

ABSTRACT

Title of Dissertation: BATTERY STATE OF CHARGE ESTIMATION BASED ON DATA-DRIVEN MODELS WITH MOVING WINDOW FILTERS AND PHYSICS-BASED MODELS WITH EFFICIENT SOLID-PHASE DIFFUSION PDES SOLVED BY THE OPTIMIZED PROJECTION METHOD

Wei He, Doctor of Philosophy, 2018

Dissertation directed by: Professor Michael Pecht,
Department of Mechanical Engineering

State of charge (SOC) estimation is one of the most important functions of battery management systems (BMSs), which is defined as the percentage of the remaining charge inside the battery to its maximum capacity. SOC indicates when the battery needs to be recharged. It is necessary for many battery management applications, for example, charge/discharge control, remaining useful time/ driving range predictions, and battery power capability estimations. Inaccurate SOC estimations can lead to user dissatisfaction, mission failures, and premature battery failures. This thesis focuses on

the development of advanced battery models and algorithms for SOC estimations. Two SOC estimation approaches are investigated, including electrochemical models and data-driven models.

Electrochemical models have intrinsic advantages for SOC estimation since it can relate battery internal physical parameters, e.g. lithium concentrations, to SOC. However, the computational complexity of the electrochemical model is the major obstacle for its application in a real-time BMS. To address this problem, an efficient solution for the solid phase diffusion equations in the electrochemical model is developed based on projection with optimized basis functions. The developed method generates 20 times fewer equations compared with finite difference-based methods, without losing accuracy. The results also show that the developed method is three times more efficient compared with the conventional projection-based method. Then, a novel moving window filter (MWF) algorithm is developed to infer SOC based on the electrochemical model. MWF converges to true values nearly 15 times faster compared with unscented Kalman filter in experimental test cases.

This work also develops a data-driven SOC estimation approach. Traditional data-driven approaches, e.g. neural network, have generalization problems. For example, the model over-fits to training data and generate erroneous results in the testing data. This thesis investigates algorithms to improve the generalization capability of the data-driven model. An algorithm is developed to select optimal neural network structure and training data inputs. Then, a hybrid approach is developed by combining the neural

network and MWF to provide stable SOC estimations. The results show that the SOC estimation error can be reduced from 8% to less 4% compared with the original neural network approach.

BATTERY STATE OF CHARGE ESTIMATION BASED ON DATA-DRIVEN
MODELS WITH MOVING WINDOW FILTERS AND PHYSICS-BASED
MODELS WITH EFFICIENT SOLID-PHASE DIFFUSION PDES SOLVED BY
THE OPTIMIZED PROJECTION METHOD

by

Wei He

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Michael Pecht, Chair
Professor Patrick McCluskey
Professor Peter Sandborn
Professor Liangbing Hu
Professor Chunsheng Wang

© Copyright by
Wei He
2018

Acknowledgements

I would like to express my deepest gratitude and respect to Prof. Michael Pecht for his guidance in this dissertation topic, continuous support towards the progress of this work and valuable help on my writing and presentation skills. I would never have been able to finish my dissertation without the guidance of Prof. Pecht.

I would also like to thank Prof. Patrick McCluskey, Prof. Peter Sandborn, Prof. Liangbing Hu and Prof. Chunsheng Wang for serving on my dissertation committee and providing valuable suggestions.

Table of Contents

Acknowledgements.....	ii
Table of Contents.....	iii
List of Tables.....	v
List of Figures.....	vi
List of Abbreviations.....	viii
Chapter 1: Introduction.....	1
1.1 Background.....	1
1.2 Motivation.....	2
1.3 Overview of the Dissertation.....	3
Chapter 2: Literature Review and Objectives.....	4
2.1 Previous Work and Research Gaps.....	4
2.1.1 Electrochemical Model.....	4
2.1.2 State Filter.....	6
2.1.3 Data Driven Model.....	9
2.2 Problem Statement and Research Objectives.....	10
Chapter 3: A Computationally Efficient Solution for Battery Diffusion PDEs in Solid Phase Particles.....	13
3.1 Introduction.....	13
3.2 Principle of Electrochemical models.....	13
3.2 Projection-based model reduction method.....	18
3.3 Orthonormalization of Basis Functions.....	22
3.4 Optimal Basis Function Construction.....	23
3.5 Results and Discussions.....	30
3.6 Summary.....	41
Chapter 4: Moving Window Filter for SOC Estimation Based on Electrochemical Models.....	42
4. 1 Introduction.....	42
4.2 State Estimation Theory and Kalman Filters.....	42
4.3 Moving Window Filter.....	49

4.4 Experimental Validation	52
Chapter 5: Improved Data-driven Approach for SOC Estimation.....	59
5.1 Introduction.....	59
5.2 Theory of Neural Network.....	60
5.3 Training and Testing Data	62
5.4 Determination of the Neural Network Structure.....	65
5.5 Training and Testing Results	67
5.6 Developed Solution by Combing NN with State Filters.....	73
5.7 Conclusions and Discussions.....	78
Chapter 6: Contributions and Suggestions for Future Research.....	80
6.1 Contributions of this Dissertation	80
6.2 Suggestions for Future Research	82
Bibliography	84

List of Tables

Table 1: Comparison of algorithm performance for diffusion PDE calculation	40
Table 2: Summary of SOC estimation results based on electrochemical model and MWF	58
Table 3: Summary of SOC estimation results based on the data-driven approach.....	77

List of Figures

Figure 1: Flowchart of model-based battery SOC estimation algorithm	7
Figure 2: Slow convergence of UKF given 40% initial SOC estimation error	8
Figure 3: Dualfoil battery electrochemical model	15
Figure 4: Flowchart of the optimal projection method for solving diffusion PDEs ...	30
Figure 5: Frequency responses of different methods compared to the true transfer function	32
Figure 6: Convergence comparison of different methods.....	34
Figure 7: 50s flux and 950s rest applied to the surface of the particle	34
Figure 8: Time domain outputs of different methods for slow dynamics.....	35
Figure 9: 5Hz pulses applied to the surface of the particle.....	36
Figure 10: Time domain outputs of different methods for 5Hz pulses.....	37
Figure 11: Errors of different methods compared to FD with 500 nodes.....	38
Figure 12: Federal driving schedule (FDS) profile.....	38
Figure 13: Fourier transform of FDS	39
Figure 14: Time-domain simulation of different algorithms based on FDS.....	39
Figure 15: Time-domain simulation errors of different algorithms based on FDS	40
Figure 16: Flow chart of Kalman filter	43
Figure 17: Unscented Kalman filter-based SOC estimation with 40% initial SOC error	45
Figure 18: OCV-SOC curve of 5 LiFePO ₄ cells	46
Figure 19: Conventional sequential estimation scheme.....	49
Figure 20: Moving window state filter	50
Figure 21: Moving window filter implementation.....	52
Figure 22: Experimental Setup	53
Figure 23: Federal driving schedule (FDS) current profile	53
Figure 24: Federal driving schedule (FDS) velocity profile.....	53
Figure 25: Effect of window size on the filter accuracy.....	54
Figure 26: SOC estimation comparison of cell 1	55
Figure 27: SOC estimation of cell 2	55
Figure 28: SOC estimation of cell 3	56
Figure 29: SOC estimation of cell 4	56
Figure 30: SOC estimation of cell 5	57

Figure 31: The structure of a multilayer feed-forward neural network	62
Figure 32: Dynamical Stress Testing (DST) profile, which will be used as training data	64
Figure 33: Federal Driving Schedule (FDS), which will be used as testing data	64
Figure 34: Experimental setup for battery tests	65
Figure 35: Neural network training results of cell #1 with 5 neurons for each layer based on DST data	69
Figure 36: Testing results of neural network with 5 neurons in each layer for Cell #1 FDS data.....	69
Figure 37: Training result of neural networks with 15 neurons in each player based on Cell #1 DST data.....	70
Figure 38: Testing results of neural network with 15 neurons in each layer for Cell #1 FDS data.....	70
Figure 39: NN SOC estimation results for Cell#2 FDS data, where NN is trained based on Cell#1 DST data.....	71
Figure 40: NN SOC estimation results for Cell#3 FDS data, where NN is trained based on Cell#1 DST data.....	71
Figure 41: NN SOC estimation results for Cell#4 FDS data, where NN is trained based on Cell#1 DST data.....	72
Figure 42: NN SOC estimation results for Cell#5 FDS data, where NN is trained based on Cell#1 DST data.....	72
Figure 43: Flowchart of the combined approach of neural network and moving window filter.....	73
Figure 44: SOC estimation results for Cell #1 FDS data based on the combined NN&MWF approach	75
Figure 45: SOC estimation results for Cell #2 FDS data based on the combined NN&MWF approach	75
Figure 46: SOC estimation results for Cell #3 FDS data based on the combined NN&MWF approach	76
Figure 47: SOC estimation results for Cell #4 FDS data based on the combined NN&MWF approach	76
Figure 48: SOC estimation results for Cell #5 FDS data based on the combined NN&MWF approach	77

List of Abbreviations

BMS	Battery management system
DAE	Differential algebraic equation
DST	Dynamic stress testing
ECM	Equivalent circuit model
EKF	Extended Kalman filter
EV	Electrical vehicle
FD	Finite difference
FDS	Federal driving schedule
IRKA	Iterative rational Krylov algorithm
LiP	Lithium-ion polymer
MWF	Moving window filter
NN	Neural network
OCV	Open circuit voltage
ODE	Ordinary differential equation
OP	Optimized projection
PDE	Partial differential equation
RC	Resistor capacitor
SOC	State of charge
SOH	State of health
UKF	Unscented Kalman filter
USABC	US Advanced Battery Consortium
VA	Volume average

Chapter 1: Introduction

1.1 Background

Batteries are electrochemical energy storage devices that are key components of many systems, ranging from Bluetooth headphones to mobile phones, laptops, electric vehicles (EVs), and satellites. Though the term “battery” originally referred to several cells connected in series, it is now also used for a single electrochemical cell [1]. A battery generally consists of an anode (negative electrode) and a cathode (positive electrode) separated by an ion-conductive, electronically insulating solution called electrolyte [1, 2]. When the battery is connected to an external load, the anode supplies a current of electrons that flow through the load and are accepted by the cathode, while the ion moves across the electrolyte from the cathode to the anode because of electroneutrality. If the electrochemical process of the battery is able to be reversed by applying an external current, then the battery is called a rechargeable or secondary battery.

State of charge (SOC) estimation is necessary for rechargeable batteries, as it provides the information to user about when the battery need to be recharged. SOC is defined as the percentage of remaining capacity relative to the maximum capacity of the battery [3]. Many SOC estimation approaches have been developed, among which Coulomb counting [4-6] is the most popular one. In Coulomb counting, the current is integrated over time to estimate SOC. Although Coulomb counting is simple and easy to implement, the measurement and calculation errors can be accumulated by the integration function. The estimation of SOC tends to drift from the actual values. In addition, Coulomb counting needs accurate initial conditions, otherwise initial error will remain in the estimation. The voltage-based method is also commonly used [7], where SOC is determined based on a voltage–SOC lookup table. However, voltage-based methods do not work well for Li-ion batteries because of their flat plateau of discharge characteristics. Recently, model-based methods have been applied to SOC estimations. They have intrinsic advantages over traditional Coulomb counting and

voltage-based methods, for example, they are robust to sensor error and do not require accurate initial SOCs. This dissertation will focus on the development of advanced battery models and state filtering algorithms for SOC estimations.

