in figure 3.16, also serves for Simulink signal routing by extracting selected elements of an input vector, matrix or a multidimensional signal, based on specified indices. [57]

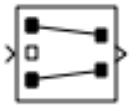

Figure 3.16: Selector

13. **PS-Simulink/Simulink-PS Converter**: These blocks, shown in figures 3.17, 3.18 are Simscape utility blocks for implementing signal conversion. While the PS-Simulink block converts an input physical signal into an output Simulink signal, the Simulink-PS takes input Simulink signal and converts it to physical signal output. [58] [59]

Figure 3.17: PS-Simulink

Figure 3.18: Simulink-PS Converter

14. **Mux**: The Mux, shown in figure 3.19 is a commonly used block for signal routing. Inputs with the same data type and complexity are combined within the block into a virtual vector. [60]

Figure 3.19: Mux

15. **1-D LookUp**: The Lookup Tables found in the Simulink library evaluates a function with a set of variables. This block maps a set of inputs to an output value by looking up or interpolating the table of variable specified in the block parameter. There are various interpolation methodos within the block parameter but he linear interpolation method will be implemented for our design. This block is shown in figure 3.20 [61]

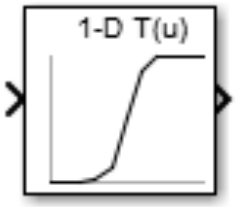

Figure 3.20: 1-D LookUp

16. **Extended Kalman filter**: Found in the state estimation section of the control system toolbox, this block as illustrated by figure 3.21 was implemented to estimate the state of the cell using first-order discrete-time extended Kalman filter algorithm. This will require the complementary use of the cell's state transition and measurement functions to generate the SOC estimates. [62]

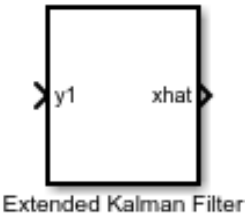

Figure 3.21: Extended Kalman filter

17. **Simulink Function**: The Simulink function block shown in figure 3.22, allows graphical definition of user-defined functions and thus will be implemented [63]

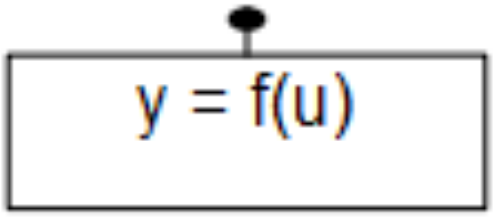

Figure 3.22: Simulink Function

18. **MATLAB Function**: The state equations for modelling the state transition function will be implemented using the MATLAB function block, illustrated by figure 3.23. This block allows for writing user-defined MATLAB functions within a Simulink model. [64]

Figure 3.23: MATLAB Function

3.4 Cell Equivalent Circuit Modelling and Characterization

A 3Ah Nickel Cobalt Alluminium (NCA) oxide battery model is implemented on Simulink and parameterized using experimental data to create the realistic model of the cell of interest. The Simulink library provides a wide scope of model blocks across various engineering domains, such as the electrical, mechanical, thermal and hydraulic systems. A table-based battery, is featured in the Simscape library, as an electrical system block for modelling and simulation. This block allows for easy

specification of battery parameters and for model characterization. The parameterization of the table-based battery is accomplished using the estimation manager tool on Simulink. Initial guesses are optionally specified for the battery parameters (voltage, resistance and capacitance). The battery is then correlated with 'inported' experimental voltage data. Hence the table-based battery is tuned to parameters that uniquely defines the battery block as a realistic battery.

3.5 The Coulomb Counting SOC Estimation

The implementation of the Coulomb Counting estimation algorithm, starts by considering the relationship between SOC and cell current. As defined in chapter two, the SOC of a cell is the ratio of instantaneous charge capacity to the total charge capacity. The discrete-time Coulomb Counting estimation equation given in (3.1) is then implemented for the SOC estimation, as a model on Simulink. This will be achieved through the connection of some Simulink Library blocks as will be seen in more detail in chapter 4. Computationally, the Coulomb Counting method offers an inexpensive method algorithm for SOC estimation and this is a reason for which the method was chosen in this research.

$$\boldsymbol{SoC(k) = SoC(k\text{-}1) + \frac{Ts * i(k)}{2.0 * 3600}} \tag{3.1}$$

3.6 The Extended Kalman Filter SOC Estimation

The extended Kalman filter block is implemented to linearize the non-linear battery model and to use the linearized model to predict the battery state of charge through predict-update steps. The extended Kalman filter block is found in the State Estimation library of the Control System Toolbox. We implement the extended

Kalman filter block with the required State transition and Measurement functions. Similar to the Coulomb Counting method, the Extended Kalman filter estimation is implemented with Simulink Library blocks as we would see in the next chapter 4. It mainly involves the implementation of the Extended Kalman filter block, and the implementation of the state transition and measurement function blocks which are the main components of the Extended Kalman filter block.

3.6.1 State Transition Function

The non-linear state transition function to be implemented, will describe the evolution of the battery state from one time step to the next. As described in chapter 2, the state transition function for the 1 R-C battery model considered in this work, is implemented according to equations (3.2) and (3.3).

$$\frac{\boldsymbol{dSoC}}{\boldsymbol{dt}} = -(\frac{1}{\boldsymbol{3600 * Capacity}})\boldsymbol{i(t)} \tag{3.2}$$

$$\frac{\boldsymbol{dV_c}}{\boldsymbol{dt}} = \frac{1}{\boldsymbol{R_1 C_1}}\boldsymbol{V_c(t)} - \frac{1}{\boldsymbol{C_1}}\boldsymbol{i(t)} \tag{3.3}$$

In this work, we consider SOC in terms of the current only. Hence the only State transition function input will be the current. Also, because the SOC estimation is implemented in discrete-time, hence the state transition function is modelled accordingly by the corresponding discrete-time equation of the general form given in equation (3.4)

$$\boldsymbol{x(k+1) = f(x(k),\ w(k), StateTransitionFcnInputs)} \tag{3.4}$$

In equation (3.4), ***'x'*** denotes the battery states, (i.e ***SoC*** and $\boldsymbol{V_c(t)}$), ***'w'*** is the process noise, ***'k'*** is the time step and the ***'StateTransitionFunctionInputs'*** are the additional inputs to the function which depends on the model design and as

aforementioned in this research, the additional input includes the current only.

The generic state equation is thus redefined for the specific implementation of the 1 R-C battery model as given in equation (3.5);

$$\boldsymbol{x(k+1)} = \boldsymbol{x(k)} + \boldsymbol{f}(\boldsymbol{x(k)i(k)})T_s \tag{3.5}$$

The modelling of the state transition function will be described in more details in chapter 4.

3.6.2 Measurement Function

The non-linear function which establishes a relationship between the state and the measurements at the current time-step is implemented as a component of the extended Kalman filter. This function is modelled according to equation (3.6)

$$\boldsymbol{v_t(k)} = \boldsymbol{E}_m - \boldsymbol{i(k)R}_0 - \boldsymbol{V}_{C1}\boldsymbol{(t)} \tag{3.6}$$

In this equation, $\boldsymbol{V}_t$ is the terminal cell voltage or the model's measured output voltage, $\boldsymbol{V}_c$ is the voltage of the parallel capacitor $\boldsymbol{C}_1$ in the 1 R-C model. $\boldsymbol{R}_0$ and $\boldsymbol{E}_m$ are both SOC dependent cell resistance and OCV respectively.

Chapter 4

Research Design

4.1 Introduction

In this chapter, a very much detailed description of the research implementation is reported. All the model blocks described in the preceding chapter are interconnected into various systems and sub-systems and their relationship is clearly established. This chapter starts by setting up the parameter estimation experiment for the battery to obtain the realistic battery model and true SOC, for which the corresponding SOC will be estimated, using the Extended Kalman filter estimation model. All descriptions in each of the sections will be presented in terms of 'Block Selection and Connection', 'Block Parameter Definition', 'Data Import and Model Configuration', and finally 'Model Simulation'.

4.2 Modelling the Battery

The 3Ah Nickel Cobalt Alluminium (NCA) oxide battery model was implemented on Simulink by creating a blank model from the 'Simulation' tab of the Simulink tool strip. The file is named 'EqiCCT.slx'.

4.2.1 Block Selection and Connection

By double-clicking within the blank model canvass, the following blocks are selected from their respective libraries and included in the model:

- Battery (Table-Based): This battery is the core of the model, for which every other block is implemented. It has three connection terminals (positive, negative and SOC) and is connected to the voltage sensor and controlled current source, via the positive terminal, while the negative terminal is connected to the electrical reference. The SOC port is connected to an Outport block. The battery block parameter settings will be described in the next subsection.

- Voltage Sensor: The voltage sensor measures the terminal voltage across the battery block with its positive terminal connected to the positive of the battery and the negative connected to the electrical reference. The voltage data port 'V' which outputs a physical system signal is connected to a Simulink signal-based outport block through a 'PS-Simulink Converter'.

- Controlled Current Source: The Controlled Current Source accepts current input data, presented at its physical signal port, from the inport block through the 'Simulink-PS Converter'. The current profile is applied for parameterization of the battery block. The parameterized battery block then becomes a realistic battery with SOC measured from the SOC port. The battery block 'parameterization' will be discussed in more detail in the the next section.

- Inport/Outport: The inport/outport blocks are numbered automatically. The model consists of two inport and two outport blocks. Ports 1 and 2 of the inport blocks, respectively conveys current and voltage data, specified in the MATLAB workspace as input variables to the model, for the purpose of parameterization of the battery (table-based). Ports 1 and 2 of the outport blocks, respectively

conveys the simulated voltage and SOC, from the model, at the end of the model simulation, to the MATLAB workspace.

- Simulink-PS/PS-Simulink Converter: The signal converter blocks were used to convert Signals between Simulink and Simscape.
- Solver: The solver block was used to initialize the model.
- Electrical Reference: The electrical reference block provided an electrical ground for the circuiting of the electrical model.
- Scope: We used the scope to visualize the actual signal output from the system, and hence analyse the system performance and estimation accuracy.

Figure 4.1 shows the full battery system model as designed.

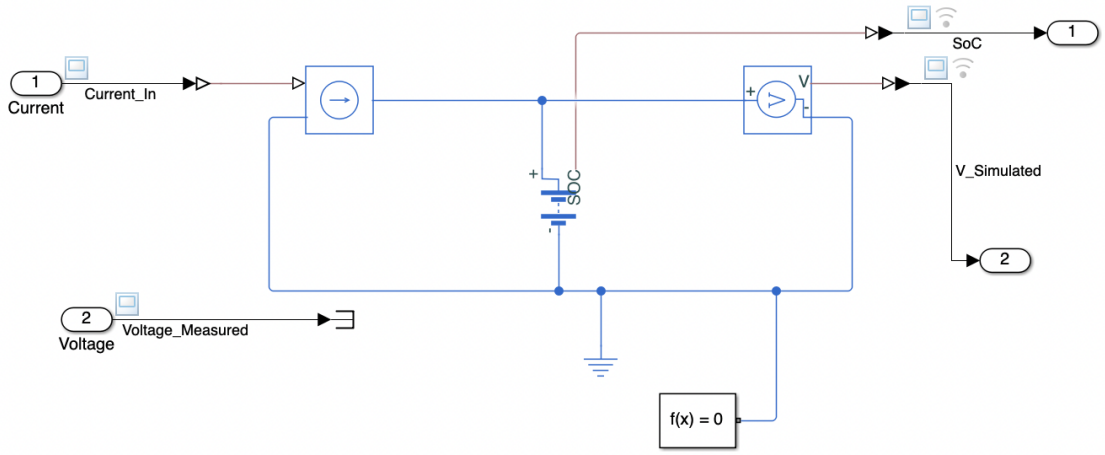

Figure 4.1: Battery System Model

4.2.2 Block Parameter Definition

- Battery (Table-Based): Various parameters were specified within the battery block parameter dialog box as shown in figure 4.2. In the implementation, specification was only provided under the 'Main' and 'Dynamics' sections of the block parmeter dialog box. This is because the model design of interest, involves only one time constant and does not consider temperature (isothermal). The impact of the number of time constants on accuracy in the ECM architecture depends on the previous knowledge of the battery system sophistication, which describes the behaviour of the real battery. For instance, when modelling for other electrochemical processes (e.g hysteresis) associated with the battery, more time constants may be modelled to represent each of these electrical processes in more sophisticated battery chemistries. The variables specified in the dialog box were correspondingly created in the MATLAB workspace. The workspace containing the specified variables and experimental data, is then saved as a .mat file named 'modelWorkspace.mat'. The file is loaded with the model callback option as a preLoad function. This ensures that the specified model variables, are automatically loaded, whenever the 'EqiCCT' file is opened. The variable names specified in the battery block parameter dialog box include; 'mSoC', 'mEm', 'mC_1', 'mR_0', and 'mR_1'. These variables are the battery parameters to be evaluated by parameterization as they describe the dynamics ('battery personality') of the battery of interest.

