12-31-2019

Study and modelling of lithium ion cell with accurate soc measurement algorithm using Kalman filter for electric vehicles

Kasthuriramanan Mahendravadi Sivaguru
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

STUDY AND MODELLING OF LITHIUM ION CELL WITH ACCURATE SOC MEASUREMENT ALGORITHM USING KALMAN FILTER FOR ELECTRIC VEHICLES

by

Kasthuriramanan Mahendravadi Sivaguru

Lithium Ion cells are preferred over lead acid cells for electric vehicles due to their energy density, higher discharge current and size. The cost of lithium ion cells is scaling down compared to ten years earlier, but as their performance characteristics increase, the need for safety and accurate modelling also increases.

The absence of a generic cell model is associated to the different makes of cells and different chemistries of Lithium ion cells behave differently under the testing conditions required for every unique application. The focus of this thesis will be on how to provide intelligence to the battery management system for calculating the state of charge of a cell so that the depth of discharge of the pack can be controlled, and to balance the voltage levels of all modules in a battery pack.

This will involve cycling of the chosen type of cell, modelling it for its parameters, analyzing the cycling data and choosing the perfect depth of discharge required for the application from the energy or capacity vs open circuit voltage (OCV) graph. The lithium ion model will be evaluated from the transient response of the battery pack. This will then be made as a working prototype on an electric vehicle car and its behavior studied practically.

STUDY AND MODELLING OF LITHIUM ION CELL WITH ACCURATE SOC MEASUREMENT ALGORITHM USING KALMAN FILTER FOR ELECTRIC VEHICLES

by
Kasthuriramanan Mahendravadi Sivaguru

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

December 2019

APPROVAL PAGE

STUDY AND MODELLING OF LITHIUM ION CELL WITH ACCURATE SOC MEASUREMENT ALGORITHM USING KALMAN FILTER FOR ELECTRIC VEHICLES

Kasthuriramanan Mahendravadi Sivaguru

Dr. Leonid Tsybeskov, Thesis Advisor — Date
Professor and Chair of Electrical and Computer Engineering, NJIT

Dr. Marek Sosnowski, Committee Member — Date
Professor and Chair of Undergraduate Studies of Electrical and Computer Engineering, NJIT

Dr. Durgamadhab Misra, Committee Member — Date
Professor and Chair of Graduate Studies of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Kasthuriramanan Mahendravadi Sivaguru

Degree: Master of Science

Date: December 2019

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2019

- Bachelor of Science in Electrical Engineering,
  Anna University, Chennai, India, 2016

Major: Electrical Engineering

ACKNOWLEDGMENT

I would like to thank Dr. Leonid Tsybeskov for giving me an opportunity to work on my dissertation and to also be able to implement my research on the NJIT Solar Car. I also thank Dr. Marek Sosnowski and Dr. Durgamadhab Misra for taking time out of their schedule to guide and oversee my thesis.

My gratitude extends to the IEEE Student Branch of New Jersey Institute of Technology for giving me access to their space, equipment that helped me continue with the research and their never-ending stock of ramen. I would like to express my special thanks to Brian Duemmer for helping me integrate my battery model with the NJIT Solar Car's firmware system even if it meant sleepless weekends to make sure the entire system was tested.

Also, a special thanks to Gaayathri Murugan for giving me the motivation and incentive to push through the difficult deadlines and odd work hours for the completion of my thesis.

I thank Dr Ashok Jhunjhunwala from the Indian Institute of Technology, Madras to have introduced me to the field of battery engineering and electric vehicles which has shaped my professional life ever since and given me something to strive towards. None of my academic accomplishments towards my master's degree would have been possible if not for the opportunity to work under the visionary he is.

A special thanks to my family, my father, mother, sister and uncle to have supported me in this venture through all the hardships and sacrifices. It has completely been a fruitful experience.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF SYMBOLS

V	Open Circuit Voltage
R0	Series internal resistance
C1	First order transient capacitance
C2	Second order transient capacitance
R1	First order transient impedance
R2	Second order transient impedance
V1	First order transient voltage
V2	Second order transient voltage
I	Cell Current
%	Percentage
ms	millisecond

LIST OF DEFINITIONS

Capacity	The quantitative ability of a lithium ion cell to store energy measured in Watt-hour or Ampere-hour.
C-Rate	The rate at which a lithium ion cell is charged/discharge with respect to its rated capacity.
Depth of Discharge	Amount of energy that has been depleted or acquired from a lithium ion cell with respect to its rated capacity
State of Charge	The amount of charge remaining in a lithium ion cell with respect to its rated capacity.
State of Health	The quantitative real time capacity of a cell with respect to its original rated capacity.
BMS	Battery Management System is the electronics hardware that monitors the status of the battery pack and shuts off the battery pack in case of fault status

CHAPTER 1

INTRODUCTION

1.1 Objective

Lithium ion battery packs have seen a sudden surge in the field of energy storage technology this 21st century. The reason being, the energy density, i.e., the capability to store energy is quantitatively higher in Lithium ion than the ones that were used like lead acid batteries for the application of electric vehicle. Advancements in electrochemistry have made the cost of lithium ion cells more affordable over the past decade. Designing the perfect lithium ion battery pack for electric vehicle is as difficult as achieving 100% efficiency. A lot of intricate detailing must be considered. The most important ones that end up deciding the longevity of the battery pack are as follows:

- Depth of Discharge
- Cell Balancing
- Charging/Discharging Current

The reason there is no generic cell model is different makes of cells and different chemistries of Lithium ion cells behave differently under the testing conditions required for every unique application.

The focus of this thesis will be on how to provide intelligence to the battery management system for calculating the state of charge of a cell so that the depth of discharge of the pack can be controlled and balance the voltage levels of all modules in a battery pack in case any module degrades faster than the rest.

This will involve cycling of the chosen type of cell, modelling it for its parameters, analyzing the cycled data and choosing the perfect depth of discharge required for the application from the energy or capacity vs open circuit voltage (OCV) graph. This will then be made as a working prototype on an electric vehicle and its behavior will be studied practically.

1.2 Background Information

State of Charge estimation is an extremely important concept concerned with energy storage devices. Living in the digital age, humans tend to base their entire schedule upon the state of charge of a device they use. It is the percentage of energy left in the energy storage system until which the device will run function normally. This depicts how accurate the estimated state of charge must be for a more efficient usage of our devices.

As far as Lithium ion battery packs are concerned, the depth of discharge (DoD) decided the life of the battery pack. The more we deplete the pack every time, the weaker the pack becomes, which in turns reduces the life of a cell. The role of the protection system for a lithium ion battery pack, the battery management system (BMS), is to make sure the battery pack does not over charge or deep discharge when the device is under working conditions.

Transients in DC system is not unheard of. Any circuitry that involves switching devices like converters, switched mode power supplies and motor drives automatically involve induced transients on its DC bus line. Inrush currents in heavy load applications are one of the most common examples of transients that can cause problems to a system. An example of how transients affects DC bus line is stated as follows.

A brushless DC motor controller is a full bridge controller that sends pulsating current to the motor phases for its operation. Due to this switching circuitry, the communication system which was run off the I2C protocol kept getting disrupted at the same frequency of the motor controller PWM frequency. On an attempt to find the source of this spikes, it was found that a 5V peak to peak transients originated at the cell terminals of the battery pack which was running the entire system. Even though being a pure DC source, the presence of transients brings questions about the equivalent model of a lithium ion cell.

The ideal lithium ion cell is a pure DC current source with a series internal resistance. But the actual model involves branches of RC circuits which cause the variation between open circuit voltage and closed-circuit voltage whenever the cell draws or supplies power. Hence, to improve the measurement accuracies, the cell model algorithm could be improved to accommodate the transient response during an application.

CHAPTER 2

STATE OF CHARGE ESTIMATION

2.1 Problem Statement

To make sure the BMS protects the battery pack from the external load, the state of charge measurement must be overly accurate. The common and simplest form of SOC measurement is by matching the open circuit voltage of the cell to the energy present with respect to the rated energy of the cell from a look up table. But this method does not consider the state of health (the degradation of the cell) of the battery pack.

For a more active way of measuring the SOC, equivalent circuit representation of a battery must be studied which involves a series impedance and a parallel capacitance all of which will be explained in detail as the thesis progresses with proper simulations and calculations. This will be done in a linear quadratic estimation technique, also known as a Kalman filter. SOC estimation using Kalman filter is a topic of research in most major institutions, either academic or industrial.

This thesis will also consider the behavior of the battery pack with passive balancing of cell modules adopted onto it to eventually develop the perfect battery pack and the steps to be followed to build one.

The case study for this thesis taken into consideration will be the concept of battery swapping technology of electric vehicles that many countries seem to adopt to overcome their lack of charging infrastructure, a detailed explanation of which is given below.

Battery swapping technology was tested out around 2010 in Asian countries such as China and Korea to compensate for their difficulty in designing an electric vehicle charging infrastructure due to cost, real estate and population parameters. Swapping a

battery pack also gave the customers and option of saving time spent in charging stations and gave option of slow charging a battery pack in bulk charging stations. Bulk chargers also had the advantage of charging through solar and other renewable energy sources rather than the conventional fossil fuel source.

As far as sustainability was concerned, swapping infrastructure was a huge win for green energy movement which played well with the reason given by the auto industry to justify the shift to electric vehicles from conventional IC engine vehicles. From a user point of view, even though the customer would be paying for energy they consume, the chances of getting a battery pack of equal health is comparatively less due to the rotational swapping system.

In any case, as customers, the driver needs to know how long a battery pack might last to plan the trip accordingly, especially since battery health is not linear or stable unlike their IC engine counterpart. This brings about a need for battery diagnostics that predict an accurate state of charge based on the battery state of health so that customers get an accurate estimate of how long (in terms of time or distance) their source of energy might last for.

2.2 Lithium Ion Equivalent Circuit

The need for an equivalent cell model of a lithium ion cell was raised when the operating voltage as a function of open circuit voltage was found to be nonlinear. The state of charge of a lithium ion cell is a very important parameter which decides the behavior of a battery pack. Even though the cut off thresholds of a cell is voltage dependent, the capacity of a cell is estimated using the state of charge. This brings about a need for an accurate measurement of this parameter. The state of charge varies non-linearly depending on the life of the cell, current drawn from/sourced to the cell and state of health that can estimate the remaining energy left in the source.

In 2014, Bing Chen from the Naval University of Engineering, Wuhan, China, considered the Lithium ion cell to be a constant current source with internal resistance because of its easiness to identify and compute the parameters. A third order transient circuit was considered for the polarization effect of the battery and Kirchoff's voltage law was used to compute the transient parameters [2].

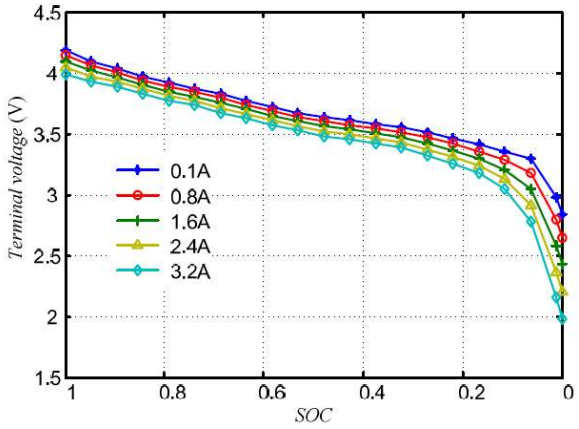

Figure 2.1 State of charge vs open circuit voltage characteristics to compute the relationship between transient cell model and charge/discharge rate.

Source: Bing Chen, Haodong Ma and Bin Fan "An Approach for State of Charge Estimation of Li-ion Battery Based on Thevenin Equivalent Circuit model".

The internal resistance was measured through electrochemical impedance spectroscopy and the corresponding closed-circuit voltage while discharging was measured. The relation between the third order Thevenin equivalent circuit and the open circuit voltage was plotted using the test data and the parameters were computed for various states of charge.

2.3 Transient Dynamic Model

Electric vehicle application requires a high rate of discharge of a cell. While most chemistries of lithium ion cells are restricted to a 2C rate of discharge, Lithium titanite oxide (LTO) can handle a high rate of discharge. Ana-Irina Stroe from Denmark characterized the equivalent cell model of an LTO rated at 13Ah through various test case scenarios ranging from 0.25C to 9C rate of discharge and charging [18].

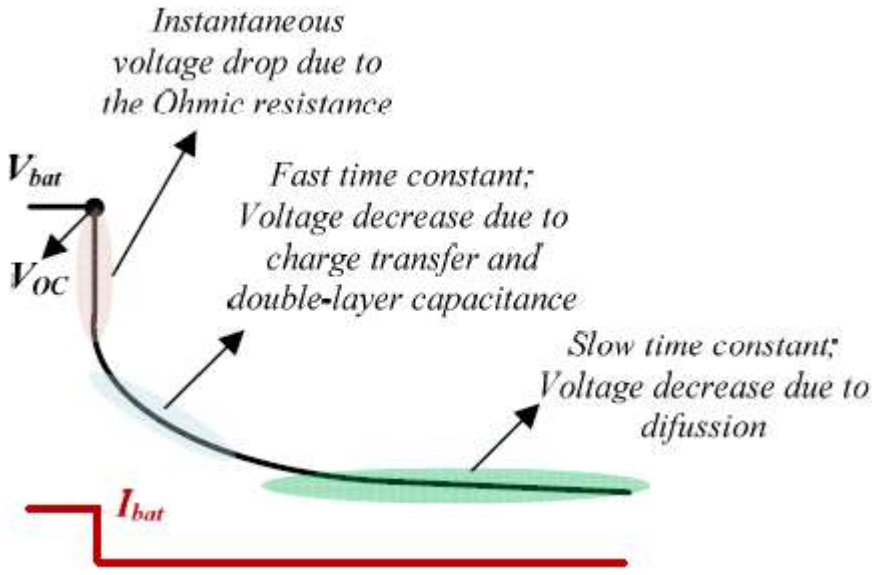

Figure 2.2 Transient response of a lithium ion cell to a discharge current pulse.