1.2 Motivation

SOC estimation is critical for battery powered systems. For instance, EV drivers need to know the remaining driving range, and SOC of the battery is the prerequisite for the remaining driving range calculation. If SOC is in-accurate, then the driving range prediction will be misleading. There has been some news reporting that EV drivers were halted in the highway. In addition, the battery management system (BMS) will need SOC information for power management, because battery power is a function of impedance, and impedance will change with SOC. Inaccurate SOC estimation will lead to erroneous estimation of the battery power capability. In the worst case, the battery will shut down suddenly if the BMS allows a power that exceeds the battery's capability. To avoid this situation, a common practice is to overdesign the battery to account for uncertainties. If the SOC can be accurately estimated, the over-engineering of batteries can be minimized. The associated benefits will be, for example, reduction of the size, weight and cost of the battery. In addition, the SOC information can be used to avoid over-charge and over-discharge of the battery. Voltage is not a good indicator of batteries' charging state, as it is also a compound function of impedance and charging state. SOC can be used to determine when the charging current should be reduced. For example, control loop can be designed to gradually reduce the charging current based on SOC in order to maximize the charging rate while avoid the degradation.

In summary, SOC estimation is one of the most important topics for battery management. It required by most of the recharged battery systems, because (1) it indicates when the battery need to be recharged; (2) it is required by real-time power management; (3) it can be used to design better charging/discharge algorithm to prolong life span; (4) it can be used to avoid over-engineering of the batteries, thereby reducing the size, weight, and cost of the battery.

1.3 Overview of the Dissertation

The structure of the remaining of the dissertation is as follows. Chapter 2 presents the literature review on state of charge estimation algorithms, including electrochemical model based methods, data-driven methods and state filter algorithms, puts forward the problem statement based on the literature review, and provides the objectives of this dissertation. Chapter 3 develops new methods for electrochemical model simplification which can be applied for the real-time SOC estimation. Chapter 4 presents an improved state filter algorithm based moving window strategy, which is then combined with electrochemical model for SOC estimation. Experimental validations of the moving window filter and the simplified electrochemical model are also presented in Chapter 4. Chapter 5 addresses the generalization issue of the data-driven models, e.g. neural network, for SOC estimation. A hybrid approach, which combines neural network and state filter is developed in Chapter 5 and then is compared with conventional data-driven approaches to demonstrate the effectiveness of the new approach. Chapter 6 draws the contributions of this dissertations and provides the future research directions.

Chapter 2: Literature Review and Objectives

2.1 Previous Work and Research Gaps

2.1.1 Electrochemical Model

The equivalent circuit model (ECM) is the most commonly used battery model in a BMS [8-16]. It is formed by an open-circuit voltage source connected with a set of electric elements, such as resistors and capacitors, to model the electrical behavior of battery. The ECM is widely used in the BMS for SOC estimation, because it is computationally efficient. ECM can be used to estimate battery voltages based on current inputs, and different resistor-capacitor (RC) branches can be used to capture different time constants inherent in the battery system. However, because ECM is an empirical model in nature, it provides little insights into the electrochemical process inside the battery, and it cannot provide highly accurate results.

In contrast, the electrochemical model uses partial differential equations (PDEs) to model the physics of the battery and it can be used to compute the electrochemical states inside a battery and provide accurate information about lithium concentrations and over-potentials, which can be used to prevent harmful side-reactions. For example, the electrochemical model developed by Doyle and Newman [17-19] includes the key chemistry and physics in a Li-ion battery, such as diffusion, migration, and reaction kinetics. The model has two scales, namely, a macro x -scale along the thickness of the electrodes, and a micro r -scale along the radius direction of the small solid-phase electrode particles. The electrochemical processes are modeled by a set of coupled

partial differential equations (PDEs). In solving PDEs, spatial discretization methods, such as finite element, finite difference, and finite volume are widely used. When using spatial discretization methods, we need to discretize the electrochemical model in both the x and r directions, resulting in thousands of states (equations) [20, 21]. Implementing a real-time BMS using the spatial discretization method is computationally prohibitive and on-board memory demanding. Therefore, in order to have an electrochemical model for real-time battery management, deriving a simplified and computationally efficient mathematical model is crucial.

The single-particle model (SPM) is a typical example of a simplified model with less than 10 states [22, 23], in which each electrode is modeled by a single spherical particle. As a result, the spatial variations of potentials are unable to be determined by SPM. In addition, the electrolyte concentration is generally assumed to be a constant in time and space. The assumptions will not hold for high current applications ($>3C$) [24, 25]. In the full electrochemical model, the diffusion PDE in solid particles is a major part of the model. It determines the lithium concentration in the electrode particles and provides information on the level of lithium available for producing an electrical charge. In the other words, it tells users about the remaining level of electrical energy in the battery. When solving the electrochemical model, the solid-phase diffusion PDE generates more than 80% of the states (equations) after discretizing the model. That is because the solid phase diffusion not only varies in the macro x -scale along the thickness of the electrode but also in the micro r -scale along the radius direction of the solid phase particles. Other PDEs, e.g. electrolyte diffusion, only changes in the macro

x -scale. For this reason, the solid phase diffusion is inherently more computational intensive than other PDEs in the electrochemical model. For example, if we discretize the battery model in the x -scale with n nodes and in the r -scale with m nodes. Then the total number of generated equations from the solid phase diffusion is $n \times m$, while other PDEs in the model will only generate n equations. Therefore, simplifying the diffusion PDEs in the solid phase is essential in providing a real-time capability. Subramanian et al. [26, 27] developed an efficient solution for the diffusion PDE by approximating the solid-phase lithium concentration using even-order monomials. The coefficients for the polynomials are obtained by volume averaging of the PDE and the derivatives of the PDE. However, Mayhew et al. [28] found that when using more than three monomials to approximate the lithium concentration, the volume-averaging technique can generate an unstable reduced-order model (ROM). Chaturvedi et al. [29] proposed a generalized projection framework to reduce the PDEs to a set of differential algebraic equations (DAEs). However, this method did not address the issue on constructing an optimal basis function.

In summary, available methods, for example, VA, can generate numerically unstable equations and lead to erroneous results. In addition, there is no available literature addresses the optimal basis function construction for the diffusion PDE reduction that is in dependent of loading conditions.

2.1.2 State Filter

In many practical cases, the physical state of a system cannot be determined by direct observation. For example, the SOC or lithium concentration cannot be directly

measured with sensor. State filter is an algorithm to infer the internal state of a given system from the measurements of the input. A system model is required to estimate system states. Below is an example of the flowchart for battery SOC estimation based on state filter. Here, the state filter compares the actual measured output and the model output, and then try to find the optimal state value, i.e. SOC, to minimize the difference between the measurement and the model output.

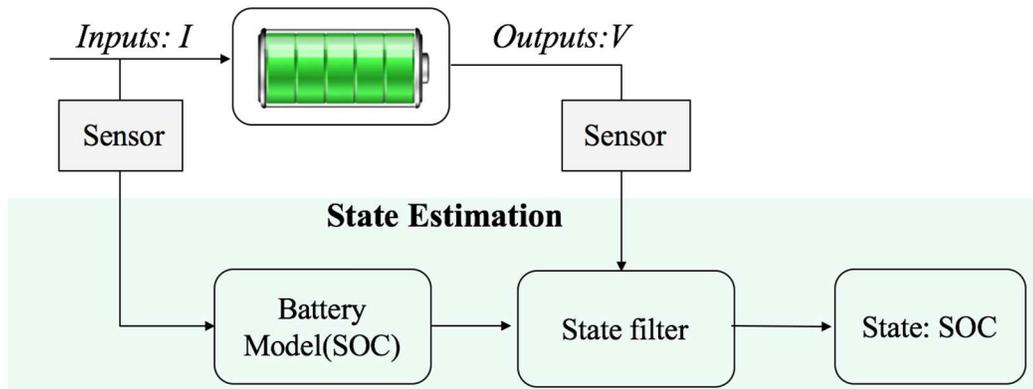


Figure 1: Flowchart of model-based battery SOC estimation algorithm

There are multiple advantages to use state filter. For example, it can handle noisy measurements, and can tolerate the modeling error due to unit to unit variations and un-modeled physics.

State filters require the system model in a state-space form as below:

$$\text{State function: } x_{k+1} = f(x_k, u_k) + w_k \quad (1)$$

$$\text{Measurement function: } y_k = h(x_k, u_k) + v_k \quad (2)$$

where w_k and v_k are process noise and measurement noise, respectively.

The commonly used state filter algorithms include: Kalman filter [30], extended Kalman filter [20, 30-37], and unscented Kalman filter (UKF) [38-47]. Corno *et al.* [48] developed a SOC estimation method based on extended Kalman filter and electrochemical model. Han *et al.* [31] proposed a method to estimate SOC based on physics-based single particle model and extended Kalman filter. Santhanagopalan *et al.* [41] used unscented Kalman filter and electrochemical model to estimate SOC of high power lithium-ion batteries. However conventional Kalman filter based approach converges slowly if initial estimation error is high, and attempts to increase the convergence rate by changing initial covariance values can result in noisy predictions for SOC. Below is an example of using UKF for SOC estimation with electrochemical model. The initial SOC error is 40%. The UKF can eventually converge to true values. However, it takes around half an hour to converge. This can cause inconvenience for users, as one need to wait for half an hour to get the accurate SOC reading. Therefore, the state filter algorithm has to be improved to ensure fast convergence.

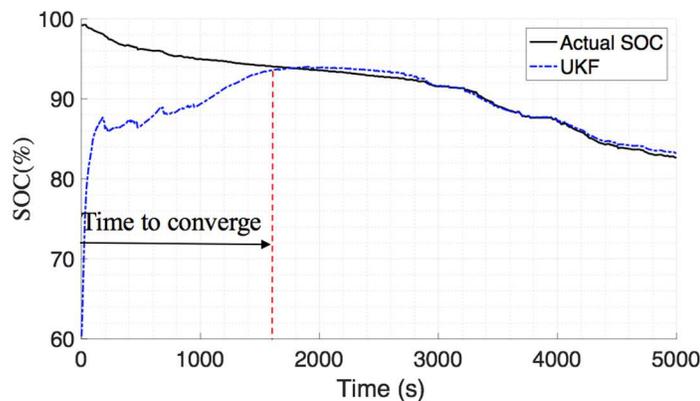


Figure 2: Slow convergence of UKF given 40% initial SOC estimation error

2.1.3 Data Driven Model

Data driven methods use the algorithms, like fuzzy-logic [11], artificial neural networks (NN) [12-16] and support vector machines (SVM) [17-19], to establish a data driven model for SOC estimation. Data-driven models do not require detailed expert knowledge on how the SOC changes with the influence factors like temperature, voltage, current, and ageing. A typical example is the SVM-based SOC estimator for a large-scale lithium-ion polymer (LiP) battery pack proposed by Hansen and Wang [17]. The features used to train the SVM were the current, voltage, SOC at the end of last second and the change in voltage during the last seconds. The SVM estimator was tested with US06 dynamical operational data from the U.S. Department of Energy's Hybrid Electrical Vehicle program. In the dynamic tests, the maximum root-mean-square error is 5.76%. Another example is the fusion approach proposed by Kozlowski *et al.*, [11], which combines the auto-regression moving average model, fuzzy logic and neural network. The features for training were battery voltage, current, temperature and electrochemical parameters obtained from electrochemical impedance spectroscopy test (EIS) [20]. The estimation error was claimed to be less than 5%.

