BATTERY LIFE AND VEHICLE RANGE PREDICTION USING MATLAB

A PROJECT REPORT

Submitted by

SAATHANA.M	**715517105033**
SOMASUNDARAM.C	**715517105039**
VIVEDHA.M.J	**715517105056**
ROHITH.R	**715517105306**

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

ANNA UNIVERSITY: 600 025

BONAFIDE CERTIFICATE

It is to certify that this project report **"BATTERY LIFE AND VEHICLE RANGE PREDICTION USING MATLAB"** is the bonafide work of **"SAATHANA M, SOMASUNDARAM C, VIVEDHA M.J, ROHITH R"** who carried out the project under my supervision.

SIGNATURE

Dr. C. L.VASU

HEAD OF THE DEPARTMENT

Department of Electrical and Electronics Engineering

PSG Institute of Technology and Applied Research.

SIGNATURE

Mr. S. Ravikrishna

SUPERVISOR

Associate Professor (Sr. Gr.)

Department of Electrical and Electronics Engineering

PSG Institute of Technology and Applied Research.

Certified that the candidate was examined in the viva voce examination held on 5-8-2021.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The primary goal of a Battery Management System (BMS) is to operate the battery under safe operating conditions within its safety zone. The present state of the battery packs gives the amount of energy available in the battery which informs us how far the vehicle can be driven before it is recharged.This can be known from State of Charge (SOC) and State of Health (SOH) of the battery and it has to be accurate for taking the right decision. In this work, the main objectives are to estimate the SOC and SOH of the battery efficiently and dynamically to increase the Electric Vehicles (EV) longevity by deducing the overcharging and over-discharging of a battery. The SOC of the battery will be determined using the sigma-point Kalman filter method since the function of the battery pack is non-linear. The SOH of the battery is determined using Total-Least-Squares considering the possible errors to get the most accurate and reliable data. This method is recursive in a closed form and allows fading memory which reduces the memory requirement. These properties are essential to enhance the performance and reliability of the battery; good estimation allows us to use the entire battery pack capacity which will, in turn, reduce the size and cost of the battery pack. An accurate battery model in a simulation platform is very important to design an efficient battery-powered system. Here, an equivalent circuit model of a lithium-ion cell is developed, and the SOC and SOH estimation algorithm is tested using MATLAB software.

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. G. Chandramohan, Principal, Dr. P. V. Mohanram, Secretary, and the management of PSG Institute of Technology and Applied Research, Neelambur, Coimbatore-641062, for providing the opportunity to carry out our project work with the best infrastructure and facilities.

We extend our sincere thanks to **Dr. C. L.Vasu**, Professor and Head of the Department of Electrical and Electronics Engineering for providing consistent support throughout the project.

We express our sincere thanks to Dr. E. Malar, Professor, our project guide Mr. S. Ravikrishna, Assistant Professor (Sr. Gr.), Department of Electrical and Electronics Engineering for giving their unfailing support throughout the project work.

We express our gratitude to all the faculty and staff members, friends, and our parents for their constant support and guidance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

EV	Electric Vehicle
NiMH	Nickel Metal Hydride
BMS	Battery Management System
SOC	State of Charge
SOH	State of Health
OCV	Open Circuit Voltage
Li-ion	Lithium-ion
LiPF6	Lithium hexafluorophosphate
PBM	Physics-Based Model
ECM	Equivalent Circuit Model
ESC	Enhanced Self-Correcting
UDDS	Urban Dynamometer Driving Schedule
SPKF	Sigma Point Kalman Filter
TLS	Total Least Square
HPPC	Hybrid Pulse Power Characterization

CHAPTER 1

Introduction

1.1 Battery-Powered Vehicles

Rising crude oil prices and worldwide awareness of environmental issues have resulted in the increased development of energy storage systems. The battery is one of the most attractive energy storage systems because of its high efficiency and low pollution. Soon, batteries will play a significant role in the transportation sector regarding vehicle electrification. Batteries in electric vehicles store electricity on board with high-capacity battery packs. That battery power is used to run the electric motor and all on-board electronics. The performance of EVs can be improved by accurate design and monitoring of the level of battery packs.

By doing so, the overall lifetime of the battery and thus that of the Electric Vehicle can be significantly improved. Electrical batteries such as Lithium-ion, NiMH have the advantages of high working cell voltage, low pollution, low self-discharge rate, and high power density. Batteries are used commonly for portable utilities, hybrid electric vehicles, and industrial applications. Lithium-ion batteries are enabling a new generation of electrified vehicles to be commercialized by global automakers.

Achieving appropriate levels of product safety is a prerequisite for designing an automotive system's battery. Lithium batteries, like any other battery, when operating only deliver some part of the chemical energy stored as electrical energy, while the rest of the total energy stored is dissipated as heat due to internal resistance mechanisms called 'over potentials'. The actual capacity of a lithium battery is often much higher than the rated capacity, to guarantee durability.

Lots of research activities are going on, trying to improve different aspects of battery technology such as more cycling life, more energy and power densities, more safety, less cost, less environmental effects after recycling, and so on. Part of this development process is focused on developing efficient battery management systems (BMSs).

1.2 Battery Management Systems (BMS)

The BMS plays a critical role in battery packs, especially for lithium-ion battery chemistry. Protecting the cells from overcharge and over-discharge, controlling the temperature at the desired level, prolonging the life of the battery pack, guaranteeing overtime safety, and indicating the available power and energy of the battery are the key functionalities of a BMS. Two important concepts of a BMS are:

- State-of-Charge (SOC)
- State-of-Health (SOH)

Battery SOC and SOH are variables that should be determined precisely to use the battery optimally and safely. Batteries are time-varying systems that behave very differently in various states. In other words, the internal states of a battery would indicate to us what to expect from it.

A BMS is responsible for different tasks such as protecting cells from damage, prolonging the life of the battery, and maintaining the battery in a state in which it can meet the requirements of the application for which it was designed. In one sentence, BMS is for ‘optimal and safe use of a battery by monitoring and control’.

1.3 Battery State Indicators

The monitoring process consists of real-time measurements of variables like current, voltage, and temperature, which are used to calculate some

‘unmeasurable’ variables like battery state-of-charge (SOC) and state-of-health (SOH).

1.3.1 State Of Charge

In simple terms, SOC is defined as an indicator of the remaining battery energy. SOC estimation is a fundamental challenge for battery use. The SOC of a battery, which is used to describe its remaining capacity, is a very important parameter for a control strategy. In general, the SOC of a battery is defined as the ratio of its current capacity (Q(t)) to the nominal capacity (Q(n)). The nominal capacity is given by the manufacturer and represents the maximum amount of charge that can be stored in the battery. The SOC can be defined as follows:

$$\mathbf{SOC(t) = Q(t) / Q(n)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.1)$$

There are many techniques in the literature developed for battery SOC estimation, and a number of them are explained in this chapter. Accurate estimation of the SOC remains very complex and is difficult to implement because battery models are limited and there are parametric uncertainties. Here different approaches are introduced, from simple book-keeping techniques, like the integration of the current signal over time, to methods that exploit some behavioural challenges during the usage range. While these methods theoretically work independently, they can be combined or used together to increase the estimation precision.

The simplest method of battery SOC calculation, called 'coulomb counting', works based on the integration of the battery current overtime to calculate the amount of energy flowing through the battery. This method is generally precise in the short term, but the integration leads to difficulties for long-term usage. The addition of small measurement errors or biases over time

can lead to large errors in this technique especially in real applications without full cycles, using less accurate sensors with more noise. Another issue can be the need for a precise initial condition, as this information might not be given at any point in practice. Because of these practical limitations, more advanced battery SOC estimation techniques have been developed.

1.3.1.1 Overview of SOC Estimation

1.3.1.1. a) Direct Measurement

Direct measurement methods refer to some physical battery properties such as the terminal voltage and impedance. Many different direct methods have been employed: open circuit voltage method, terminal voltage method, impedance measurement method, and impedance spectroscopy method.

1.3.1.1. b) Open Circuit Voltage Method

There is approximately a linear relationship between the SOC of the lead-acid battery and its open-circuit voltage (OCV) given by:

$$V_{OC}(t) = a_1 \times SOC(t) + a_0 \quad \text{.....(1.2)}$$

The OCV method based on the OCV of batteries is proportional to the SOC when they are disconnected from the loads for a period longer than two hours. However, such a long disconnection time may be too long to be implemented for the battery. The OCV relationship with SOC was determined from applying a pulse load on the Li-ion battery, then allowing the battery to reach equilibrium.

1.3.1.1. c) Terminal Voltage Method

The terminal voltage method is based on the terminal voltage drops because of the internal impedances when the battery is discharging, so the Electro-Motive Force (EMF) of the battery is proportional to the terminal voltage. Since the EMF of a battery is approximately linearly proportional to the SOC, the terminal voltage of the battery is also approximately linearly proportional to the SOC. The terminal voltage method has been employed at different discharge currents and temperatures. But at the end of battery discharge, the estimated error of the terminal voltage method is large, because the terminal voltage of the battery suddenly drops at the end of discharge.

1.3.1.1. d) Impedance Method

Among the techniques which have been employed, impedance measurements provide knowledge of several parameters, the magnitudes of which may depend on the SOC of the battery. Although the impedance parameters and their variations with SOC are not unique for all battery systems, it appears to be imperative to perform a wide range of impedance experiments for identification and use of impedance parameters for estimating the SOC of a given battery.

1.3.1.1. e) Impedance Spectroscopy Method

The impedance spectroscopy method measures battery impedances over a wide range of ac frequencies at different charge and discharge currents. The values of the model impedances are found by least-squares fitting to measured impedance values. SOC may be indirectly inferred by measuring present battery

impedances and correlating them with known impedances at various SOC levels.

1.3.1.1. f) Book-Keeping Estimation

Book-keeping estimation method uses battery discharging current data as input. This method permits to include some internal battery effects as self-discharge, capacity-loss, which do not influence and discharging efficiency. Two kinds of book-keeping estimation methods have been employed: The coulomb counting method and the modified Coulomb counting method.

1.3.1.1. g) Coulomb Counting Method

The Coulomb counting method measures the discharging current of a battery and integrates the discharging current over time to estimate SOC. Coulomb counting method is done to estimate the SOC(t), which is estimated from the discharging current, I(t), and previously estimated SOC values. SOC is calculated by the following equation:

$$SOC(t) = SOC(t\text{-}1) + (I(t) / Q_n)\Delta t \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.3)$$

1.3.1.1. h) Kalman Filtering Method

Using real-time measurement road data to estimate the SOC of a battery would normally be difficult or expensive to measure. The application of the Kalman filter method is shown to provide verifiable estimations of SOC for the battery via real-time state estimation.

Recently, a journal presented a Kalman filter-based SOC estimation method for lithium-ion batteries. Experimental results validate the effectiveness of the Kalman filter during the online application. Barbarisi et al. presented an Extended Kalman filter (EKF) to estimate the concentrations of the main

chemical species which are averaged on the thickness of the active material to obtain the SOC of the battery, by using the terminal current and voltage measurements.

Based on an Unscented Kalman Filter (UKF) theory and a comprehensive battery model, a novel SOC estimation method is proposed. The results show that the UKF method is superior to the extended Kalman filter method in SOC estimation for batteries. They presented an adaptive UKF method to estimate the SOC of a lithium-ion battery for battery electric vehicles. The adaptive adjustment of the noise covariance in the SOC estimation process is implemented by an idea of covariance matching. Followed by UKF, the Sigma Point Kalman Filter (SPKF) is best suited for highly nonlinear systems which are robust and reliable for dynamic applications.

1.3.2 State of Health

Battery SOH is defined as a quantitative indicator of the healthiness of the battery and is determined based on the battery end-of-life definition. Typically, a battery's SOH will be 100% at the time of manufacture and will decrease over time and use. However, there is not a single definition of battery end-of-life that is accepted by all scientists in this area. Consequently, different definitions exist in the literature such as “calendar life' which presents the age of a battery as months or years. So, battery end-of-life is confirmed based on a period.

$$SOH = (C_{max} / C_{rated}) * 100\% \quad (1.4)$$

Where,

C_{max} - maximum capacity of the cell

C_{rated} - rated capacity of the cell

The life of a battery is affected by different scenarios of usage. So, another definition of battery life has been proposed called 'cycling life'.The age of a battery is presented in the sense of battery use depending on the charge/discharge profile. So, battery life is calculated based on the number of cycles. Such a definition can be used in practice when the load condition is consistent and repeating.

1.3.2.1 Need for health estimates

Over time, cells in a battery pack will age and their performance will degrade. Eventually the cells will reach a point where they no longer meet the performance requirements of the battery pack, the pack's end of life is considered. Between a battery pack's beginning of life and end of life, it is important to know the present degradation status of its cells to be able to make accurate calculations of state of charge, available energy, and available power.

Normal aging is only one cause of cell failure. Failures can also occur because of cell design faults, poorly controlled manufacturing processes, or impurities in the materials used during manufacture, abuse, and uncontrolled operations. Cells that have design or manufacturing faults or are abused often appear normal for a while and then fail very rapidly. Proper battery management will prevent uncontrolled operation while the BMS is active, but has no influence over external factors or ambient conditions while the BMS is inactive.