Source: Ana-Irina Stroe, Daniel-Ioan Stroe, Department of Energy Technology, Aalborg University "Lithium-Ion Battery Dynamic Model for Wide Range of Operating Conditions".

A second order lithium ion cell was modelled, and the transient circuits were determined by sending in pulsated current for charge and discharge and measuring the fall time/rise time of open circuit voltage with respect to the state of charge. An optimal RC network model was formed on MATLAB to minimize measurement errors.

An accurate model of a cell needs to be devoid of variations with test data. This flexibility is sometimes difficult to achieve in real time systems for electric vehicle applications. Rafael M. S. Santos from the Federal University of Para´ıba, Brazil sought out to the use of look up tables that determine the state of health of a lithium ion cell which will in turn develop the parameters on a cell for state of charge estimation. Instead of a linear function system, real time computation of the parameters was achieved by having an operating system control the charge and discharge rate and duration and measuring the transients of the cell voltage [16].

This system was run real time on a 20Ah cell for various state of charge ranges. One disadvantage with the method of look up tables is that the processing speed and power are limitations to the functioning system due to the number of values plotted on the look up table.

2.4 Energy Estimation

A study was conducted by IEEE members Kaiyuan Li, Feng Wei, King Jet Tseng, and Boon-Hee Soong on the ageing effects of lithium ion cells on the energy remaining. Cycle-ageing is the most dominant reason for reduction of capacity of a Li-ion cell. The aim of this study was to find the relation between the capacity degradation and number of cycles run on the cell. Cells of various state of health were chosen to be tested and were studied as to how the state of charge varied with the open circuit voltage [6].

The deviation from coulomb counting method of estimation brought about the need for an online state of charge estimator by changing the state space model of the cell with respect to the state of health of said lithium ion cell. This was achieved by the adapted unscented Kalman filter by using the look up tables from performed tests as input parameters and comparing them with the real time test case scenarios.

The proposed model was not efficient enough for a microprocessor on the battery management system to handle. The one drawback from the method used in this study is that the state of energy estimator relied on the energy consumed by the load while discharging and this is considered tedious since there are a lot of parameters that come into account like ambient temperature, initial state of charge and state of health.

2.5 State of Charge Estimation

Large battery packs of lithium ion require cell balancing algorithms to maintain pack stability. This brings about the need for an accurate state of charge measurement to define an algorithm function for balancing. Fan Zhang and Dragan Maksimovic from University of Colorado Boulder worked on a study of active balancing by estimating the state of charge using the sigma point Kalman filter which was implemented on a microcontroller for a 3-cell pack. The aim was to improve the health of a cell by implementing active cell balancing and still maintain the accuracy in state of charge estimation [24].

Xia Xiaohu and Wei Yun from Hefei University, China, proposed an algorithm for state of charge estimation using a combination of the Extended Kalman filter and the Interactive Multi Model filter. The simulation of the nonlinear system concluded that integrating the multi model filter resulted in a system with better accuracy than just using the Extended Kalman Filter [21].

Dr Gregory L Plett from the University of Colorado Boulder worked on an advanced battery management system algorithm that predicts real time state of charge by taking into consideration the primary functions of a battery management system for a hybrid electric vehicle the results of which will be considered in developing a real time BMS algorithm for the SOC estimation [12].

The method adopted for this study of modelling a cell is from the Volume 1 of Gregory L Plett's Battery Management System that deals with explaining the mathematical model of the lithium ion cell with a clear explanation of the transient network [13].

From the above literature surveys, it has been observed that state of charge estimation using open circuit voltage does not take into consideration the relaxation time of a nonideal cell. The transients observed during the relaxation time while charging or discharging can be used to compute the state of health of a lithium ion cell. The reason for opting to model a transient network is that the relation between open circuit voltage and the capacity of the cell is dynamic and changed over change in state of health.

CHAPTER 3

IMPLEMENTATION

3.1 Testing Requirements

The lithium ion cell to be used for this study is the LG MJ1 18650 type 3500mAh cell with the cell chemistry of Nickel Manganese commonly known as the NMC type lithium ion cell with a nominal voltage of 3.6. The minimum cell voltage is 3.00V and maximum voltage is 4.2V with a deep discharge voltage of 2.5V.

The LG MJ1 can handle a discharge rate of 2C and a charge rate of 1C and this specification for a lithium ion cell is commonly used for electric vehicle application bring cost considerate as well. The LG MJ1 come with equipped PTCs which add on to the cell internal resistance rated at 70mohms. The internal resistance is set to be non-linear and varies with respect to time and discharge rate.

3.2 Methodology

The method of coulomb counting is considered one of the most efficient ways to measure the energy capacity of a cell, but it is useful only when the application has a fixed charge and discharge rates. Electric vehicle applications have an erratic rate of discharges, although charging rate is usually fixed. The high rate of discharge makes it difficult to estimate health of a cell since it degrades if the lithium ion cell is abused by high discharge current.

In this study, the higher-level functioning SOC estimation algorithm will have two models for the cell. The first model is the equivalent cell circuit which comprises of the RC transient networks which give us an estimate of the variation in open circuit voltage.

The modelling will be done by charging and discharging the cell by using pulses of current and build a second order RC network by measuring the relaxation time.

The internal impedance measured with an electronic spectrometer will be used to determine the series internal resistance for the simulation. The input parameters for the model will be the state of health and the current rate.

The second model will be the state of health model of the cell which will be updated real time based on the total capacity during each charge cycle. Since the charging rate is controlled, the battery management system will update the model with coulomb counting and residual energy left in the battery pack before charging and estimate the state of health.

The state of charge estimation algorithm will be performed using Texas Instruments TM4C series of microcontroller which has enough processing power for the complex state space equation and state update equation computations which will estimate the SOC of the cell.

For cycle testing, a custom designed cell tester is used which can handle up to 5A during charging and 10Amps discharge rate. A synchronous closed loop buck converter with an input of 12V and an output of 5V is used for the charger circuit. The charging current is varied by varying the switching frequency of the buck converter. To reduce ripple on the output of the charger, a snubber circuit is added so that the switching transients does not affect the lithium ion cell.

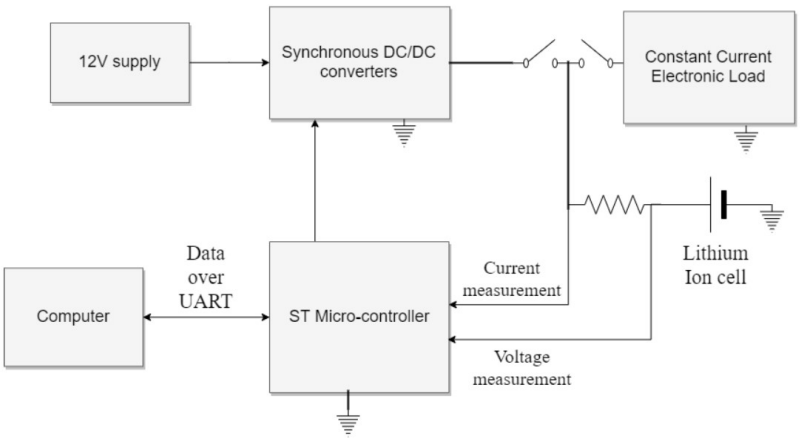

Figure 3.1 Block diagram of single channel cell tester for cycle testing.

The ST Microcontroller reads the current and voltage of the cell and communicates with a computer via Universal Asynchronous Receiver/Transmitter in the form of comma separated values along with a time stamp. This data is used as the input parameters for the mathematical cell modelling on MATLAB.

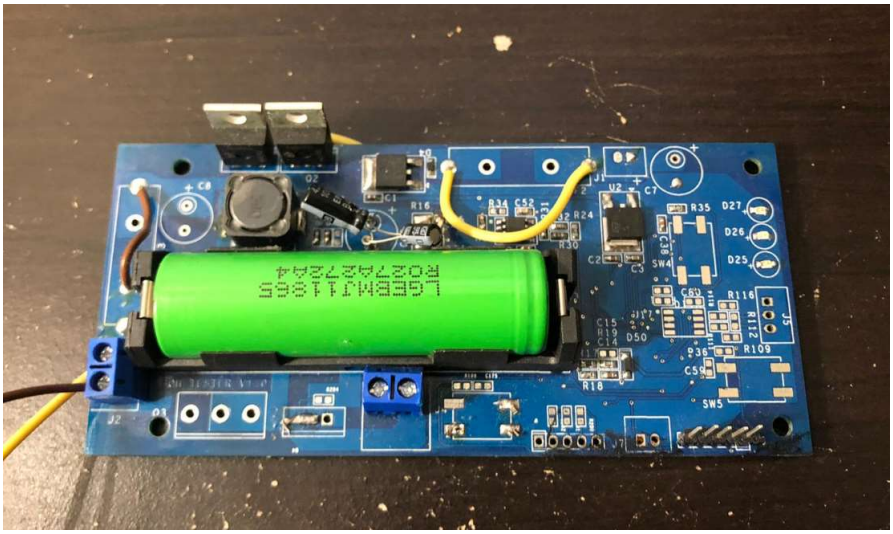

Figure 3.2 30 Watts single channel Lithium Ion cell tester.

3.3 Mathematical Modelling and Circuit Analysis

The lithium ion cell equivalent circuit consists of RC network. For this study, we will consider a second order functional state space equation. This means that we will consider 2 sets of parallel RC network as shown in the figure below.

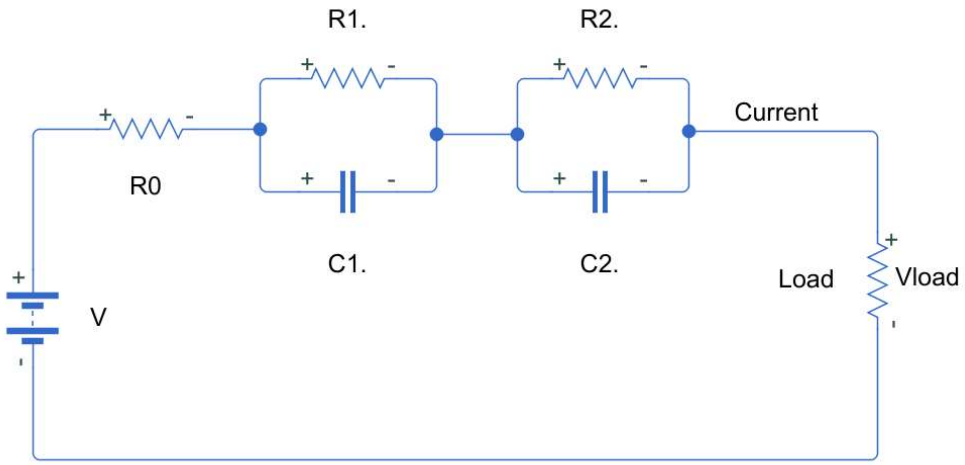

Figure 3.3 Second degree R-C network equivalent cell model of a lithium ion cell schematic representation.

The output voltage equation of this network is given by,

$$V = (R_0 \times I) + V_2 + V_1 + V_{load} \tag{3.1}$$

The exogenous inputs for this system are the RC network values, current and load voltage. The current and voltage and obtained from the battery management system's measurement circuit. The passive element values of the RC network are obtained from the relaxation time monitored by the BMS.

$$I = C_1 \frac{dV_1}{dt} + V_1 R_1 \tag{3.2}$$

$$I = C_2 \frac{dV_2}{dt} + V_2 R_2 \tag{3.3}$$

3.4 Energy Estimation

To understand the behavior of cycle testing and the relation between voltage vs watthour and voltage vs state of charge, initial cycle testing was done, and the test data was acquired through a digital multimeter. The first cycle testing involved charging and discharging the cell at the rate of 0.5Amperes from 3.1V to 4.1V and the voltage and current was logged every 5 seconds. The energy stored/received from the cell is calculated by the formula below.

$$Energy\ (\ Watt - hour) = \sum Voltage(volts) \times Current(amperes) \qquad (3.4)$$

This method of calculating the energy stored in a lithium ion cell is called coulomb counting. The summation of energy at every instant of the measurement frequency in ampere-hour gives the total accumulated value of energy stored in the cell. The curve between the energy with its respective voltage is called the Voltage vs Watt-hour curve. This is usually used to find the operating voltage limits for any application where the maximum energy can be sourced. A higher slope yields the best operating limits for a cell.