Data driven models can be accurate if the training data is sufficient to cover the loading conditions of the battery. However, the loading conditions of battery powered systems, e.g. EVs, can be complex, and the battery system response can change with current, temperature, SOC and SOH. Collecting training data that has good coverage on all loading conditions can be expensive and time consuming. In addition, there are unit to

unit variations among the cells due to the manufacturing uncertainties, which add difficulties of the implementing of data driven methods to real applications. Therefore, develop methods to improve the generalization capability of data driven model is critical for real-applications.

2.2 Problem Statement and Research Objectives

SOC estimation provides necessary information to the user about how much charge lefts in the battery, and it is essential for battery power/discharging/charging control. SOC estimation is challenging due to many reasons, for example, the sensor measurements can contain noise or bias; SOC itself is a complex function of impedance and loading conditions. Model-based methods are promising for SOC estimation as it is able to handle the sensor noise problem and do not require initial SOC values. Model-based methods include physics-based or data-driven methods.

For physics-based methods, the major problem for real-time SOC estimation is the computational complexity of the coupled partial differential equations (PDEs), which are used to describe the physical process inside the battery. Simplification of the diffusion PDEs in the solid-phase particles has been found to be the key to reduce the computation time and memory requirement of the physics-based models. The available methods for diffusion PDE simplifications suffer from lots of drawbacks. For instants, the commonly used volume averaging method can generate unstable equations, which will lead to unacceptable error. Projection-based methods are promising for the model

reduction of diffusion PDEs. However, the basis function used in the projection will greatly influence the performance, and how to construct an optimal basis function has not been discussed in the literature. In this research, we will address these problems in the diffusion PDE reduction. First, we will develop a methodology to reduce the PDEs to ODEs, which guarantees numerical stability of the generated equations. In addition, we will investigate a method to construct optimal basis functions, which aims to further reduce the computation requirement and improve the accuracy compared with original projection-based methods.

State-filters are required when using physics-based model for SOC estimations. Available state filters are effective to handle measurement noise and modeling uncertainties. However, as we discussed previously, the state-filter can converge slow if the initial error is high, which can cause inconvenience to battery users. Investigating approaches to improve the convergence rate of the state filters is another major objective of this research.

Data-driven models is an alternative to physics-based methods, especially for new battery chemistries, where their material properties are unknown and physical processes are not fully understood. Data-driven models only require the current, voltage, temperature data, and pre-defined training target values, and do not need the detailed physics of the battery. Implantation and development of data-driven model is easier than physics-based methods. However, the generalization of data-driven model is a problem, as data-driven models highly rely on training data. To collect data that covers

all conditions are not practical. Therefore, how to improve generalization capability of data-driven models becomes significant for its applications. Therefore, in this research, we will study algorithms to mitigate the generalization issue of the data-driven approaches. We aim to develop methods to reduce the variations and outliers generated from the data-driven model, which may due to over-fitting of the training data. The developed methods should enable the data-driven model to estimate SOC for different cells and various loading conditions, while learn the model from a cell and under a particular load condition.

Chapter 3: A Computationally Efficient Solution for Battery Diffusion PDEs in Solid Phase Particles

3.1 Introduction

This section develops a computationally efficient numerical solution for the diffusion PDEs in solid phase particles, which contributes to more than 80% of the computation cost of the electrochemical model. The assumption of the developed method is that the reaction flux is homogeneous and the diffusivity is a constant as function of SOC. The PDEs take reaction flux as the input and Li concentration as the output.

The developed numerical solution is based on the projection method with optimized basis functions. The optimized basis functions are constructed with three steps: firstly, define elementary basis functions, e.g. even order monomials; secondly, orthonormalize the elementary basis functions to ensure the numerical stability of the generated equations; thirdly, the optimized basis function is constructed by the linear transformation of the orthonormal basis function. The performance of the new numerical solution is analyzed in frequency domain and time domain with comparison to available methods in the literature.

3.2 Principle of Electrochemical models

Li-ion batteries are electrochemical systems. The performance of Li-ion batteries can be determined by the electrochemical states. For example, the battery output voltage and capacity depend on the lithium concentration in the solid electrochemical particles.

For SOC estimation, the lithium concentration in solid phase can be used to calculate electrode SOC as below:

$$SOC(t) = \frac{3}{LR^3} \int_0^L \int_0^R r^2 \frac{c_s(x, r, t)}{c_{s,max}} dr dx \quad (3)$$

where $c_s(x, r, t)$ is the solid phase concentration, $c_{s,max}$ is the max concentration in solid phase, L is the thickness of the electrode and R is the particle radius.

Electrochemical model can also be used to determine the states that related to battery degradation. The degradation and failure of Li-ion batteries are generally caused by the harmful side-reactions inside the battery. Lithium plating is a typical example that consumes cyclable lithium. The occurrence of side-reactions depends on the overpotential. If the overpotential can be estimated, algorithms can be developed to minimize the side-reactions inside the cell, which results in improving battery reliability. To estimate the electrochemical states, an accurate and computational efficient electrochemical model is required.

The electrochemical model consists of coupled partial differential equations to describe the diffusion, migration and reaction kinetics inside the battery. Solving the electrochemical models is challenging. Traditionally, t

Doyle and Newman [17] have developed an electrochemical model that takes mass transfer, diffusion, migration, and reaction kinetics into account. In the literature, this model is also called the “dualfoil” model. It is the most commonly used model for

simulating the electrochemical process of the Li-ion battery, and it has been extensively validated by experiments [49, 50]. In this model, the battery is divided into three domains, namely, negative electrode, separator, and positive electrode. Figure 3 shows the structure of a Li-ion battery and the governing equations of the physics process in each part of the battery. The lattice structures of the electrode active materials are modeled by small spherical particles. The transport of Li-ions in the solid electrode partials (solid phase) and electrolyte phase is modeled by diffusion equations and coupled by Butler–Volmer reaction kinetics. Using this model, the lithium concentrations and overpotentials inside the battery can be calculated. The major equations describing the physics of the battery are introduced below.

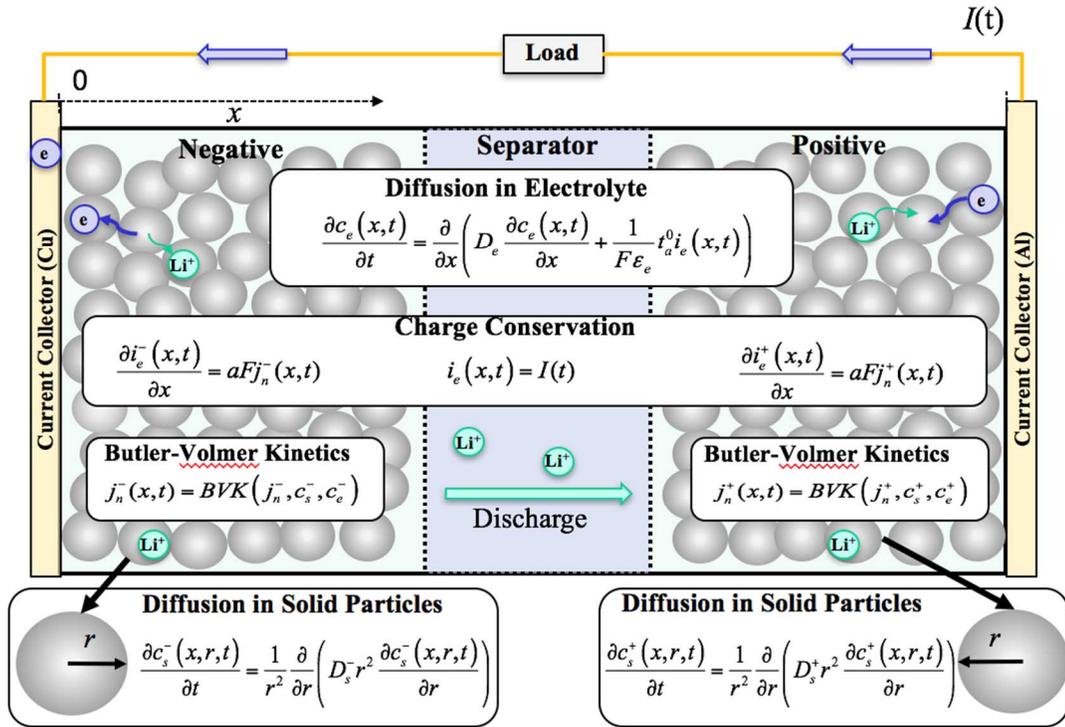


Figure 3: Dualfoil battery electrochemical model

We define an x -axis starting from the negative current collector to the positive current collector, and an r -axis along the radius direction of the solid electrode particle. The transportation of the Li-ions in the solid-phase particles can be modeled by a diffusion equation as follows [17]:

$$\frac{\partial c_s(x, r, t)}{\partial t} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(D_s r^2 \frac{\partial c_s(x, r, t)}{\partial r} \right) \quad (4)$$

with boundary conditions and initial conditions:

$$\left. \frac{\partial c_s}{\partial r} \right|_{r=0} = 0 \quad (5)$$

$$\left. \frac{\partial c_s}{\partial r} \right|_{r=R_p} = -\frac{1}{D_s} j_n \quad (6)$$

$$c_s(x, r, 0) = c_s^0 \quad (7)$$

where c_s is the solid-phase concentration, j_n is the molar flux, D_s is the diffusion coefficients, and R_p is the radius of the solid-phase particle.

The transportation of Li-ions in the electrolyte can be modeled by

$$\frac{\partial c_e(x, t)}{\partial t} = \frac{\partial}{\partial x} \left(D_e \frac{\partial c_e(x, t)}{\partial x} + \frac{1}{F \varepsilon_e} t_a^0 i_e(x, t) \right) \quad (8)$$

where c_e is the lithium concentration in the electrolyte, D_e is the effective diffusion coefficient, ε_e is the volume fraction of the electrolyte, F is Faraday's constant, and t_a^0 is the transference number for the anion.