4.2.3 Data Import and Model Configuration

Data is imported into the battery model on two instances which are the parameterization data and the simulation data, however the same data used for parameterization is also used for simulation. The parameter estimation data import procedure will be described in the Battery Parameter Estimation section of this chapter. To

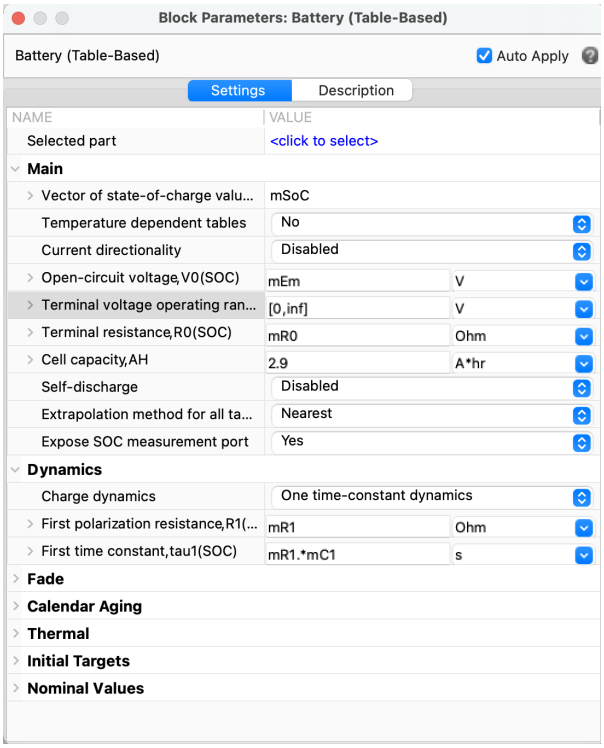

Figure 4.2: Definition of Block Parameters

import the simulation data, which is saved in the mat-file named 'SimData', a 'Root Inport Mapper' scenario was created. The Root Inport Mapper scenario is created and accessible through the 'Connect Inputs' button of the model 'Configuration Parameters' (Settings) dialog shown in figure 4.3. Through the Edit tab of the 'Root Inport Mapper' window, a new MAT-file was created in the signal editor and the simulation data from the MATLAB workspace, is used to specify the model input signals as an 'Author Signal' in the signal editor dialog (See figure 4.4). Further specifications are provided in the Configuration Parameter dialog.

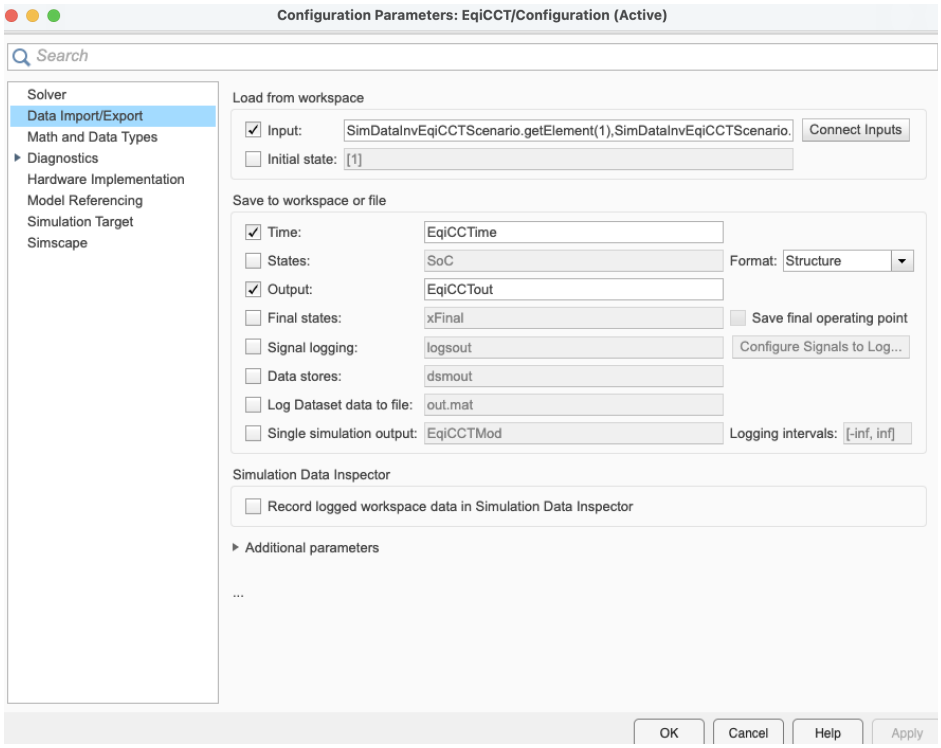

Figure 4.3: Inport Data Specification

4.2.4 Model Simulation

The battery model is simulated to view the model outputs on the scope, and to ensure that the model compiles correctly and without errors. The model output result will be discussed in the proceeding chapter (5). The simulation time specification must be implemented for the full duration of the specified model signals, as it influences the model performance and result. This means that the ending time value in the imported time data, should be specified as the the simulation duration in seconds.

4.3 Battery Parameter Estimation

Parameter estimation is a technique used for obtaining the parameter values of the components of an electrical equivalent circuit which describes the behaviour of the battery it represents. The aim of parameter estimation is to generate the look-up table for each of the cell parameters including open circuit voltage (E_m), terminal resistance (R_0), parallel resistance (R_1) and the time constant (tau), according to their correlation with each level of SOC.

The 'Parameter Estimation' App is launched through the 'Apps' tab in the Simulink toolstrip as shown in figure 4.4. In the App session, a parameter estimation experiment named 'SimDataInvParam', is created in the 'Experiments' section of the app. Parameters from the battery model to be estimated, are also specified in the 'Parameters' section of the app. This is illustrated in figure 4.5. The battery model is parameterized using the data, imported from a .xls file named 'ParamData' (which is different from the mat-file 'SimData').

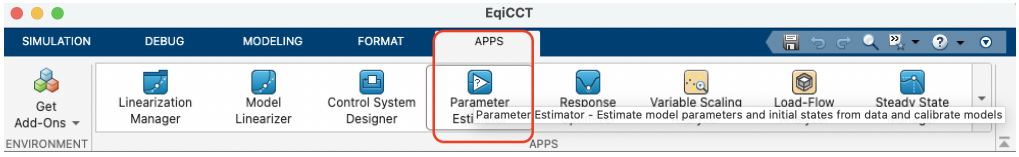

Figure 4.4: Launching Parameter Estimation App

The parameter values (E_m, R_0, R_1 and tau) of the cell block are specified as initial guesses. The initial guesses are simply computed (for a 1R-C Equivalent circuit), using equations (4.1), (4.2), and (4.3).

$$V_0 = R_0 \Delta i \tag{4.1}$$

$$V_\infty = (R_0 + R_1)\Delta i \tag{4.2}$$

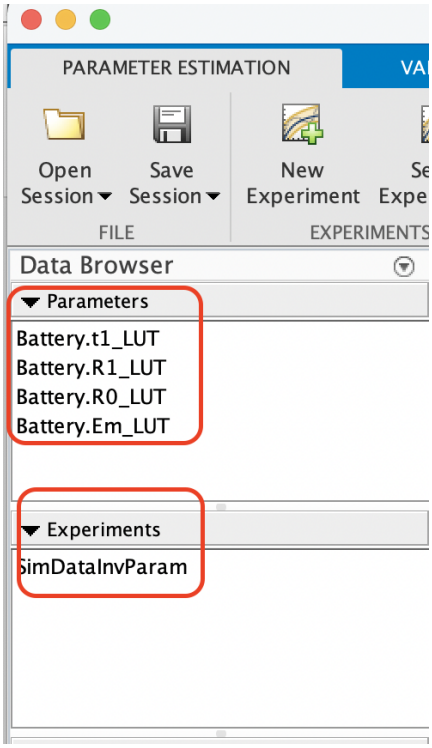

Figure 4.5: Experiment/Parameter Specification

$$\Delta t = 5R_1C_1 \tag{4.3}$$

The variables for computing the initial guesses may be obtained from a graphical analysis of the pulse discharge plot as shown in figure 4.6. The figure (4.6), describes the pules-discharge plot of the battery, showing important data points. The 'Y-axis' represents the voltage and current in each sub-plot, while the 'X-axis' represents the time in seconds, in both sub-plots. Using the data points, the change in current 'Δi', is computed and with the nominal voltage of the battery 'V_0', (4.1) is further computed to find the initial guess for R_0. By substituting for R_0 in (4.2), the initial guess for R_1 is computed. (4.3) is likewise evaluated to obtain the initial guess for the time constant (R_1C_1). The initial guesses were thus evaluated to the following

values;

- E_m = 3.6 V (Nominal voltage for the cell of interest)
- R_0 = 0.002584 Ohms
- R_1 = 0.00471 Ohms
- tau1 = 12.36

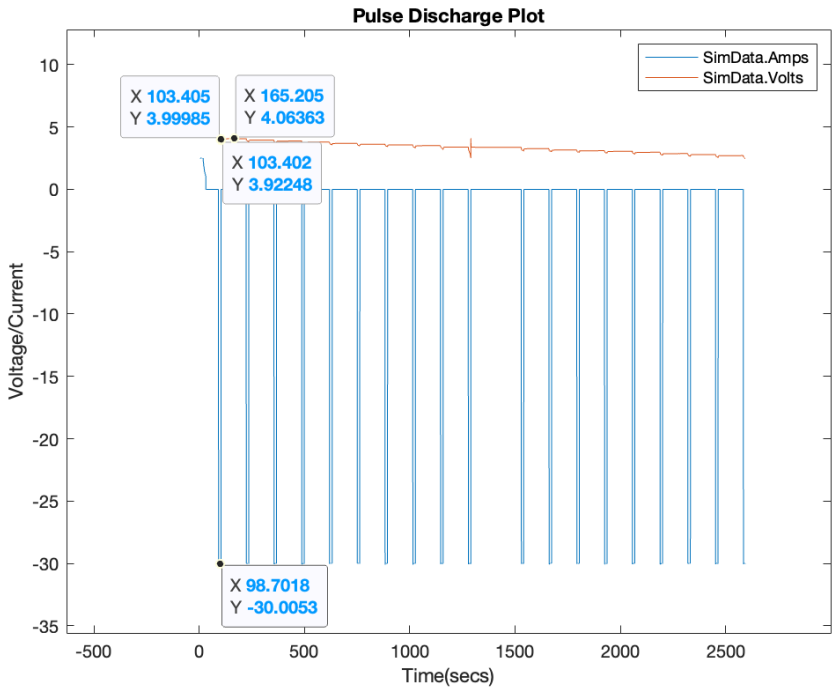

Figure 4.6: Pulse Discharge plot showing data tips

The cell is excited (stimulated) using the same discharge pulse profile, as in the experimental setup. The simulation model response for the cell voltage is correlated with the measured experimental data and the parameter values are adjusted (tuned)

until convergence is achieved between the measured simulated response plot. Look-up tables resulting from the parameterization session, are saved for use in the SOC estimation step.