The total energy calculated was 11.988 Watt-hour. This value is going to be approximated to 12 Watt-hour for testing purposes. The LG MJ1 18650 cell was found to store 12 watt-hours of energy at 100% state of health. Below is the plot between energy and watt-hour. In this case, the highest linear slope is found between 3.42V to 4.07V which holds the maximum amount of energy.

Since the test was done on a cell at its 100% health, this data will be processed on MATLAB as the ideal cell behavior and the initial state space model for state of charge estimator.

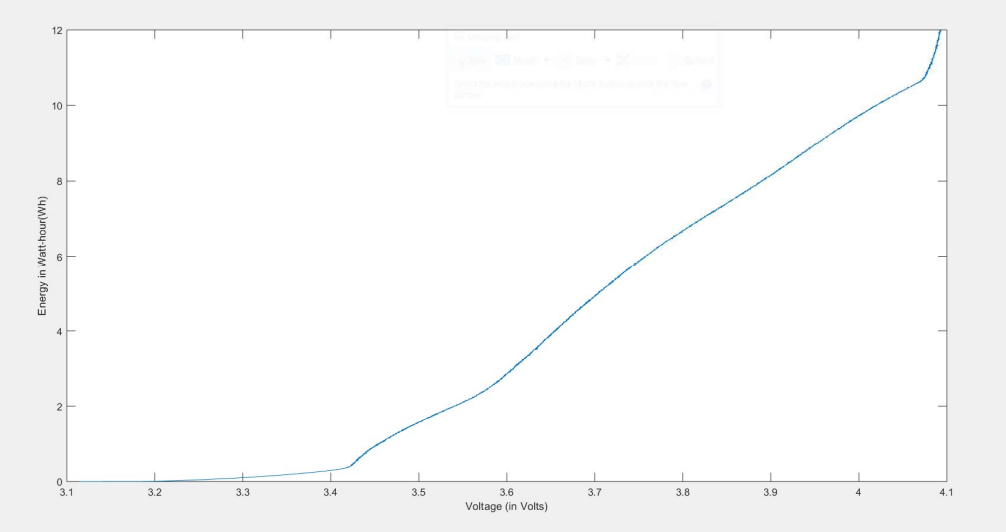

Figure 3.4 Energy vs open circuit voltage of 100% SOH test cell

The above method was repeated and after 'n' number of cycles, the capacity test was computed. The reason to consider the number of cycle an unknown variable is that we are interested in the decline of health of a cell in any cycle than estimate the number of cycles left because the rate of discharge affects the total life cycle in a nonlinear way. After 'n' number of cycles, the total capacity of the cell was found to be 9.514209107 Watt-hour. For testing purposes, this value is approximated as 9.5Watt-hour.

For the same depth of discharge of 3.1V to 4.1V, the ability of the lithium ion cell to store energy reduced from 12 watt-hour to 9.5 watt-hour. Considering 12 watt as full capacity of the cell, there was a 20% decline in energy storage capacity. The state of health of a battery is approximated at 80% by the given formula,

$$State\ of\ Health\ (\%) = \frac{Total\ Energy\ (from\ coulomb\ counting)}{Rated\ Capacity} \times 100 \qquad (3.5)$$

It was noticed that the decline of health did not affect the operating voltage limits of the cell. Although this did affect the linearity of the voltage vs watt-hour curve.

3.5 State of Charge Estimation

The traditional method to compute the state of charge of cell is by plotting the state of charge vs voltage curve and embedding the characteristic equation to the firmware of the BMS which matches the lowest cell voltage of a pack to a look up table to its corresponding state of charge value. The state of charge is estimated by the following equation,

$$State\ of\ Charge\ (i) = \frac{Energy\ (i)}{Rated\ Capacity} \times 100 \tag{3.6}$$

The variable 'i' stands for the duration of how long the cell has been charged or discharged stands for the energy calculated using coulomb counting for the voltage at the given time. This set of state of charge value is then plotted and the characteristic equation goes on to define the look up table and estimate the state of charge. The Voltage vs State of Charge curve for the cell at 100% state of health is as follows,

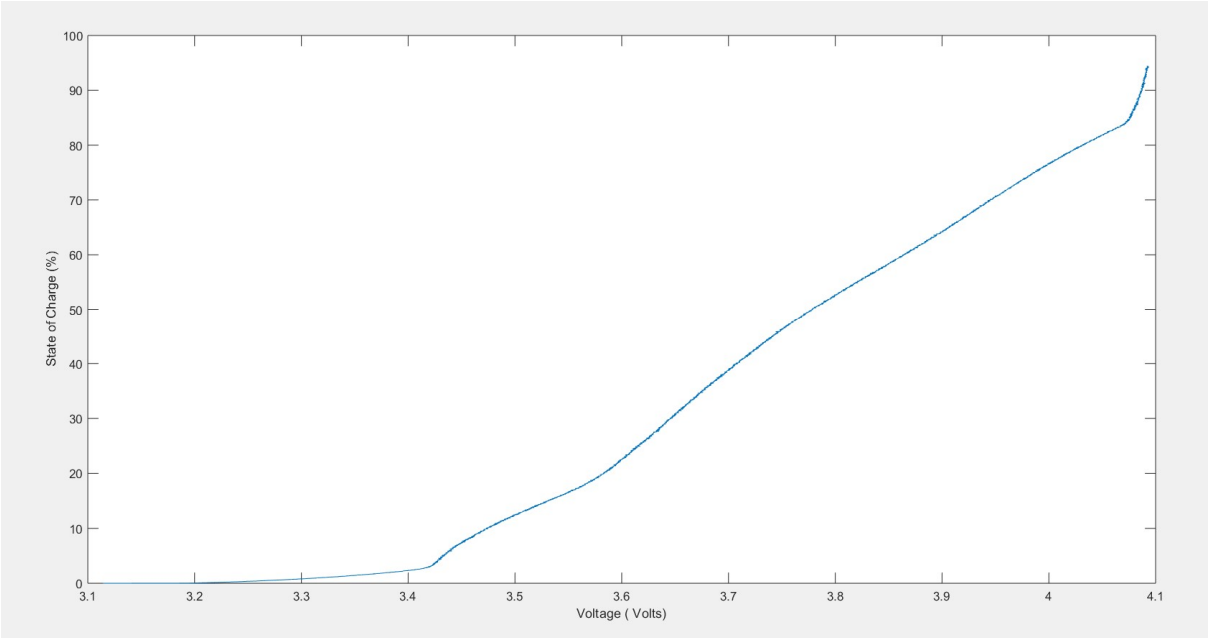

Figure 3.5 State of Charge vs open circuit voltage of 100% SOH test cell.

Nominal methods adopted by products that implement a lithium ion cell and the ones where measurement accuracy is trivial follow a method called a state of charge vs open circuit voltage where the microcontroller is fed a look up table data of voltage versus corresponding state of charge from coulomb counting test. This data from the table is then matched to a quadratic equation generally knowns as the characteristic equation of a lithium ion cell. The characteristic curve equation for the cell tested at 100% state of health is found to be

$$SOC\% = -20x^3 + 230x^2 - 850x + 1000 \quad (3.7)$$

The variable x stands for the open circuit voltage value read by the battery management system. The curve equation was verified by applying voltage values and checking the look up table of the test data. This method of state of charge estimation is known as coulomb counting. Unfortunately, this equation does not hold up for the voltage vs SOC curve of the cell at 80% SOH. This is since the energy storage capacity of the cell has reduced by 20%. This is corroborated by the voltage vs energy curve.

The linearity of the curve had reduced, and the curve had a new characteristic equation which went on to define the state of charge. Ideally, a system should be intelligent enough to update the state of charge estimation equation in real time depending on the state of health. This is not possible by the method of coulomb counting. Further test data will provide an error percentage in state of charge between the 100% SOH cell and the 80% SOH cell tested under the same conditions.

This brings about a need for a different method to estimate state of charge. The algorithm needs to be sophisticated enough to calculate the state of health of a cell and update the parameters of the characteristic equation. This is done by forming the relation

between voltage, current, energy and state of health. It is not viable to run cycle test on a battery pack under use to estimate the state of health and does not take into consideration the relaxation time of a cell.

The state of health can be determined by observing the transients of the lithium ion cell. The equivalent circuit of the cell parameters varies over time, which end up determining the state of health of the cell. Hence, by updating the equivalent RC-network of a cell, the state of health and the state of charge characteristic equation can be estimated. In later stages, the paper will discuss about the control algorithm required to implement Kalman filtering to update the state space equation of the initial conditions of the cell.

3.6 Transient Parameters

The estimation of the RC network inputs to the state space equation will involve testing the cell through pulses of current. The proposed methodology to obtain the transient curve is to start at t=0 mins, until t=5 mins cell remains at rest; t=5 mins, cell starts discharging at 0.5C for 15 mins, t=20 mins, discharge cut off, t=20 to 25 mins, cell rest. The same is continued and the ‘rest’ period voltage curve is captured on an oscilloscope to measure the time constant.

Hysteresis in the current response of a lithium ion cell takes place when a pulsated current is sourced or loaded from a lithium ion cell, the voltage waveform does not follow the rectangular wave pulsated wave form. It is noticed that the cell voltage undergoes hysteresis due to the capacitance of the materials that undergo chemical reaction. This hysteresis loss brings about a loss of accuracy in voltage measurements.

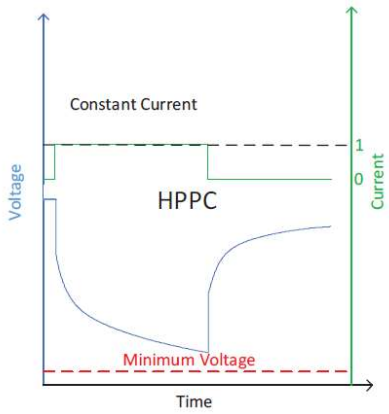

Figure 3.6 Voltage characteristics with time for a discharge current pulse of a lithium ion cell

Source: W. Wang, P. Malysz, K. Khan, L. Gauchia and A. Emadi, "Modeling, parameterization, and benchmarking of a lithium ion electric bicycle battery," 2016 IEEE Energy Conversion Congress and Exposition (ECCE), Milwaukee, WI, 2016, pp. 1-7.

A transient analysis on the voltage waveform gives us a better picture as to what causes the hysteresis.

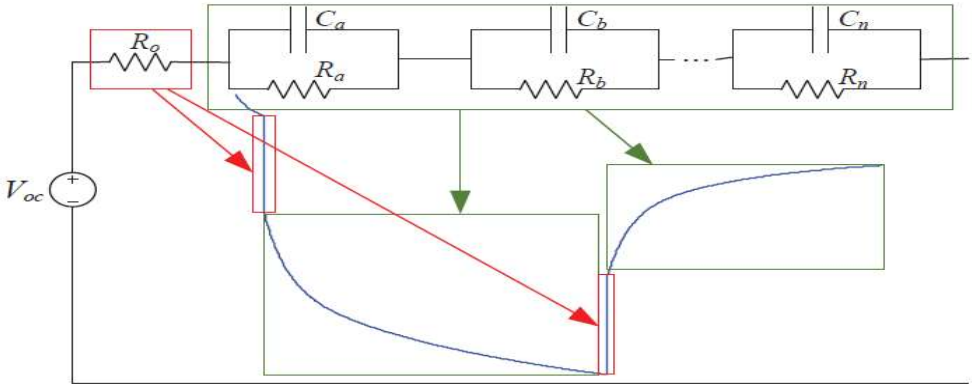

Figure 3.7 Transient analysis of open circuit voltage wave form for a current pulse

Source: W. Wang, P. Malysz, K. Khan, L. Gauchia and A. Emadi, "Modeling, parameterization, and benchmarking of a lithium ion electric bicycle battery," 2016 IEEE Energy Conversion Congress and Exposition (ECCE), Milwaukee, WI, 2016, pp. 1-7.

From the above figure, it is noticed that once flow of current is engaged, there is an instantaneous drop in the cell voltage. This is due to the series internal resistance that is caused majorly due to the presence of a positive thermal coefficient protection that adds up the internal impedance. The RC networks are responsible for the time constants for the voltage drop to saturate. In this study, we consider a 2 RC network to account for a slow time constant and a fast time constant.

The fast time constant occurs when charge transfer occurs from one electrode to the other which gradually builds up a double layer capacitance across the membranes of the electrolyte. This effect is associated to a fast time constant because of the continuous discharge of electrons from the cell.

During the off period, it is noticed that when the flow of current is disrupted, there is an instantaneous voltage rise which is evidently because of internal series resistance. Since there is no charge transfer happening, there is a steady voltage rise which is due to the diffusion of electrons of the chemical reaction in the lithium ion cell until the voltage saturates. This effect is associated with a slow time constant due to the time taken for the hysteresis to saturate [18].

The time constant is determined by the battery management system by measuring the voltage of individual cell with an accuracy of 10mV and a sampling period of 10ms. The input to the Kalman filter will be the fast and slow time constant which will be used to compute the transient parameters such as the RC-network and in turn, forming the equivalent circuit of the lithium ion cell.