On the surface of solid-phase particles, an electrochemical reaction occurs when Li-ions are transferred from (to) the solid phase to (from) the electrolyte phase. The reaction flux j_n depends on the overpotential η_s , and this relationship can be described by the Butler–Volmer equation Eq. (9). As

$$j_n(x, t) = \frac{i_0(x, t)}{F} \left[\exp\left(\frac{\alpha_a F}{RT} \eta_s(x, t)\right) - \exp\left(\frac{\alpha_c F}{RT} \eta_s(x, t)\right) \right] \quad (9)$$

where α_a and α_c are transport coefficients; and i_0 is the exchange current density and is given by

$$i_0(x, t) = r_{eff} c_e(x, t)^{\alpha_a} \times (c_{s,max} - c_{ss}(x, t))^{\alpha_a} c_{ss}(x, t)^{\alpha_c} \quad \begin{matrix} (1 \\ 0) \end{matrix}$$

where c_{ss} and $c_{s,max}$ are the surface concentration and the maximum possible concentration in the solid particles of the electrode, and r_{eff} is a reaction rate constant. As seen from (9), the overpotential η_s determines the rate of electrochemical reactions or flux j_n , and it varies with time t and location x . The overpotentials calculated by the solid-phase potential Φ_s , electrolyte potential Φ_e , open-circuit potential $U(c_{ss}(x, t))$, and the flux j_n , as

$$\eta_s(x, t) = \Phi_s(x, t) - \Phi_e(x, t) - U(c_{ss}(x, t)) - FR_f j_n(x, t) \quad (11)$$

where $FR_f j_n(x, t)$ is the potential drop due to the film resistance R_f at the solid electrolyte interface.

In each electrode, the flux j_n is dependent upon the divergence of current density in the electrolyte i_e , as follows:

$$\frac{\partial i_e(x, t)}{\partial x} = aFj_n(x, t) \quad (12)$$

where a is the specific interfacial area of the particle. The i_e should be zeroes at the current collectors and should be equal to the applied current density, I , at the separator.

3.2 Projection-based model reduction method

Traditional finite element and finite difference methods to solve the electrochemical model can generate thousands of states making these methods virtually impossible for on-line applications. To implement the electrochemical model in a real-time BMS, where the memory and computational power is limited, an efficient numerical solution for the diffusion PDEs must be developed. Projection-based methods have been developed to reduce the PDEs to a low-order DAEs system [9]. The projection-based method approximates the concentration $c_s(r, t)$ by a set of basis functions with time-varying coefficients as follows:

$$c_s(r, t) = \mathbf{\Phi}^T(r) \cdot \mathbf{a}(t) \quad (13)$$

where $\mathbf{\Phi}(r) = [\phi_1(r), \phi_2(r), \dots, \phi_N(r)]^T$ and $\mathbf{a}(t) = [a_1(t), a_2(t), \dots, a_N(t)]^T$. Once $a_i(t)$ is determined, $c_s(r, t)$ can be found accordingly. Therefore, the problem is converted into a problem of finding time-varying coefficients $a_i(t)$. First, N equations are needed, as there are N unknown $a_i(t)$. The projection-based method is used to generate the N equations based on the diffusion equation.

A function $P: U \rightarrow U$ is called a projector if it is: (1) linear: $P(a \cdot f_1 + b \cdot f_2) = a \cdot P(f_1) + b \cdot P(f_2)$; and (2) idempotent: $P^2(f) = P(f)$. For example, the Galerkin projector for the function $f(r)$ is

$$P_{\beta_i}(f(r)) = \frac{\int_0^{R_p} r^2 f(r) \cdot \beta_i dr}{\int_0^{R_p} r^2 \beta_i \cdot \beta_i dr} \cdot \beta_i \quad (14)$$

where β_i is called the test function of the projector. With different test functions, the projection results are different. This property is useful for constructing different equations to calculate the time-varying coefficients in (13).

In this paper, we design the basis function to satisfy the first boundary condition at the center of the particle in (5). An example of such a basis function is the even-order monomials (i.e., $1, r^2, r^4 \dots$). Plugging in the approximation to the boundary condition at the surface of the particle, we obtain the first equation

$$D_s \left(\frac{\partial \Phi^T(x)}{\partial r} \Big|_{r=R_p} \right) \cdot \mathbf{a}(t) = -j_n \quad (15)$$

The remaining $N - 1$ equations are obtained by applying the projection $N - 1$ times to both sides of the solid-phase diffusion PDE using different basis functions β_i , $i = 1, \dots, N - 1$,

$$\begin{aligned} P_{\beta_i} \left(\frac{\partial c_s(r, t)}{\partial t} \right) &= P_{\beta_i} \left(\frac{1}{r^2} \frac{\partial}{\partial r} \left(D_s r^2 \frac{\partial c_s(r, t)}{\partial r} \right) \right) \\ &\Rightarrow \int_0^{R_p} r^2 \frac{\partial c_s(r, t)}{\partial t} \beta_i(r) dr \\ &= \int_0^{R_p} r^2 \left(\frac{1}{r^2} \frac{\partial}{\partial r} \left(D_s r^2 \frac{\partial c_s(r, t)}{\partial r} \right) \right) \beta_i(r) dr \end{aligned} \quad (16)$$

Integrating by parts for the right-hand side of (16) and applying the boundary condition in (5) and (6), we have

$$\begin{aligned} &\int_0^{R_p} r^2 \frac{\partial c_s(r, t)}{\partial t} \beta_i(r) dr \\ &= - \int_0^{R_p} D_s r^2 \frac{\partial c_s(r, t)}{\partial r} \frac{\partial \beta_i(r)}{\partial r} dr - R_p^2 \beta_i(R) j_n, \end{aligned} \quad (17)$$

$i = 1, \dots, N - 1.$

Plugging the approximation (13) into (17), we obtain

$$\begin{aligned}
& \left(\int_0^{R_p} r^2 \boldsymbol{\Phi}^T(r) \beta_i(r) dr \right) \cdot \dot{\mathbf{a}}(t) \\
& = \left(- \int_0^{R_p} D_s r^2 \frac{\partial \boldsymbol{\Phi}^T(r)}{\partial r} \frac{\partial \beta_i(r)}{\partial r} dr \right) \cdot \mathbf{a}(t) \\
& + \left(-R_p^2 \beta_i(R_p) \right) j_n, \\
& \quad i = 1, \dots, N-1
\end{aligned} \tag{18}$$

Then, the obtained differential algebraic equations are summarized as follows:

$$\begin{aligned}
\mathbf{M} \cdot \dot{\mathbf{a}}(t) &= \mathbf{A} \cdot \mathbf{a}(t) + \mathbf{B} \cdot j_n \\
\mathbf{E} \cdot \mathbf{a}(t) &= -j_n \\
\mathbf{c}_{ss}(t) &= \mathbf{C} \cdot \mathbf{a}(t)
\end{aligned} \tag{19}$$

where

$$\begin{aligned}
\mathbf{M} &= \left[\int_0^{R_p} r^2 \boldsymbol{\beta}(r) \boldsymbol{\Phi}^T(r) dr \right]_{N-1 \times N} \\
\mathbf{A} &= \left[- \int_0^{R_p} D_s r^2 \frac{\partial \boldsymbol{\beta}(r)}{\partial r} \frac{\partial \boldsymbol{\Phi}^T(r)}{\partial r} dr \right]_{N-1 \times N} \\
\mathbf{B} &= \left[-R_p^2 \boldsymbol{\beta}(R_p) \right]_{N-1 \times 1} \\
\mathbf{E} &= \left[D_s \frac{\partial \boldsymbol{\Phi}^T(r)}{\partial r} \Big|_{r=R_p} \right]_{1 \times N} \\
\mathbf{C} &= \left[\boldsymbol{\Phi}^T(R_p) \right]_{1 \times N}
\end{aligned} \tag{20}$$

3.3 Orthonormalization of Basis Functions

To ensure numerical stability and avoid large condition number, Φ and β have to be orthonormal basis functions, such that

$$\text{Proj}_{\beta_i}(\phi_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (21)$$

and then the matrix \mathbf{M} becomes

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \quad (22)$$

Let us assume that the original elementary basis set is $\mathcal{S} = \{\Phi_1, \Phi_2, \dots, \Phi_k\}$. The Gram–Schmidt algorithm is used to generate a set of orthonormal basis functions $\hat{\mathcal{S}} = \{u_1, u_2, \dots, u_k\}$ that spans the same k -dimensional space of \mathbb{R}^n as \mathcal{S} , and

$$\langle u_i, u_j \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (23)$$

The steps of the Gram–Schmidt algorithm are listed as follows:

	<u>Orthogonalizing</u>	Normalizing
Step 1:	$u_1 = \phi_1$	$u_1 = \frac{u_1}{\ u_1\ }$
Step 2:	$u_2 = \phi_2 - \text{Proj}_{u_1}(\phi_2)$	$u_2 = \frac{u_2}{\ u_2\ }$
Step 3:	$u_3 = \phi_3 - \text{Proj}_{u_1}(\phi_3) - \text{Proj}_{u_2}(\phi_3)$	$u_3 = \frac{u_3}{\ u_3\ }$
\vdots	\vdots	\vdots
Step k :	$u_k = \phi_k - \sum_{j=1}^{k-1} \text{Proj}_{u_j}(\phi_k)$	$u_k = \frac{u_k}{\ u_k\ }$

3.4 Optimal Basis Function Construction

The selection of basis function has a significant effect on the approximation accuracy. In this study, we develop a method to construct an optimal basis function from a set of elementary basis functions. First, we define a set of high-order orthonormal elementary basis functions, \mathbf{U} , formed by applying the Gram–Schmidt algorithm to even-order monomials (i.e., $1, r^2, r^4 \dots$)

$$\mathbf{U} = [u_1, u_2, \dots, u_N]^T \quad (24)$$

We also choose the same \mathbf{U} as the elementary test functions in the projection to satisfy the orthonormality between the basis and test functions in (21). Then the \mathbf{M} , \mathbf{A} , \mathbf{B} , \mathbf{E} , and \mathbf{C} matrices in the DAE (19) become

$$\begin{aligned}
\mathbf{M} &= \left[\int_0^{R_p} r^2 \mathbf{U}(r) \mathbf{U}^T(r) dr \right]_{N-1 \times N} \\
\mathbf{A} &= \left[- \int_0^{R_p} D_s r^2 \frac{\partial \mathbf{U}(r)}{\partial r} \frac{\partial \mathbf{U}^T(r)}{\partial r} dr \right]_{N-1 \times N} \\
\mathbf{B} &= \left[-R_p^2 \mathbf{U}(R_p) \right]_{N-1 \times 1} \\
\mathbf{E} &= \left[D_s \frac{\partial \mathbf{U}^T(r)}{\partial r} \Big|_{r=R_p} \right]_{1 \times N} \\
\mathbf{C} &= \left[\mathbf{U}^T(R_p) \right]_{1 \times N}
\end{aligned} \tag{25}$$

The construction of the optimal basis function Ψ and test function β is accomplished by using a linear transformation given by

$$\begin{aligned}
\Psi &= \mathbf{V} \times \mathbf{U} \\
\beta &= \mathbf{W} \times \mathbf{U}
\end{aligned} \tag{26}$$

where \mathbf{V} and \mathbf{W} are k by N matrices, and N and k are the original and reduced dimensionality with $k \ll N$. To ensure the orthonormality of the Ψ and β , we must have $\mathbf{W}^T \mathbf{V} = \mathbf{I}$.