4.3.1 Block Parameter Definition

The 'SimDataInvParam' experiment is edited with the appropriate specifications of the parameter estimation and the data is imported, using the 'edit experiment' dialog. shown in figure 4.7.

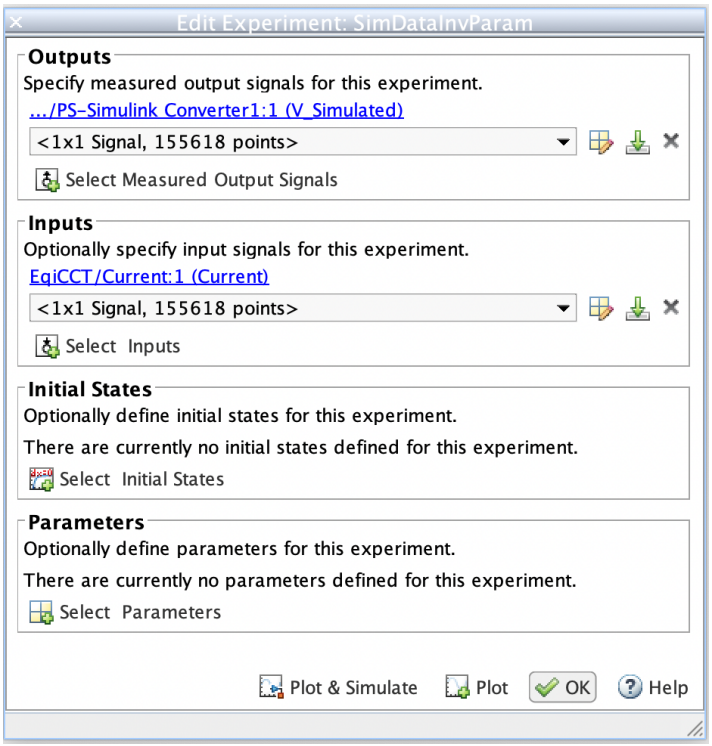

Figure 4.7: Experiment Specification for Parameter Estimation

4.3.2 Data Import and Model Simulation

The parameter estimation data is imported via the import icon in the 'SimDataInvParam' dialog from the 'ParamData' file. The time-current data is specified for the input while the time-voltage data is specified for the output. The 'plot and simulate' button allows to visualize and compare the simulated voltage to the measured voltage of the parameter estimation data. The parameter estimation initial plot is shown in figure 4.8. The result from the parameter estimation final plot will also be discussed in chapter 5.

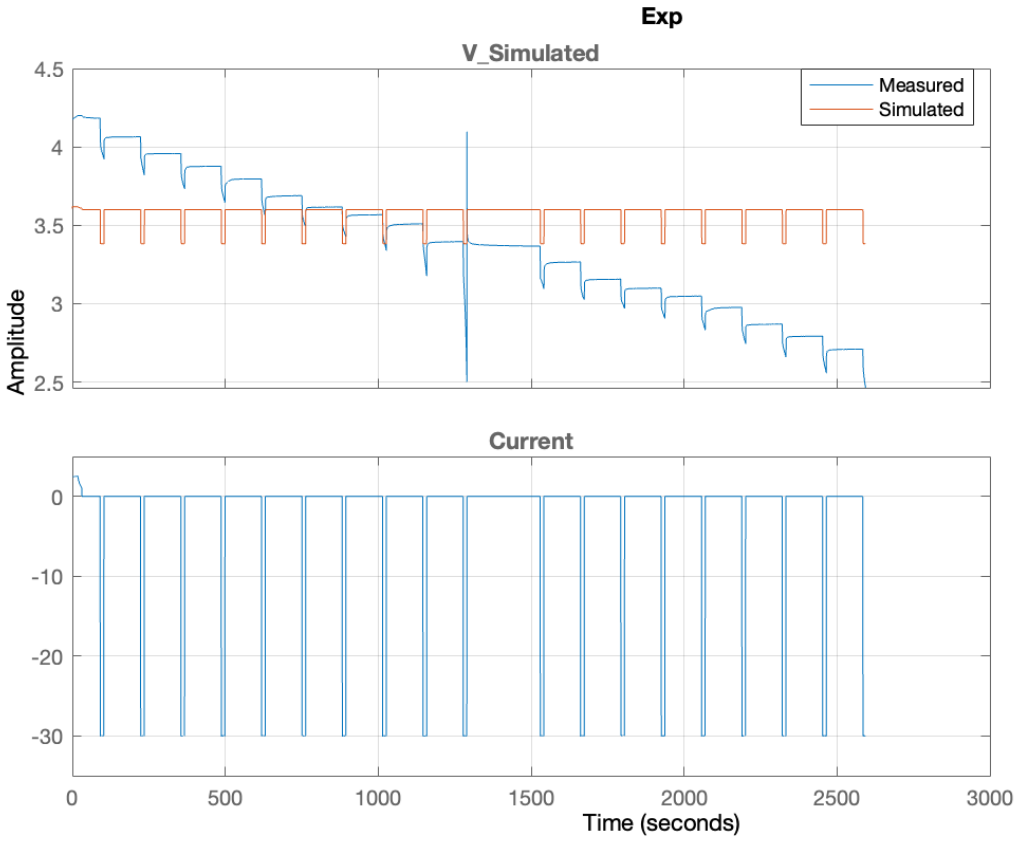

Figure 4.8: Experimental Plot before Parameterization

4.4 The Coulomb Counting Model

The Coulomb Counting Model was implemented by creating a separate Simulink model file named 'CC_Model_fin.slx'. This model is basically a graphical implementation of the Coulomb Counting equation as in (4.4)

$$\boldsymbol{SoC(k) = SoC(k\text{-}1) + \frac{Ts * i(k)}{2.9 * 3600}} \tag{4.4}$$

4.4.1 Block Selection and Connection

The Coulomb Counting model is built from the following model blocks which are selected from the Simulink library.

- Unit Delay: The Unit Delay block implements the term for the previous SoC time-step '***SoC(k-1)***' in the Coulomb Counting equation. The unit delay block takes input of SOC and returns a time delayed SOC output as a feedback.
- Add: The 'Add' block is a design block which performs the sum operation at the input on the terms ***SoC(k-1)*** and $\frac{Ts*i(k)}{2.9*3600}$.
- Divide: The 'Divide' block implements the ratio between the terms ***Ts*i(k)*** and $2.9 * 3600$.
- Constant: The 'Constant' block evaluates the capacity of the battery of interest in Ampere-seconds.
- Sample Period: The 'Sample Period' block multiplies the input current by the sampling period.
- Weighted Sample Time: The Weighted Sample Time block measures and outputs the sampling period of the input data.

- Inport/Outport: The inport block provide the channels for linking external current input data to the model while the outport block links the specified model output(s) to the MATLAB workspace.
- Scope: The scope is implemented in the viewer mode and configured to open at the initialization of the model. The scope enables visualization of the signals processed within the model, for the purpose of analysis.

4.4.2 Block Parameter Definition

- Unit Delay: The performance of the Coulomb Counting model largely depends on the definition of the the initial SOC condition which is specified by assumption, within the unit delay block parameter as shown in figure 4.9.

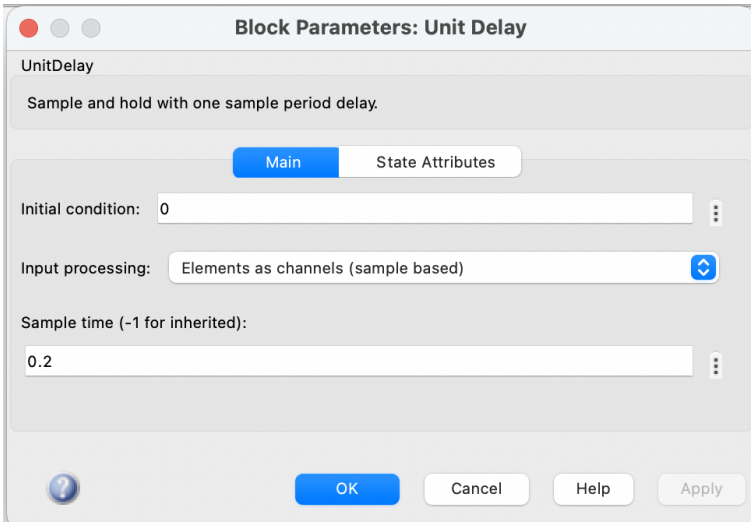

Figure 4.9: Unit Delay Block Definition

4.4.3 Data Import and Model Configuration

The Coulomb Counting model comprises of only one inport block, this block provides a root import link for importing experimental data into the model. The

signal is defined just as described for the battery model in subsection (4.3.2) and then mapped according to the port number.The imported simulation data file is names 'SimData'.

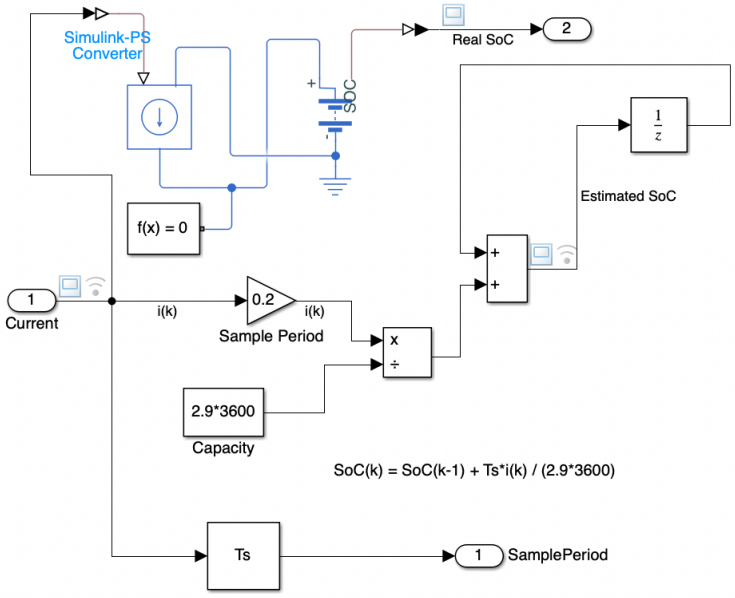

Figure 4.10: Coulomb Counting Model Design

4.4.4 Model Simulation

To simulate the Coulomb Counting model, the model simulation time was specified as the last value of the simulation time data, using the syntax 'SimData.TestTimeNorm(end)'. The simulation time value is 2593,43683 seconds. By clicking the 'Run' button, the model simulation begins and the result from the simulation is visualized using the model scope which is set to open at simulation. The implemented Coulomb Counting model is illustrated in figure 4.10

4.5 The Extended Kalman Filter Model

A separate model file named 'SoCEstimation.slx' was created for the extended Kalman filter estimation model. This model compares the real SOC to the estimated SOC by implementing the extended Kalman filter.