Table 3.1 Coefficient of Time Constant

SOC%	90	70	60	50
R0	0.0184	0.0127	0.0208	0.0110
R1	0.0025	0.0024	0.0033	0.0041
R2	0.0035	0.0029	0.0038	0.0041
T1	1527	1562	2580	1785
T2	147.1	146.9	199.4	209.1
C1	599060	656580	781460	435150
C2	41825	50533	52192	51572

Source: R. M. S. Santos, C. L. G. d. S. Alves, E. C. T. Macedo, J. M. M. Villanueva and L. V. Hartmann, "Estimation of lithium-ion battery model parameters using experimental data," 2017 2nd International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT), Fortaleza, 2017, pp. 1-6.

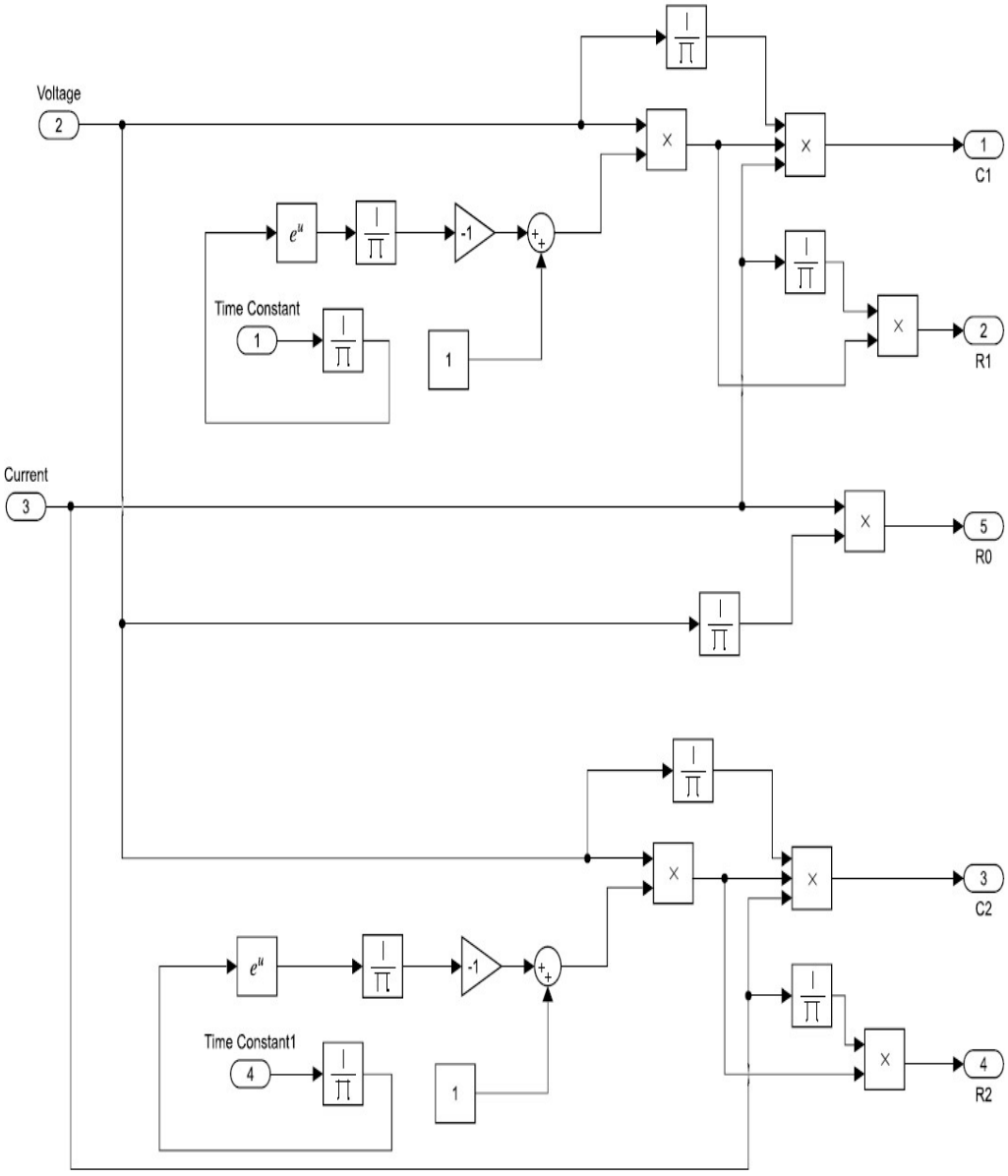

Figure 3.8 MATLAB model for computation of 2nd degree RC network parameters of a lithium ion cell.

The transient parameters characteristics are computed from the lithium ion cell model for a fixed time constant. The time constants are chosen for a total of 500ms being constrained at the operational time taken by the microcontroller. From initial cell tests of a 100% state of health lithium ion cell, the ratio between T1 and T2 was noticed to be 30:70 and hence for the comparison of transient parameters at a range of state of charge (3.2V to 4.2V), the cell parameters were computed. It is observed that the relation between the transient parameters and open circuit voltage (which can be extrapolated as state of charge) is linear for a fixed state of health.

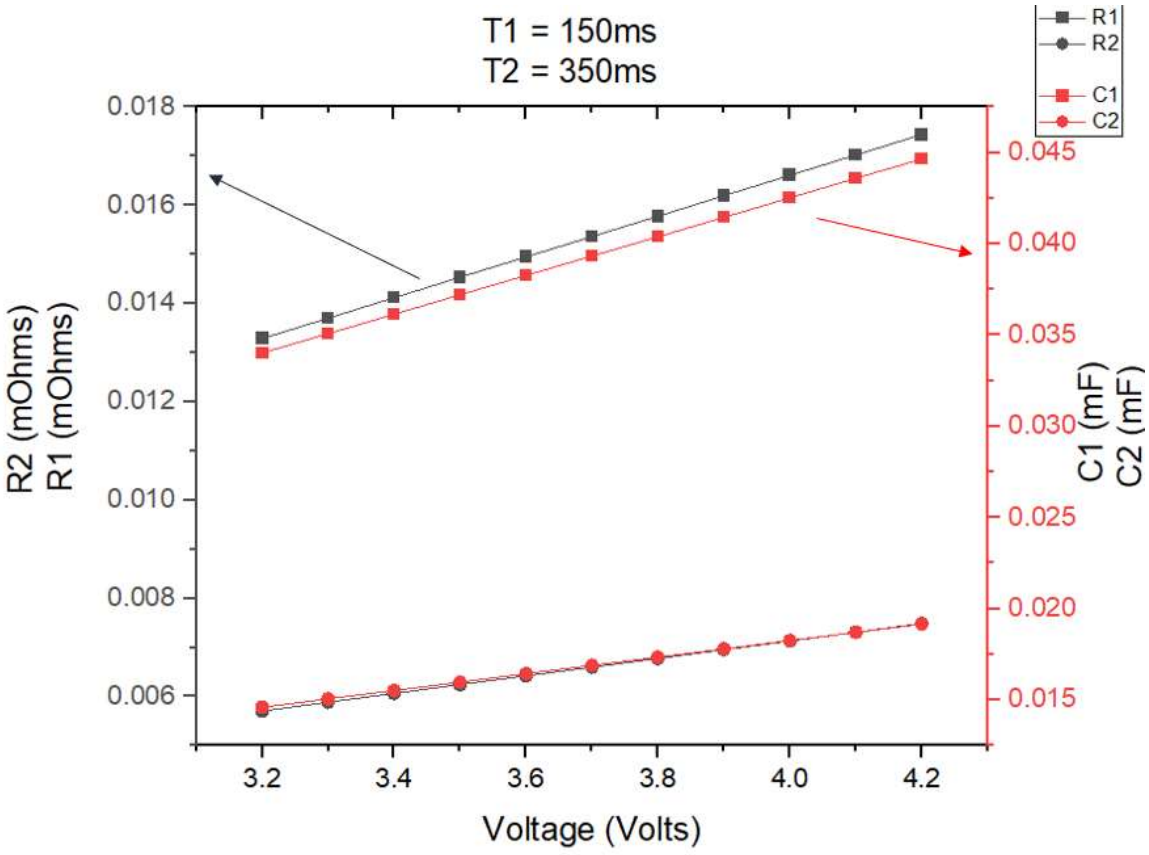

Figure 3.9 Transient parameters (R1, R2,C1,C2) vs Open Circuit Voltage characteristics

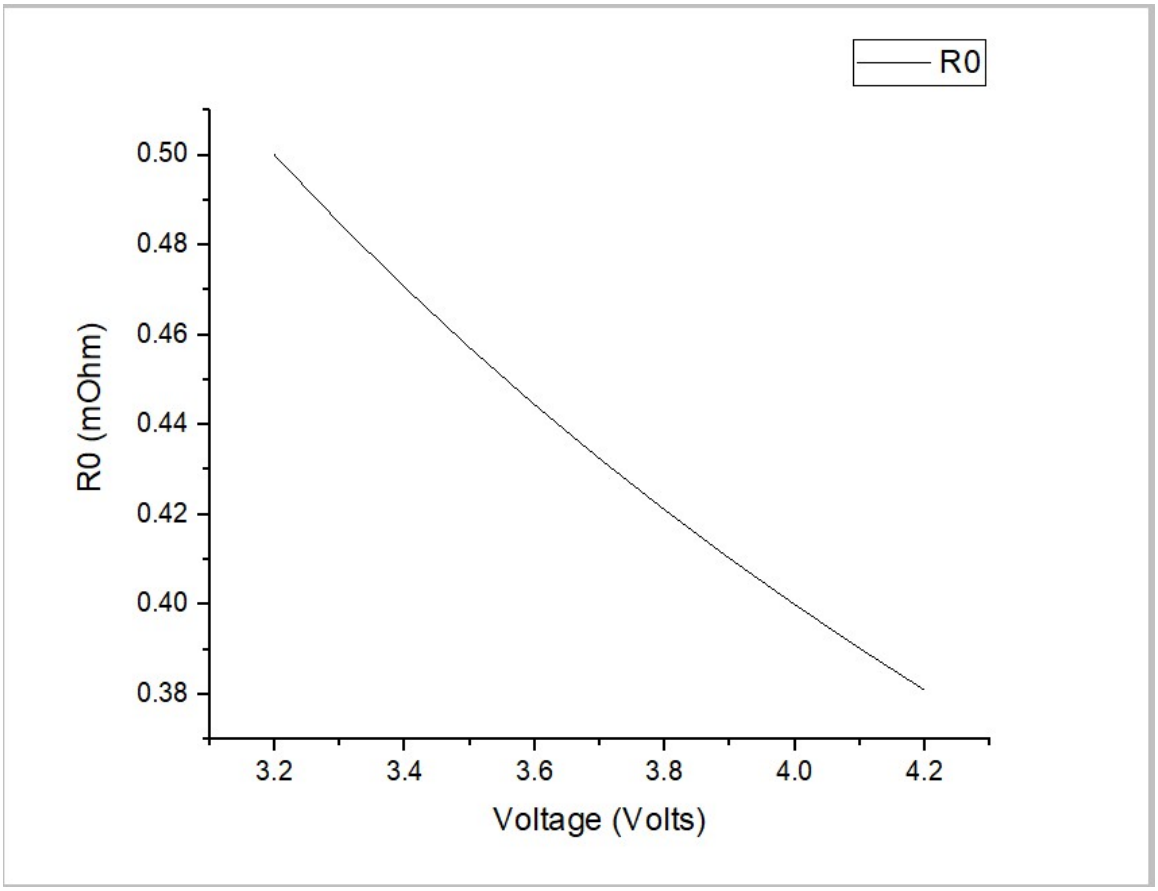

Figure 3.10 Transient parameter R0 vs Open Circuit Voltage characteristics

The above model (figure 3.8) is simulated for a fixed voltage and the relaxation times are varied. The constraints of the chosen microcontroller again reflect on choosing a time period of 500ms and since T1 and T2 and complimentary of each other from a value of 0 to 500, the range of T1 increases from 0 to 500ms while T2 simultaneously decreases from 500 to 0 to their corresponding cell voltage from the model.

Since R0 is a purely resistive loss and is voltage dependent, the value for R0 remains constant for a varied range of state of health. This is corroborated from the parameters simulated in the cell model.

It is noticed that a cell of lower state of health (T1 and T2 at maximum values) produces a higher transient effect thereby increasing the relaxation time of the cell. By studying the duration of T1 and T2 and computing the corresponding RC network parameters, the state of charge can be updated real time.