The concentration $c_s(r, t)$ can then be approximated by

$$c_s(r, t) = \Psi^T \mathbf{a}_r(t) \tag{27}$$

Plugging the basis and test functions in (26) into (19) and (20), the DAEs generated by the optimized basis and test functions are

$$\begin{aligned}
\mathbf{M}_r \cdot \dot{\mathbf{a}}_r(t) &= \mathbf{A}_r \cdot \mathbf{a}_r(t) + \mathbf{B}_r \cdot j_n \\
\mathbf{E}_r \cdot \mathbf{a}_r(t) &= -j_n \\
c_{ss}(t) &= \mathbf{C}_r \cdot \mathbf{a}_r(t),
\end{aligned} \tag{28}$$

where

$$\begin{aligned}
\mathbf{M}_r &= \left[\mathbf{W} \left(\int_0^{R_p} r^2 \mathbf{U}(r) \mathbf{U}^T(r) dr \right) \mathbf{V}^T \right]_{k-1 \times k} \\
\mathbf{A}_r &= \left[\mathbf{W} \left(- \int_0^{R_p} D_s r^2 \frac{\partial \mathbf{U}(r)}{\partial r} \frac{\partial \mathbf{U}^T(r)}{\partial r} dr \right) \mathbf{V}^T \right]_{k-1 \times k} \\
\mathbf{B}_r &= \left[\mathbf{W} \left(-R_p^2 \mathbf{U}(R_p) \right) \right]_{k-1 \times 1} \\
\mathbf{E}_r &= \left[\left(D_s \frac{\partial \mathbf{U}^T(r)}{\partial r} \Big|_{r=R_p} \right) \mathbf{V}^T \right]_{1 \times k} \\
\mathbf{C}_r &= \left[\left(\mathbf{U}^T(R_p) \right) \mathbf{V}^T \right]_{1 \times k}.
\end{aligned} \tag{29}$$

Therefore, the number of DAEs is further reduced to k .

Suppose the transfer function of the DAEs based on the high-order elementary basis function is $H(s)$, and the transfer function of the DAEs generated by the low-order

optimized basis function is $H_r(s)$. The objective of the optimization is to find the transform matrices \mathbf{V} and \mathbf{W} to minimize the 2-norm error between $H(s)$ and $H_r(s)$:

$$\|H - H_r\|_2 = \int_{-\infty}^{+\infty} |H(i\omega) - H_r(i\omega)|^2 d\omega \quad (30)$$

The Meier–Luenberger condition [51] provides the first-order necessary optimality condition for the optimization problem (30), which is stated as follows:

Theorem 1.1 Let $H(s)$ be the full-order system, and $H_r(s)$ be the minimizer of $\|H(s) - H_r(s)\|_2$ with simple poles $\hat{\lambda}_1, \dots, \hat{\lambda}_r$. Then $H(-\hat{\lambda}_k) = H_r(-\hat{\lambda}_k)$ and $H'(-\hat{\lambda}_k) = H_r'(-\hat{\lambda}_k)$ for $i = 1, \dots, r$.

To satisfy this necessary condition, we need to interpolate $H(s)$ and its derivative at the mirror image of the poles of the reduced-order system. For linear stable ODEs, the rational Hermite interpolation provides a way to determine the \mathbf{W} and \mathbf{V} matrices to obtain H_r , and below is its definition:

Given $H(s) = \mathbf{C}^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B}$, and r distinct points $\sigma_1, \dots, \sigma_r \in \mathbb{C}$, let

$$\begin{aligned} \text{Ran}(\mathbf{V}) &= \text{span}\{(\sigma_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, \dots, (\sigma_r\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\} \\ \text{Ran}(\mathbf{W}) &= \text{span}\{(\sigma_1\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{C}, \dots, (\sigma_r\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{C}\}. \end{aligned} \quad (31)$$

Define $\mathbf{A}_r = (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{B}_r = (\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T \mathbf{B}$, $\mathbf{C}_r = \mathbf{C} \mathbf{V}$. Then $H_r(s) = \mathbf{C}_r^T (s \mathbf{I}_r - \mathbf{A}_r)^{-1} \mathbf{B}_r$ is a rational Hermite interpolate to $H(s)$ at $\sigma_1, \dots, \sigma_r$, and $H(\sigma_i) = H_r(\sigma_i)$ and $H'(\sigma_i) = H_r'(\sigma_i)$, for $i = 1, \dots, r$.

The rational Hermite interpolate can be directly used to find \mathbf{V} and \mathbf{W} to construct H_r , if the poles of H_r are known. However, this information is not known as a prior. In this case, the IRKA can be used to iteratively correct the interpolation points until the local minimum is found [52]. The IRKA is defined below.

Iterative Rational Krylov Algorithm (IRKA)

1. Make an initial selection of σ_i for $i = 1, \dots, r$ that is closed under conjugation.
2. Choose \mathbf{V} and \mathbf{W} such that:

$$\text{Ran}(\mathbf{V}) = \text{span}\{(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{B}, \dots, (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{B}\}$$

$$\text{Ran}(\mathbf{W}) = \text{span}\{(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{C}, \dots, (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{C}\}$$

and $\mathbf{W}^T \mathbf{V} = \mathbf{I}$.

3. While relative change in $\{\sigma_i\} > \text{tol}$

- a. $\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}$
- b. Assign $\sigma_i \leftarrow -\lambda_i(\mathbf{A}_r)$ for $i=1, \dots, r$
- c. Update \mathbf{V} and \mathbf{W} , such that

$$\text{Ran}(\mathbf{V}) = \text{span}\{(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{B}, \dots, (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{B}\}$$

$$\text{Ran}(\mathbf{W}) = \text{span}\{(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{C}, \dots, (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{C}\}$$

and $\mathbf{W}^T \mathbf{V} = \mathbf{I}$.

4. $\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{B}_r = \mathbf{W}^T \mathbf{B}$, $\mathbf{C}_r = \mathbf{C} \mathbf{V}$

The IRKA was developed originally for asymptotically stable systems, i.e., all of the eigenvalues of the pencil (\mathbf{A}, \mathbf{I}) must be in the left half of the complex plane. However, our system is unstable, therefore the IRKA is not directly applicable.

To solve this problem, we first convert the DAEs into ODEs. Let us assume that $\mathbf{R} = [\mathbf{M}; \mathbf{E}]$. If \mathbf{R} is invertible, then we can transform the DAEs in (19) to ODEs [28]. Let

$$\begin{aligned} \mathbf{z} &= \mathbf{M} \cdot \mathbf{a} \\ \mathbf{R}^{-1} &= [\mathbf{Q}_1 \quad \mathbf{Q}_2] \end{aligned} \quad (32)$$

where $\mathbf{Q}_1 \in \mathbb{R}^{N-1 \times N}$ and $\mathbf{Q}_2 \in \mathbb{R}^{N-1 \times 1}$, then we have

$$\mathbf{a} = \begin{bmatrix} \mathbf{M} \\ \mathbf{E} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z} \\ -j_n \end{bmatrix} = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{z} \\ -j_n \end{bmatrix} = \mathbf{Q}_1 \mathbf{z} - \mathbf{Q}_2 j_n \quad (33)$$

Plugging (33) into (19), we obtain the ODEs:

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{A}(\mathbf{Q}_1 \mathbf{z} - \mathbf{Q}_2 j_n) + \mathbf{B} j_n = \mathbf{A} \mathbf{Q}_1 \mathbf{z} + (\mathbf{B} - \mathbf{A} \mathbf{Q}_2) j_n \\ c_{ss}(t) &= \mathbf{C}(\mathbf{Q}_1 \mathbf{z} - \mathbf{Q}_2 j_n) = \mathbf{C} \mathbf{Q}_1 \mathbf{z} - \mathbf{C} \mathbf{Q}_2 j_n \end{aligned} \quad (34)$$

Since $[u_1, u_2, \dots, u_N]$ are orthonormal, $[\mathbf{M}; \mathbf{E}]$ is

$$\begin{bmatrix} \mathbf{M} \\ \mathbf{E} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & D_s u_2' & D_s u_3' & \cdots & D_s u_{N-1}' & D_s u_N' \end{bmatrix} \quad (35)$$

It is straightforward to prove that the inverse of (35) is

$$\begin{bmatrix} \mathbf{M} \\ \mathbf{E} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & -\frac{u_2'}{u_N'} & -\frac{u_3'}{u_N'} & \cdots & -\frac{u_{N-1}'}{u_N'} & \frac{1}{D_s u_N'} \end{bmatrix} \quad (36)$$

Because $u_1=1$, the first column and first row of \mathbf{A} in (20) are zeros. Therefore, the first row and first column of $\mathbf{A}\mathbf{Q}_1$ are zeros, and the ODEs are obtained as follows:

$$\begin{aligned} \begin{bmatrix} \dot{z}_1(t) \\ \dot{\mathbf{z}}_2(t) \end{bmatrix} &= \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{A}}_2 \end{bmatrix} \mathbf{z}(t) + \begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{B}}_2 \end{bmatrix} \cdot j_n(t) \\ c_{ss}(t) &= [\bar{\mathbf{C}}_1 \quad \bar{\mathbf{C}}_2] \cdot [z_1(t), \mathbf{z}_2(t)]^T + \bar{\mathbf{D}} \cdot j_n(t) \end{aligned} \quad (37)$$

where $\mathbf{z}_2 = [z_2, z_3, \dots, z_N]^T$. Then, we decompose the system into a stable part and an unstable part, and the IRKA is only applied to the stable part of the system.

$$\begin{cases} \dot{z}_1(t) = \bar{\mathbf{B}}_1 \cdot j_n(t) \\ \dot{\mathbf{z}}_2(t) = \bar{\mathbf{A}}_2 \cdot \mathbf{z}_2(t) + \bar{\mathbf{B}}_2 \cdot j_n(t) \end{cases} \quad (38)$$

$$y(t) = \bar{\mathbf{C}}_1 \cdot z_1(t) + \bar{\mathbf{C}}_2 \cdot \mathbf{z}_2(t) + \bar{\mathbf{D}} \cdot j_n(t)$$

The transfer function of (38) is

$$H(s) = \bar{\mathbf{C}}_1 s^{-1} \bar{\mathbf{B}}_1 + \underbrace{\bar{\mathbf{C}}_2 (s \cdot \mathbf{I} - \bar{\mathbf{A}}_2)^{-1} \bar{\mathbf{B}}_2}_{\text{to be reduced by IRKA}} + \bar{\mathbf{D}} \quad (39)$$

In the IRKA, only $\bar{\mathbf{C}}_2(s\mathbf{I} - \bar{\mathbf{A}}_2)^{-1}\bar{\mathbf{B}}_2$, will be reduced, and the remaining part stays the same. Supposing that the transformation matrices generated in IRKA are \mathbf{V}_2 and \mathbf{W}_2 , then the transformation matrices \mathbf{V} and \mathbf{W} are

$$\mathbf{V} = \begin{bmatrix} 1 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{V}_2 & \mathbf{0} \\ 0 & \mathbf{0} & 1 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 1 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{W}_2 & \mathbf{0} \\ 0 & \mathbf{0} & 1 \end{bmatrix} \quad (40)$$

Then we can use (26) to obtain the optimized basis function. Figure 4 summaries the steps of the optimal projection method developed in this thesis for solving diffusion PDEs