4.5.1 Block Selection and Connection

The model is constituted using the following blocks obtained from the Simulink block library.

- Battery(Table-Based): This battery block is identical to the one implemented under 'Modelling the Battery' section. The battery is a 3 Ah Nickel Cobalt Aluminium (NCA) oxide battery which was validated with a pre-parameterized battery block of the same specification, used as the reference to simulate the real SOC to which the Extended Kalman filter estimate SOC is correlated. The battery block is connected to the controlled current source and voltage sensor blocks.

- Controlled Current Source: The controlled current source supplies the current specified at the current inport block by linking the current inport block to the to the battery block.

- Voltage Sensor: The voltage sensor measures the voltage potential at the battery terminals and outputs the signal as voltage input to the state transition function block.

- Band-Limited White Noise: The noise block, simulates the process noise which is specified as an additive to the Exteded Kalman filter block. The noise is added using the add block, to the current and voltage outport blocks.

- State Transition Table: The state transition table enables the specification of control signal which controls the charge-discharge cycle of the battery. The battery SOC range is set between a lower SOC of 0.3 and an upper SOC of 0.9. When the battery SOC reaches upper SOC, the current switches to discharge mode and when the battery SOC reaches lower SOC, the current switches to charge mode.

- Inport/Outport: These blocks enables the root import mapping of the external battery simulation data linked to the model.

- Solver Configuration: The solver configuration defines the solver settings required for simulation of the model.

- Switch: The switch as controlled by the charge discharge control signal switches the model input current between charge and discharge.

- Rate Transition: The rate transition block ensures the data transfer integrity form the battery subsystem to the Simulink function blocks.

- Selector: As the signal processed within the Simulink function blocks are multidimensional, the selector blocks enables the selection of specific elements from the multidimensional signal.

- Scope: The scope processes the final output signal for visualization and analysis.

- Simulink Function: The Simulink function blocks enables the implementation of the state transition and measurement functions required by the Kalman filter.

- Extended Kalman Filter: The Kalman filter block performs the state of charge estimation of the non-linear battery model.

4.5.2 Block Parameter Definition

The major model constituting blocks are thus specified to achieve interoperability within the system model.

- Battery(Table-Based): The variables for the battery block are first created as a 'struct' named 'Battery', in the MATLAB workspace. The battery block parameters are then specified from the 'Battery struct' variables in the model workspace named 'Battery'.

- Band-Limited White Noise: A default noise power of 0.1 and sample time of 0.2 are specified for the noise block.

- Simulink Function: The state transition function is defined with the function name **'batteryStateFcn(x)'** while the measurement function is defined with the function name **batteryMeasurementFcn(x)**. The state transition function block implements equations (3.2), (3.3) as shown in figure 4.11. The variables for dynamic resistance (R_1) and time constant (**tau** = R_1C_1) are defined as look-up tables while the cell capacity is defined using a constant block.

  Similarly, the measurement function block, implements equation (3.6) as seen in figure 4.12. The variables for instantaneous resistance (R_0) and open circuit voltage (E_m), are also defined as look-up tables.

- Extended Kalman Filter: The EKF block parameters consists of the state transition and measurement sections which requires the calling of the state transition and measurement functions, using their function names. Also, additive process noise and measurement noise along with their corresponding covariance are respectively defined for the EKF block. In the block parameter setting section, the EKF signal data type and sampling time are specified. The EKF model design is shown in figure 4.13.

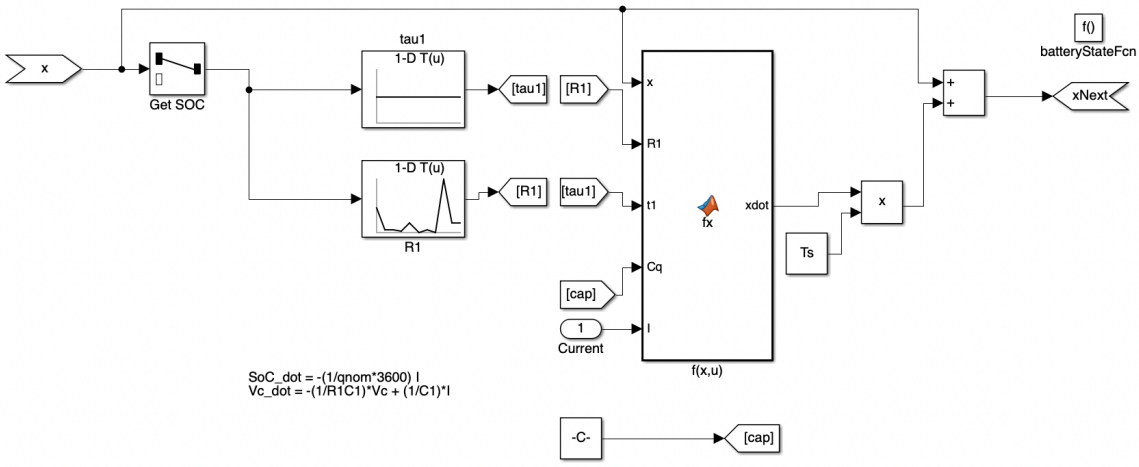

Figure 4.11: State Transition Function Block

4.5.3 Data Import and Model Configuration

There are two model inport blocks which allows the linking of the model to external data. The input signals are connected via the 'Data Import/Export' tab of the model configuration parameter dialog, which lunches the root-inport mapper tool where the signal import file is created and the signal scenario is specified for the model, with the help of the signal editor.

4.5.4 Model Simulation

The EKF model is set to simulate for a period of 2593.4 secs. This time value is the value of the last time data point available in the simulation data and is set to allow the simulation process to run for the entire data points. The simulation result is visualized using the scope.

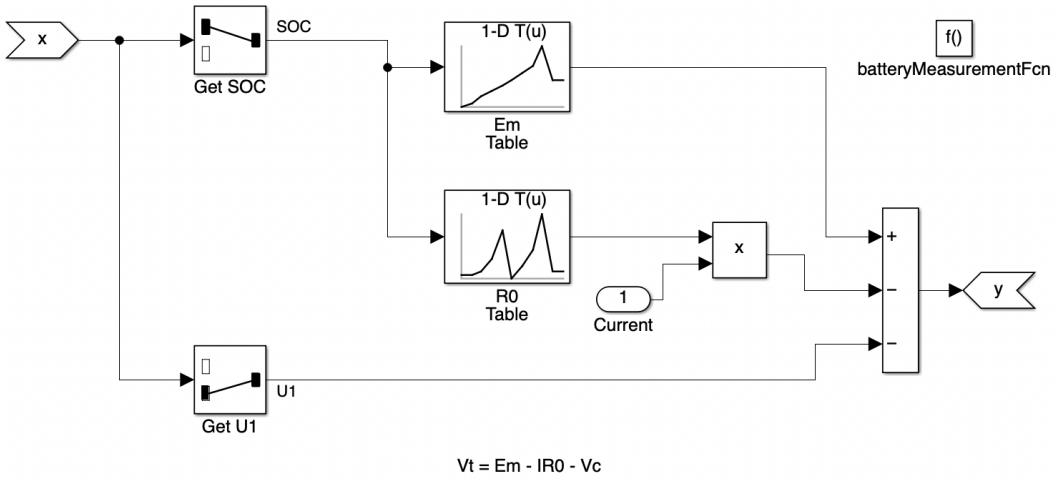

Figure 4.12: Measurement Function Block

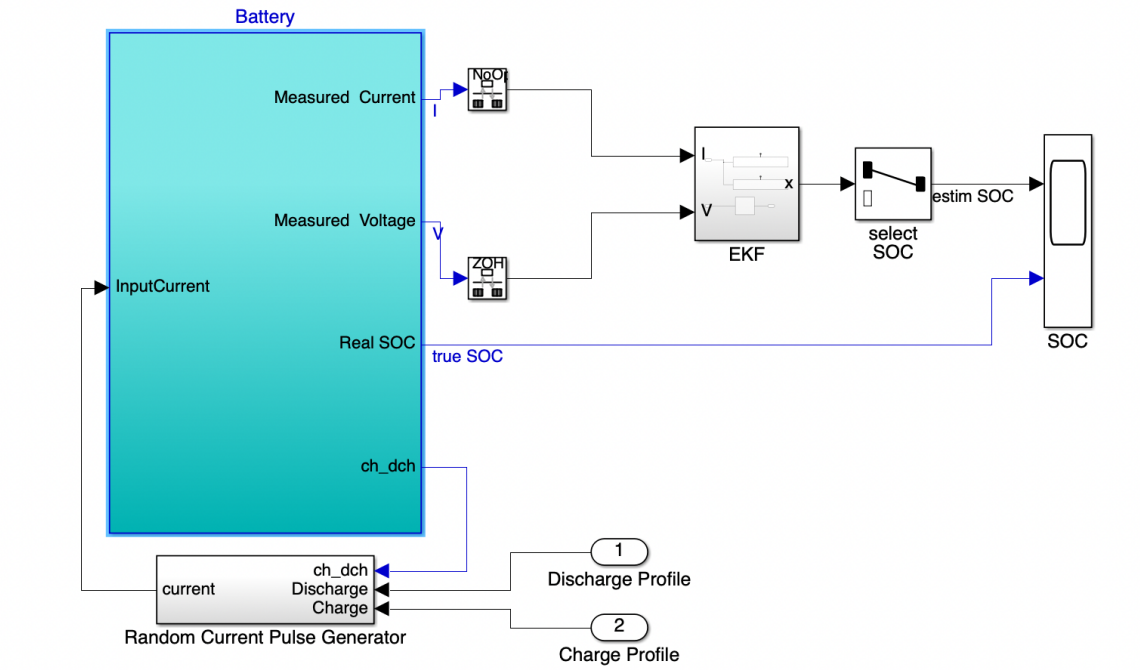

Figure 4.13: EKF Model Design

Chapter 5

Analysis of Results

In this chapter, the resulting plots from the Coulomb Counting estimation method, and the parameterization and estimation tasks for the EKF estimation method are presented and described. The results are analysed in terms of the initial SOC, convergence time and root-mean square error (RMSE) as measured on the plots.

5.1 Coulomb Counting Result

The figure 5.1, shows the estimation plot for the Coulomb Counting estimation method as visualised on the scope. The scope consists of three sub-plots showing the estimated SOC, true SOC (measured per unit) and input current (measured in Amps). The true SOC was obtained from the pre-parameterized battery reference block while the estimated SOC is obtained from the Coulomb Counting model. In both the estimated SOC and true SOC plots, identical results were obtained as a decreasing SOC of the cell is first observed as a response to the input pulse discharge current from initial 0.800 to 0.783 SOC for the estimated SOC plot and from initial 0.700 to 0.683 for the true SOC plot, in about 1526 seconds. The input signal then switches to pulse charge current which charges the battery from 0.783 to 0.797 for the

estimated SOC and 0.683 to 0.697 for the true SOC in 1067 seconds, summing up to a total simulation time of 2593.4 seconds. As with Coulomb Counting, a limitation is that the initial SOC must be known and specified for the estimation model to obtain accurate estimation, however it can be observe that once the initial SOC is not specified correctly, the entire SOC estimation will also be incorrect as there is no convergence observed between the estimated and true SOC plots, which means that the initial state error is propagated throughout the entire simulation time. From the signal statistics, the root mean square (RMS) value for the true SOC plot is 0.691 while the RMS value for the estimated SOC plot is 0.791, which evaluates to a RMSE of 0.1.