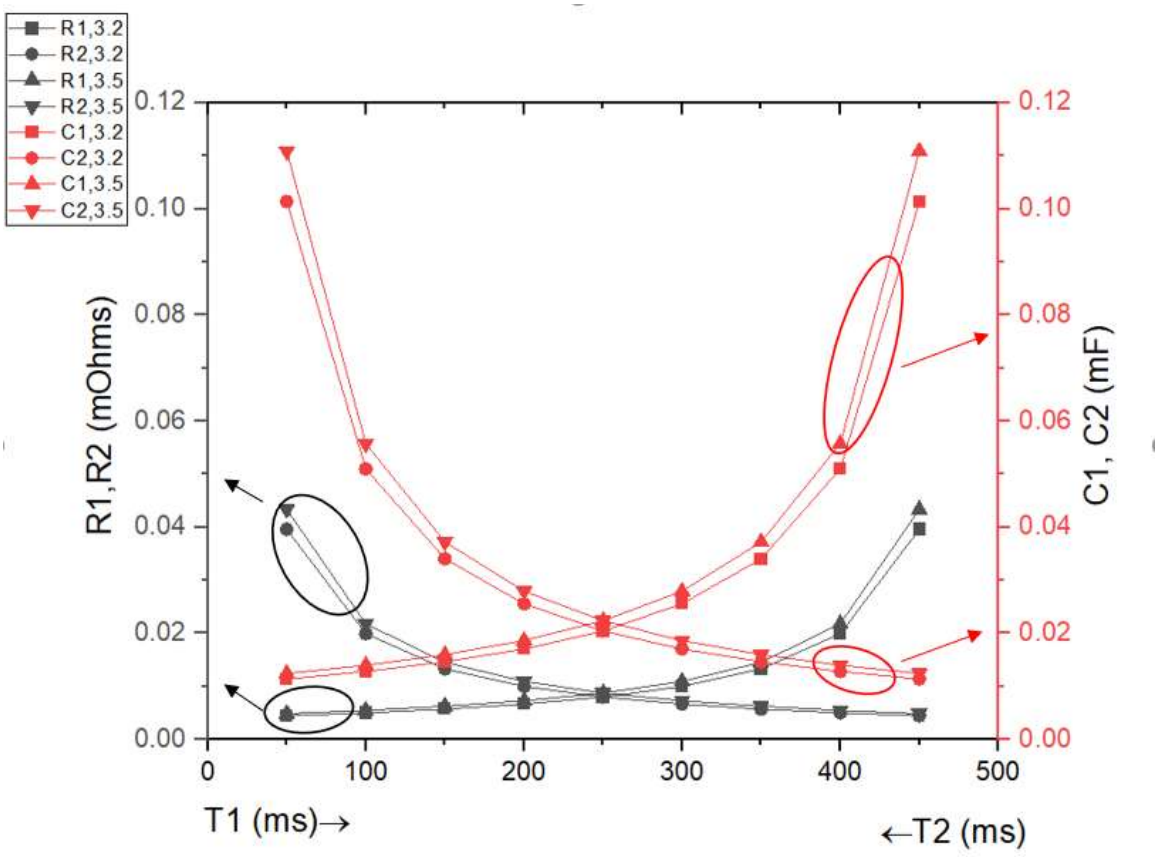

Figure 3.11 Transient parameter (R1, R2, C1,C2) vs State of health Characteristics

Table 3.2 Transient parameters for a fixed state of charge (3.2V) from lithium ion model

Voltage (Volts)	T1 (ms)	T2 (ms)	R1 (mOhms)	C1 (mF)	R2 (mOhms)	C2 (mF)	R0 (mOhms)
3.2	50	450	0.039602653	0.101382793	0.00443951	0.01136515	0.5
3.2	100	400	0.019900333	0.050944851	0.004993755	0.01278401	0.5
3.2	150	350	0.013288987	0.034019808	0.00570613	0.01460769	0.5
3.2	200	300	0.009975042	0.025536107	0.006655568	0.01703825	0.5
3.2	250	250	0.007984021	0.020439095	0.007984021	0.02043909	0.5
3.2	300	200	0.006655568	0.017038254	0.009975042	0.02553611	0.5
3.2	350	150	0.00570613	0.014607693	0.013288987	0.03401981	0.5
3.2	400	100	0.004993755	0.012784013	0.019900333	0.05094485	0.5
3.2	450	50	0.00443951	0.011365145	0.039602653	0.10138279	0.5

3.7 State of Health vs Transient Parameters

A discharge pulse current of 1.7A (0.5C) was applied to the cell at 100% state of health and the voltage response was captured on the Keysight Multimeter.

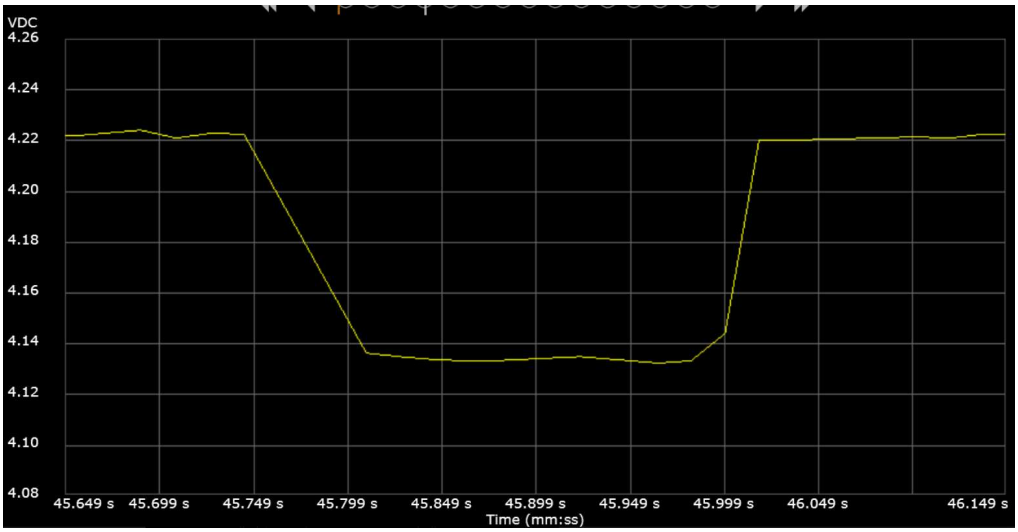

Figure 3.12 The open circuit voltage response of the LG MJ1 3500mAh lithium ion cell at 100% state of charge and 100% state of health to a 0.5C discharge rate current pulse captured on a digital multimeter.

The terminal voltage of the cell is compared to the transient dynamic curve of a lithium ion cell and it is noticed that the waveform resembles the theorized behavior of a lithium ion cell under a discharge pulse. The ohmic resistance drop from 4.22V to 4.14V

comes into effect due to the series resistance. The fast time constant due to charge transfer and double layer capacitance is observed to be from 4.14V to 4.13V with a time constant of 50ms and the slow time constant due to diffusion with a time constant of 150ms.

It is noticed that the series impedance and fast time constant reduce the voltage of the cell gradually over time and once the pulse current is removed the cell does not take a long time to rise back up to the open circuit voltage. The lesser relaxation time, the healthier the cell is. Below is the voltage response curve for the cell at 75% state of health.

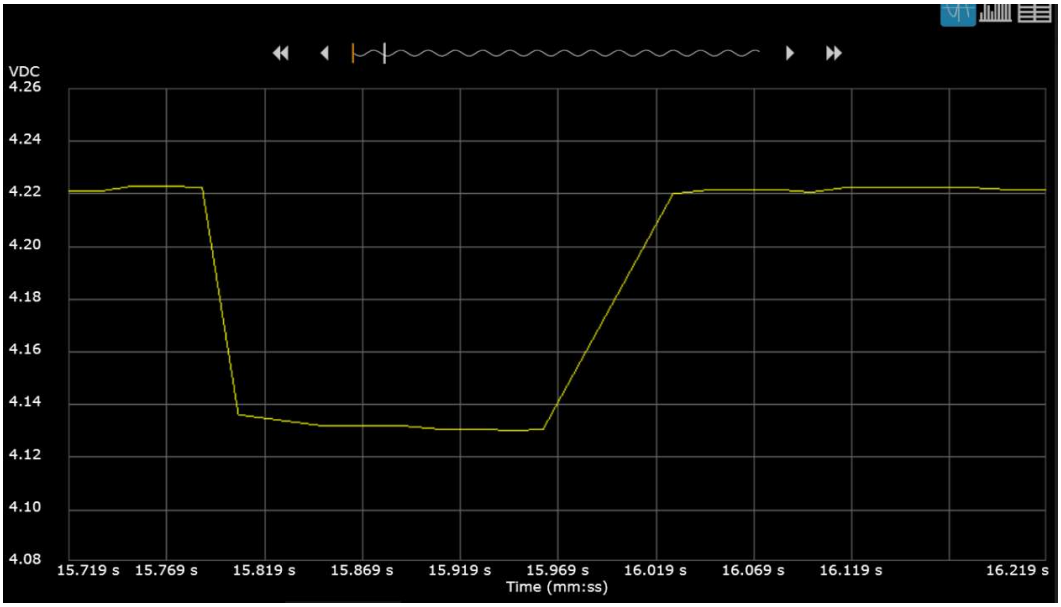

Figure 3.13 The open circuit voltage response of the LG MJ1 3500mAh lithium ion cell at 100% state of charge and 25% state of health to a 0.5C discharge rate current pulse captured on a digital multimeter.

Due to the degradation of the cell's chemical reaction by 25%, there is an instantaneous drop in cell voltage associated with the series impedance. The fast time constant exists if the current pulse exists and once discharging stops, the cell takes a rather longer time to reach and stabilize its open circuit voltage. This brings about measurement errors and miscalculations in the state of charge.

For a cell at 75% SOH, the ohmic voltage drop is noticed to remain the same. The time constants on the other hand have changed to a fast time constant of 45ms and a slow time constant of 125ms. This difference can be associated with the decline of health of the lithium ion cell by 25%.

The voltage-time plot with 1 microsecond time accuracy from figure 4 allows us to measure the fast time constant and slow constant. This can be plugged into the state space equation as inputs for the mathematical model of the cell.

To analyses how the time constant affect the open circuit voltage for a constant current discharge, the cell was set discharged at a constant rate of 0.5C for 60 seconds and the voltage response was captured.

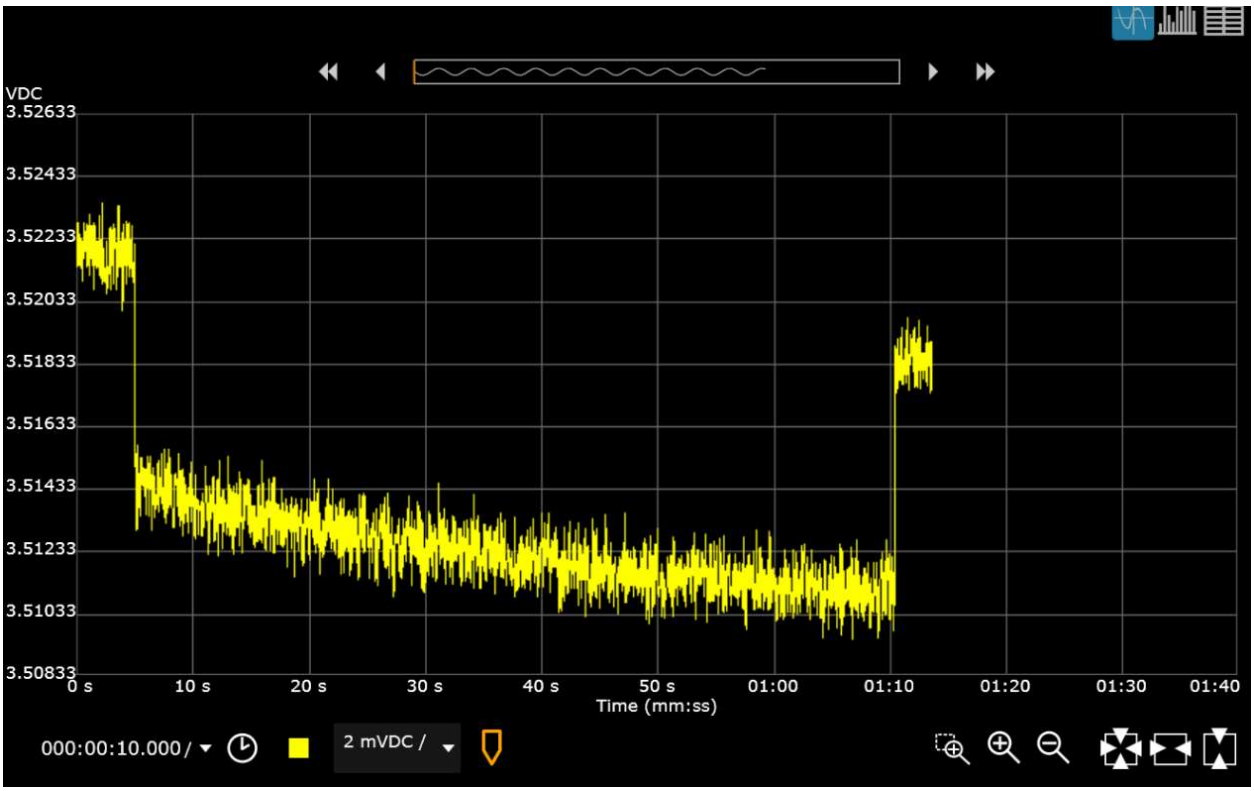

Figure 3.14 The open circuit voltage response of the LG MJ1 3500mAh lithium ion cell at 100% state of charge and 25% state of health to a 0.5C discharge rate continuous current captured on a digital multimeter.

It is noticed that the cell does not relax back to the voltage it started at and this can be associated with the function of coulomb counting. The time constants and voltage difference limits from these test graphs will be fed into the state space model to act as the 'Lithium Ion cell Model".