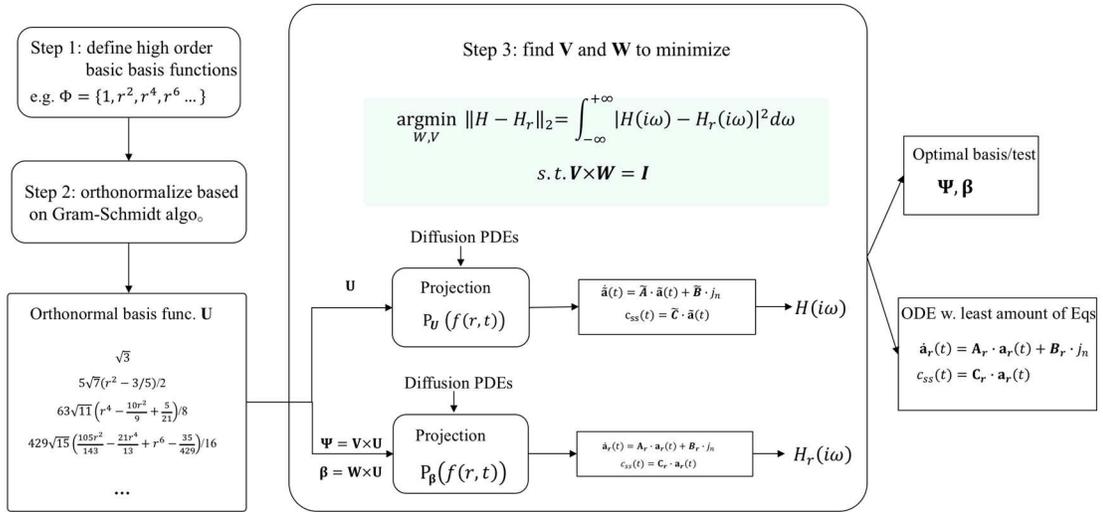


Figure 4: Flowchart of the optimal projection method for solving diffusion PDEs

3.5 Results and Discussions

In this study the optimal basis function is constructed from $\Phi(x) = [1, r^2, r^4, \dots, r^{20}]^T$. First, the Gram-Schmidt algorithm is applied to Φ to obtain the orthonormal elementary basis and test functions, $\mathbf{U} = [u_1, u_2, \dots, u_N]^T$. The model parameters, including the diffusion coefficient $D_s = 3.9 \times 10^{-10}$ cm²/s and particle radius $R_p = 12.5$

$\times 10^{-4}$ cm, were obtained from [49]. These model parameters have shown good agreement with experimental data of $\text{Li}_x\text{C}_6\backslash\text{Li}_y\text{Mn}_2\text{O}_4$ cells in [10].

To compare the developed approach with available methods, the frequency responses of different methods are calculated, including finite difference (FD), volume averaging (VA), projection with even-order monomials (P), and our proposed method: projection with optimized basis functions (OP). The actual transfer function of the diffusion equation is given by [53]

$$H_T(s) = \frac{R_p}{D_s} \frac{\tanh(R_p \sqrt{s/D_s})}{(\tanh(R_p \sqrt{s/D_s}) - R_p \sqrt{s/D_s})}. \quad (41)$$

Figure 5 shows the frequency responses of different methods compared with the true transfer function. At low frequency (e.g., $<10^{-2}$ rad/sec), all methods well match the true transfer function. As the frequency increases, we see different performances for different methods. Overall, the FD method with 500 nodes (FD500) shows the best performance. However, it requires solving 500 equations, which is not feasible for real-time applications. Our proposed projection method with optimized basis functions (OP) shows the best balance between accuracy and computational burden among those methods. Figure 5 shows that OP5 outperforms all other methods excepting for FD500. For example, OP5 is better than FD100 in high frequency domain, while it only requires solving about 20 times fewer equations compared to FD100. As expected, OP5 also outperforms VA and the projection with un-optimized basis functions (P5). The VA method developed in [26] performs well in the low-frequency domain but generates

large errors in the high-frequency domain. In addition, VA3 with 3 basis functions is worse than VA2 with 2 basis functions, although more basis functions are used in VA3, because the system equations generated by VA become unstable when using high order monomials.

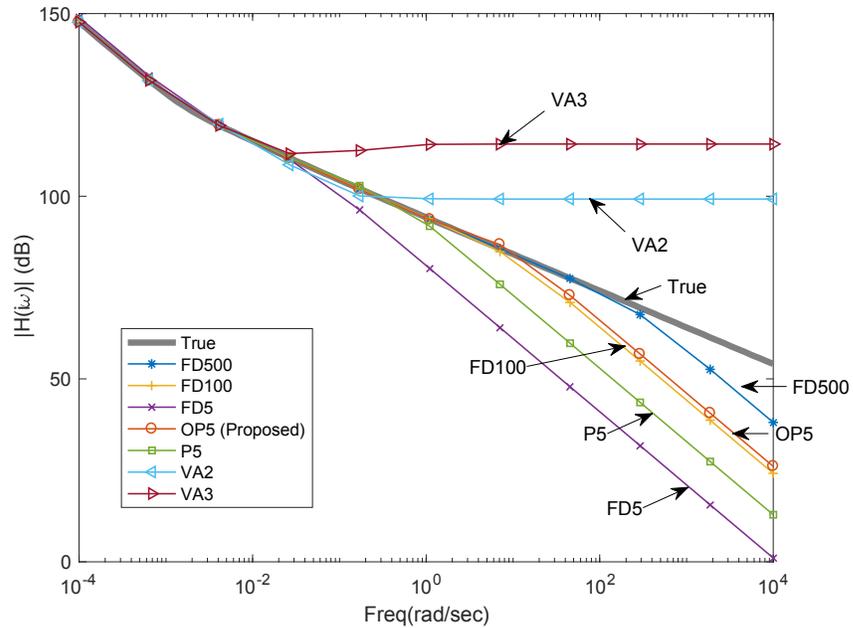


Figure 5: Frequency responses of different methods compared to the true transfer function

(FD5: finite difference with 5 nodes; FD100: finite difference with 100 nodes; FD500: finite difference with 500 nodes; P5: projection with un-optimized basis; OP5: our proposed projection method with optimized basis; VA2: volume averaging with 2 basis functions; VA3: volume averaging with 3 basis functions)

More simulations were conducted to illustrate how the error changes with the number of states (or the convergence rate). Based on this investigation, we can determine how

many states are really necessary. Figure 6 shows a convergence rate comparison of the different methods. The relative error is calculated by the following equation in the frequency domain:

$$Error = \int_0^w \frac{\|H(i\omega) - H_T(i\omega)\|_2}{\|H_T(i\omega)\|_2} d\omega \quad (42)$$

The OP shows an exponential convergence rate, while the finite difference and the projection with original un-optimized basis present a linear convergence rate. The advantage of the proposed OP method is obvious. From Figure 6, the accuracy of the proposed OP with 3 states is similar to that of the un-optimized basis with 10 states and is 2 times better than FD with 10 states. With 5 states, the OP is almost converged, and there is no need to add more states.

To compare the performance in time domain, three simulation studies are conducted with the time-varying flux $j_n(t)$. The first case is a 50s discharge step followed by a 950s rest step as shown in Figure 7. This is to evaluate the algorithms' performance in slow dynamics and steady-state simulations. The benchmarking result is obtained by using a FD method with 500 nodes, which is an accurate solution due to the high number of nodes used. The time-domain outputs using different methods are calculated and plotted in Figure 8. In the 50s discharge step, the results of FD100, P5 and OP5 are on top of that of FD500, meaning they are accurate for slow dynamics. VA2 and FD5 are less accurate as they show obvious discrepancy from FD500 in the concentration estimation. In the rest step, after 950s rest, the system dynamics should set down and

converge to steady state. In this step, most algorithms including FD500, FD100, P5, OP5, and VA2 are converged to the same steady-state solution. However, FD5 shows a different steady-state solution, which can cause issues in actual applications, for example, errors in state of charge estimation.