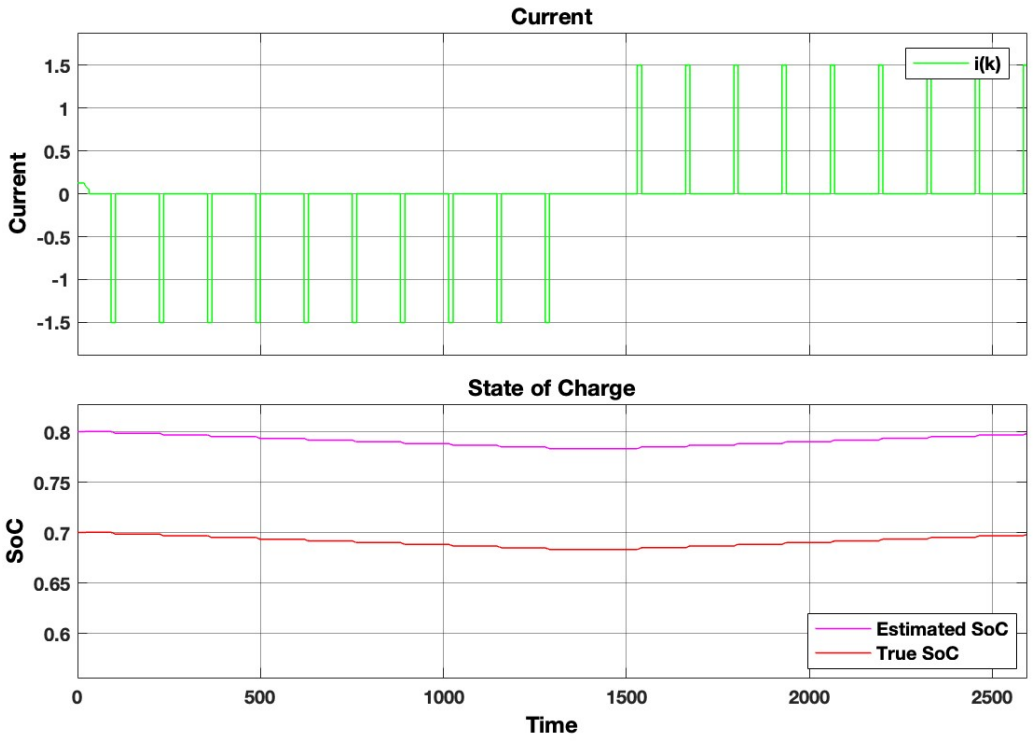

Figure 5.1: Coulomb Counting Estimation

5.2 Parameterization Result

The initial simulation of the parameterization experimental plot is shown in figure 5.2. It can be observed that the simulated voltage curve does not fit the measured voltage curve. This is because the initial guesses of the cell's equivalent circuit parameters specified for the simulated battery, does not correctly define the characteristics of the real battery which was tested. However, the initial guesses provide a reasonable starting point for the parameter estimation task. The error between the true and estimated voltage in the parameter estimation task may be minimize by performing a thorough analysis of the experimental pulse discharge plot with the aim of extracting best initial guess. Also, the error may be minimized by addition of time constants in the equivalent circuit model, which better describes the dynamic behavior of the battery.

The parameter estimation task was ran for a total of 5.8598 hours, covering initialization time, execution time and termination time. The parameter estimation iteration plot in figure 5.3 shows that the parameter estimation task was completed in a total of 3 iterations. The mapping of the parameter values from the initial guesses until convergence to the actual parameter values are also seen on the plot.

Having completed the parameter estimation task, the simulated voltage curve now fits the measured voltage curve as shown in figure 5.4, although the fitting is not very perfect. This implies that the estimated parameter values have converged to the actual parameter values (with some error). These resulting parameter values defines the battery model as possessing the fingerprint of the real battery of interest from which the pulse discharge test data was obtained. The parameter estimation result is saved for the battery model.

Now that the battery has been parameterized, the battery model is then simulated for 2593.4 seconds. The figure 5.5 features the voltage plot, the current plot and the SOC plot. The result in figure 5.5, correspondingly shows that the simulated voltage

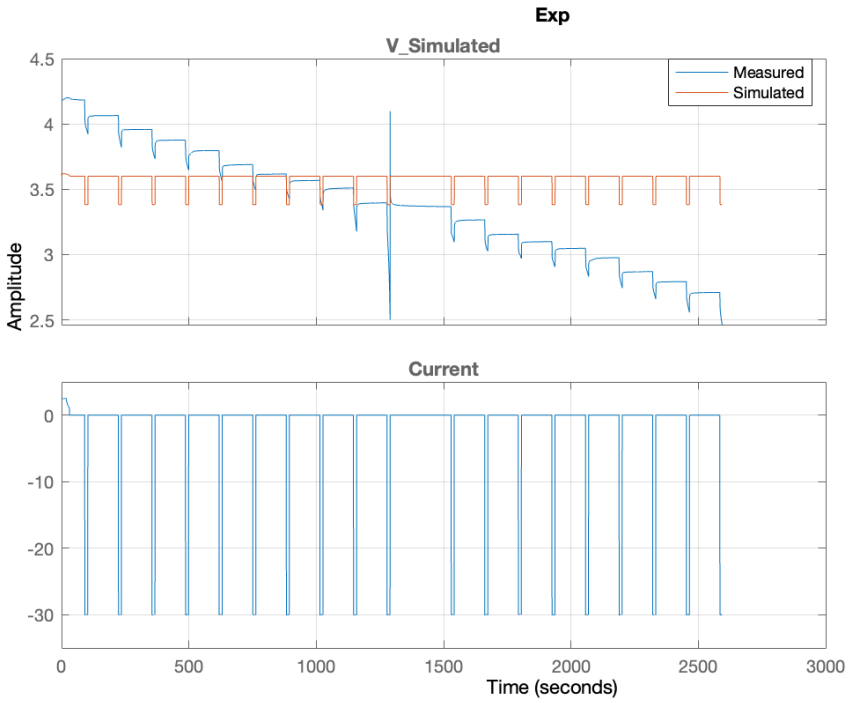

Figure 5.2: Experimental Plot before Parameterization

curve fits the measured voltage curve. This implies that the battery equivalent circuit model's voltage response to input current, is the same as in the case of the real battery in the experimental test and that the input current with an amplitude of (-)30 Amps, discharges the battery model from 0.7 SOC to 0.09 SOC. The parameterized equivalent circuit battery model can then be implemented with the extended Kalman filter for corresponding estimation.

5.3 Extended Kalman Filter Result

The resulting plot from the EKF SOC estimation is presented in figure 5.6. The plot consists of the true SOC curve and the estimated SOC curve. Firstly, by visual

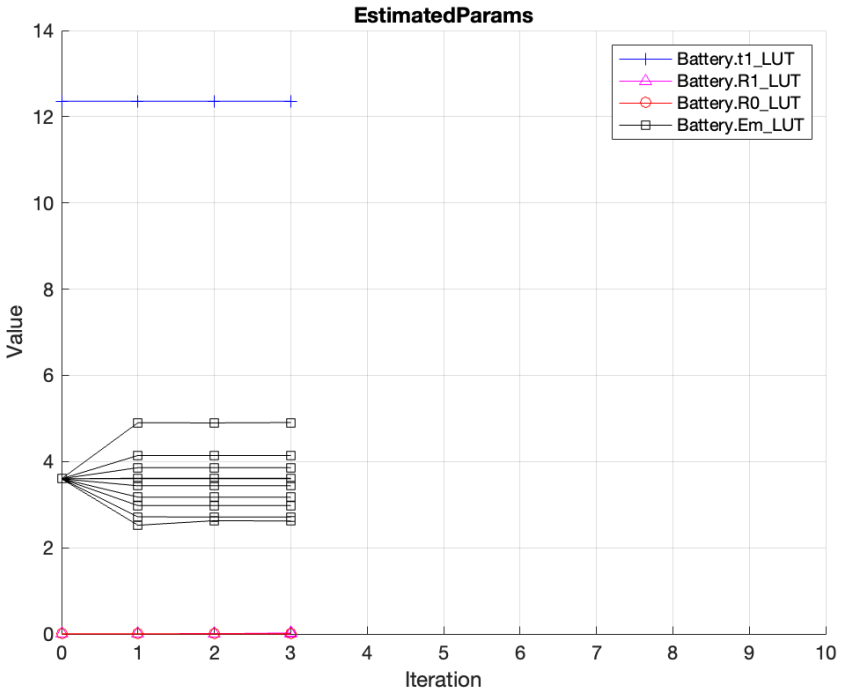

Figure 5.3: Parameterization Iteration Plot

inspection, it is observed that both curves are not only identical, but also shows convergence. By analysis of the resulting plot using the measurement and signal statistics, it is observed that the true SOC initializes at 0.7 and as the battery is in discharge mode, the SOC decreases to 0.3 as prescribed in the model design. Then the battery switches to charge mode and the SOC correspondingly rises to 0.517. The estimated SOC is comparatively observed to be initialized at 0.4 SOC. The extended Kalman filter model, tracks the true SOC until convergence in 223 seconds while the whole simulation is executed in 2593.4 seconds. Although the convergence between the estimated and true SOC shows that some distortion and minimized precision which can be described by the root mean square error (RMSE). However, optimal convergence is observed between times 2043.158 seconds and 2338.927 seconds which

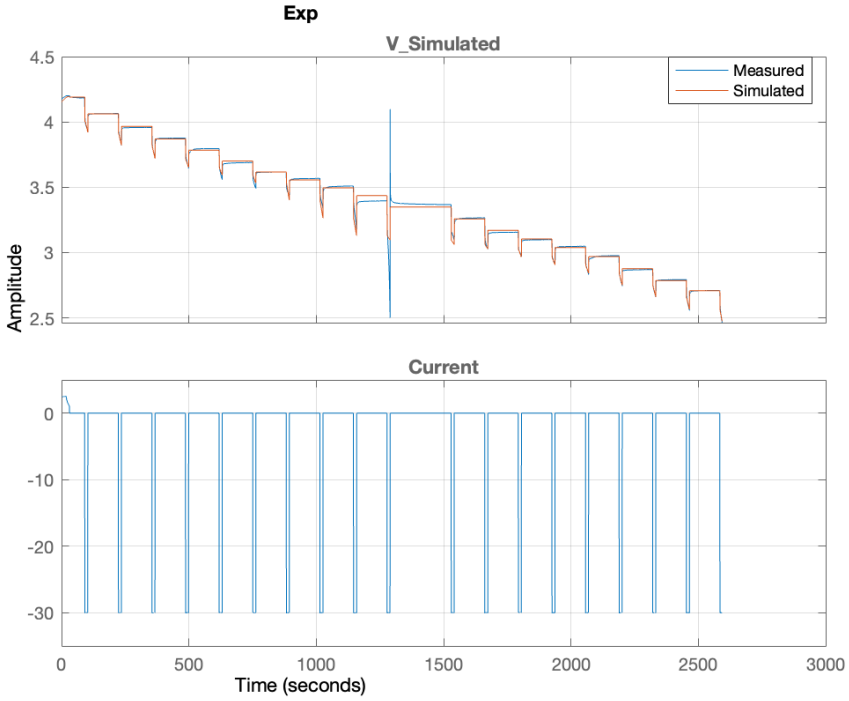

Figure 5.4: Experimental Plot after Parameterization

corresponds to the duration when the input charge switched from discharge to charge mode. The RMS of the true SOC is found to be 0.4814 while the RMS of the estimated SOC is found to be 0.4795. These values evaluates to RMSE of 0.0019.