The focus is on understanding and tracking normal aging processes as well as uncontrolled operations in the sense that overvoltage, over temperature, and so forth accelerate the normal degradation mechanisms. The battery pack can still be used during the period when it is aging normally until its end of life, but at reduced levels of performance. Internal faults and abuse, which are usually detected using other means and may require shutting down parts or all of

the battery pack for immediate servicing to prevent the propagation of the failure.

The BMS designer should work closely with the battery-cell electrochemists and production engineers to understand all of the probable failure mechanisms and their characteristics fully to be able to adapt the knowledge from this chapter into their designs. In particular, those quantities that reflect a change in the performance that the battery pack can deliver. These are indicators of battery-pack SOH. There is no universally agreed-upon definition of SOH, but the most commonly estimated quantities used to summarize battery pack health include the present total capacity and present equivalent series resistance of every cell. Accurate estimates of total capacity and equivalent series resistance allow us to compute reliable total energy and available power estimates for the battery pack over its service life.

1.3.2.1. a) Total capacity

As a battery cell ages, its total capacity Q decreases. In a lithium-ion cell, this is due primarily to unwanted side reactions that consume lithium that could otherwise be used during charge and discharge of the cell, and to structural deterioration of the electrode active materials that eliminate lithium storage sites.

This slow reduction in capacity is often referred to as capacity fade. Algorithms that track capacity fade to provide the other battery-management-system algorithms with up-to-date estimates of every cell's total capacity are required. This knowledge is critical to be able to calculate battery-pack available energy accurately, where total capacity is a major contributing factor. If coulomb counting is being used for SOC estimation, an accurate estimate of total capacity is also needed; however, if Kalman filters are being used instead, the state of charge estimates turn out to be fairly insensitive to a poor total-

capacity estimate since the built-in feedback correction mechanism can compensate for moderate errors in the total-capacity estimate. The dependence of available power estimates on the value of total capacity also turns out to be minimal.

1.3.2.1. b) Equivalent Series Resistance (ESR)

As a battery cell ages normally, its equivalent series resistance R_0 increases. This is also due primarily to unwanted side reactions and structural deterioration. The side reactions tend to form resistive films on the surface of the active material particles that impede the ionic conductivity. Structural deterioration serves electronic pathways between particles and decreases the electronic conductivity. Having an up-to-date knowledge of equivalent-series resistance is important because it turns out to be a major contributing factor to the available-power calculation. It can be a major contributing factor to the state of charge estimation for some voltage-based methods; however, for Kalman-filter-based methods, it is a minor factor, and it does not affect coulomb-counting methods at all. It does not have a significant role in available energy estimation.

1.3.2.1. c) Other cell parameters

As the cell ages, other cell-model parameter values will change as well. For example, while the open-circuit-potential relationships of each electrode remain fixed by the chemistry of the crystal structures, a cell's overall OCV relationship can change due to shifts in the stoichiometric operating windows used by each electrode that occurs when capacity is lost due to side reactions and structural deterioration. This effect tends not to be large but may need to be tracked by future BMS. Other cell-model parameter values almost certainly change as well; however, few if any present BMS make efforts to estimate the

changes. Overall, changes to the equivalent-series resistance and total capacity have the dominant impact on BMS performance.

1.3.2.2 Negative-electrode aging

Degradation in the negative electrode is examined, with ageing results visible on three scales. First, any ageing happens at the interface between the solid and the electrolyte, at the surface of the electrode active-material particles. Second, other ageing processes occur within the active material particles interiors. Third, modifications to the active materials, conductive additives, binder materials, current collectors, porosity, and other components of the overall composite electrode structure may occur as a result of ageing. Table 1.1 states the causes and effects of the negative electrode aging.

Table 1.1 - Principal aging mechanisms at the negative electrode

Cause	Effect	Leads to	Enhanced by
Continuous low-rate electrolyte decomposition reaction builds SEI	Lithium loss, impedance rise	**Capacity fade, Power fade**	High temp-eratures, high cell SOC
Solvent co-interca-lation, gas evolution and subsequent graphite exfoliation	Loss of active material, lithium loss	Capacity fade	Overcharge
Decrease of accessible surface area due to continuous SEI growth	Impedance rise	**Power fade**	High temp-eratures, high cell SOC
Changes in porosity due to volume change and SEI growth	Impedance rise, larger overpotentials	Power face	High cycling rate, high cell SOC
Contact loss of active material particles due to volume changes during cycling	Loss of active material	**Capacity fade**	High cycling rate, low cell SOC
Decomposition of binder	Lithium loss, loss of mechanical stability	Capacity fade	High cell SOC, high temperatures
Current collector corrosion	Larger over-potentials and impedance; Inhomogeneous distribution of current and potential	Power fade, Enhances other aging mechanisms	Overdischarge, low cell SOC
Metallic lithium plating and subsequent electrolyte decomposition by metallic lithium	Lithium loss (electrolyte loss)	Capacity fade (power fade)	Low temp-erature, high charge rates, geometric misfits

1.3.2.3 Positive-electrode aging

As with the negative electrode, aging occurs in three locations in the positive electrode: at the surface of the particles, within the active material particles themselves, and in the bulk positive electrode.

1.3.2.3. a) Positive electrode aging at the surface of particles

In the positive electrode, researchers have found that a film can grow on the surface of the active materials and the particles as well. In part, this is due to a chemical reaction between the solvent in the electrolyte and the positive-electrode active materials; however, this mechanism is not as pronounced as it is in negative electrodes. A bigger factor is the dissolution of metals from the electrode crystal structures into the electrolyte and products formed from these metals that can reprecipitate onto the particle surface as a high resistance film. This dissolution is accelerated by hydrofluoric acid in the electrolyte, initiated by trace amounts of water that combine with the LiPF6 salt.

1.3.2.3. b) Positive electrode aging in bulk

When lithium intercalates into and deintercalation out of the positive electrode active particles, stresses cause strains known as phase transitions that distort the shape of the crystal structure of the electrode materials without changing the overall structure itself. Transitions in phase are caused by the presence or absence of lithium in the storage sites, leading to different local molecular forces. Some of these phase transitions are normal and reversible, but others lead to collapse of the electrode structure and rapid capacity decrease due to the resulting loss of lithium storage sites. This is most common when overcharging a cell: too much lithium is removed from the positive electrode, which causes the lithium pathways to collapse.

1.3.2.3. c) Positive electrode aging in composite electrode

The composite positive electrode experiences degradation in the same ways as the composite negative electrode. Over time, the binder can decompose, the conductive additives can become oxidized, the current collector can become corroded, and contact among particles and between particles and the current collector can become lost due to volume changes.Table 1.2 shows the aging effects of the positive electrode.

Table 1.2 - Principal aging mechanisms at positive electrode

Cause	Effect	Leads to	Enhanced by
Phase transitions	Cracking of active particles	Capacity fade	High rates, high/low SOC
Structural disordering	Lithium sites lost and lithium trapped	Capacity fade	High rates, high/low SOC
Metal dissolution and/or electrolyte decomposition	Migration of soluble species,	Capacity fade	High/low SOC, high temp-erature
	Re-precipitation of new phases,	Power fade	
	Surface layer formation	Power fade	
Electrolyte decomposition	Gas evolution		High temp-erature
Binder decomposition	Loss of contact	Power fade	
Oxidation of conductive agent	Loss of contact	Power fade	
Corrosion of current collector	Loss of contact	Power fade	High SOC

CHAPTER 2

Literature Review

B. S. Bhangu et al., in their paper "Nonlinear Observers for Predicting State-of-Charge and State-of-Health of Lead-Acid Batteries for Hybrid-Electric Vehicles" [1] had presented an approach to estimating the SOC and SOH of a cell pack by the application of a KF and EKF respectively, only the measurements of cell terminal quantities are used as input.

Padmanaban Dheenadhayalan et al., in their work "A Novel Method for Estimation of State of Charge of Lithium-ion Battery using Extended Kalman Filter" [2] showed a demonstration and explanation of EKF which serves to accurately estimate the value of SOC under noise is carried out. An equivalent circuit of lithium-ion batteries was identified and Extended Kalman Filter was designed for SOC estimation purposes. Estimation of SOC was simulated by different methods and the accuracy of the designed EKF was validated by statistical analysis. Finally, they conclude that EKF is an effective approach to estimate the SOC for lithium-ion batteries as it has a high level of immunity towards noise that occurs during vehicle operation.

He Lin et al., in their paper, "State of charge estimation by finite difference extended Kalman filter with HPPC parameters identification" [3] had proposed a finite-difference extended Kalman filter (FDEKF) with Hybrid Pulse Power Characterization (HPPC) parameters identification to estimate the SOC. The finite difference (FD) algorithm is beneficial to compute the partial derivative of a nonlinear function, which can reduce the linearization error generated by the extended Kalman filter (EKF) both constant current discharge (CCD) and urban dynamometer driving schedule (UDDS), are utilized to validate the FDEKF algorithm. Comparing convergence rate and accuracy

between the FDEKF and the EKF algorithm, it can be seen that the former is a better candidate to estimate the SOC.

Wei He et al., in their work “State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation” [4] had developed an artificial neural network-based battery model to estimate the SOC, based on the measured current and voltage. An unscented Kalman filter is used to reduce the errors in the neural network-based SOC estimation.

Zhiwei He et al., in their work “Battery Model Parameters Estimation with the Sigma Point Kalman Filter” [5] had developed a Sigma Point Kalman Filter based battery model parameters estimation method. The parameters can be estimated accurately while efficiently. Compared to the classical least squares method, this method consumes much less memory and calculation time, which makes it suitable for embedded applications.

Khalid Khan et al., in their paper “Real-Time Application of Battery State of Charge and State of Health Estimation” [6] presented a method of estimating SOC and SOH through the incorporation of current integration, voltage translation, and Ah-throughput. SOC estimation utilizing current integration is inadequate due to the accumulation of errors throughout usage. Thus voltage translation of SOC is applied to rectify the current integration method which improves the accuracy of estimation. Voltage translation data is obtained by subjecting the battery to a hybrid pulse power characterization (HPPC) test. The Battery State of Health was determined by a semi-empirical model combined with an accumulated Ah-throughput method. Battery state of charge was employed as an input to estimate damages accumulated to battery aging through a real-time model. This method allows us to monitor battery operating conditions instantaneously.

ZenabMoussalli et al., in their work “State of Charge estimation algorithms in Lithium-ion battery-powered Electric Vehicles” [7] discussed the different methods of SOC estimation and their weakness and strengths. Finally, they have proposed that the SOC of the battery be kept within appropriate limits; 20%≤ SOC(t) ≤95% to keep the battery performance.

Kong Soon Ng et al., in their paper “Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries” [8] had proposed an enhanced estimation method based on coulomb counting, is proposed to improve the estimation accuracy.

Chi Nguyen Van and Thuy Nguyen Vinh, in their paper “SOC Estimation of the Lithium-Ion Battery Pack using a Sigma Point Kalman Filter Based on a Cell’s Second-Order Dynamic Model” [9] discussed the state of charge (SOC) estimation of a lithium-ion battery pack (LiBP) connected by some cells in series and parallel. The voltage noise, noise, and current bias of current through the LiBP are taken into account in the SOC estimation problem. The second dynamic order model of the cell is used to estimate the SOC of the LiBP. By applying the sigma point Kalman filter (SPKF), the average SOC of the pack and bias current of current measurement are estimated by first estimator; the second estimator estimates the SOC differences of the cell modules from average SOC of the pack. The SOC of the cell modules is the sum of the average SOCs of the pack and the SOC differences. By only using two estimators, the calculation complexity for SOC estimation is reduced; this is very useful for the LiBP, which has the number of cells connected in a large series.

Sabine Piller et al., in their paper “Methods for state-of-charge determination and their applications” [10] had discussed methods for SOC determination and a relationship between the advantages of different methods and most common applications. Finally, they have suggested that Ah counting is

the most direct and transparent method and quite easily implemented and also gives satisfyingly accurate results for short-time applications, especially if used in the range of low to medium SOC.

Venu Sangwan et al., in their paper "Estimation of Model Parameters and State-of-Charge for Battery Management System of Li-ion Battery in EVs" [11] presented a method of identifying the unknown parameters of the battery model using the least square algorithm for Dynamic Stress Test (DST), validation of estimation is conducted for Federal Urban Driving Schedule (FUDS) and concluded that the error between predicted terminal voltage form model and voltage from DST profile was less than 0.08V for defined conditions. SOC estimation, recursive Bayesian estimation method based Extended Kalman Filtering (EKF), and Sigma-Point Kalman Filtering (SPKF) approaches were adopted. To quantify the performance of the estimators, Root Mean Square Error (RMSE) and execution time at different temperatures were evaluated. Finally, they conclude that the maximum error in the case of EKF is 2.43% whereas SPKF is 1.2% and the maximum execution time taken by EKF is 3.57 sec whereas SPKF is 4.53 sec. The results reported that SPKF provides accurate and robust SOC estimation in comparison to EK.