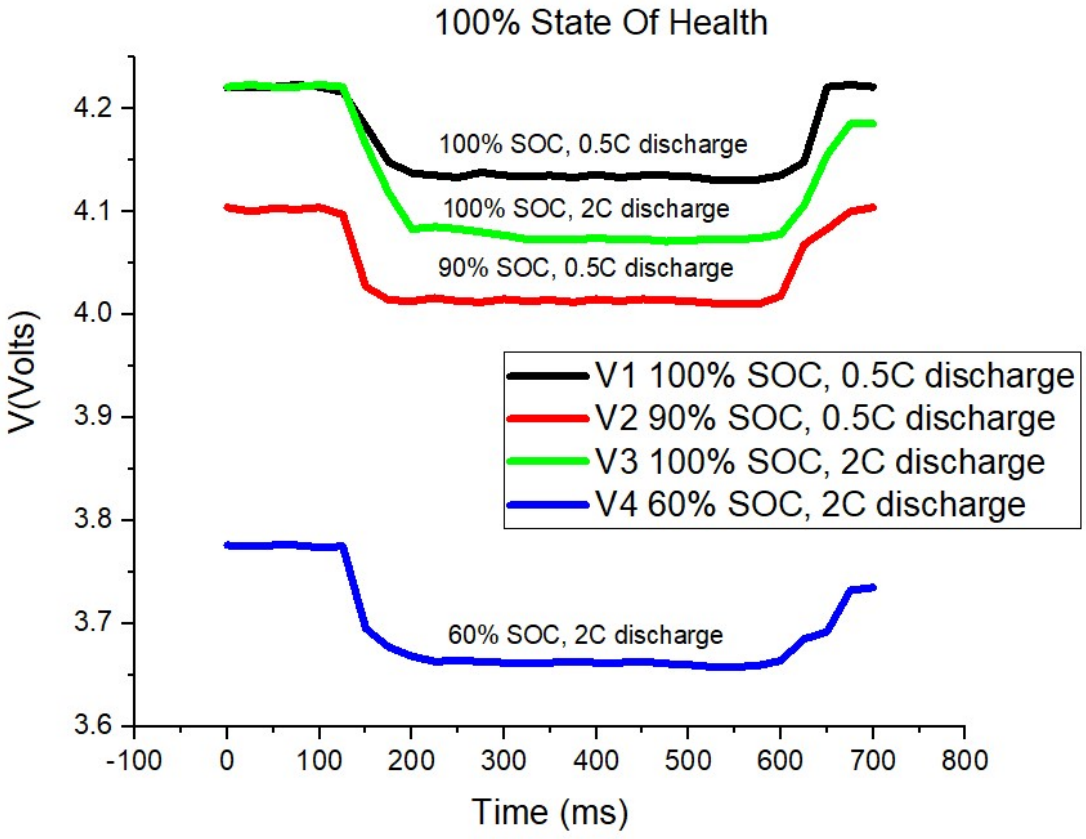

Figure 3.15 Experimental data of Cell open voltage curve for a discharge pulse of 500ms

CHAPTER 4

STATE OF CHARGE ESTIMATION BY KALMAN FILTERING

4.1 State Space Model

The state space equation extrapolated from the mathematical model of the equivalent lithium ion cell is used to obtain the RC network parameters with voltage, current and time period as input to the model. The series impedance being linear acts as the exogenous input for the state space model.

The state space equation of the of equation is as follows,

$$\dot{v}_1 = -v_1 \frac{R_1}{C_1} + \frac{I}{C_1} \tag{4.1}$$

$$\dot{v}_2 = -v_2 \frac{R_2}{C_2} + \frac{I}{C_2} \tag{4.2}$$

$$\dot{v}_3 = 0 \tag{4.3}$$

$$\dot{v} = Av + Bu \tag{4.4}$$

$$\dot{y} = Cv + Du + Ex_0 \tag{4.5}$$

$$A = \begin{bmatrix} \frac{-R_1}{C_1} & 0 \\ \frac{-R_2}{C_2} & 0 \end{bmatrix} \tag{4.6}$$

$$B = \begin{bmatrix} \frac{1}{C_1} \\ \frac{1}{C_2} \end{bmatrix} \tag{4.7}$$

$$C = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.8}$$

$$D = [R_0] \tag{4.9}$$

$$E = [1] \tag{4.10}$$

An open loop system of the state space equations that govern the open circuit of the cell is designed on MATLAB.

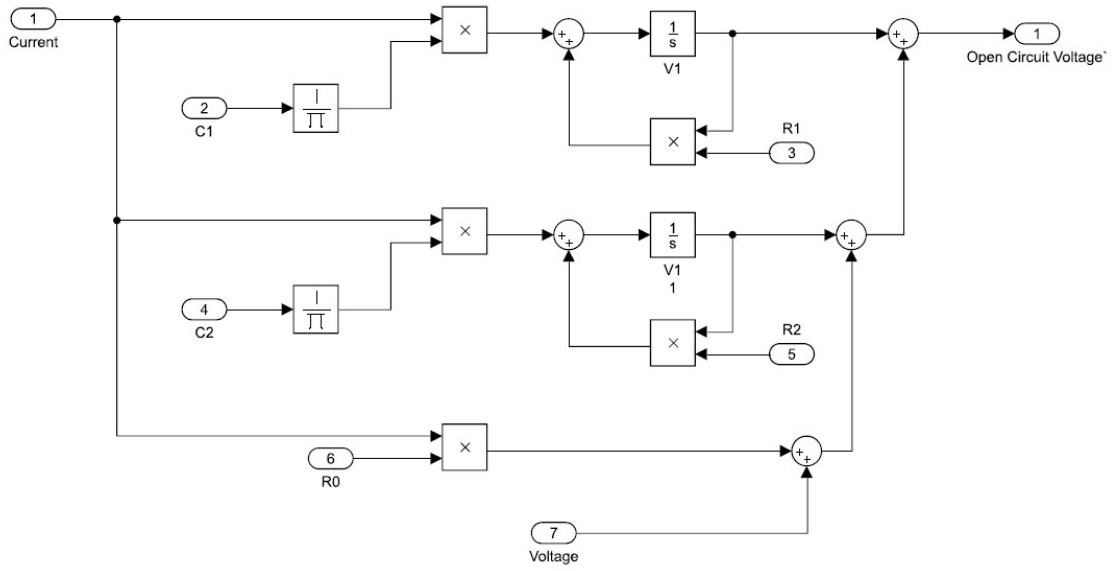

Figure 4.1 State space equation for the time domain equation of the RC network circuit to compute open circuit voltage.

4.2 Kalman Filter

Kalman filtering is a control estimation theory to calculate the stability of a system with the presence of an external observer, in this case, the closed-circuit voltage which varies from the open circuit voltage dynamically. This dynamic system controlled by the inputs to the system converges to the time domain analysis. Below is the state space model for the entire system to estimate the open circuit voltage with a better accuracy.

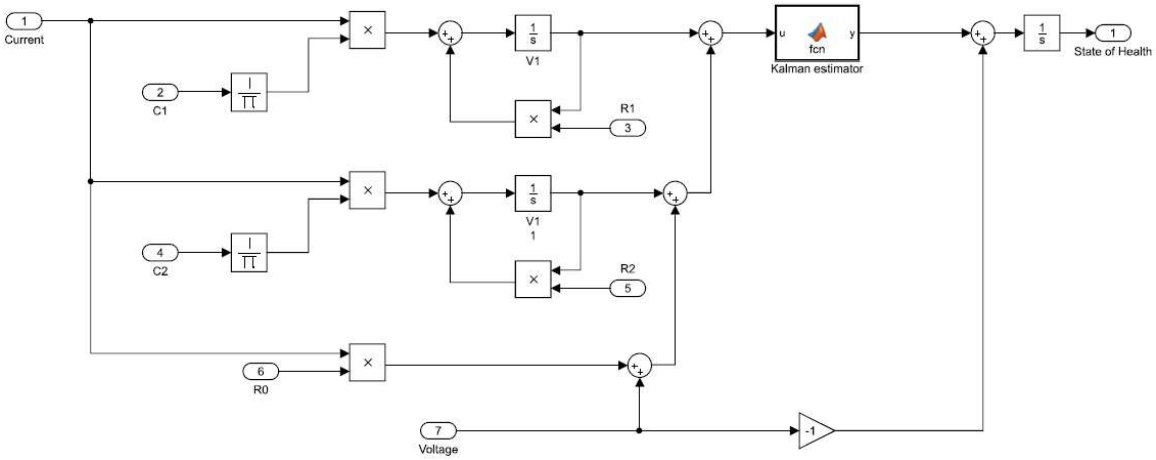

Figure 4.2 Kalman estimator state space model for lithium ion cell to estimate state of health.

$$P'' = AP + PA' + C'EC \tag{4.11}$$

$$K = PC'W^{-1} \tag{4.12}$$

The characteristic equation obtained from equation 3.7 denoted the open circuit voltage of the lithium ion cell for the corresponding state of health.

4.3 State of Charge Estimation

The state of charge of a lithium ion cell is a function of the open circuit voltage, temperature, rate of charge/discharge and the state of health. The linear dependence of state of charge with the open circuit voltage depends on the state of health of a cell. As seen from fig 3.10, it is noted that the characteristic equation of the State of charge versus voltage curve varies with respect to the state of health of a cell.

By using Kalman estimation, the state of health of a lithium ion cell is calculated by finding the RC network parameters and comparing them with the ideal cell parameters at different state of health with corresponding rate of discharge. The input to the entire state of charge estimation model are as follows,

1. Closed circuit cell voltage
2. Rate of charge/Discharge
3. Slow time constant
4. Fast time constant

The cell test data used to model the ideal cell parameters from 100% state of health to 0% state of health at 0.5C, 1C and 2C rate of charge and discharge are fed as an input to the state of charge estimating function.

4.4 Firmware Implementation

The bq76PL455A-Q1 is a battery monitoring and protection integrated circuit for a 16-cell battery system, scalable up to 16 modules. The BQ chip provides cell voltage and sampling time along with cell temperature with sophisticated fault detection system and 6 general purpose input-output pins. This BQ chip is used for the sampling of the cell voltage which is obtained through UART protocol by a microcontroller.

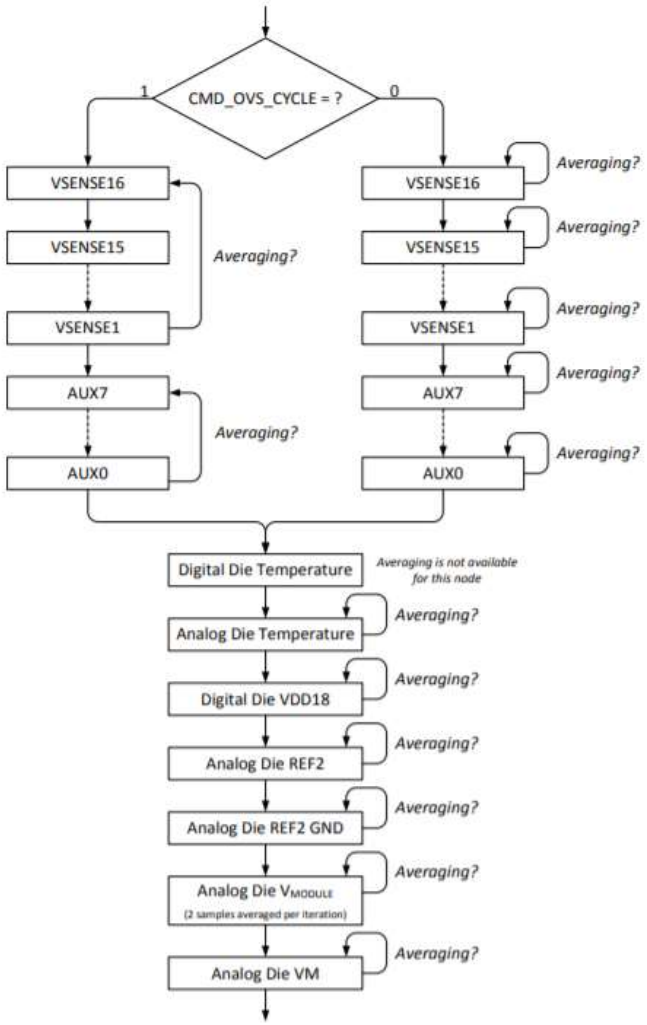

Figure 4.3 Cell voltage measurement algorithm of battery management system.

During every charge/discharge cycle, the battery management system reads the cell voltage every 10-millisecond including the open circuit voltage. During charge/discharge, the energy of the battery pack is calculated by coulomb counting which is embedded in the battery management firmware. The BQ chip acts as an analog front end while controlled by the master device, the TM4C microcontroller chosen for its computational speed and ability to execute signal processing commands.

CHAPTER 5

CONCLUSION

Being in the digital age, accuracy drives everything. Right from time accuracy to quantity accuracy, the precision of what we do end up governing how effective our task is. The same concept applies to energy storage device which are the frontrunners of electric vehicles. Lithium ion cell technology has revolutionized electric automotive technology and energy backup application.

The transients in a pure DC source such as a lithium ion cell end up governing measurement accuracy and load behavior which takes a toll on the accuracy. From the transient study and modelling of the parameters of the cell, the voltage versus energy characteristic equation can be updated with the model by measuring the time constants of the relaxation time during hysteresis of the cell voltage.