Table 5.1: SOC Estimation Performance.

SOC Estimation Methods	Performance Metric
Coulomb Counting	RMSE = 0.1 Convergence Time = ∞ Simulation Time = 2593.4 seconds
Extended Kalman Filter	RMSE = 0.0019 Convergence Time = 223 seconds Simulation Time = 2593.4 seconds

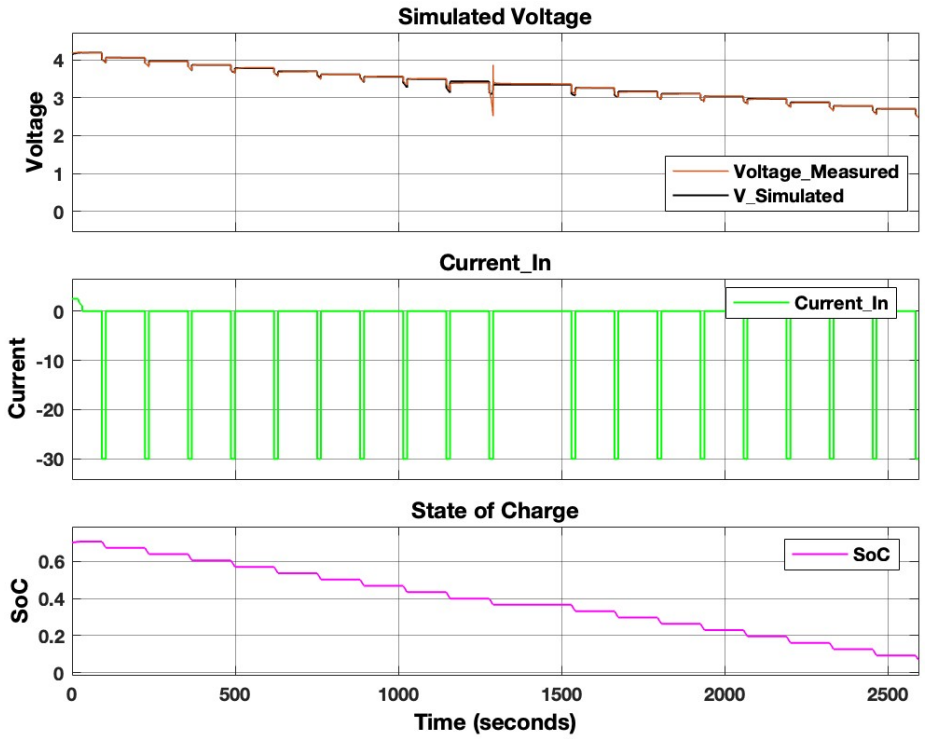

Figure 5.5: Equivalent Circuit Simulation Plot

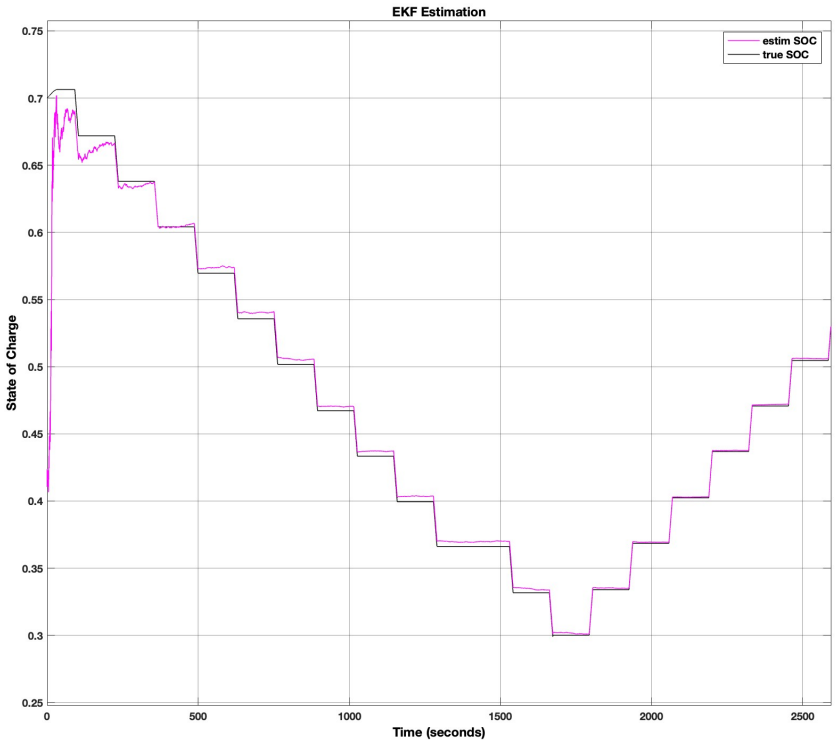

Figure 5.6: EKF SoC Estimation Simulation Plot

Chapter 6

Conclusion

This chapter presents an overview of the accomplishments and findings made in this research and also highlights some limitations of the investigated methods.

6.1 Conclusions

In this research, the concept of estimating the state of charge for lithium-ion cell chemistry is investigated. An in-depth review of literature on the topic of interest is first explored to establish an understanding of the research area and appreciate the research problem, in terms of the scope and gaps in the literature with objectives positioned towards a strategic solution. State of charge estimation was considered as a functional aspect of the battery management system while focusing on lithium-ion cells as the battery chemistry of interest.

Having reviewed various techniques for state of charge estimation for the lithium cell chemistry, and finding the gap of a detailed and comprehensive report in the literature, the research then focused on the comparative analysis of two methods which are the Coulomb Counting and Extended Kalman filter for SoC estimation. The research goes further to contribute a detailed report which attempts to close the

knowledge gap in the literature.

Models were developed for both methods in consideration and data was adopted from the University of Wisconsin Madison, McMaster University and the Battery Intelligence Laboratory at the University of Oxford for the NMC and NCA batteries respectively. After performing the modelling and simulations, the following conclusions were drawn towards answering the research questions.

- How can we know the true SOC to evaluate our estimators? The research showed that the true SOC was computed by parameterization of a battery model with laboratory measured data and using the battery model as the true SOC reference in comparison with estimated SOC from both the Coulomb Counting and Extended Kalman Filter models

- How can experimental data be acquired for SOC estimation? It was established that experimental data is acquired for SOC estimation through a controlled laboratory experimental setup

- What computing capabilities and tools are required for performing an SOC estimation task? In this research, it was established that the MATLAB/Simulink/Simscape software, has sufficient computing capability as a tool relevant for SOC estimation.

- What performance metrics are relevant for validating SOC estimation accuracy? The estimation methods considered in this research where compared based on the root mean square error and the convergence time.

- How does the Coulomb Counting estimation results compare to the Extended Kalman filter results? The results of the SOC estimation methods show that considering the tracking capability, convergence time and root mean square errors resulting from both estimation methods, it can then be concluded that the

Extended Kalman filter outperforms the coulomb counting method. However considering the need for parameterization, it can be infered that the EKF is however more computationally complex than the Coulomb counting method.

The battery industry is plagued with concerns of optimal battery management. This concern is centered around the battery's SOC. Some of the challenges arise and are seen in battery over-discharge and over-charge scenarios. To prevent such constraints, and ensure the reliability of battery systems, the battery management functionality of SOC estimation is a basic function that must be implemented in every power system application including EVs, that features the use of battery systems. When batteries are over-discharged, they suffer a loss in their energy storage and delivery capability and by extension degrades their reliability. A practical benefit of this work in the battery and EV industry is the improvement it brings in the aspect of battery reliability and integrity. Therefore ensuring and assuring the user of promised battery operating run-time and expected durability.

Bibliography

[1] "Global supply chains of ev batteries – analysis - iea," https://www.iea.org/reports/global-supply-chains-of-ev-batteries, (Accessed on 02/07/2023).

[2] C. Vidal, P. Malysz, P. Kollmeyer, and A. Emadi, "Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-art," *IEEE Access*, vol. PP, pp. 1–1, 03 2020.

[3] "Library genesis," https://libgen.li/file.php?md5=05d29af0e92fb0ff22f14de6507a088f, (Accessed on 02/07/2023).

[4] "Battery management systems, volume ii: Equivalent-circuit methods - gregory l. plett - google books," https://books.google.com.ng/books?id=1kSPCwAAQBAJ, (Accessed on 02/07/2023).

[5] "Ece5720: Battery management and control," http://mocha-java.uccs.edu/ECE5720/index.html, (Accessed on 09/15/2022).

[6] A. Caliwag, K. L. Muh, S. H. Kang, J. Park, and W. Lim, "Design of modular battery management system with point-to-point soc estimation algorithm," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 701–704.

[7] "Net zero by 2050 - a roadmap for the global energy sector - summary for policy makers," https://iea.blob.core.windows.net/assets/7ebafc81-74ed-412b-9c60-5cc32c8396e4/NetZeroby2050-ARoadmapfortheGlobalEnergySector-SummaryforPolicyMakers_CORR.pdf, (Accessed on 09/06/2022).

[8] "Ev battery health - what 6,000 batteries tell us — geotab," https://www.geotab.com/blog/ev-battery-health/, (Accessed on 09/10/2022).

[9] B. Bilgin, P. Magne, P. Malysz, Y. Yang, V. Pantelic, M. Preindl, A. Korobkine, W. Jiang, M. Lawford, and A. Emadi, "Making the case for electrified transportation," *IEEE Transactions on Transportation Electrification*, vol. 1, no. 1, pp. 4–17, 2015.

[10] "Knovel - battery management systems and inductive balancing," https://app.knovel.com/kn/resources/kpBMSIB003/toc, (Accessed on 02/07/2023).

[11] J. Turner, *3. Simplified Extended Kalman Filter Observer for SOC Estimation of Commercial Power-Oriented LFP Lithium Battery Cells (2013-01-1544)*, 2016, pp. 19–28.

[12] G. Plett, *Battery Management Systems, Volume II: Equivalent-Circuit Methods*, ser. Artech House power engineering and power electronics. Artech House, 2015. [Online]. Available: https://books.google.com.ng/books?id=1kSPCwAAQBAJ

[13] S. Chon, "What it takes to do efficient and cost-effective real-time control with a single microcontroller: The c2000™ advantage," *Texas Instruments*, 2011.

[14] N. Voigt, T. Wawer, and T. Albert, "Optimizing prosumers' battery energy storage management using machine learning," in *2019 16th International Conference on the European Energy Market (EEM)*, 2019, pp. 1–6.

[15] "Rolls battery user manual v7.2," https://rollsbattery.com/wp-content/uploads/2019/11/Rolls-Battery-User-Manual.pdf, (Accessed on 09/19/2022).

[16] C. Chibuzor Francis and J. Bertha Mwangama, "Accurate state of charge estimation for lithium iron phosphate battery cell using equivalent circuit model, parameter tuning and unscented kalman filter," in *2022 5th International Conference on Energy, Electrical and Power Engineering (CEEPE)*, 2022, pp. 199–204.

[17] "Battery management system — springerlink," https://link.springer.com/chapter/10.1007/978-3-662-53071-9_14, (Accessed on 02/07/2023).

[18] "Knovel - khtml viewer," https://app.knovel.com/web/view/khtml/show.v/cid:kt011FHFT1/viewerType:khtml//root_slug:batteries-electric-vehicles/url_slug:battery-models/?view=collapsed, (Accessed on 02/07/2023).

[19] "Knovel - systems approach to lithium-ion battery management," https://app.knovel.com/kn/resources/kpSALIBM02/toc, (Accessed on 02/07/2023).