Nikolaos Wassiliadisa et al., in their paper "Revisiting the dual extended Kalman filter for battery state-of-charge and state-of-health estimation: A use-case life cycle analysis" [12] had developed a dual extended Kalman filter (DEKF), it consists of two extended Kalman filters (EKFs), that synchronously estimate both the battery states and parameters. This work investigates the DEKF performance from a high-level perspective, involving different load dynamics and SOH stages. A numerical optimization-based approach for the crucial filter parameterization is employed. The DEKF partly improves the accuracy of the SOC estimation compared to the simple EKF over battery lifetime within the operational limits of an automotive application. The work

investigates the DEKF performance from a high-level perspective, involving different load dynamics and SOH stages. A numerical optimization-based approach for the crucial filter parameterization is employed. Finally, they conclude that the DEKF partly improves the accuracy of the SOC estimation compared to the simple EKF over battery lifetime within the operational limits of an automotive application.

Bingjun Xiao et al., in their work "A Universal State-of-Charge Algorithm for Batteries" [13] had developed an efficient yet accurate OCV algorithm that applies to all types of batteries. They have used a linear system analysis but without a circuit model, the OCV is calculated based on the sampled terminal voltage and discharge current of the battery. The algorithm is numerically stable, robust to history-dependent error, and obtains SOC with less than 4% error compared to a detailed battery simulation for a variety of batteries.

Fei Zhang et al., in their paper "A Battery State of Charge Estimation Method with Extended Kalman Filter" [14] had proposed a battery State of Charge (SOC) estimation method based on the extended Kalman filter. The proposed model assumption matches better with the real battery behaviour. A battery is modelled as a nonlinear system, with the SOC defined as a system state and the extended Kalman filter is applied to estimate SOC directly for a lithium battery pack.

Wenhui Zheng et al., in their paper "State of Charge Estimation for Power Lithium-Ion Battery Using a Fuzzy Logic Sliding Mode Observer" [15] proposed a new fuzzy logic sliding mode observer for SOC estimation. The second-order resistor-capacitor equivalent circuit model is used to describe the discharging/charging behaviour of the battery. The exponential fitting method is applied to determine the parameters of the model. The fuzzy logic controller is introduced to improve the performance of the sliding mode observers forming

the fuzzy logic sliding mode observer (FLSMO). Finally conclude that the FLSMO algorithm has better performance than the sliding mode observer and the extended Kalman filter in terms of robustness against measurement noise and parameter disturbances.

Yuan Zou et al., in their work “Combined State of Charge and State of Health estimation over lithium-ion battery cell cycle lifespan for electric vehicles” [16] developed a first-order RC(resistor-capacitor) model to analyse the SOC dependency of the nominal parameters and the performance degradation of the nominal model over the battery lifetime is quantified. Two extended Kalman filters with different time scales are used for combined SOC/SOH monitoring.

CHAPTER 3

Equivalent Cell Model

3.1 Simulation of Battery Cells

One of the important functions of a battery management system is to compute estimates of a variety of fundamental quantities such as SOC, SOH, remaining power, and available energy. The methods to produce these estimates require highly accurate and computationally simple mathematical sets of equations or models of cell current/voltage dynamics. These applications require insights into cell internal electrochemical dynamics. Two fundamentally different kinds of models can be used for these applications; physics-based model and equivalent-circuit model. Either of these models can be used to describe the operation of lithium-ion battery cells.

3.1.1 Physics-Based Models (PBMs)

Physics-based models are developed based on the observation of electrochemical variations that occurs in a lithium-ion cell during its operation. The governing equations of this model are derived from physical laws that describe the internal cell electrochemical variables in response to an input stimulus. These variables can then be used to compute the cell voltage. This model can predict a wide range of operating conditions and can predict the internal electrochemical state of the cell. But it has convergence issues as it is expressed as coupled partial-differential equations.

3.1.2 Equivalent-Circuit Models (ECMs)

ECMs use an electrical circuit analogy to explain the behaviour of the lithium-ion cell. This makes use of simple electrical equations to explain the complex electrochemical reactions. Laboratory tests are conducted to collect

data for optimizing the circuit element parameter values. These data are used to model the cell to closely match the current/voltage behaviours of the true cell. Because ECMs amount to an empirical fit of data collected from a cell to a model structure comprising electronic circuit components, they share the same features of other types of the curve fit. These optimized curve-fitted ECM models cannot be used in applications where the cell is operated very differently from the lab-test scenario. ECMs can yield fast and robust simulations of the cell input/output (current/voltage) behaviour but they cannot predict the internal electrochemical states.

In this project, an equivalent circuit model of Li-ion battery cell is used as the input for estimating SOC and SOH of the battery. ECMs use electrical-circuit analogues to define a behavioural or phenomenological approximation to how a cell's voltage responds to different input-current stimuli.

That is, knowledge of common electronic elements is applied to define a circuit that has behaviour that closely matches the observed behaviour of the battery cell. The equations that describe the circuit therefore also closely describe the observed operation of the cell. Such models are called equivalent-circuit models.

At present, the majority of battery management systems for large battery packs use some kind of equivalent-circuit cell model as a basis for maintaining the proper operating boundaries for the cells and for estimating critical internal cell states. The simplicity and robustness of equivalent-circuit models are the main reasons for this.

3.2 Components of Equivalent Circuit Model

The Equivalent Circuit Model of the Li-ion cell model that is developed in this project consists of several components, namely: a dependent voltage source, a series resistor, a diffusion voltage component, and a hysteresis voltage component.

3.2.1 Dependent Voltage Source

The terminal voltage of the battery is higher when it is fully charged and it is lower when it is fully discharged. Therefore there exists a relationship between SOC and the terminal voltage of the battery. But the terminal voltage also depends on the dynamic factors. However, the equilibrium unloaded rest voltage or open-circuit voltage of a fully charged cell is higher than that of a fully discharged cell. This relation is depicted from the test taken in 3 different lithium-ion cells shown in Fig. 3.1.

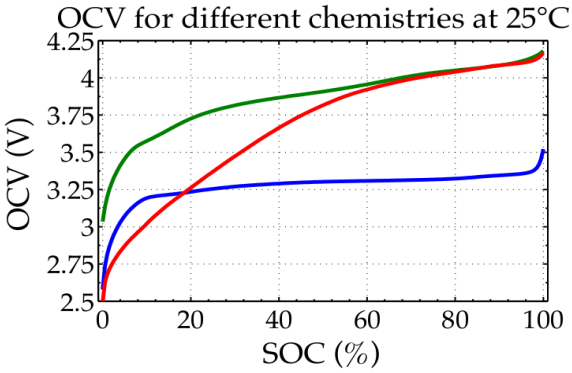

Fig. 3.1 - Variation of OCV with SOC

3.2.2 Equivalent Series Resistance

Equivalent series resistance is added to include the dynamic behaviour of the cell when it is loaded or charged. When a cell is loaded, the terminal voltage drops below the open-circuit voltage of the cell and while charging, the terminal

voltage shoots above the open-circuit voltage. These two dynamic behaviours can be included in the cell model by adding series of resistance to the dependent voltage source.

3.2.3 Diffusion Voltage Component

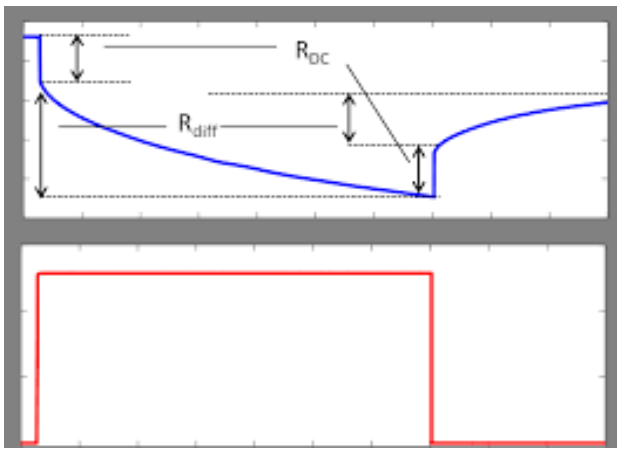

Fig. 3.2 - Observation of Diffusion in Practical Lithium-ion Cell

In Fig. 3.2, the red curve shows the discharge pulse to a lithium-ion battery and the blue curve shows the variation of terminal voltage of the battery. Here, the slow variation in terminal voltage of the battery with the discharge is called the diffusion voltage caused by diffusion in the battery (R_{diff}). And the immediate rise in the terminal voltage after the discharge is removed is due to the stored charge in between the junction of the electrodes (R_{DC}).

To include the slow variation that is observed during pulse discharge of a Li-ion cell, a third component is included. This is caused by the slow diffusion of Lithium in Li-ion cells. And this slow voltage variation is called diffusion voltage (V_{diff}). This effect can be closely approximated by one or more parallel capacitor-resistor sub circuits.

For an exact equivalence, an infinite number of resistor-capacitor networks are needed; but, the circuit can often be modelled very well over some frequency range of interest using a small number of resistor-capacitor pairs.

3.2.4 Hysteresis Voltage Component

Hysteresis voltage in a lithium-ion cell can be realized by observing a practical cell, in which the cell voltage settles down to a different value than that of the OCV of the cell. This variation depends on the recent history of the cell usage; whether it was recently charged or discharged. If it was recently charged, then the settled voltage is greater than OCV. If it was previously discharged, then the settled voltage is lesser than OCV. This hysteresis voltage is shown by the following Fig. 3.3.

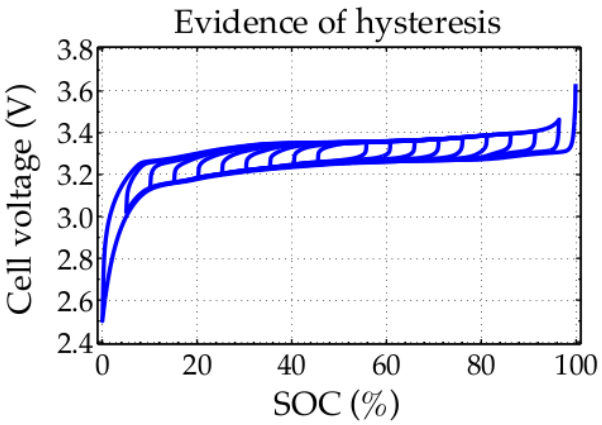

Fig. 3.3 - Evidence of Hysteresis

3.3 Enhanced Self-Correcting Cell Model

The model which includes all the above-mentioned components is called the Enhanced Self-Correcting Cell Model.

- *Enhanced* - includes hysteresis effect.
- *Self-Correcting* - includes both static and dynamic characteristics.

The complete equivalent circuit model of the enhanced self-correcting cell model is given in Fig. 3.4. This model serves as the input for the estimation algorithms used in our project. All the components in this circuit do not have constant values but are a table of values depending on the SOC and operating temperature of the cell.

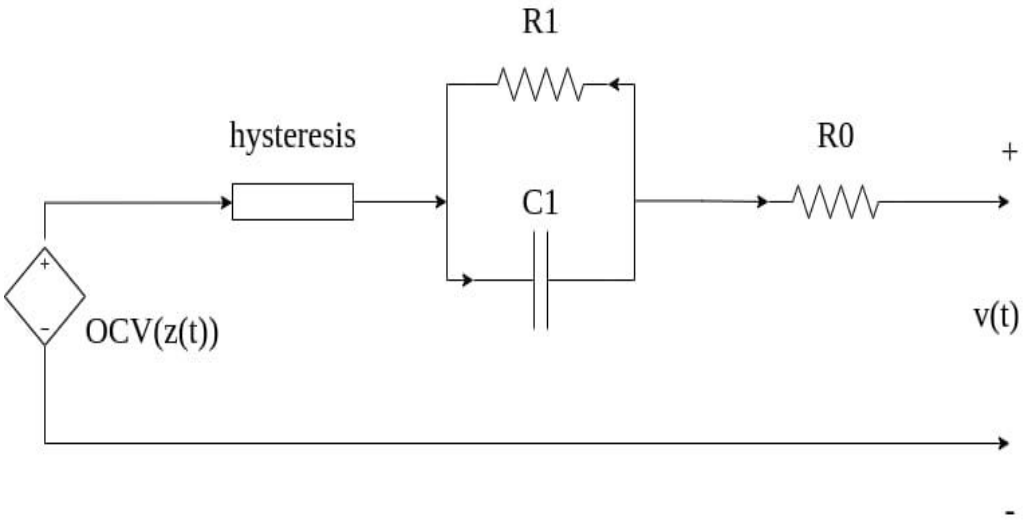

Fig. 3.4 - Equivalent Circuit Diagram of Enhanced Self-Correcting Cell model

3.3.1 Mathematical Equations describing the ESC cell model

The state equation of the ESC model is given by the following matrix equation (3.1)

$$\begin{bmatrix} z[k+1] \\ i_R[k+1] \\ h[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & A_{RC} & 0 \\ 0 & 0 & A_H[k] \end{bmatrix} \begin{bmatrix} z[k] \\ i_R[k] \\ h[k] \end{bmatrix} + \begin{bmatrix} -\frac{\eta[k]\Delta t}{Q} & 0 \\ B_{RC} & 0 \\ 0 & A_H[k]-1 \end{bmatrix} \begin{bmatrix} i[k] \\ sgn(i[k]) \end{bmatrix} \quad \text{..... (3.1)}$$

Where A_{RC} and B_{RC} are matrices depending on the number of RC sub circuits and are given by equation (3.2)

$$i_R[k+1] = \begin{bmatrix} F_1 & 0 & ... \\ 0 & F_2 & ... \\ \vdots & \vdots & \ddots \end{bmatrix} i_R[k] + \begin{bmatrix} (1-F_1) \\ (1-F_2) \\ \vdots \end{bmatrix} \quad \text{..... (3.2)}$$

And $F_j = \exp\left(\frac{-\Delta t}{R_j C_j}\right)$, where j is the number of RC pairs.