It has been observed that the traditional method of coulomb counting is inefficient to use in real time applications such as electric vehicles. The lithium ion cell used in this application, the NJIT electric vehicle program, has been modelled and tested to reflect the deterioration of the state of health which in turn provides a more accurate state of health with Kalman estimation which can be integrated with the method of coulomb counting without having to opt for a processor of higher specifications in battery management systems. The MATLAB simulations were verified for a cell of 75% state of health which reflected the appropriate state of charge of the cell which coulomb counting failed to estimate.

APPENDIX A

MATLAB SOURCE CODE AND SIMULINK MODEL

```
function OG = fcn(soh)

coder.extrinsic('lqe');

A = [-r1/c1 0 ;  -r2/c2 0 ];

B = [1/c1 ; 1/c2];

C = [1 ; 1];

D = R0;

I = [1 0; 0 1];

F = [1 ; 0];

poles = eig(A-BI);

P = [p1 p2 ; p3 p4];

G = place(A,B,P);

[K,P, ekf] = lqe(A,B,C,D,F)

[kest , L , P] = Kalman(K, P, ekf)

tf(soh);
```

APPENDIX B

FIRMWARE INTEGRATION WITH BMS

```
/*
 * bq76.c
 *
 *  Created on: Nov 10, 2018
 *      Author: Duemmer
 */

#include <stdint.h>
#include <stdbool.h>

#include <driverlib/uart.h>
#include <driverlib/gpio.h>
#include <driverlib/debug.h>
#include <driverlib/timer.h>


#include "bq.h"
#include "fault.h"
#include "types.h"
#include "util.h"

static uint8_t g_pui8ReadBuf[BQ_READ_BUF_SIZE]; // Read buffer for UART frames

static uint32_t g_ui32ReadBufPtr = 0; // Relative pinter to tail of the read buffer

static bool g_bCmdInProgress = false; // If true, a command is currently in progress

static bool g_bCmdHasResponse = false; // If true, the current command expects a
response

static tSampleType g_sSampleType = NONE; // Current voltage sampling mode

// MCU peripheral data
static uint32_t g_ui32ReadDoneVec; // IRQ vector number to trigger when a UART read
is finished
static uint32_t g_ui32UARTBase; // Base address of the UART peripheral used for bq76
communications
static uint32_t g_ui32WTimerBase; // Base address of the general purpose wide timer
allocated for BQ use
```

```
/**
 * Converts a numeric transmission frame length into the form
 * required by the frame header
 */
static inline uint8_t bq76_len2frame(uint8_t ui8Len) {
   if(ui8Len < 7)
      return ui8Len;
   else
      return BQ_DATA_SIZE_8;
}



/**
 * CRC-16-IBM checksum algorithm, as used by the bq76 to verify user
 * transmissions. Copied from the BQ76 datasheet, section 7.5.2 on page
 * 55
 */
uint16_t bq76_checksum(uint8_t *pui8Buf, uint16_t ui16Len) {
   uint16_t crc = 0;
   while (ui16Len--) {
   crc ^= *pui8Buf++;
   for (uint16_t j = 0; j < 8; j++)
   crc = (crc >> 1) ^ ((crc & 1) ? 0xa001 : 0);
   }
   return crc;
}


/**
 * General interrupt routine for processing interrupts
 * from the bq76 UART module.
 */
void _bq76_uartISR() {
   // if we make it this far we won't time out
   TimerDisable(g_ui32WTimerBase, TIMER_A);

   // Copy data to read buffer
   while(UARTCharsAvail(g_ui32UARTBase))
      g_pui8ReadBuf[g_ui32ReadBufPtr++] = UARTCharGet(g_ui32UARTBase) &
0xFF;

   // Receive timed out, so we can assume its done
   if(UARTIntStatus(g_ui32UARTBase, 1) & UART_INT_RT) {
      g_bCmdInProgress = false;
```

```
    }
}


/**
 * ISR that runs whenever the timeout period on a command expires.
 * For commands without response, this will happen normally, after
 * every command, and serves to give the bq76 chips some time to
 * process the command before receiving a new one. For systems with
 * response, however, the UART receive ISR should stop the timer, therefore
 * if this runs after a response command, we have a communication fault.
 */
void _bq76_timerISR() {
    g_bCmdInProgress = false;
    if(g_bCmdHasResponse) {
        tFaultInfo sInfo;
        sInfo.ui64TimeFlagged = util_usecs();
        sInfo.data.ui32 = 0; // Not really any data we can package with this
        fault_setFault(FAULT_BQ_COM, sInfo);
    }
}




/**
 * Writes a frame of data to the BQ76 bank over UART. At the moment,
 * this always returns true, but this may change as more fault handling
 * capabilities are added.
 */
bool bq76_write(
        uint8_t ui8Flags,
        uint8_t ui8Len,
        uint8_t ui8Addr,
        uint8_t *pui8Data)
{
    uint8_t pui8Writebuf[BQ_WRITE_BUF_SIZE];
    uint16_t ui16Checksum;
    uint32_t i = 0; // write buffer pointer

    // Wait until we're ready to run a command, then reset global flags
    bq76_waitResponse();
    g_ui32ReadBufPtr = 0;
    g_bCmdInProgress = true;
    g_bCmdHasResponse =
            !(ui8Flags & BQ_REQ_TYPE_SGL_NORESP)    &&
```

```
        !(ui8Flags & BQ_REQ_TYPE_GRP_NORESP)    &&
        !(ui8Flags & BQ_REQ_TYPE_BC_NORESP);

    // Build write buffer
    if(ui8Flags & BQ_FRM_TYPE_COMMAND) {

        // header
        ui8Flags |= bq76_len2frame(ui8Len);
        pui8Writebuf[i++] = ui8Flags;

        // if we aren't broadcasting, we need to have an address byte
        // in the packet, so add it in
        if(
            !(ui8Flags | BQ_REQ_TYPE_BC_RESP)  &&
            !(ui8Flags | BQ_REQ_TYPE_BC_NORESP)
        ) {
            pui8Writebuf[i++] = ui8Addr;
        }

        // data
        for(int j=0; j<ui8Len; j++)
            pui8Writebuf[i++] = pui8Data[j];

        // checksum
        ui16Checksum = bq76_checksum(pui8Writebuf, ui8Len);
        pui8Writebuf[i++] = ui16Checksum >> 8;
        pui8Writebuf[i++] = ui16Checksum & 0xFF;

        // write the buffer
        for(int j=0; j<i; j++)
            UARTCharPut(g_ui32UARTBase, pui8Writebuf[j]);

        // Run the timeout timer
        TimerLoadSet(g_ui32UARTBase, TIMER_A, BQ_WRITE_TIMEOUT);
        TimerEnable(g_ui32UARTBase, TIMER_A);
    } else {} // TODO: throw a fault, we shouldn't be transmitting without the command
flag
    return true; // TODO: add proper return exception handling
}



/**
 * Writes a value to a register of one or more bq76 modules. Can
 * write a variable number of bytes to the register, from 0 to
 * BQ_WRITEREG_MAX_MSG. If ui8DataLength is greater than
```

```
BQ_WRITEREG_MAX_MSG,
 * then the write will abort and return 0. Otherwise, it will return
 * whatever bq76_write returns.
 */
bool bq76_writeReg(
      uint8_t ui8Flags,
      uint8_t ui8Addr,
      uint16_t ui16Reg,
      uint8_t ui8Len,
      uint8_t *pui8Data)
{
   if(ui8Len > BQ_WRITEREG_MAX_MSG) {
      // TODO: Assert proper software fault, this is illegal
      return false;
   }

   // 2 for register address, the rest for the max possible message length
   uint8_t pui8WriteBuf[2 + BQ_WRITEREG_MAX_MSG];
   int i=0;

   if(ui8Flags & BQ_ADDR_SIZE_16)
      pui8WriteBuf[i++] = ui16Reg >> 8;
   pui8WriteBuf[i++] = ui16Reg & 0xFF;

   for(int j=0; j<ui8Len; j++)
      pui8WriteBuf[i++] = pui8Data[j];

   return bq76_write(ui8Flags, i, ui8Addr, pui8WriteBuf);
}




/**
 * Extracts information from a raw response stored in the read buffer,
 * starting with ui8Start
 * Checks that the response checksum is correct, and that a command with response
 * is not in progress, and provides the caller with the response
 * length.
 * Returns true if the checksum is correct and the command
 * in progress flag is false, false otherwise. If a zero flag is
 * returned, there is no guarantee that the calculated length is correct
 */
bool bq76_parseResponse(uint8_t ui8Start, uint8_t *pui8Len) {

   // if a response command is in progress, the buffer may be clobbered
```

```
    if(g_bCmdInProgress && g_bCmdHasResponse)
        return false;

    *pui8Len = g_pui8ReadBuf[ui8Start];

    // Extract the checksum
    uint16_t ui16RespChecksum = g_pui8ReadBuf[*pui8Len + ui8Start] << 8;
    ui16RespChecksum |= g_pui8ReadBuf[*pui8Len + 1 + ui8Start];

    // Now compare it with the calculated checksum
    uint16_t ui16CalcChecksum = bq76_checksum(g_pui8ReadBuf+ui8Start, *pui8Len);
    return ui16CalcChecksum == ui16RespChecksum;

}


/**
 * Parses the response of commands from multiple targets, from either broadcast
 * or group writes with responses. This will loop over the buffer, extracting the
 * lengths and start pointers of each block, up to ui8NumResponses.
 * In addition, it will individually verify the checksum of each response, returning
 * a 0 for a mismatch. Note that a read fault, and missed checksum,
 * could possibly shift the data, therefore corrupting all responses
 * past that point. Returns 0 if a command with response is in progress
 * or the read buffer is overrun, false otherwise.
 */
bool bq76_parseMultiple(
        uint8_t ui8NumResponses,
        uint8_t *pui8StartPtrs,
        uint8_t *pui8Lengths,
        uint8_t *pui8ChecksumGood)
{
    if(g_bCmdInProgress && g_bCmdHasResponse)
        return false;

    int iBufPtr = 0;
    uint8_t ui8RespOn = 0;
    while(ui8RespOn < ui8NumResponses &&
            iBufPtr < BQ_READ_BUF_SIZE)
    {
        // Verify the checksum, parse the block, and get its length
        pui8ChecksumGood[ui8RespOn] =
                bq76_parseResponse(iBufPtr, pui8Lengths+ui8RespOn);

        pui8StartPtrs[ui8RespOn] = (uint8_t) iBufPtr;
```

```
        // Advance the buffer pointer
        // add 3 bytes on top of data length: 2 for checksum,
        // 1 for frame initializer
        iBufPtr += pui8Lengths[ui8RespOn] + 3;
        ui8RespOn++;
    }

    // Verify any overruns
    return iBufPtr < BQ_READ_BUF_SIZE;
}



/**
 * Reads the value of a single (variable length) register on a single
 * bq76 module. Returns a nonzero value if the register was read
 * with a correct checksum, zero otherwise
 */
uint8_t bq76_readRegSingle(
        uint8_t ui8Addr,
        uint16_t ui16Reg,
        uint8_t ui8Len,
        uint8_t *pui8Response)
{
    // Single device write with response, either 8 or 16 bit
    // register addressing, based on register address size
    uint8_t ui8Flags =
            BQ_FRM_TYPE_COMMAND    |
            BQ_REQ_TYPE_SGL_RESP   |
            (ui16Reg & 0xFF00) ? BQ_ADDR_SIZE_16 : BQ_ADDR_SIZE_8;

    uint8_t pui8WriteBuf[5];
    int i=0;

    if(ui8Flags & BQ_ADDR_SIZE_16)
        pui8WriteBuf[i++] = ui16Reg >> 8;
    pui8WriteBuf[i++] = ui16Reg & 0xFF;
    pui8WriteBuf[i++] = ui8Len-1;

    bq76_write(ui8Flags, i, ui8Addr, pui8WriteBuf);

    uint8_t ui8RespLen;
    if(!bq76_waitResponse(BQ_WRITE_RESP_TIMEOUT) &&
        bq76_parseResponse(0, &ui8RespLen))
    { // got a response, copy it over
```

```
        for(int j=0; j<ui8RespLen; j++)
            pui8Response[j] = g_pui8ReadBuf[j+1];

        // if the lengths don't match, that's probably an issue,
        // so flag it in the return
        return ui8Len == ui8RespLen;
    }

    return false; // Bad checksum or response timeout
}




/**
 * Waits until the g_ui8CmdInProgress flag goes low,
 * or a timeout occurs. The flag should be automatically set
 * after a command is sent, and reset after either the entire
 * response is received (if applicable), or a certain amount
 * of time has elapsed (for no response commands).
 *
 * The timeout is determined based on whether or not the command in
 * progress expects a response
 *
 * Returns true if it timed out waiting for a response,
 * false if the flag was cleared.
 *
 * NOTE: This will never affect flag or rx buffer state!
 */
bool bq76_waitResponse() {
    uint64_t ui64Started = util_usecs();
    uint64_t ui64Timeout = BQ_WRITE_TIMEOUT;
    while(g_bCmdInProgress) {
        if(util_usecs() + ui64Timeout > ui64Started)
            return true;
    }
    return false;
}

/**
 * Executes the auto addressing sequence to the onboard BQ76 modules.
 * Taken from the  BQ76 Software Developer's reference, pg. 2. This
 * will assign addresses starting from 0 up to BQ_NUM_MODULES-1, with
 * 0 being the device on the bottom of the stack (connected to the UART)
 * and BQ_NUM_MODULES-1 being the device at the top of the stack
 * (with no high side differential transmitter). In addition, this will
```

```
 * also clear all faults asserted on the modules.
 *
 * Returns a nonzero value if addressing was successful, 0 otherwise
 */
bool bq76_autoAddress(uint32_t ui32nModules) {
   uint8_t pui8RegData[4]; // temp buffer for multi-byte register writes

   // Fully Enable Differential Interfaces and Select Auto-Addressing Mode
   uint16_t ui16ComCfgDat =
         BQ_BAUD_INIT_CODE   |
         (1 << 7)          |   // UART transmission enable
         (1 << 6)          |   // high side receiver enable
         (1 << 5);             // low side transmitter enable

   bq76_writeReg(
         BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_NORESP,
         0,
         BQ_REG_COMCONFIG,
         2,
         (uint8_t *) &ui16ComCfgDat);

   // Put Devices into Auto-Address Learning Mode
   pui8RegData[0] = 1 << 4;
   bq76_writeReg(
         BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_NORESP,
         0,
         BQ_REG_DEVCONFIG,
         1,
         pui8RegData);

   pui8RegData[0] = 1 << 3;
   bq76_writeReg(
         BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_NORESP,
         0,
         BQ_REG_DEV_CTRL,
         1,
         pui8RegData);

   // Assign addresses to each potential device
   for(int i=0; i<BQ_MAX_NUM_MODULE; i++) {
      pui8RegData[0] = i;
      bq76_writeReg(
         BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_NORESP,
         0,
         BQ_REG_ADDR,
         1,
```

```
        pui8RegData);
}

// Verify that the modules are responding correctly. If any one doesn't respond,
// assert a level 1 (non-halting fatal) fault
uint8_t ui8AddrResponse;
for(int i=0; i<ui32nModules; i++) {
   if(!bq76_readRegSingle( // Read failed
         i,
         BQ_REG_ADDR,
         0,
         &ui8AddrResponse))
   {
      // Failed to address the modules.
      // TODO: assert a fault
      return false;
   }
}

// if we make it here, everything is addressed correctly

// Turn off the high side of the differential interface on
// the device on the top of the stack. See Section 1.2.5 of
// the software developer's manual for reference on these data
// values
pui8RegData[0] = 0x10;
pui8RegData[1] = 0x20;
bq76_writeReg(
   BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_SGL_NORESP,
   ui32nModules-1,
   BQ_REG_COMCONFIG,
   2,
   pui8RegData);

// Turn off the low side of the differential interface on
// the device on the bottom of the stack. See Section 1.2.6 of
// the software developer's manual for reference on these data
// values
pui8RegData[0] = 0x10;
pui8RegData[1] = 0xC0;
bq76_writeReg(
   BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_SGL_NORESP,
   0,
   BQ_REG_COMCONFIG,
   2,
   pui8RegData);
```

```
    // Clear all faults on each device, starting from the top down.
    // See Section 1.2.6 of the software developer's manual for
    // reference on these data values
    pui8RegData[0] = 0xFF;
    pui8RegData[1] = 0xC0;
    for(int i=0; i<BQ_MAX_NUM_MODULE; i++) {
        bq76_writeReg(
            BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_SGL_NORESP,
            i,
            BQ_REG_FAULT_SUM,
            2,
            pui8RegData);
    }

    // At this point the devices should be ready to go with further
    // configuration
    return true;
}


/**
 * Tries to start running a cell voltage sample with response on the
 * BQ76 stack. Will Configure the sampling channels, and send a
 * broadcast write request, using the current BQ configurations.
 * will return true if sampling is started, false otherwise
 */
bool bq76_StartCellVoltageSample(tConf *psConf) {
    if(g_bCmdInProgress)
        return false;

    bool bALlGood = true;

    // Set the cells to sample on each device. Broadcast a read all
    // channels select, then on the last one, round it out to the correct
    // number of cells
    uint32_t ui32SampleConf = 0xFFFF0000;
    bALlGood = bq76_writeReg(
        BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_NORESP,
        0,
        BQ_REG_CHANNELS,
        4,
        (uint8_t *)&ui32SampleConf);

    // Determine the mask for the last module
    uint32_t ui32nCellsLast = psConf->ui32nCells % BQ_MAX_SAMPLE;
```

```
    ui32SampleConf = 0;
    for(int i=0; i<ui32nCellsLast; i++)
        ui32SampleConf |= 1 << (i+16);

    bALlGood = bq76_writeReg(
        BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_SGL_NORESP,
        psConf->ui32NumBQModules-1,
        BQ_REG_CHANNELS,
        4,
        (uint8_t *)&ui32SampleConf);

    if(!bALlGood)
        return false;

    // Send the broadcast cell sample command on all devices
    uint8_t ui8Data = BQ_CMD_SAMPLE | (psConf->ui32NumBQModules & 0xF);
    bALlGood = bq76_writeReg(
        BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_RESP,
        0,
        BQ_REG_CMD,
        1,
        &ui8Data);

    return bAlGood;
}




/**
 * Tries to start running a thermo voltage sample with response on the
 * BQ76 stack. Will Configure the sampling channels, and send a
 * broadcast write request, using the current BQ configurations.
 * will return true if sampling is started, false otherwise
 */
bool bq76_startThermoSample(tConf *psConf, bool bMuxState) {
    if(g_bCmdInProgress)
        return false;

    bool bALlGood = true;

    // Set the channels to sample on each device
    uint32_t  ui32SampleConf = 0xFF00;

    bALlGood = bq76_writeReg(
        BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_NORESP,
```

```
        0,
        BQ_REG_CHANNELS,
        4,
        (uint8_t *)&ui32SampleConf);

    if(!bALlGood)
        return false;

    // Send the broadcast cell sample command on all devices
    uint8_t ui8Data = BQ_CMD_SAMPLE | (psConf->ui32NumBQModules & 0xF);
    bALlGood = bq76_writeReg(
        BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_RESP,
        0,
        BQ_REG_CMD,
        1,
        &ui8Data);

    if(bALlGood)
        g_sSampleType = bMuxState ? THERM2 : THERM1;
    else
        g_sSampleType = NONE;

    return bALlGood;
}


bool bq76_setCellVoltageThreshold(float fUnder, float fOver) {

    // The registers are adjacent, can do it in one write
    uint16_t pui16Conf[2];
    pui16Conf[0] = BQ_VOLTS_TO_ADC(fUnder);
    pui16Conf[1] = BQ_VOLTS_TO_ADC(fOver);

    return bq76_writeReg(
            BQ_FRM_TYPE_COMMAND | BQ_REQ_TYPE_BC_NORESP,
            0,
            BQ_REG_CELL_UV,
            4,
            (uint8_t *)pui16Conf);
}
```