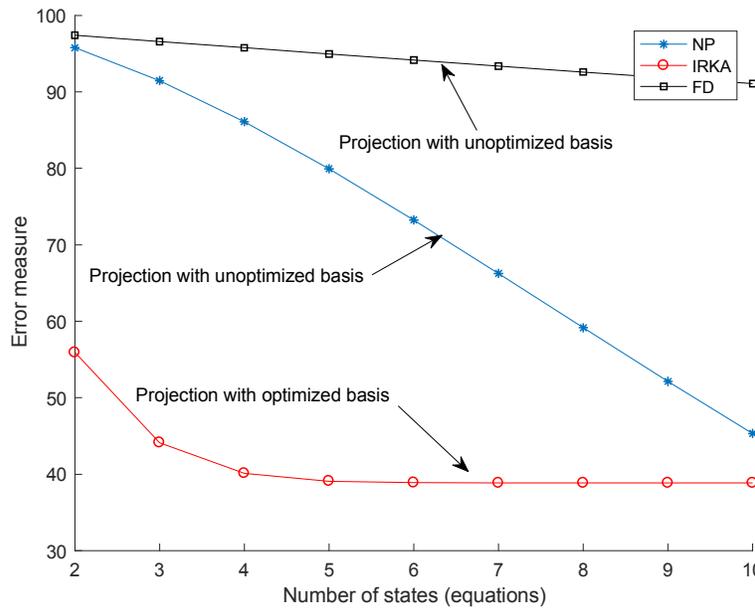


Figure 6: Convergence comparison of different methods

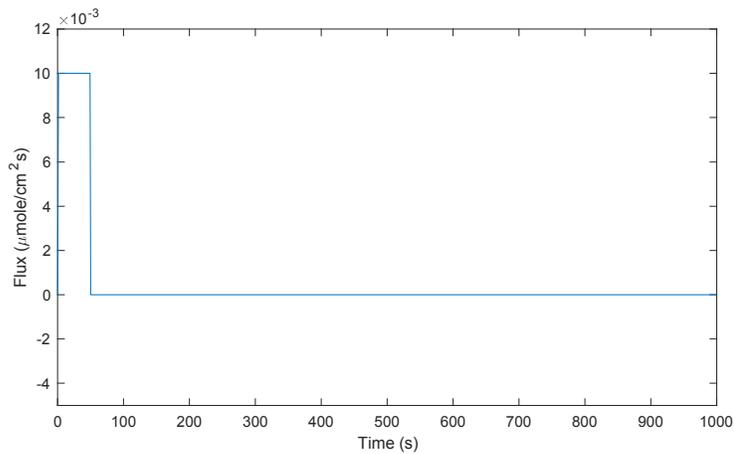


Figure 7: 50s flux and 950s rest applied to the surface of the particle

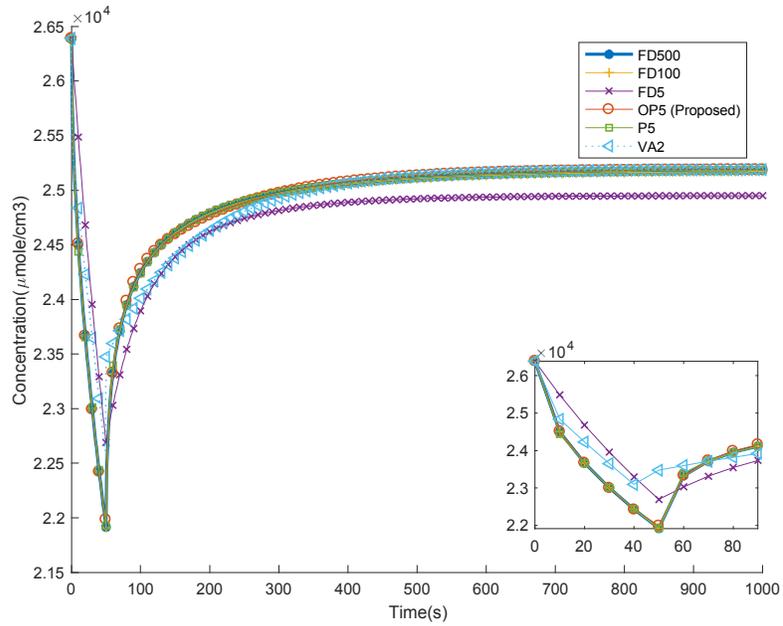


Figure 8: Time domain outputs of different methods for slow dynamics

The second simulation uses a pulse series with frequency 5Hz and duration 100s, which is to evaluate the algorithms' accuracy in the simulation for fast dynamics. The waveform of the pulses is shown in Figure 9 (only the first second data is plotted in order to show the data clearly). The time-domain results are plotted in Figure 10 and the error comparing to baseline (FD500) is shown in Figure 11. Similar to the frequency domain result in Figure 5, in the high frequency simulations, our proposed OP5 shows obvious advantages comparing to P5, VA2, and FD5 as it better matches with the results of FD500. The VA2 and FD5 perform rather poorly and show large swings from the baseline.

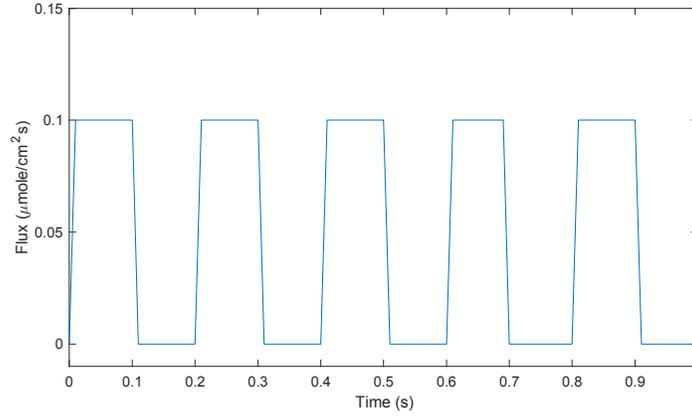


Figure 9: 5Hz pulses applied to the surface of the particle

The third case is based on federal driving schedule (FDS), which is to simulate the actual loading condition of a EV battery. The time-domain current waveform is shown in Figure 12, and its Fourier transform is shown in Figure 13. From Figure 13, we know that the main frequency components of FDS are less than 2 Hz. In Figure 5, the transfer function of FD500 overlaps with the true transfer function for frequencies < 2 Hz. Hence, here we can still use FD500 as baseline for the comparison. Figure 14 shows the simulation results of FDS based on different algorithms, and Figure 15 shows the simulation errors. Similar to the previous case, FD100 and OP5 are still the best among all methods. Table 1 summaries the comparison of the Root Mean Square Error (RMSE), computational time, and memory requirements (for the equation storage) of difference algorithms in the simulation of the FDS. OP5 has the least RMS error among all algorithms. OP5 reduces the computational time and memory by over 20 and 200 times respectively compared to FD 100, and in the mean time, it is more accurate. With the same number of states, the proposed OP5 is also much better than the conventional P5 method, where the RMS error is reduced from 92 to 14.

The simulations above demonstrate that FD100, P5, and the proposed OP5 performs similarly for low frequency inputs in terms of accuracy. For high-frequency inputs, our developed method is superior to FD100 and P5. For applications where the current profile is stationary (e.g., constant current discharge), either FD100, P5, or OP5 should provide good performance. For dynamical applications, for instance, EVs, where the current consist of various transients due to starting, braking, and accelerating, the proposed OP5 method is the best among all the methods investigated in terms of accuracy and computational efficiency.