The output equation of the ESC model is given by equation (3.3)

$$v[k] = OCV(z[k],T[k]) + M_0 s[k] + Mh[k] + \sum R_j i_{Rj}[k] + R_0 i[k] \quad \text{..... (3.3)}$$

Where,	$OCV\,(z[k], T[k])$	-	Open Circuit Voltage of the cell
	$M_0 s[k]$	-	Instantaneous hysteresis voltage
	$Mh[k]$	-	Hysteresis voltage
	$\sum R_j i_{Rj}[k]$	-	Voltage drop across RC sub circuit
	$R_0 i[k]$	-	Voltage drop across series resistance

3.4 Collecting Cell Data for Modelling

Parameter values for the ESC model are found by fitting the model equations to data collected from experiments performed on the cells of interest. These experiments are conducted with the aid of specially designed laboratory equipment known as battery-cell cyclers or battery-cell test equipment. Besides, most tests must be conducted in controlled temperature settings. This is maintained using environmental chambers.

A cell's open-circuit voltage is a static function of its state of charge and temperature. All other aspects of a cell's performance are dynamic in some sense. So, separate experiments have to be performed to collect data for the OCV versus SOC relationship and the dynamic relationship.

3.4.1 Lab tests to determine OCV relationship

The general idea is straightforward. Before the start of the test, the cell must be fully charged. Then, the cell is very slowly discharged to a minimum operating voltage while continuously measuring cell voltage and accumulated ampere-hours discharged. The cell is then very slowly charged to a maximum operating voltage while continuously measuring cell voltage and accumulated ampere-hours charged.

The purpose of the slow rate is to minimize the excitation of the dynamic parts of the cell model. C/30 rate is used, which is a compromise between the desire to have zero current to have a true equilibrium and the practical

realization that the test will already require on the order of 60 hours completing with a C/30 discharge followed by a C/30 charge.

3.4.2 Lab tests to determine the dynamic relationship

The cell must be exercised with profiles of current versus time that are representative of the final application for the resulting model to work best in that application. The Urban Dynamometer Driving Schedule (UDDS) profile is used to represent the driving pattern of the electric vehicle. An example UDDS profile is given in Fig. 3.5. Before any dynamic testing, the cell is first fully charged until it is at 100 % state of charge at 25˚C. Then experiments are done at several temperatures spread over the operational range of the cell.

Both the tests are done at different temperatures so that the model can be used in real-time electric vehicle applications over a wide range of operating conditions.

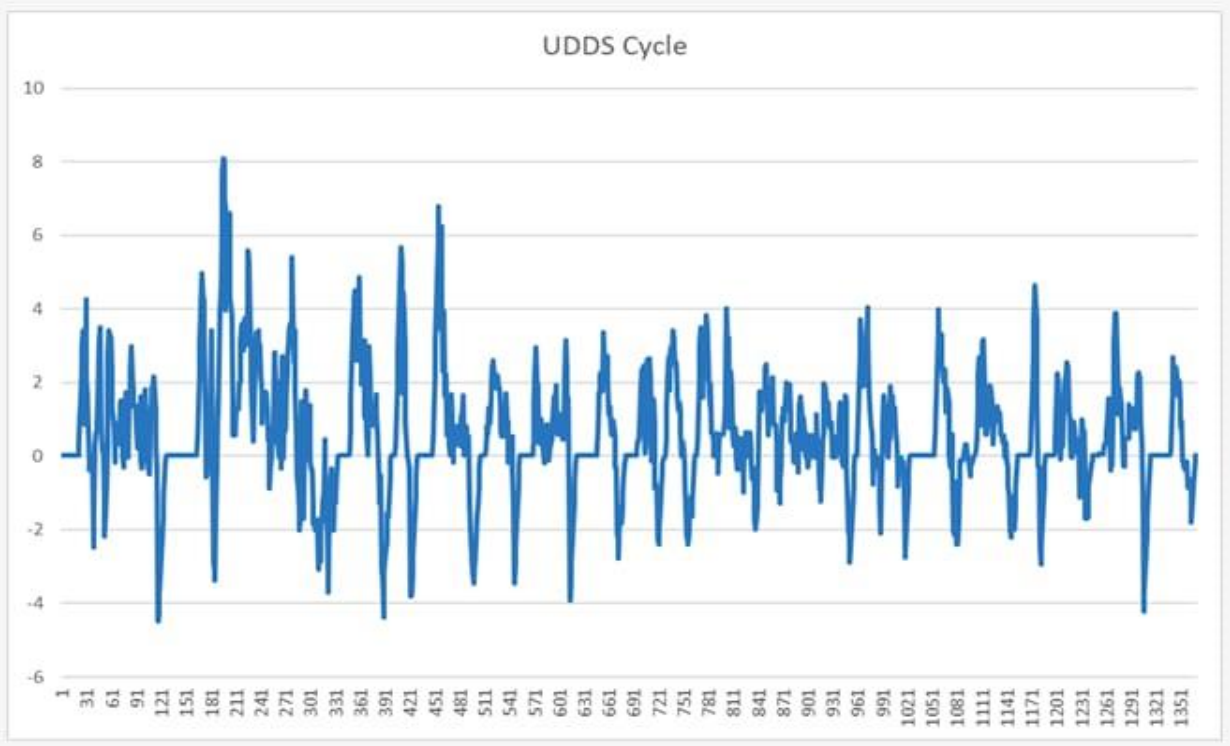

Fig. 3.5 - Current Profile of UDDS Drive Cycle

3.5 Creating ESC model using MATLAB

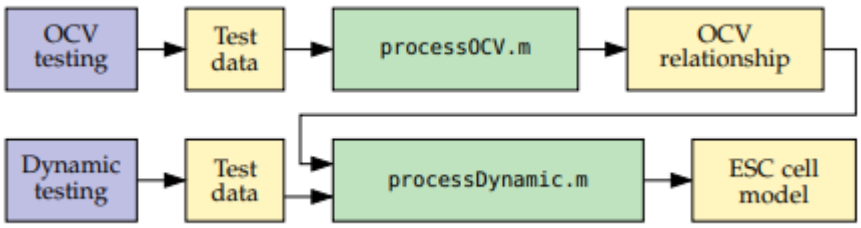

Fig. 3.6 - Flow Chart for the Creation of ESC Cell Model

The flow chart in Fig.3.6 depicts the overall process for creating an enhanced self-correcting (ESC) cell model. The blue boxes indicate laboratory processes; the yellow boxes indicate data files, and the green boxes indicate processing by MATLAB functions. The "OCV testing" box denotes OCV relationship cell tests run at several temperatures. The "Dynamic testing" box denotes dynamic relationship cell tests also run at several temperatures.

The "Test data" boxes abstractly denote the output of the laboratory testing. The procesSOCV.m function processes the raw cell-test data to produce an OCV relationship. The processDynamic.m function likewise processes the raw cell-test data to produce the final ESC cell model.

3.6 Output of ESC Cell Model

model

1x1 struct with 18 fields

Field	Value
OCV0	*1x201 double*
OCVrel	*1x201 double*
SOC	*1x201 double*
OCV	*1x178 double*
SOC0	*1x178 double*
SOCrel	*1x178 double*
OCVeta	[0.9805;0.9891;0.9934;0.9931;0.9924;0.9834;0.9926;0.9864]
OCVQ	[5.2204;5.2322;5.2501;5.2463;5.2521;5.2127;5.2763;5.2417]
name	'E2'
temps	[-25,-15,-5,5,15,25,35,45]
etaParam	[-1.3799,0.3899,0.8174,0.9776,0.9892,0.9929,0.9958,0.9976]
QParam	[5.1230,5.1234,5.1167,5.1555,5.1298,5.1344,5.1514,5.1255]
GParam	[56.5073,5.1269,250.0000,106.2428,82.1689,61.7498,58.6469,1.0000]
M0Param	[0.0267,0.0129,0.0066,0,2.1149e-04,0.0025,0.0035,0.0026]
MParam	[0.2471,0.1485,0.0817,0.0805,0.0471,0.0443,0.0339,0.0982]
R0Param	[0.2157,0.1269,0.0655,0.0313,0.0173,0.0112,0.0081,0.0058]
RCParam	[0.9019;0.8745;1.2296;1.8220;2.3494;2.4107;2.5693;2.7938]
RParam	[0.0988;0.0487;0.0209;0.0071;0.0043;0.0025;0.0020;0.0022]

Fig. 3.7 - Calculated Values of ESC Model from MATLAB

The values shown in Fig. 3.7 are the calculated model values of the ESC cell model for **lithium iron phosphate ($LiFePO_4$)** or **LFP** battery.

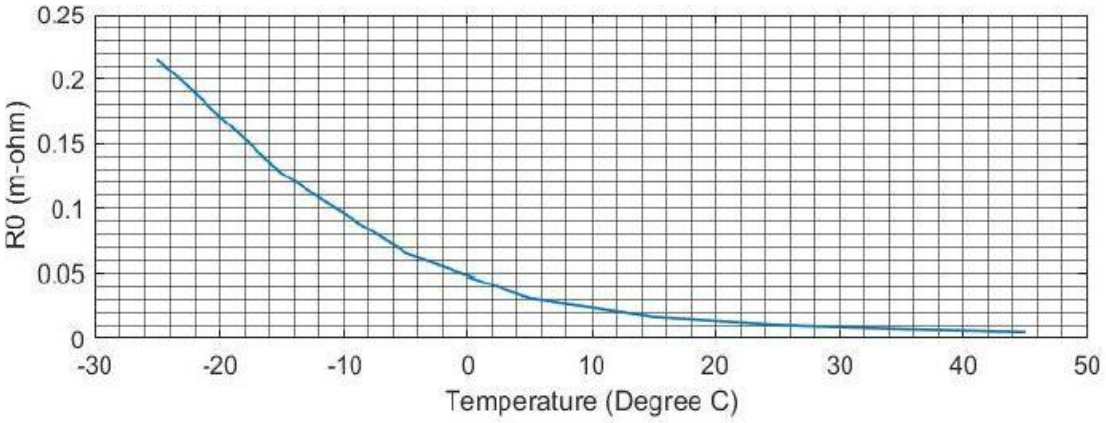

Fig. 3.8 - Plot between Series Resistance and Temperature of ESC Model

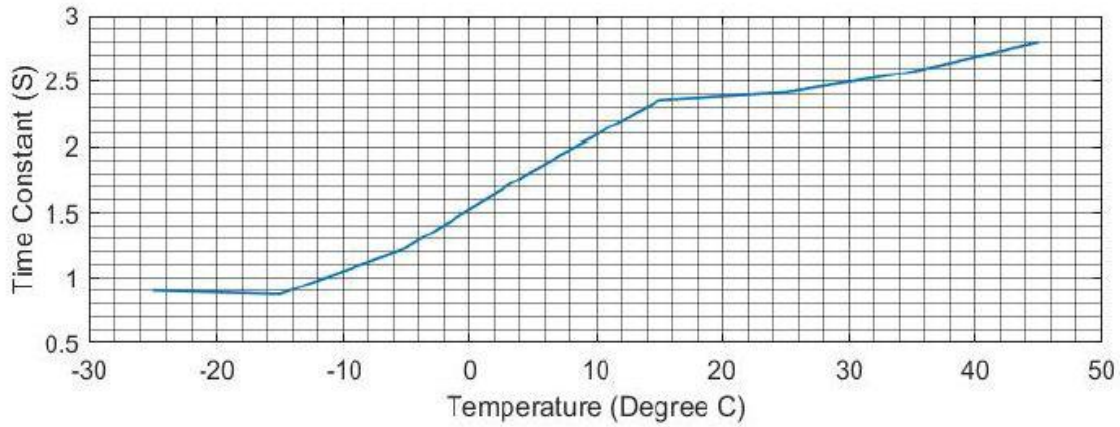

Fig. 3.9 - Plot between RC Time Constant and Temperature of ESC Model

The output graphs in Fig. 3.8 and Fig. 3.9 shows the variation of series resistance and RC time constant concerning variation in ambient temperature respectively.