REFERENCES

[1] Z. Cen, P. Kubiak and I. Belharouak, "Online parameter estimation/tracking for Lithium-ion battery RC model," *2016 International Renewable and Sustainable Energy Conference (IRSEC)*, Marrakech, 2016, pp. 936-940.

[2] B. Chen, H. Ma, H. Fang, H. Fan, K. Luo and B. Fan, "An approach for state of charge estimation of Li-ion battery based on Thevenin equivalent circuit model," 2014 Prognostics and System Health Management Conference (PHM-2014 Hunan), Zhangiiaijie, 2014, pp. 647-652.

[3] H. Hinz, "Evaluation of a lithium-ion battery model for power systems," PCIM Asia 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, Shanghai, China, 2016, pp. 1-8.

[4] Jingyu, Yan & Xu, Guoqing & Qian, Huihuan & Yangsheng, Xu. (2010). Robust State of Charge Estimation for Hybrid Electric Vehicles: Framework and Algorithms. Energies. 3. 10.3390/en3101654.

[5] K. Li and K. J. Tseng, "An electrical model capable of estimating the state of energy for lithium-ion batteries used in energy storage systems," 2016 IEEE 2nd Annual Southern Power Electronics Conference (SPEC), Auckland, 2016, pp. 1-8.

[6] K. Li, F. Wei, K. J. Tseng and B. Soong, "A Practical Lithium-Ion Battery Model for State of Energy and Voltage Responses Prediction Incorporating Temperature and Ageing Effects," in IEEE Transactions on Industrial Electronics, vol. 65, no. 8, pp. 6696-6708, Aug. 2018.

[7] W. Li, L. Liang, W. Liu and X. Wu, "State of Charge Estimation of Lithium-Ion Batteries Using a Discrete-Time Nonlinear Observer," in IEEE Transactions on Industrial Electronics, vol. 64, no. 11, pp. 8557-8565, Nov. 2017.

[8] Y. Ma, B. Li, G. Li, J. Zhang and H. Chen, "A nonlinear observer approach of SOC estimation based on hysteresis model for lithium-ion battery," in IEEE/CAA Journal of Automatica Sinica, vol. 4, no. 2, pp. 195-204, April 2017.

[9] J. Meng et al., "An overview of online implementable SOC estimation methods for Lithium-ion batteries," 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), Brasov, 2017, pp. 573-580.

[10] J. Meng et al., "An Overview and Comparison of Online Implementable SOC Estimation Methods for Lithium-Ion Battery," in IEEE Transactions on Industry Applications, vol. 54, no. 2, pp. 1583-1591, March-April 2018.

[11] Q. Ouyang, J. Chen, J. Zheng and H. Fang, "Optimal Cell-to-Cell Balancing Topology Design for Serially Connected Lithium-Ion Battery Packs," in IEEE Transactions on Sustainable Energy, vol. 9, no. 1, pp. 350-360, Jan. 2018.

[12] Plett, Gregory. (2004). Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background. Journal of Power Sources. 134. 252-261. 10.1016/j.jpowsour.2004.02.031.

[13] Plett, G. L. (2015). Battery management systems: Battery modeling.

[14] J. Poonsuk and S. Pongyupinpanich, "Design and estimation of state-charging applied for lithium-ion battery based on Matlab-Simulink," 2016 Management and Innovation Technology International Conference (MITicon), Bang-San, 2016, pp. MIT-176-MIT-179.

[15] Saidani, Fida & Hutter, Franz & Scurtu, Rares-George & Braunwarth, Wolfgang & Burghartz, J.N.. (2017). Lithium-ion battery models: A comparative study and a model-based powerline communication. Advances in Radio Science. 15. 83-91. 10.5194/ars-15-83-2017.

[16] R. M. S. Santos, C. L. G. d. S. Alves, E. C. T. Macedo, J. M. M. Villanueva and L. V. Hartmann, "Estimation of lithium-ion battery model parameters using experimental data," 2017 2nd International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT), Fortaleza, 2017, pp. 1-6.

[17] Y. Shi, K. Smith, R. Zane and D. Anderson, "Life prediction of large lithium-ion battery packs with active and passive balancing," 2017 American Control Conference (ACC), Seattle, WA, 2017, pp. 4704-4709.

[18] A. Stroe, D. Stroe, M. Swierczynski, R. Teodorescu and S. K. Kær, "Lithium-ion battery dynamic model for wide range of operating conditions," 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), Brasov, 2017, pp. 660-666.

[19] T. Takegami and T. Wada, "State-of-charge and parameter estimation of lithium-ion battery using dual adaptive filter," 2017 IEEE Conference on Control Technology and Applications (CCTA), Mauna Lani, HI, 2017, pp. 1332-1337.

[20] Texas Insturments bq76PL455A-Q1 16-Cell EV/HEV Integrated Battery Monitor and Protector

[21] X. Xia and Y. Wei, "Lithium-ion batteries State-of-charge estimation based on interactive multiple-model Extended Kalman filter," 2016 22nd International Conference on Automation and Computing (ICAC), Colchester, 2016, pp. 204-207.

[22] N. Yang, J. Feng, Q. Sun, T. Liu and D. Zhong, "Online estimation of state-of-health for lithium ion batteries based on charge curves," 2016 11th International Conference on Reliability, Maintainability and Safety (ICRMS), Hangzhou, 2016, pp. 1-8.

[23] W. Wang, P. Malysz, K. Khan, L. Gauchia and A. Emadi, "Modeling, parameterization, and benchmarking of a lithium ion electric bicycle battery," 2016 IEEE Energy Conversion Congress and Exposition (ECCE), Milwaukee, WI, 2016, pp. 1-7.

[24] F. Zhang et al., "State-of-charge estimation based on microcontroller-implemented sigma-point Kalman filter in a modular cell balancing system for Lithium-Ion battery packs," 2015 IEEE 16th Workshop on Control and Modeling for Power Electronics (COMPEL), Vancouver, BC, 2015, pp. 1-7.

[25] Y. Zhang, R. Xiong, H. He and W. Shen, "Lithium-Ion Battery Pack State of Charge and State of Energy Estimation Algorithms Using a Hardware-in-the-Loop Validation," in IEEE Transactions on Power Electronics, vol. 32, no. 6, pp. 4421-4431, June 2017.

[26] Z. Zhang, X. Cheng, Z. Lu and D. Gu, "SOC Estimation of Lithium-Ion Battery Pack Considering Balancing Current," in IEEE Transactions on Power Electronics, vol. 33, no. 3, pp. 2216-2226, March 2018.