[20] H. Beaty and B. Taylor, "Units, symbols, constants, definitions, and conversion factors," 01 2012.

[21] "What is the difference between positive and negative electrodes, anode and cathode: battery basics - biologic," https://www.biologic.net/topics/anode-cathode-positive-and-negative-battery-basics/, (Accessed on 10/18/2022).

[22] T. Huria, M. Ceraolo, J. Gazzarri, and R. Jackey, "High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells," in *2012 IEEE International Electric Vehicle Conference*, 2012, pp. 1–8.

[23] D. N. T. How, M. A. Hannan, M. S. Hossain Lipu, and P. J. Ker, "State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review," *IEEE Access*, vol. 7, pp. 136 116–136 136, 2019.

[24] F. Yang, X. Song, F. Xu, and K.-L. Tsui, "State-of-charge estimation of lithium-ion batteries via long short-term memory network," *IEEE Access*, vol. 7, pp. 53 792–53 799, 2019.

[25] W.-Y. Chang, "The state of charge estimating methods for battery: A review," *ISRN Applied Mathematics*, vol. 2013, 01 2013.

[26] M. Danko, J. Adamec, M. Taraba, and P. Drgona, "Overview of batteries state of charge estimation methods," *Transportation Research Procedia*, vol. 40, pp. 186–192, 2019, tRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146519301905

[27] J. Meng, M. Ricco, G. Luo, M. Swierczynski, D.-I. Stroe, A.-I. Stroe, and R. Teodorescu, "An overview and comparison of online implementable soc estimation methods for lithium-ion battery," *IEEE Transactions on Industry Applications*, vol. 54, no. 2, pp. 1583–1591, 2018.

[28] "Online full-parameter identification and soc estimation of lithium-ion battery pack based on composite electrochemical - dual circuit polarization modeling - iopscience," https://iopscience.iop.org/article/10.1088/1755-1315/675/1/012192, (Accessed on 02/07/2023).

[29] A. I. Ilieş, G. Chindriş, and D. Pitică, "A comparison between state of charge estimation methods: Extended kalman filter and unscented kalman filter," in *2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2020, pp. 376–381.

[30] V. Dao, M.-C. Dinh, C. Kim, M. Park, C.-H. Doh, J. Bae, M.-K. Lee, L. Jianyong, and B. Zhiguo, "Design of an effective state of charge estimation method for a lithium-ion battery pack using extended kalman filter and artificial neural network," *Energies*, vol. 14, pp. 1–20, 04 2021.

[31] V. Vaideeswaran, S. Bhuvanesh, and M. Devasena, "Battery management systems for electric vehicles using lithium ion batteries," *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, vol. 1, pp. 1–9, 2019.

[32] T. Huria, M. Ceraolo, J. Gazzarri, and R. Jackey, "Simplified extended kalman filter observer for soc estimation of commercial power-oriented lfp lithium battery cells," 04 2013.

[33] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for electric vehicle batteries using unscented kalman filtering," *Microelectronics Reliability*, vol. 53, p. 840–847, 06 2013.

[34] H. Shi, S. Wang, C. Fernandez, C. Yu, Y. Fan, and W. Cao, "Online full-parameter identification and SOC estimation of lithium-ion battery pack based on composite electrochemical - dual circuit polarization modeling," *IOP Conference Series: Earth and Environmental Science*, vol. 675, no. 1, p. 012192, feb 2021. [Online]. Available: https://doi.org/10.1088/1755-1315/675/1/012192

[35] H. Pan, Z. Lü, W. Lin, J. Li, and L. Chen, "State of charge estimation of lithium-ion batteries using a grey extended kalman filter and a novel open-circuit voltage model," *Energy*, vol. 138, 07 2017.

[36] K. Prasad and D. Bp, "Real time estimation of soc and soh of batteries," *International Journal of Renewable Energy Research*, vol. 8, pp. 44–55, 01 2018.

[37] D. Neoh, M. A. Hannan, M. S. Hossain Lipu, and P. J. Ker, "State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review," *IEEE Access*, vol. PP, pp. 1–1, 09 2019.

[38] J. Wehbe and N. Karami, "Battery equivalent circuits and brief summary of components value determination of lithium ion: A review," 04 2015, pp. 45–49.

[39] "Panasonic 18650pf li-ion battery data - mendeley data," https://data.mendeley.com/datasets/wykht8y7tg/1, (Accessed on 02/07/2023).

[40] G. dos Reis, C. Strange, M. Yadav, and S. Li, "Lithium-ion battery data and where to find it," *Energy and AI*, vol. 5, p. 100081, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546821000355

[41] T. Raj, "Path Dependent Battery Degradation Dataset Part 1," 2020, type: dataset. [Online]. Available: https://ora.ox.ac.uk/objects/uuid:de62b5d2-6154-426d-bcbb-30253ddb7d1e

[42] "Tabulated battery model - matlab," https://www.mathworks.com/help/sps/ref/batterytablebased.html, (Accessed on 12/07/2022).

[43] "Ideal current source driven by input signal - matlab," https://www.mathworks.com/help/simscape/ref/controlledcurrentsource.htmlsearchHighlight=controlled/20current/20source&s_tid=srchtitle_controlled/2520current/2520source_1, (Accessed on 12/07/2022).

[44] "Voltage sensor in electrical systems - matlab," https://www.mathworks.com/help/simscape/ref/voltagesensor.html?s_tid=doc_ta, (Accessed on 12/07/2022).

[45] "Create input port for subsystem or external input - simulink," https://www.mathworks.com/help/simulink/slref/inport.html?searchHighlight=inport&s_tid=srchtitle_inport_1, (Accessed on 12/07/2022).

[46] "Create output port for subsystem or external output - simulink," https://www.mathworks.com/help/simulink/slref/outport.html?s_tid=doc_ta, (Accessed on 12/07/2022).

[47] "Physical network environment and solver configuration - matlab," https://www.mathworks.com/help/simscape/ref/solverconfiguration.html?s_tid=doc_ta, (Accessed on 12/07/2022).

[48] "Display signals generated during simulation - simulink," https://www.mathworks.com/help/simulink/slref/scope.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[49] "Delay signal one sample period - simulink," https://www.mathworks.com/help/simulink/slref/unitdelay.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[50] "Add or subtract inputs - simulink," https://www.mathworks.com/help/simulink/slref/add.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[51] "Divide one input by another - simulink," https://www.mathworks.com/help/simulink/slref/divide.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[52] "Multiply and divide scalars and nonscalars or multiply and invert matrices - simulink," https://www.mathworks.com/help/simulink/slref/product.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[53] "Generate constant value - simulink," https://www.mathworks.com/help/simulink/slref/constant.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[54] "Multiply input by constant - simulink," https://www.mathworks.com/help/simulink/slref/gain.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[55] "Pass block input to from blocks - simulink," https://www.mathworks.com/help/simulink/slref/goto.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[56] "Accept input from goto block - simulink," https://www.mathworks.com/help/simulink/slref/from.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[57] "Select input elements from vector, matrix, or multidimensional signal - simulink," https://www.mathworks.com/help/simulink/slref/selector.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[58] "Convert physical signal into simulink output signal - matlab," https://www.mathworks.com/help/simscape/ref/pssimulinkconverter.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[59] "Convert simulink input signal into physical signal - matlab," https://www.mathworks.com/help/simscape/ref/simulinkpsconverter.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[60] "Combine input signals of same data type and complexity into virtual vector - simulink," https://www.mathworks.com/help/simulink/slref/mux.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[61] "Approximate one-dimensional function - simulink," https://www.mathworks.com/help/simulink/slref/1dlookuptable.html?s_tid=doc_ta, (Accessed on 12/10/2022).

[62] "Estimate states of discrete-time nonlinear system using extended kalman filter - simulink," https://www.mathworks.com/help/control/ref/ekf_block.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[63] "Graphically define a function with simulink blocks - simulink," https://www.mathworks.com/help/simulink/slref/simulinkfunction.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[64] "Include matlab code in models that generate embeddable c code - simulink," https://www.mathworks.com/help/simulink/slref/matlabfunction.html?s_tid=doc_ta, (Accessed on 12/09/2022).

[65] "Path dependent battery degradation dataset part 1 - ora - oxford university research archive," https://ora.ox.ac.uk/objects/uuid:de62b5d2-6154-426d-bcbb-30253ddb7d1e, (Accessed on 02/07/2023).

[66] "Global Supply Chains of EV Batteries – Analysis." [Online]. Available: https://www.iea.org/reports/global-supply-chains-of-ev-batteries

[67] S. Wang and D. Deng, "State-of-charge estimation of lithium-ion batteries using lstm and ukf," 06 2022.

[68] Z. Zhang, X. Zhang, Z. He, C. Zhu, W. Song, M. Gao, and Y. Song, "State of charge estimation for lithium-ion batteries using simple recurrent units and unscented kalman filter," *Frontiers in Energy Research*, vol. 10, p. 938467, 07 2022.

[69] M. A. F. Bossche, Alex Van den. Institution of Engineering and Technology (The IET), 2021. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpBMSIB003/battery-management-systems/battery-management-systems

[70] R. Dorn, R. Schwartz, and B. Steurich, *Battery management system*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 165–175. [Online]. Available: https://doi.org/10.1007/978-3-662-53071-9_14

Chapter 7

Appendix

The codes appended hereunder are snippets generated from the Simulink models for the Equivalen Circuit model with file name 'EqiCCT', the Coulomb Counting model with file name $'CC_Model_fin'$, and the Extende Kalman Filter model with file name 'SoCEstimation'. The code snippets describes the initialization program of the models. The full stack code for all the implemented models may be found in the google drive with link as follows;
https://drive.google.com/drive/folders/1NEHuDRnaaHYXRJ3vR2CoewJKA1J7bODe?usp=sharing