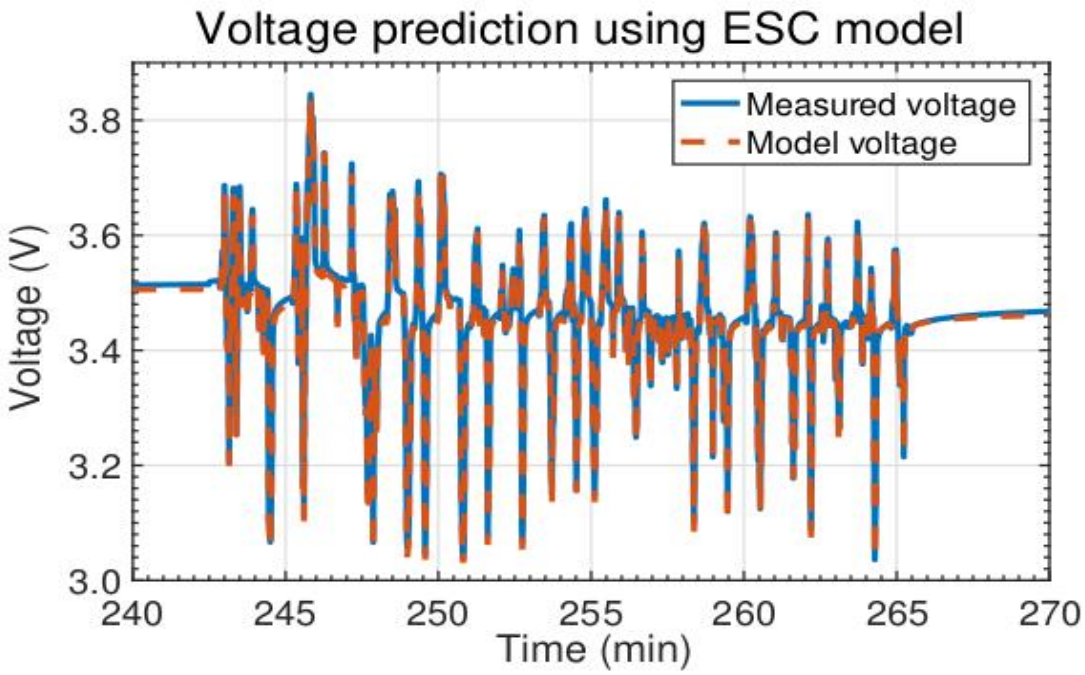

Fig. 3.10 - Plot between Measured Voltage during the Test and Estimated Voltage of the Cell Model

Fig. 3.10 gives the comparative result of the measured value of the battery when it was under test to the terminal voltage developed by the ESC model for the same current stimulus. This gives us the closeness of the developed ESC model to the practical cell for the same input.

CHAPTER - 4

Battery State Estimation

4.1 SOC Estimation

The various algorithms are initialised when the battery-management device programme is started. As the driver turns the key to the "on" location in any vehicle application, this happens, and it can also necessitate taking any initial measurements, loading stored parameter values from a non-volatile memory, doing safety tests, and closing the contactors.

The BMS then enters its main loop control, which loops at a constant interval after all of these initializations. The current in the battery pack, cell voltages, and temperatures are all determined first. The condition of each battery cell is then calculated, which involves calculating a SOC calculation. Then, each time, a health prediction for each cell is revised. At this stage, cells with unequal SOCs would be matched, and also the battery-pack energy and power limits are calculated and transmitted to the load-management system.

Finally, when the application is terminated, the contactors are reopened and data showing the current state of the battery pack is saved to non-volatile memory, which must be reloaded and used the next time the battery pack is needed. The cell's SOC is an integral part of the battery-model state vector. A SOC estimate is required as input to balancing strategies and to both energy and power calculations.

Although estimating the entire battery-model state vector can be helpful, SOC is estimated as the first step. Any basic methods produce inaccurate figures, although more sophisticated methods produce excellent results.

4.2 Estimation Techniques of SOC

Estimation techniques can be of methods mainly focused on voltage measurements, or methods purely based on current measurements, or a more general approach combining both voltage and current measurements with an effective cell model.

4.2.1 Voltage-based methods to estimate SOC

Based on the ESC cell model, cell's terminal voltage is given as a function of its SOC via equation (4.1):

$$v[k] = OCV(z[k],T[k]) + M_0 s[k] + Mh[k] + \sum R_j i_{Rj}[k] + R_0 i[k] \quad \text{..... (4.1)}$$

Assuming the cell is at rest and neglecting the effect of hysteresis, the basic relationship can be obtained as v[k] ≈ OCV (z[k]). The open-circuit voltage can be calculated as a function of SOC with the help of a lookup table. Approximating state of charge with the help of lookup table and using the present terminal voltage is truly accurate. However, it is nonetheless a simple operation. It is okay to use this method even when the cell is under load to find an approximate state of charge from the loaded terminal voltage. However, doing so gives poor results.

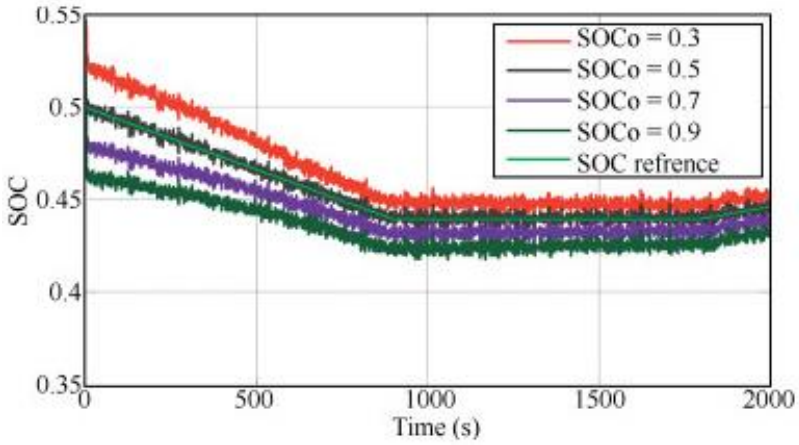

Fig. 4.1 Variation of SOC from Minimum to Maximum

The variance of the battery's State of Charge with respect to time (i.e. the battery's lifecycle) is shown in Fig. 4.1. The values on the vertical axis are the SoC values expressed as a proportion of 100%.

4.2.2 Poor, current-based method to estimate SOC

To use current as the primary estimator of state of charge, the following equation (4.2) is used:

$$z_k = z_0 - (\Delta t / Q) \sum_{j=0}^{k-1} \eta_j i_j \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.2)$$

This equation is, in fact, precise. It measures the amount of charge added to or removed from the cell, normalizes this net amount by total capacity, and updates the state of charge based on this net flow of charge.

The noise and nonlinear errors should be assumed to have a zero mean and thus have little impact on the state of charge estimate's expected value. They do, however, allow the estimate's uncertainty to rise over time. Since bias, self-discharge, and leakage errors do not have a zero mean, the state of charge calculation will continue to decay, and measurement error variance will further increase the uncertainty of the SOC estimate.

4.2.3 Model-based state estimation

A voltage-only or a current-only device may be replaced by combining the approaches in either way. It is done by using a model of cell input/output - current/voltage behaviour in a model-based estimation method. The resulting method will be able to estimate SOC as well as all of the model's internal states, along with providing some additional benefits. The electrical current received by the cell is its input, and the cell's output is its terminal-voltage response. Inside the cell, there is a true SOC value and a true set of diffusion currents and hysteresis voltages.

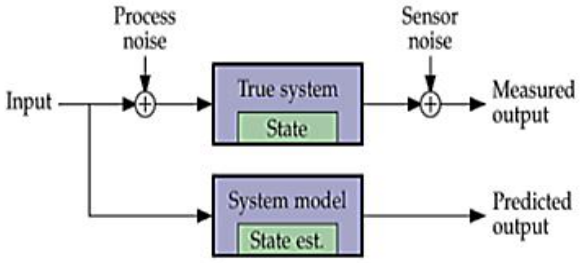

Fig. 4.2 General Block Diagram of Kalman Filter

The Fig. 4.2 shows the general block diagram representation of SPKF. It is clear from the figure that the Kalman filter compares the true system and model to produce respective outputs.

Since the system's true state cannot be calculated directly, model-based calculations are used to determine the input given to the true system i.e. the current and it propagates the same input into the model of the system. The model-based method is analogous to coulomb counting.

Extreme caution is needed while applying feedback. The voltage-prediction error will rise due to one or many things. State prediction errors, which must be corrected, calculation errors that arise due to sensor noise and modelling errors are all examples of above mentioned prediction errors. Since the model is not a perfect representation of the true cell dynamics, and henceforth to account for these causes of error, the state-estimate update must be carefully computed as much as possible.

Sequential probabilistic inference

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$$
$$y_k = h(x_k, u_k, v_k) \quad \text{..... (4.3)}$$

The sequential-probabilistic-inference problem given in equation (4.3) seeks to find an efficient recursive estimate of the present state of the dynamic

system using knowledge of all inputs and measurements of all outputs up until time k. The solution will be sequential in the sense that it implements a recursion that computes the new estimate based on the prior estimate and on the new information measured at this time step; hence, a sequence of steps provides a sequence of estimates. The solution is probabilistic in the sense that it must take into account the randomness of the process noise and the sensor noise when computing the estimates.

4.3 The six-step process of Kalman filter

1. The state estimate: The best approximation of this current state value is computed, which is $\hat{x}$ + *k*, at the end of each iteration.

2. The covariance estimate: The covariance matrix gives the uncertainty of $\hat{x}$ + *k* which could be used to compute confidence intervals or error bounds on the estimate.

General step 1a: State-prediction time update.

Each time step, the computed prediction is updated of this present value of $\hat{x}$+k based on prior information using equation (4.4).

$$\hat{x}_k^{-} = \mathbb{E}[x_k \mid \mathbb{Y}_{k-1}]$$

$$= \mathbb{E}[f(x_{k-1}, u_{k-1}, w_{k-1} \mid \mathbb{Y}_{k-1}] \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.4)$$

General step 1b: Error-covariance time update.

Based on prior information, the predicted state-estimate error covariance matrix is determined using the equation (4.5), and therefore the system model is updated.

$$\sum_{\tilde{x},k}^{-} = \mathbb{E}[(\tilde{x}_k^{-})(\tilde{x}_k^{-})^T] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.5)$$

General step 1c: Predict system output $\hat{y}_k$.

The system's output is predicted with the help of equation (4.6) and using prior information and the model.

$$\hat{x}_k^- = \mathbb{E}[x_k \mid \mathbb{Y}_{k-1}]$$
$$= \mathbb{E}[f(x_{k-1}, u_{k-1}, w_{k-1} \mid \mathbb{Y}_{k-1}] \quad \text{..... (4.6)}$$

General step 2a: Estimator gain matrix L_k.

The estimator gain matrix L_k is computed by evaluating the subsequent equation (4.7) which is given as follows:

$$\hat{y}_k = \mathbb{E}[y_k \mid \mathbb{Y}_{k-1}]$$
$$= \mathbb{E}[h(x_k, u_k, v_k \mid \mathbb{Y}_{k-1}] \quad \text{..... (4.7)}$$

General step 2b: State-estimate measurement update.

The posterior i.e., the state after the measurement is taken, state estimate is updated by the prediction as in equation (4.8), using the gain vector L_k.

$$\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k) \text{..... (4.8)}$$

General step 2c: Error-covariance measurement update.

Finally, determine the posterior state-estimate error covariance matrix using equation 4.9.

$$\sum_{\tilde{x},k}^+ = \sum_{\tilde{x},k}^- - L_k \sum_{\tilde{y},k} L_k^T \text{..... (4.9)}$$

After performing all six stages, the estimator waits until the next sample interval, updates k, and then moves on to step 1a.

4.3.1 Deriving the six steps

To address the state estimation problem, the sigma point method is used to spread statistics through non-linearity, as well as the sequential probabilistic inference discussed previously.

For the previous time moment, the augmented posterior state estimate is first established and then the augmented posterior state-estimation-error covariance matrix. The p + 1 augmented sigma points are generated using these variables.

4.3.2 Implementing SPKF using the ESC cell model

In order to generate the prior state estimates and related noises, the augmented sigma points defining are generated in the first phase of SPKF. First, the prior approximation $\hat{x}$ is used to construct the augmented state-estimate vector xhat_a. The Cholesky component of the previous estimation-error covariance SigmaX is then calculated.

The biggest difference is that the model state and output equations are not embedded inline but is separated as independent functions.

The BMS must have a real-time clock to record the amount of time the battery pack has been idle for proper initialization. The battery cells are assumed to be in electrochemical equilibrium and that cell voltage is equal to OCV since the pack is assumed as rested for a long time before being started. So, based on the calculated OCV, the SOC figures are reset, the diffusion currents are set to zero, and keep the prior hysteresis condition because hysteresis does not change when the cell is resting.

In other case, if the pack has been resting for a short period, this method will not work well since the battery cells are not in a steady-state. It often takes tens of minutes or sometimes even hours to reach the state equilibrium. In fact, due to aggressive driving, a large regeneration pulse was sent into the battery pack just before turning the vehicle off, and then the battery management system has to store cell state estimates to non-volatile memory.

Even if the stored calculations are closely accurate, the current measured voltages would be far higher than OCV. It is actually not that easy to initialise the pack in the same way if it were in equilibrium.

Instead, it is set up and made to execute a simple time and measurement update via a simple Kalman filter for the state equations involving SOC and diffusion currents. Hysteresis voltages do not change, since the pack has been resting. A single-step execution of this Kalman filter will update the state estimate and its covariance matrix based on the total time the battery pack has been resting.

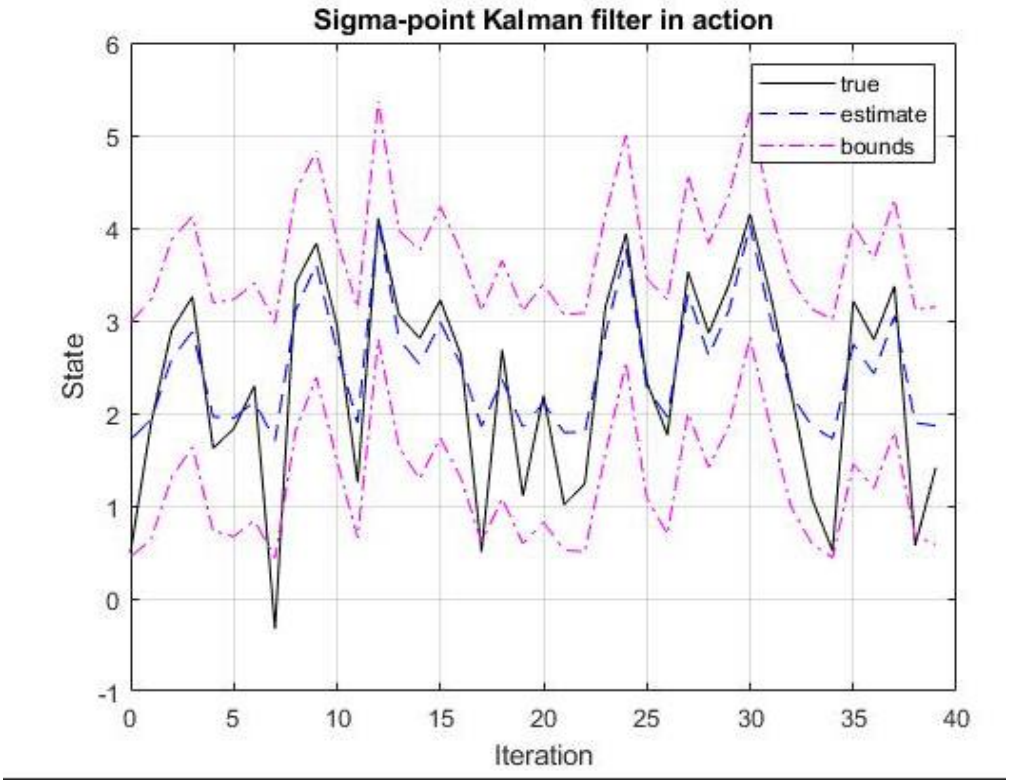

Fig. 4.4 Sigma Point Kalman Filter in Action

Fig. 4.4 shows the simple action of Sigma point Kalman Filter. It simply depicts that the estimated value trying close to match up with true values and also to lie within the bounds.

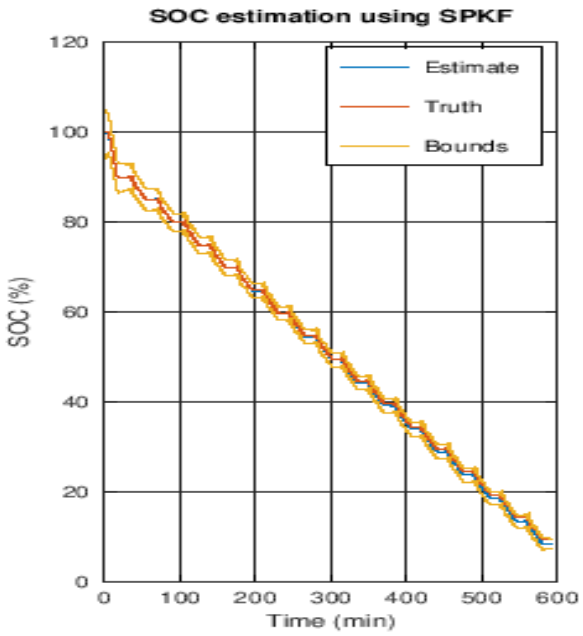

Fig. 4.5 SOC Determination using SPKF

Fig. 4.5 shows the result of SOC estimation using the Sigma Point Kalman Filter algorithm. The various SOC values are plotted against time from initial cycle of the battery. It is clear from the graph that the SOC value degrades with respect to battery cycles i.e. time. And also, it is abruptly clear that the estimated true values are well within the error bounds.

CHAPTER 5

Battery Health Estimation

5.1 Sensitivity of voltage to R_0

To estimate total capacity and resistance, input/output (current/voltage) data from the cell must be used in some form. As a measure of how easily a good estimate can be made, the sensitivity of the cell voltage signal to resistance and capacity changes can be considered. It gives an idea of the amount of estimation that is easy or difficult and reveals the voltage.

It is relatively easy to estimate the cell's equivalent series resistance because voltage measurements are highly observable. Consider the equation (5.1) of cell voltage,

$$v[k] = OCV(z[k], T[k]) + M_0 s[k] + Mh[k] + \sum R_j i_{Rj}[k] + R_0 i[k] \text{.....} \quad (5.1)$$

A cell's resistance is typically measured in milliohms, and its voltage is typically measured in volts. The absolute sensitivity of voltage to R_0 is relatively high when multiplied by cell current ik, which can be very broad (a magnitude of 0.01 would not be out of the ordinary). This means that the voltage signal resistance should be measured relatively easily.

Subtracting voltages at two adjacent time samples is one way to estimate R0. When $\Delta i_k = i_k - 1 - i_k = 0$, as during a constant-current event or cell rest, Immediate divide-by-zero problem is shown when using this form. As a result, when $|\Delta i_k|$ is small, the updates to $\hat{R}_{0,k}$ are skipped avoiding the divide-by-zero problem as well as amplification of measurement and approximation noise in Equation.

The estimate of $\hat{R}_{0,k}$ from Equation is very noisy due to the ESC cell model's imperfect fidelity in terms of true cell activity and the inaccuracy

introduced by the unique approximations. The $\hat{R}_{0,k}$ signal can be filtered to get a better estimate of R_0.

Another point to mention is that a cell's resistance is affected by both the SOC and the temperature. As scalar resistance estimation is revised, it adapts to model the cell resistance at the current SOC and temperature as those two variables shift over time. If, instead of an adaptive scalar, an adaptive function that describes the entire relationship of resistance versus state of charge is required, a functional form must be proposed, and the coefficients of the form must be adjusted based on the current state of charge, temperature, and operating resistance.

$$R_0 = R_{0,ref}\exp\left(\frac{R_{0,ref}}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.2)$$

To overcome the temperature dependence of the equation (5.2), only the $R_{0,ref}$ and $E_{R0,ref}$ values must be adjusted online. However, if the battery pack stays near one temperature for a prolonged period of time, the adaptation algorithm must be carefully designed, as adaptation will overcorrect predictions at that temperature, causing resistance predictions at other temperatures to become biased. Resistance modeling with local basis functions can also manage temperature and SOC dependency. Changing the parameters of these basis functions will only affect the resistance relationship locally, and will not affect resistance predictions at other temperatures or states of charge. However, the resistance of operating points that are not regularly visited by the application will not be changed as frequently, and the figures will become out of date over time.

5.2 Sensitivity of voltage to total capacity Q

The voltage's sensitivity to capacity is low due to the analogue hysteresis concept. As a result, individual voltage measurements provide very little capacity information. To accurately estimate total capacity, several voltage measurements must be combined, and it helps if the current is usually in the same direction for the majority of the time between measurements. That is, a big change in SOC over time makes estimating total capacity with the data collected during that time better than a small change in SOC.

Two separate, more difficult methods. First, a nonlinear Kalman filter is considered, which can be effective and has the added advantage of simultaneously estimating all cell-model parameter values as they shift over time. These Kalman-filter-based methods, on the other hand, are fairly complex, and some algorithm stability issues must be resolved before the estimates can be trusted. To estimate total power, simpler regression approaches can be used, but the most straightforward ordinary least-squares method produces biased results.

5.3 Ensuring correct convergence

The state estimate $\hat{x}_k^+$ and the parameter estimate $\widehat{\theta}_k^+$ are adjusted such that the model input/output relationship closely matches the calculated input/output results. It is possible for state and parameter estimates to deviate from their true values while still having good agreement with input/output measurements, at least for a time, due to cancellations between errors caused by incorrect state and parameter estimates.

Although the errors in $\hat{z}_k$ caused by ignoring short-term hysteresis bias and diffusion voltages prevent it from being used as the primary SOC estimator, its predicted long-term behavior in a complex system is correct, and the state of charge state in the dual and joint filters is accurate.

5.4 Weighted total least squares.

The TLS method assumes that both x_i and y_i measurements are inaccurate and models the data as $(y - \Delta y) = Q(x - \Delta x)$. The error bars on the data points reflect the degree of uncertainty in each dimension. x is a zero-mean Gaussian random vector with known element variances σ^2_{xi} and that y is a zero-mean Gaussian random vector with known element variances σ^2_{yi}, where σ^2_{xi} is not always equal to or similar to σ^2_{yi} and where both of the σ^2_{xi} and σ^2_{yi} can be distinct is assumed.

TLS tries to find an approximation $\hat{Q}_n$ of the true cell total potential Q that minimises the number of squared errors Δx_i plus Δy_i. The method here to allow us to find $\hat{Q}_n$ that minimizes the number of weighted squared errors, where the weighting accounts for calculation uncertainty is generalized.
Unfortunately, it lacks all of the desirable features of the WLS solution. Specifically,

1. The number of significant figures in the solution doubles with each iteration of the update of this Newton–Raphson process. In testing, four iterations yield results with double precision is found. And, since the metric function χ^2_{WTLS} is convex, this iterative procedure is guaranteed to converge to the global solution.
2. In the general case, there is no recursive update. This has consequences for both storage and computation. To use WTLS, the whole x and y vectors must be held, which means more storage is required as the number of measurements grows. Furthermore, as n increases, the number of computations increases. WTLS, on the other hand, is not ideally designed for embedded-system applications that must run in real-time and with little memory.

3. There is no memory recursive change that fades with time. A non-recursive fading memory weighted total least squares (FMWTLS) cost function, on the other hand, can be defined as follows:

$$X^2_{FMWTLS} = \sum_{i=1}^{n} \gamma^{n-i} \frac{(y_i - \hat{Q}_n x_i)^2}{\hat{Q}_n \sigma^2_{x_i} + \sigma^2_{y_i}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.3)$$

This cost function's Jacobian is

$$\frac{\partial X^2_{FMWTLS}}{\partial \hat{Q}_n} = 2 \sum_{i=1}^{n} \gamma^{n-i} \frac{(\hat{Q}_n x_i - y_i)\left(\hat{Q}_n y_i \sigma^2_{x_i} + x_i x_i^2{}_{y_i}\right)}{\left(\hat{Q}_n{}^2 \sigma^2_{y_i} + \sigma^2_{y_i}\right)^2} \ldots\ldots\ldots\ldots\ldots\ldots \quad (5.4)$$

The Hessian is

$$\frac{\partial^2 X^2_{FMWTLS}}{\hat{Q}_n{}^2} = 2 \sum_{i=1}^{n} \gamma^{n-i} \left(\frac{\sigma^4_{y_i} x_i^2 + \sigma^4_{x_i}\left(3\hat{Q}_n{}^2 y_i^2 - 2\hat{Q}_n{}^3 x_i y_i\right)}{\left(\hat{Q}_n{}^2 \sigma^2_{x_i} + \sigma^2_{y_i}\right)^3} \right.$$

$$\left. - \frac{\sigma^2_{x_i} \sigma^2_{y_i}\left(3\hat{Q}_n{}^2 x_i^2 - 6\hat{Q}_n x_i y_i + y_i^2\right)}{\left(\hat{Q}_n{}^2 \sigma^2_{x_i} + \sigma^2_{y_i}\right)^3} \right) \ldots\ldots\ldots\ldots\ldots \quad (5.5)$$

A Newton–Raphson search is used to find the solution to the fading-memory cost function which is used to find an approximation of Q by using the Jacobian and Hessian of this cost function.

A special case of WTLS that gives a closed-form solution with recursive updates and fading memory is given by equation (5.5). Then an approximation of a solution is given to the general WTLS problem that has the same features as the WLS solution.

5.5 Goodness of Model Fit

It's also necessary to be able to determine the degree of confidence when calculating cell total capacity estimation$\hat{Q}_n$. The variance $\sigma^2\hat{Q}_n$ of the total capacity estimate so that confidence intervals can be computed like three-sigma bounds ($\hat{Q}_n - 3\sigma\hat{Q}_n$, $\hat{Q}_n + 3\sigma\hat{Q}_n$) within which the true value of cell total capacity Q lies is qualified.

5.6 Simplified Total Least Squares

The properties of the proportional-total-least-squares solution are similar to those of the WLS solution:

1. It yields a closed-form answer forQ_n. It is not necessary to iterate advanced algorithms; only basic mathematical operations are needed.
2. In a recursive way, the solution can be conveniently computed. The three running sums $c_{1,n}$, $c_{2,n}$, and $c_{3,n}$ are kept track of. The amounts and calculate a modified total-capacity calculation as new data becomes available is updated.
3. It's easy to incorporate fading memories.
4. Unfortunately, this solution does not allow for arbitrary values for $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ they must be proportionally connected for each data point by the scaling factor σ_{x_i} = kσ_{y_i}. The following section defines a PTLS approximation that allows an arbitrary relationship.