7.1 Equivalent Circuit Model Code (Initialization)

```
  /*
 * EqiCCT.c
 *
 * Academic License - for use in teaching, academic research, and meeting
 * course requirements at degree granting institutions only.  Not for
 * government, commercial, or other organizational use.
 *
```

```
 * Code generation for model "EqiCCT".
 *
 * Model version              : 1.33
 * Simulink Coder version : 9.8 (R2022b) 13-May-2022
 * C source code generated on : Wed Feb  8 12:23:19 2023
 *
 * Target selection: grt.tlc
 * Note: GRT includes extra infrastructure and instrumentation for prototyping
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "EqiCCT.h"
#include "rtwtypes.h"
#include <string.h>
#include <stddef.h>
#include "EqiCCT_private.h"
#include "rt_nonfinite.h"

/* Block signals (default storage) */
B_EqiCCT_T EqiCCT_B;

/* Continuous states */
X_EqiCCT_T EqiCCT_X;

/* Block states (default storage) */
```

```
DW_EqiCCT_T EqiCCT_DW;

/* External inputs (root inport signals with default storage) */
ExtU_EqiCCT_T EqiCCT_U;

/* External outputs (root outports fed by signals with default storage) */
ExtY_EqiCCT_T EqiCCT_Y;

/* Mass Matrices */
MassMatrix_EqiCCT_T EqiCCT_MassMatrix;

/* Real-time model */
static RT_MODEL_EqiCCT_T EqiCCT_M_;
RT_MODEL_EqiCCT_T *const EqiCCT_M = &EqiCCT_M_;



  /* initialize */
  {
    int_T i;
    real_T *f = EqiCCT_M->intgData.fac;
    for (i = 0; i < (int_T)(sizeof(EqiCCT_M->odeFAC)/sizeof(real_T)); i++) {
      f[i] = 1.5e-8;
    }
  }

  EqiCCT_M->intgData.DFDX = EqiCCT_M->odeDFDX;
  EqiCCT_M->intgData.W = EqiCCT_M->odeW;
```

```
EqiCCT_M->intgData.pivots = EqiCCT_M->odePIVOTS;
EqiCCT_M->intgData.xtmp = EqiCCT_M->odeXTMP;
EqiCCT_M->intgData.ztmp = EqiCCT_M->odeZTMP;
EqiCCT_M->intgData.M = EqiCCT_M->odeMASSMATRIX_M;
EqiCCT_M->intgData.M1 = EqiCCT_M->odeMASSMATRIX_M1;
EqiCCT_M->intgData.xdot = EqiCCT_M->odeXDOT;
EqiCCT_M->intgData.Edot = EqiCCT_M->odeEDOT;
EqiCCT_M->intgData.fminusMxdot = EqiCCT_M->odeFMXDOT;
EqiCCT_M->intgData.isFirstStep = true;
rtsiSetSolverExtrapolationOrder(&EqiCCT_M->solverInfo, 4);
rtsiSetSolverNumberNewtonIterations(&EqiCCT_M->solverInfo, 1);
EqiCCT_M->contStates = ((X_EqiCCT_T *) &EqiCCT_X);
EqiCCT_M->massMatrixType = ((ssMatrixType)3);
EqiCCT_M->massMatrixNzMax = (3);
EqiCCT_M->massMatrixIr = (EqiCCT_MassMatrix.ir);
EqiCCT_M->massMatrixJc = (EqiCCT_MassMatrix.jc);
EqiCCT_M->massMatrixPr = (EqiCCT_MassMatrix.pr);
rtsiSetSolverMassMatrixType(&EqiCCT_M->solverInfo, (ssMatrixType)3);
rtsiSetSolverMassMatrixNzMax(&EqiCCT_M->solverInfo, 3);
rtsiSetSolverData(&EqiCCT_M->solverInfo, (void *)&EqiCCT_M->intgData);
rtsiSetIsMinorTimeStepWithModeChange(&EqiCCT_M->solverInfo, false);
rtsiSetSolverName(&EqiCCT_M->solverInfo,"ode14x");
rtmSetTPtr(EqiCCT_M, &EqiCCT_M->Timing.tArray[0]);
rtmSetTFinal(EqiCCT_M, 2593.4);
EqiCCT_M->Timing.stepSize0 = 0.2;
```

7.2 Coulomb Counting Model Code (Initialization)

```
   /*
 * CC_Model_fin.c
 *
 * Academic License - for use in teaching, academic research, and meeting
 * course requirements at degree granting institutions only.  Not for
 * government, commercial, or other organizational use.
 *
 * Code generation for model "CC_Model_fin".
 *
 * Model version              : 2.6
 * Simulink Coder version : 9.8 (R2022b) 13-May-2022
 * C source code generated on : Wed Feb  8 11:40:40 2023
 *
 * Target selection: grt.tlc
 * Note: GRT includes extra infrastructure and instrumentation for prototyping
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objective: Debugging
 * Validation result: Not run
 */

#include "CC_Model_fin.h"
```

```
#include "rtwtypes.h"
#include <string.h>
#include <stddef.h>
#include "CC_Model_fin_private.h"
#include "rt_nonfinite.h"

/* Block signals (default storage) */
B_CC_Model_fin_T CC_Model_fin_B;

/* Continuous states */
X_CC_Model_fin_T CC_Model_fin_X;

/* Block states (default storage) */
DW_CC_Model_fin_T CC_Model_fin_DW;

/* External inputs (root inport signals with default storage) */
ExtU_CC_Model_fin_T CC_Model_fin_U;

/* External outputs (root outports fed by signals with default storage) */
ExtY_CC_Model_fin_T CC_Model_fin_Y;

/* Mass Matrices */
MassMatrix_CC_Model_fin_T CC_Model_fin_MassMatrix;

/* Real-time model */
static RT_MODEL_CC_Model_fin_T CC_Model_fin_M_;
RT_MODEL_CC_Model_fin_T *const CC_Model_fin_M = &CC_Model_fin_M_;
```

```
/* initialize */
{
  int_T i;
  real_T *f = CC_Model_fin_M->intgData.fac;
  for (i = 0; i < (int_T)(sizeof(CC_Model_fin_M->odeFAC)/sizeof(real_T)); i++)
  {
    f[i] = 1.5e-8;
  }
}

CC_Model_fin_M->intgData.DFDX = CC_Model_fin_M->odeDFDX;
CC_Model_fin_M->intgData.W = CC_Model_fin_M->odeW;
CC_Model_fin_M->intgData.pivots = CC_Model_fin_M->odePIVOTS;
CC_Model_fin_M->intgData.xtmp = CC_Model_fin_M->odeXTMP;
CC_Model_fin_M->intgData.ztmp = CC_Model_fin_M->odeZTMP;
CC_Model_fin_M->intgData.M = CC_Model_fin_M->odeMASSMATRIX_M;
CC_Model_fin_M->intgData.isFirstStep = true;
rtsiSetSolverExtrapolationOrder(&CC_Model_fin_M->solverInfo, 4);
rtsiSetSolverNumberNewtonIterations(&CC_Model_fin_M->solverInfo, 1);
CC_Model_fin_M->contStates = ((X_CC_Model_fin_T *) &CC_Model_fin_X);
CC_Model_fin_M->massMatrixType = ((ssMatrixType)1);
CC_Model_fin_M->massMatrixNzMax = (2);
CC_Model_fin_M->massMatrixIr = (CC_Model_fin_MassMatrix.ir);
CC_Model_fin_M->massMatrixJc = (CC_Model_fin_MassMatrix.jc);
CC_Model_fin_M->massMatrixPr = (CC_Model_fin_MassMatrix.pr);
```

```
rtsiSetSolverMassMatrixType(&CC_Model_fin_M->solverInfo, (ssMatrixType)1);
rtsiSetSolverMassMatrixNzMax(&CC_Model_fin_M->solverInfo, 2);
rtsiSetSolverData(&CC_Model_fin_M->solverInfo, (void *)
                  &CC_Model_fin_M->intgData);
rtsiSetIsMinorTimeStepWithModeChange(&CC_Model_fin_M->solverInfo, false);
rtsiSetSolverName(&CC_Model_fin_M->solverInfo,"ode14x");
rtmSetTPtr(CC_Model_fin_M, &CC_Model_fin_M->Timing.tArray[0]);
rtmSetTFinal(CC_Model_fin_M, 2593.4);
CC_Model_fin_M->Timing.stepSize0 = 0.2;
```

7.3 Extended Kalman Filter Model (Initialization)

```
   /*
* SoCEstimation.c
*
* Academic License - for use in teaching, academic research, and meeting
* course requirements at degree granting institutions only.  Not for
* government, commercial, or other organizational use.
*
* Code generation for model "SoCEstimation".
*
* Model version              : 7.20
* Simulink Coder version : 9.8 (R2022b) 13-May-2022
```

```
 * C source code generated on : Wed Feb  8 12:47:29 2023
 *
 * Target selection: grt.tlc
 * Note: GRT includes extra infrastructure and instrumentation for prototyping
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "SoCEstimation.h"
#include "rtwtypes.h"
#include "rt_nrand_Upu32_Yd_f_pw_snf.h"
#include <string.h>
#include <math.h>
#include "xnrm2_Talqnny2.h"
#include "rt_hypotd_snf.h"
#include "EKFCorrector_correctSt_OSoz5Fz4.h"
#include <emmintrin.h>
#include "xnrm2_2UnwFLhW.h"
#include "xgemv_Kyotrcue.h"
#include "xgerc_vy68e9U4.h"
#include <stddef.h>
#include "SoCEstimation_private.h"
#include "rt_nonfinite.h"
#include "batteryMeasurementFcn.h"
#include "batteryStateFcn.h"
```

```
/* Named constants for State Transition Table: '<S1>/State Transition Table' */
#define SoCEstimation_IN_Charge        ((uint8_T)1U)
#define SoCEstimation_IN_Discharge     ((uint8_T)2U)

/* Block signals (default storage) */
B_SoCEstimation_T SoCEstimation_B;

/* Continuous states */
X_SoCEstimation_T SoCEstimation_X;

/* Block states (default storage) */
DW_SoCEstimation_T SoCEstimation_DW;

/* External inputs (root inport signals with default storage) */
ExtU_SoCEstimation_T SoCEstimation_U;

/* Mass Matrices */
MassMatrix_SoCEstimation_T SoCEstimation_MassMatrix;

/* Real-time model */
static RT_MODEL_SoCEstimation_T SoCEstimation_M_;
RT_MODEL_SoCEstimation_T *const SoCEstimation_M = &SoCEstimation_M_;


  /* initialize */
  {
    int_T i;
```

```
    real_T *f = SoCEstimation_M->intgData.fac;
    for (i = 0; i < (int_T)(sizeof(SoCEstimation_M->odeFAC)/sizeof(real_T)); i++)
    {
      f[i] = 1.5e-8;
    }
  }

  SoCEstimation_M->intgData.DFDX = SoCEstimation_M->odeDFDX;
  SoCEstimation_M->intgData.W = SoCEstimation_M->odeW;
  SoCEstimation_M->intgData.pivots = SoCEstimation_M->odePIVOTS;
  SoCEstimation_M->intgData.xtmp = SoCEstimation_M->odeXTMP;
  SoCEstimation_M->intgData.ztmp = SoCEstimation_M->odeZTMP;
  SoCEstimation_M->intgData.M = SoCEstimation_M->odeMASSMATRIX_M;
  SoCEstimation_M->intgData.M1 = SoCEstimation_M->odeMASSMATRIX_M1;
  SoCEstimation_M->intgData.xdot = SoCEstimation_M->odeXDOT;
  SoCEstimation_M->intgData.Edot = SoCEstimation_M->odeEDOT;
  SoCEstimation_M->intgData.fminusMxdot = SoCEstimation_M->odeFMXDOT;
  SoCEstimation_M->intgData.isFirstStep = true;
  rtsiSetSolverExtrapolationOrder(&SoCEstimation_M->solverInfo, 4);
  rtsiSetSolverNumberNewtonIterations(&SoCEstimation_M->solverInfo, 1);
  SoCEstimation_M->contStates = ((X_SoCEstimation_T *) &SoCEstimation_X);
  SoCEstimation_M->massMatrixType = ((ssMatrixType)3);
  SoCEstimation_M->massMatrixNzMax = (3);
  SoCEstimation_M->massMatrixIr = (SoCEstimation_MassMatrix.ir);
  SoCEstimation_M->massMatrixJc = (SoCEstimation_MassMatrix.jc);
  SoCEstimation_M->massMatrixPr = (SoCEstimation_MassMatrix.pr);
  rtsiSetSolverMassMatrixType(&SoCEstimation_M->solverInfo, (ssMatrixType)3);
```

```
rtsiSetSolverMassMatrixNzMax(&SoCEstimation_M->solverInfo, 3);
rtsiSetSolverData(&SoCEstimation_M->solverInfo, (void *)
                  &SoCEstimation_M->intgData);
rtsiSetIsMinorTimeStepWithModeChange(&SoCEstimation_M->solverInfo, false);
rtsiSetSolverName(&SoCEstimation_M->solverInfo,"ode14x");
rtmSetTPtr(SoCEstimation_M, &SoCEstimation_M->Timing.tArray[0]);
rtmSetTFinal(SoCEstimation_M, 2593.4);
SoCEstimation_M->Timing.stepSize0 = 0.2;
```