5.7 Deriving the AWTLS Cost Function

An approximate solution to the WTLS problem that allows $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ to be nonproportional and that yields a recursive solution for feasible implementation in an embedded system, which illustrates the geometry of the

WTLS and PTLS solutions and motivates the geometry of the approximate solution to be developed in this section.

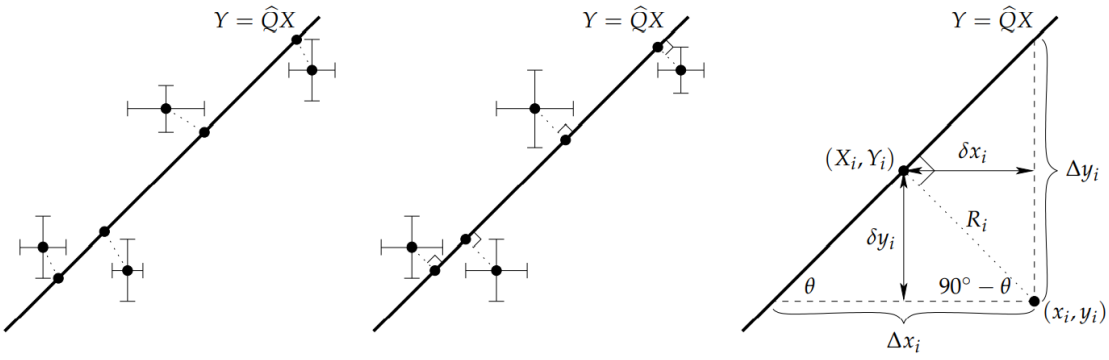

Fig 5.1 - Geometry of WTLS, PTLS and AWTLS.

When $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ are random, the WTLS relationship between data point (x_i, y_i) and its optimised map (X_i, Y_i) on $Y_i = \hat{Q}_n X_i$is seen in the left frame of the diagram. The error bars are drawn around the data points (x_i, y_i) to show the uncertainties in each axis, which are proportional to $x_i$andy_i. On the line Y_i=$\hat{Q}_n X_i$, a dotted line connects each data point (x_i, y_i) to its map (X_i, Y_i). The difference between x_i and X_i is not always equal to the distance between y_i andY_i. If the x_i measurement's quality is higher (worse) than the $y_i$measurement's quality, the distance from x_i to its map $X_i$should be shorter (greater) than the distance from $x_i$to its map X_i.

The PTLS relationship between data point (x_i, y_i) and its optimized map (X_i, Y_i) on Y_i=$\hat{Q}_n X_i$when $\sigma_{x_i}^2$and $\sigma_{y_i}^2$are equal is seen in the middle frame. The distance between $x_i$and $X_i$is the same as the distance between $y_i$and $Y_i$in this case, and the line connecting the data point (x_i, y_i) and its map (X_i, Y_i) is perpendicular to the line Y_i=$\hat{Q}_n X_i$. If $x_i$and $y_i$are not equal but additive, so either the x- or y-axes can be scaled to produce transformed data points with equal variances, and the same principle applies.

The concepts that will be used to obtain an estimated weighted total-least-squares (AWTLS) solution are seen in the right frame of the diagram. The line connecting data points (x_i, y_i) and (X_i, Y_i) be perpendicular to the line $Y_i=\hat{Q}_n X_i$ is required. This would produce a solution that can be solved in a recursive manner. However, in the optimization cost equation, the distance between $x_i$and $X_i$differently than the distance between $y_i$andY_i, as in the WTLS solution is weighted.

These AWTLS solutions have the following features in common with the WLS solution:

1. ForQ_n, they have a closed-form solution. It is not necessary to iterate. The only complication is the need to locate the roots of a quartic polynomial, but this can be done in a manageable manner.
2. In a recursive way, the solution can be conveniently computed. The six moving quantities $c_{1,n}$ through $c_{6,n}$ are kept track of. The amounts and calculate a modified total-capacity calculation as new data becomes available is updated.
3. Fading memory can be conveniently applied to allow the calculation Q:n to put more focus on recent measurements than earlier measurements, enabling Q:n to respond to true cell total capacity changes.
4. Furthermore, unlike the PTLS approach, this method allows for independent weighting of x_i and $y_i$data points.

5.8 Output of System

The Fig.5.2 shows the efficiency of the WLS, WTLS, PTLS, and AWTLS total-capacity estimation approaches is compared by examining multiple utilization scenarios. The fading memory variant of the four methods is used in all cases, but the prefix "FM" is omitted for brevity. The fading-memory forgetting factor is set to 1.0 unless otherwise specified.

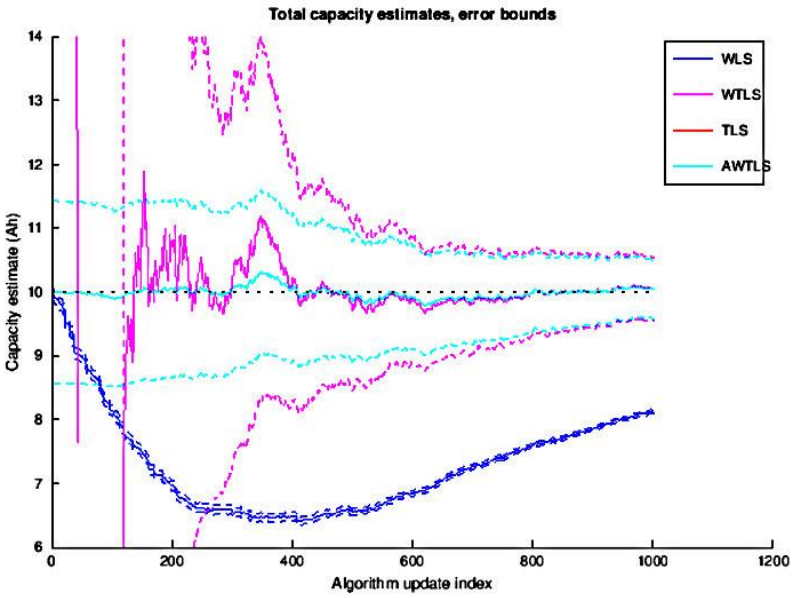

Fig. 5.2 - Comparison of the estimates of each method.

Finally, the AWTLS approach is implemented. First, the roots of the quadratic equation are searched. Any roots that aren't positive and genuine are discarded. Then, using the remaining candidate roots, the cost function of the equation is calculated and holds the root that minimizes the cost function. A capacity degradation factor is added to optimize the final result. Fig. 5.3 shows the final state of health of the battery during the battery life time

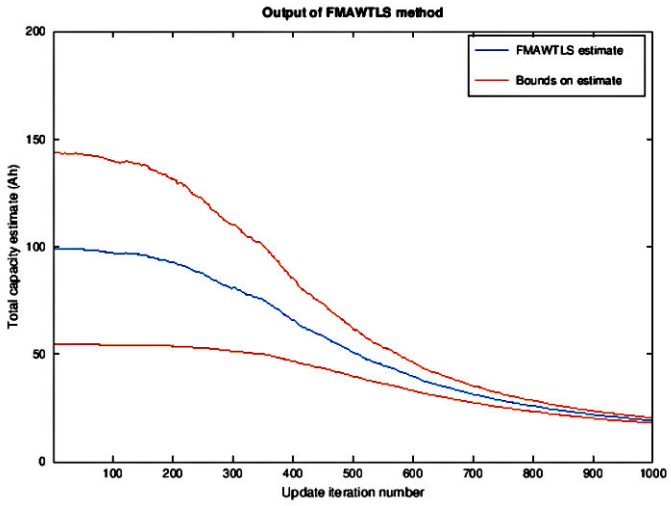

Fig.5.3 - State of Health Output

CHAPTER 6

CONCLUSION

Thus the project work involves successful estimation of the state of charge and the state of health of the lithium iron phosphate (LiFePO4) or LFP battery with the help of enhanced self-correcting cell model using MATLAB software. The output graphs show the estimated values of SOC and SOH of the cell along with their true values measured in the laboratory during testing.

Future Works

This project can be extended to any chemistry of the lithium-ion cells just by changing the test data in the ESC cell model program without having to change any other program. This work can be used for the Real-Time online estimation of SOC and SOH of an electric vehicle battery. For online estimation, one input to the algorithms must be from the ESC cell model and another input must be from the real battery pack.

APPENDIX

1. OCVfromSOCTemp

```
% Set up data for problem
load read-only/Modeldata.mat

% FMAWTLS - recursive, scaled by initial standard deviations
K = sqrt(Sigmax(1)/Sigmay(1));     % Constant of proportionality
Q0=100;
Qnom = 0.99*Q0;                % Initialize nominal capacity
gamma = 1.00;                  % Fading-memory rate constant. Default = 1.00
SYinit = 1e-2;                  % Uncertainty of Q with respect to Qnom
c1 = 1/(K^2*SYinit);             % Correct initialization of c1 recursive value
c2 = K*Qnom/(K^2*SYinit);      % Correct initialization of c2 recursive value
c3 = K^2*Qnom^2/(K^2*SYinit);% Correct initialization of c3 recursive value
% Init c4...c6 assuming SXinit = K^2*SYinit to match TLS
c4 = 1/(K^2*SYinit);          % Correct initialization of c4 recursive value
c5 = K*Qnom/(K^2*SYinit); % Correct initialization of c4 recursive value
c6 = K^2*Qnom^2/(K^2*SYinit);% Correct initialization of c4 recursive value
Qhat = 0*x;                    % Initialize storage for output
SigmaQ = 0*x;
for k = 1:length(x),
        c1 = gamma*c1 + x(k)^2/(K^2*Sigmay(k));
        c2 = gamma*c2 + K*x(k)*y(k)/(K^2*Sigmay(k));
        c3 = gamma*c3 + K^2*y(k)^2/(K^2*Sigmay(k));
        c4 = gamma*c4 + x(k)^2/Sigmax(k);
        c5 = gamma*c5 + K*x(k)*y(k)/Sigmax(k);
        c6 = gamma*c6 + K^2*y(k)^2/Sigmax(k);

         r = roots([c5 (-c1+2*c4-c6) (3*c2-3*c5) (c1-2*c3+c6) -c2]);
         r = r(r==conj(r));
         r = r(r>0);
        Jr = ((1./(r.^2+1).^2).*(r.^4*c4-2*c5*r.^3+(c1+c6)*r.^2-2*c2*r+c3))';
        Q = r(Jr==min(Jr));
        J = min(Jr);
         H=(2/(Q^2+1)^4)*(-2*c5*Q^5+(3*c1-6*c4+3*c6)*Q^4+(-
        12*c2+16*c5)*Q^3+(-8*c1+10*c3+6*c4-8*c6)*Q^2+(12*c2-
        6*c5)*Q+(c1-2*c3+c6));
        Qhat(k) = Q/K;
        SigmaQ(k) = 2/H/K^2;
end
xvals = 1:length(Qhat);
```

```
plot(xvals,Qhat,[xvals,NaN,xvals],[Qhat+3*sqrt(SigmaQ),NaN,Qhat-
3*sqrt(SigmaQ)]);
xlabel('Update iteration number'); ylabel('Total capacity estimate (Ah)');
title('Output of FMAWTLS method');   xlim([1 1000]);
legend('FMAWTLS estimate','Bounds on estimate','location','northeast');
fprintf('Final FMAWTLS estimate (using all data): %f +/-
%f\n',Qhat(end),3*sqrt(SigmaQ(end)));
```

2. SIMCELL

```
function [vk,irk,hk,zk,sik,OCV] = simCell(ik,T,deltaT,model,z0,iR0,h0)
% Force data to be column vector(s)
ik = ik(:); iR0 = iR0(:);
% Get model parameters from model structure
RCfact = exp(-deltaT./abs(getParamESC('RCParam',T,model)))';
G = getParamESC('GParam',T,model);
Q = getParamESC('QParam',T,model);
M = getParamESC('MParam',T,model);
M0 = getParamESC('M0Param',T,model);
RParam = getParamESC('RParam',T,model);
R0Param = getParamESC('R0Param',T,model);
etaParam = getParamESC('etaParam',T,model);
etaik = ik;
% compensate for coulombic efficiency
etaik(ik<0) = etaParam*ik(ik<0);
% Simulate the dynamic states of the model
if exist('ss','file') , % use control-system-toolbox method, if available
        sysd= ss(diag(RCfact),1-RCfact,eye(length(RCfact)),0,-1);
        irk = lsim(sysd,etaik,[],iR0);
else
        irk=zeros([length(ik) length(iR0)]); irk(1,:) = iR0;
        for k = 2:length(ik),
        irk(k,:) = RCfact'.*irk(k-1,:) + (1-RCfact')*etaik(k-1);
        end
end
zk = z0-cumsum([0;etaik(1:end-1)])*deltaT/(Q*3600);
if any(zk>1.1),
        warning('Current may have wrong sign as SOC> 110%');
end
```

```
% Hysteresis stuff
hk=zeros([length(ik) 1]); hk(1) = h0; sik = 0*hk;
fac=exp(-abs(G*etaik*deltaT/(3600*Q)));
for k=2:length(ik),
hk(k)=fac(k-1)*hk(k-1)-(1-fac(k-1))*sign(ik(k-1));
        sik(k) = sign(ik(k));
        if abs(ik(k))<Q/100, sik(k) = sik(k-1); end
end
% Compute output equation
  OCV = OCVfromSOCtemp(zk,T,model);
vk = OCV - irk*RParam' - ik.*R0Param + M*hk + M0*sik;
```

3. XLSALGOS

```
% = xLSalgos(measX,measY,SigmaX,SigmaY,gamma,Qnom)
%       Tests the recursive performance of the xLS algorithms on a particular
%       dataset.
%       Inputs:
%      measX:  noisy z(2)-z(1)
%       measY:  noisy integral(i(t)/3600 dt)
%       SigmaX: variance of X
%       SigmaY: variance of Y
%       gamma:  geometric forgetting factor (gamma = 1 for perfect memory)
%       Qnom:   nominal value of Q: used to initialize recursions if nonzero
%
%       Outputs:
%       Qhat:   estimate of capacity at every time step
%      column 1 = WLS - weighted, recursive
%      column 2 = WTLS - weighted, but not recursive
%      column 3 = SCTLS - scaled confidence TLS; recursive and
%      weighted, but using SigmaX(1) and SigmaY(1) only
%      to determine factor by which all SigmaX and
%       SigmaY are assumed to be related
%       column 4 = AWTLS - recursive and weighted
%       SigmaQ: variance of Q, computed via Hessian method (columns
%      correspond to methods in the same way as for Qhat)
%       Fit:    goodness of fit metric for each method (columns
%      correspond to methods in the same way as for Qhat)
function[Qhat,SigmaQ,Fit] =
xLSalgos(measX,measY,SigmaX,SigmaY,gamma,Qnom)
% Reserve some memory
Qhat = zeros(length(measX),4); SigmaQ = Qhat; Fit = Qhat;
  K = sqrt(SigmaX(1)/SigmaY(1));
```

```
% Initialize some variables used for the recursive methods
  c1 = 0; c2 =0; c3 = 0; c4 = 0; c5 = 0; c6 = 0;
  C1 = 0; C2 =0; C3 = 0; C4 = 0; C5 = 0; C6 = 0;
ifQnom ~= 0
        c1 = 1/SigmaY(1); c2 = Qnom/SigmaY(1); c3 = Qnom^2/SigmaY(1);
        c4 = 1/SigmaX(1); c5 = Qnom/SigmaX(1); c6 = Qnom^2/SigmaX(1);
        C1 = 1/(K^2*SigmaY(1)); C2 = K*Qnom/(K^2*SigmaY(1));
        C3 = K^2*Qnom^2/(K^2*SigmaY(1));
        C4 = 1/SigmaX(1); C5 = K*Qnom/SigmaX(1); C6 =
K^2*Qnom^2/SigmaX(1);
end
foriter = 1:length(measX)
        % Compute some variables used for the recursive methods
        c1 = gamma*c1 + measX(iter)^2/SigmaY(iter);
        c2 = gamma*c2 + measX(iter)*measY(iter)/SigmaY(iter);
        c3 = gamma*c3 + measY(iter)^2/SigmaY(iter);
        c4 = gamma*c4 + measX(iter)^2/SigmaX(iter);
        c5 = gamma*c5 + measX(iter)*measY(iter)/SigmaX(iter);
        c6 = gamma*c6 + measY(iter)^2/SigmaX(iter);
        C1 = gamma*C1 + measX(iter)^2/(K^2*SigmaY(iter));
        C2 = gamma*C2 + K*measX(iter)*measY(iter)/(K^2*SigmaY(iter));
        C3 = gamma*C3 + K^2*measY(iter)^2/(K^2*SigmaY(iter));
        C4 = gamma*C4 + measX(iter)^2/SigmaX(iter);
        C5 = gamma*C5 + K*measX(iter)*measY(iter)/SigmaX(iter);
        C6 = gamma*C6 + K^2*measY(iter)^2/SigmaX(iter);

        % Method 1: WLS
        Q = c2./c1; Qhat(iter,1) = Q;
        H = 2*c1;   SigmaQ(iter,1) = 2/H;
        J = Q.^2.*c1 -2*Q.*c2 + c3;
        Fit(iter,1) = gammainc(J/2,(iter-1)/2,'upper');
end
  r = roots([c5 (-c1+2*c4-c6) (3*c2-3*c5) (c1-2*c3+c6) -c2]);
   r = r(r==conj(r));
   r = r(r>0);
Jr = ((1./(r.^2+1).^2).*(r.^4*c4-2*c5*r.^3+(c1+c6)*r.^2-2*c2*r+c3))';
   Q = r(Jr==min(Jr));
   J = min(Jr);
   H = (2/(Q^2+1)^4)*(-2*c5*Q^5+(3*c1-6*c4+3*c6)*Q^4+(-
12*c2+16*c5)*Q^3 ...
+(-8*c1+10*c3+6*c4-8*c6)*Q^2+(12*c2-6*c5)*Q+(c1-2*c3+c6));
Qhat(iter,4) = Q;
```

```
SigmaQ(iter,4) = 2/H;
Fit(iter,4) = gammainc(J/2,(2*iter-1)'/2,'upper');
  Fit = real(Fit);
Return
```

4. AWLTS

```
%% Set up data for problem
loadreadonly/Lesson425data.mat
%% WTLS -- not recursive, uses all data to produce single estimate
sigmax = sqrt(Sigmax); % std-dev of x
sigmay = sqrt(Sigmay); % std-dev of y
Qhat = 5; % purposefully bad initialization of Qhat
for k = 1:10
jacobian = sum((2*(Qhat*x-y).*(Qhat*y.*sigmax.^2+x.*sigmay.^2))./...
             ((Qhat^2*sigmax.^2+sigmay.^2).^2));
hessian = sum((2*sigmay.^4.*x.^2+sigmax.^4.*...
        (6*Qhat^2*y.^2-4*Qhat^3*x.*y) - ...
        sigmax.^2.*sigmay.^2.*...
            (6*Qhat^2*x.^2-12*Qhat*x.*y+2*y.^2))./...
            ((Qhat^2*sigmax.^2+sigmay.^2).^3));
Qhat = Qhat - jacobian/hessian;
SigmaQ = 2/hessian;
boundsQ = 3*sqrt(SigmaQ);
end
fprintf('WTLS estimate after %d Newton-Raphson iterations: %f +/-
%f\n',k,Qhat,boundsQ);

%% AWTLS -- recursive, updates estimate every time new data become
available
Qnom = 10;              % Initialize nominal capacity
SYinit = 1e-4;          % Uncertainty of Q with respect to Qnom
c1 = 1/SYinit;          % Correct initialization of c1 recursive value
c2 = Qnom/SYinit;           % Correct initialization of c2 recursive value
c3 = Qnom^2/SYinit;         % Correct initialization of c3 recursive value
% Init c4...c6 assuming SXinit = SYinit to match TLS
c4 = 1/SYinit;          % Correct initialization of c4 recursive value
c5 = Qnom/SYinit;           % Correct initialization of c4 recursive value
c6 = Qnom^2/SYinit;         % Correct initialization of c4 recursive value
Qhat = 0*x;             % Initialize storage for output estimate
SigmaQ = 0*x;               % Initialize storage for estimation-error
variance
```

```
for k = 1:length(x)
  c1 = c1 + x(k)^2/Sigmay(k);    % Update c1 recursive parameter
  c2 = c2 + x(k)*y(k)/Sigmay(k);  % Update c2 recursive parameter
  c3 = c3 + y(k)^2/Sigmay(k);    % Update c3 recursive parameter
  c4 = c4 + x(k)^2/Sigmax(k);    % Update c4 recursive parameter
  c5 = c5 + x(k)*y(k)/Sigmax(k);  % Update c5 recursive parameter
  c6 = c6 + y(k)^2/Sigmax(k);    % Update c6 recursive parameter
  r = roots([c5 (-c1+2*c4-c6) (3*c2-3*c5) (c1-2*c3+c6) -c2]);
  r = r(r==conj(r));
  r = r(r>0);
Jr = ((1./(r.^2+1).^2).*(r.^4*c4-2*c5*r.^3+(c1+c6)*r.^2-2*c2*r+c3))';
  Q = r(Jr==min(Jr));
  J = min(Jr);
  H = (2/(Q^2+1)^4)*(-2*c5*Q^5+(3*c1-6*c4+3*c6)*Q^4+(-
12*c2+16*c5)*Q^3 ...
+(-8*c1+10*c3+6*c4-8*c6)*Q^2+(12*c2-6*c5)*Q+(c1-2*c3+c6));
Qhat(k) = Q;
SigmaQ(k) = 2/H;
end

xvals = 1:length(Qhat);
plot(xvals,10*ones(size(Qhat)),xvals,Qhat,[xvals,NaN,xvals],[Qhat+3*sqrt(Sig
maQ),NaN,Qhat-3*sqrt(SigmaQ)]);
xlabel('Update iteration number'); ylabel('Total capacity estimate (Ah)');
title('Output of AWTLS method');   xlim([1 1000]);
legend('True total capacity','AWTLSestimate','Bounds on
estimate','location','northeast');

fprintf('Final AWTLS estimate (using all data): %f +/-
%f\n',Qhat(end),3*sqrt(SigmaQ(end)));
```

REFERENCES

1. Bhangu, B.S., Bentley, P., Stone, D. A. & Bingham, C. M. (2005) 'Nonlinear Observers for Predicting State-of-Charge and State-of-Health of Lead-Acid Batteries for Hybrid-Electric Vehicles' - IEEE Transactions on Vehicular Technology Vol. 54, No.3, pp.783-794.
2. Dheenadhayalan, P., Nair, A., Manalikandy, M., Reghu, A., John, J., & Rani, V. S. (2015) 'A Novel Method for Estimation of State of Charge of Lithium-ion Battery using Extended Kalman Filter' - SAE Technical Paper 2015-01-1183.
3. He, L., Hu, M., Wei, Y., Liu, B., & Shi, Q. (2019) 'State of charge estimation by finite difference extended Kalman filter with HPPC parameters identification' - Science China Technological Sciences, Sci. China Technol. Sci. Vol.63, pp.410–421.
4. He, W., Williard, N., Chen, C., &Pecht, M. (2014) 'State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation' - International Journal of Electrical Power & Energy Systems, Vol.62, pp.783–791.
5. He, Z., Gao, M., Xu, J., & Liu, Y. (2009) 'Battery Model Parameters Estimation with the Sigma Point Kalman Filter' - 2009 International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, China.
6. Khan, K., Zhou, B., and Rezaei, A., 'Real-Time Application of Battery State of Charge and State of Health Estimation' - SAE Technical Paper 2017-01-1199, 2017.

7. Moussalli, Z., BrahimSedra, M., & Ait Laachir, A. (2018) ‘State of Charge estimation algorithms in Lithium-ion battery-powered Electric Vehicles’ - 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS).
8. Ng, K. S., Moo, C.-S., Chen, Y.-P., & Hsieh, Y.-C. (2009) ‘Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries’ - Applied Energy, Vol. 86, No. 9, pp.1506-1511.
9. Nguyen Van, C., & Nguyen Vinh, T. (2020) ‘SOC Estimation of the Lithium-Ion Battery Pack using a Sigma Point Kalman Filter Based on a Cell’s Second-Order Dynamic Model’ - Applied Sciences, Vol.10, No.5, pp.1896.
10. Piller, S., Perrin, M., &Jossen, A. (2001) ‘Methods for state-of-charge determination and their applications’ - Journal of Power Sources Vol. 96, No.1, pp.113–120.
11. Sangwan, V., Kumar, R., &Rathore, A. K. (2017) ‘Estimation of model parameters and state-of-charge for the battery management system of Li-ion battery in EVs’ - 2017 IEEE Transportation Electrification Conference (ITEC-India).
12. Wassiliadis, N., Adermann, J., Frericks, A., Pak, M., Reiter, C., Lohmann, B., &Lienkamp, M. (2018) ‘Revisiting the dual extended Kalman filter for battery state-of-charge and state-of-health estimation: A use-case life cycle analysis’ - Journal of Energy Storage, Vol.19, pp.73–87.
13. Xiao, B., Shi, Y., & He, L. (2010) ‘A universal state-of-charge algorithm for batteries’ - Proceedings of the 47th Design Automation Conference on - DAC ’10.
14. Zhang, F., Liu, L., &Fang , L. (2008) ‘A Battery State of Charge Estimation Method with Extended Kalman Filter’ - 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.

15. Zheng, W., Xia, B., Wang, W., Lai, Y., Wang, M., & Wang, H. (2019) ‘State of Charge Estimation for Power Lithium-Ion Battery Using a Fuzzy Logic Sliding Mode Observer’ - Energies, Vol. 12, No.13, pp.2491.
16. Zou, Y., Hu, X., Ma, H., & Li, S. E. (2015) ‘Combined State of Charge and State of Health estimation over lithium-ion battery cell cycle lifespan for electric vehicles’ - Journal of Power Sources, Vol.273, pp.793–803.