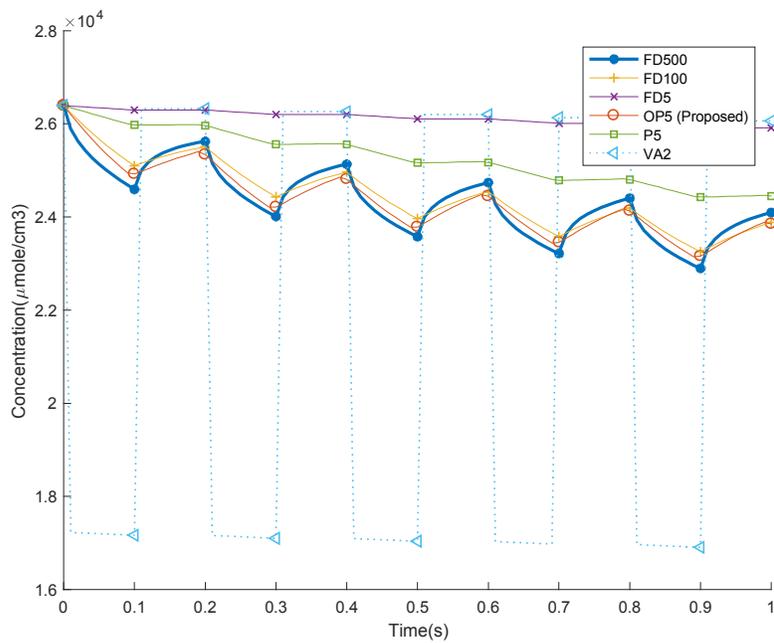


Figure 10: Time domain outputs of different methods for 5Hz pulses

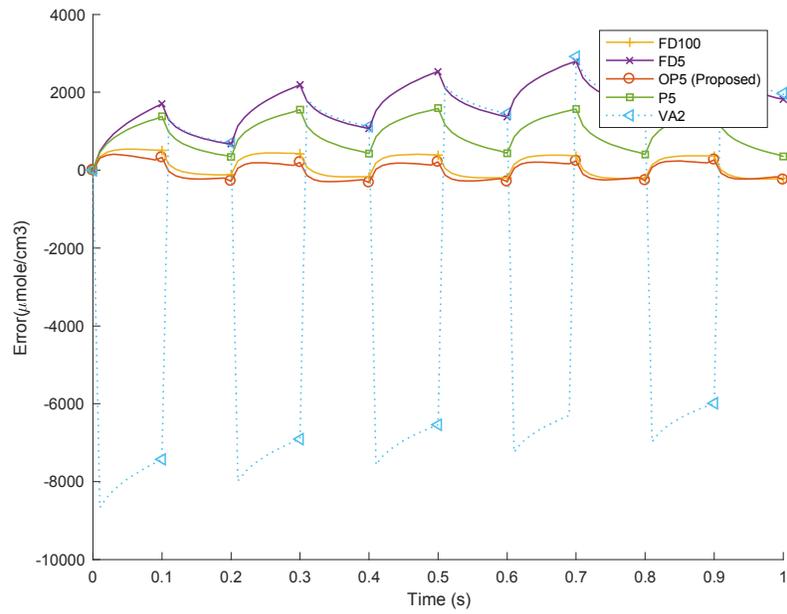


Figure 11: Errors of different methods compared to FD with 500 nodes

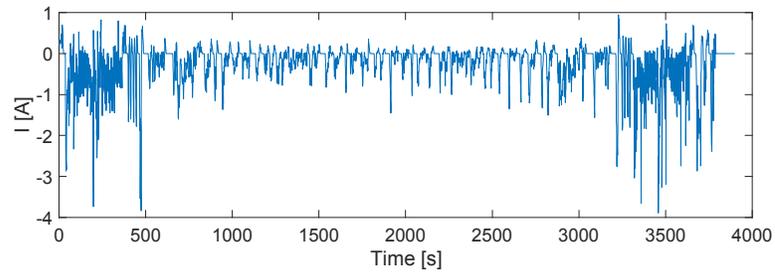


Figure 12: Federal driving schedule (FDS) profile

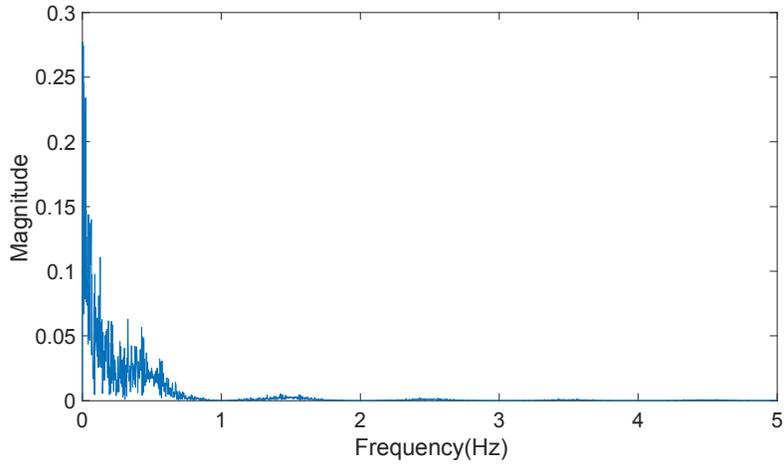


Figure 13: Fourier transform of FDS

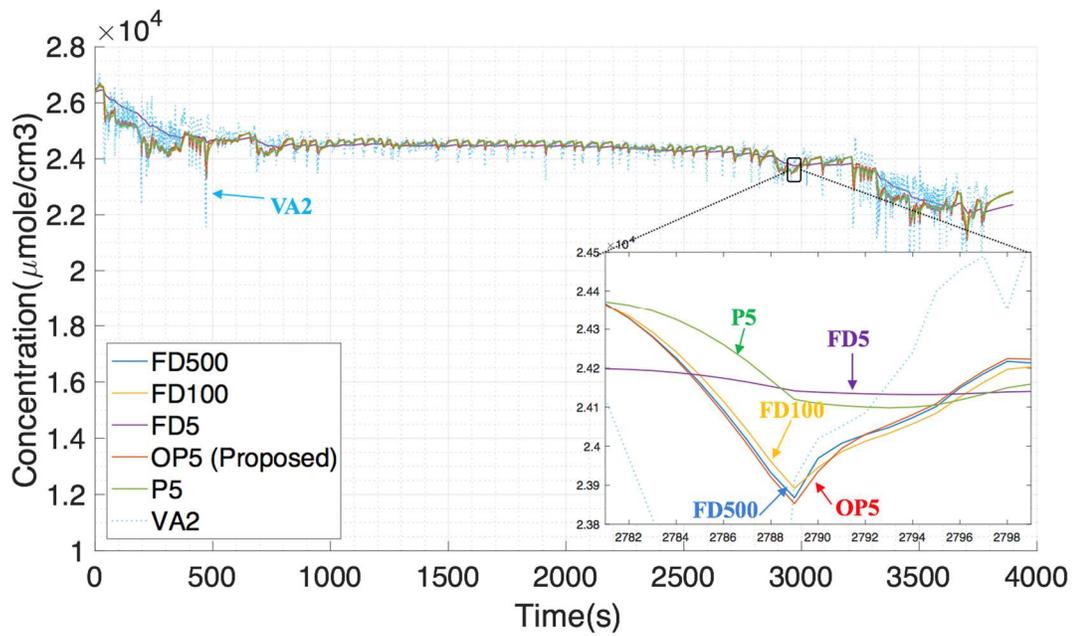


Figure 14: Time-domain simulation of different algorithms based on FDS

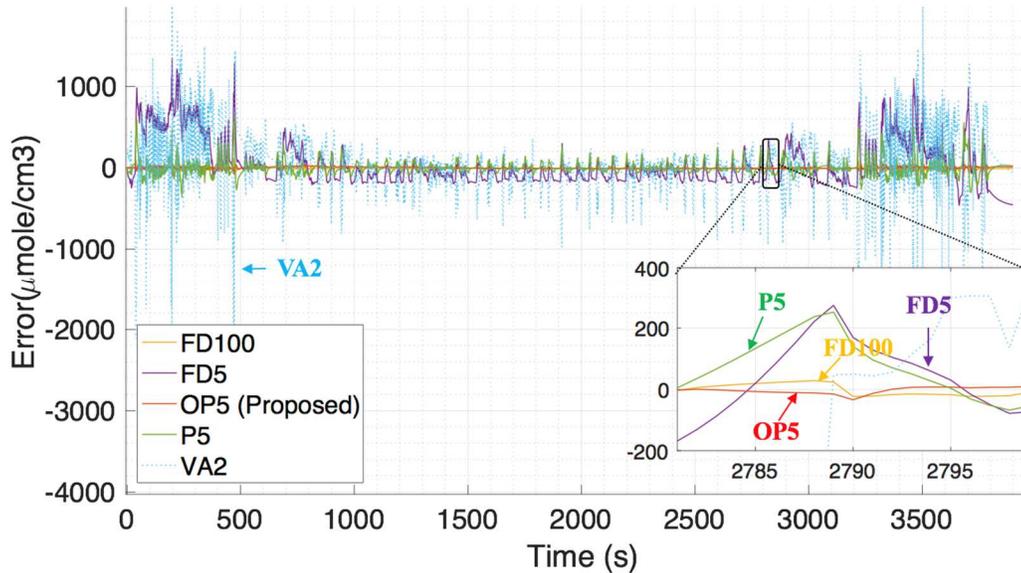


Figure 15: Time-domain simulation errors of different algorithms based on FDS

Table 1: Comparison of algorithm performance for diffusion PDE calculation

Algorithm	Simulation time (s)	# equations	RMS error ($\mu\text{mol}/\text{cm}^3$)	Memory (byte)
FD100	12	100	18	81680
FD5	0.44	5	271	288
OP5 (Proposed)	0.42	5	14	288
P10	1.21	10	16	1768
P5	0.43	5	92	488
VA2	0.41	2	382	88

3.6 Summary

Computational complexity is the major bottleneck for the application of any realistic electrochemical model in a real-time battery management system. In particular, the coupled solid-phase partial differential diffusion equations are the most computationally intensive. In this paper, a projection-based method with optimized orthonormal basis functions was developed to generate a reduced order model (ROM) from the solid-phase diffusion equations. The optimized basis function was constructed by linear transforming a set of orthonormal elementary basis functions. The linear transformation matrices were found by using the iterative rational Krylov algorithm. Simulation studies were conducted to compare the performance of the developed method with other commonly used algorithms, including finite difference, volume average and the projection method with conventional basis functions. The results show that the proposed method with optimized basis functions is the best in terms of accuracy and computational complexity among all the investigated algorithms. Particularly, in the simulation of high frequency inputs, which is common for EVs, our proposed method can reduce the root mean square error by more than 5 times compared to the conventional basis functions. In addition, compared with finite difference method, our proposed method can cut the computational time and memory requirement by over 20 times and 200 times, respectively, while it does not lose any accuracy. The developed method can also be applied to solve electrolyte diffusion PDEs, and with proper modification, the developed method can be extended to solve 2-D or 3-D diffusion PDEs by adding basis functions in other dimensions.

Chapter 4: Moving Window Filter for SOC Estimation Based on Electrochemical Models

4.1 Introduction

This section develops a new state filter algorithm based on the moving window to ensure the fast convergence of the SOC estimation. The filter gain of the moving window filter (MWF) is derived based on the maximum likelihood theory. The derivation of the filter gain assumes: (1) the measurement noise and process noise are Gaussian distributed with zero mean; (2) the system equations are continuously differentiable. The developed MWF is validated based on the federal driving schedule data from five cells and it shows improved performance compared with unscented Kalman filters.

4.2 State Estimation Theory and Kalman Filters

Information about system state is necessary to solve many control theory problems. For example, the power control of batteries will need the state of charge (SOC) information. In many practices, state can not be directly measured. For example, SOC of the battery cannot be directly observed. In control theory, state observer is an algorithm to estimate the internal state of a real system based on the input and output of the system. State filter is an improved approach of state observer as it takes the measurement noise and process noise into account. The measurement noise is normally caused by sensor errors, and process noise is usually due to modeling errors, e.g. unit to unit variations and un-

modeled physics. The measurement noise and process noise are often modeled as Gaussian noise with zero mean and a certain standard deviation. The commonly used state filter algorithms include: Kalman filter [30], extended Kalman filter [20, 30-34], and unscented Kalman filter (UKF) [38-41]. Figure 16 shows the flow chart of a standard Kalman filter. u_k and y_k are the input and output to the system respectively at time k . At each time step k , the input u_k feeds into the state space model with an initial guess of the state \hat{x}_k to provide an estimation of the output y_k , and then the state filter tries to find an updated state value that minimizes the difference between the model predicted output and actual output. The initial state guess \hat{x}_k is corrected by $x_k = \hat{x}_k + K(y_k - \hat{y}_k)$, where the K is called filter gain and used to penalize the state error in order to correct the state in the right directions.

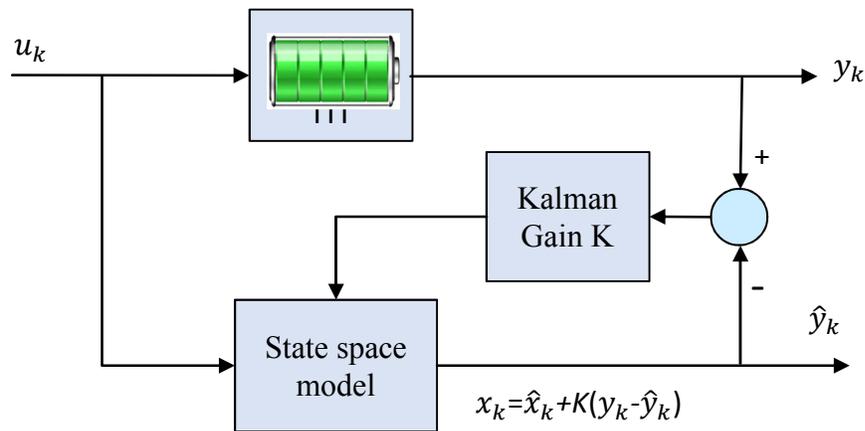


Figure 16: Flow chart of Kalman filter

Kalman filter is used for linear systems and it is the optimal filter for Gaussian measurement and Gaussian process noise. For nonlinear systems, extended Kalman filter and unscented Kalman filter are commonly used. State of charge estimation is a nonlinear problem, because the OCV of the battery is nonlinear. Unscented Kalman

filter has been demonstrated in the literature that it can provide better results compared with extended Kalman filter. However, problems still exist for state of charge estimation using unscented Kalman filter. When the initial state of charge guess presents high error, e.g. $> 40\%$ SOC error, the filter convergence could be slow. Here is an example shown in Figure 17. The unscented Kalman filter is used with electrochemical model for SOC estimation of a LiFePO₄ battery. The initial SOC error is 40%. Black line is the actual SOC value, while red line is the UKF estimation. The UKF can gradually converge to true value. However, it takes around 30 minutes for the UKF to converge. The convergence is too slow and can cause inconvenience usage experience. For example, if you are planning for a trip with electric car, you need to wait 30 minutes to get accurate range prediction, which is undesirable. To overcome SOC estimation error, a common practice is to over-design the battery to have buffer to ensure users have enough juice to accomplish the mission. Therefore, a state estimation algorithm that converges faster can not only improve the customer experience but also can reduce the over-design of the battery to save the cost.

In this thesis, we developed a new state filter algorithm which converges faster than Kalman filters and applied the developed approach for SOC estimations. We compared the developed approach with UKF, to quantify the improvements. In next section, we will present a brief introduction of UKF, which we will use it as the benchmark for performance comparison, and after that, we will discuss the theoretical background and derivations of the new moving window filter (MWF); In final section, we will compare the developed new filter with UKF with realistic test cases.

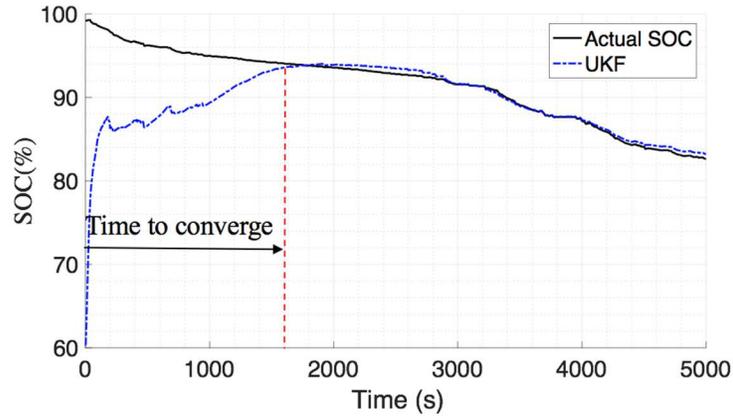


Figure 17: Unscented Kalman filter-based SOC estimation with 40% initial SOC error

SOC estimation is a nonlinear problem. The nonlinearity can be seen in the measurement model, where $OCV(SOC[k])$ is highly nonlinear, as shown in Figure 18. For the nonlinear state estimation problem, the extended Kalman filter (EKF) is a standard approach. The problem with EKF is that it only uses first-order or second-order terms of the Taylor series expansion to approximate the nonlinear functions. Large errors are produced if the model is highly nonlinear. In this study, we utilized an unscented Kalman filter (UKF), which is accurate to the third order, in the sense of a Taylor series expansion for any nonlinearity [54]. UKF is a direct application of the unscented transform (UT), which is a statistical tool. In UT, a Gaussian distribution is represented by a set of carefully chosen sample points called sigma points. These sigma points capture the mean and covariance of the Gaussian random variables (GRVs) when propagated through a nonlinear function. UKF has been applied to system estimation [55, 56], anomaly detection [57, 58], and prognostics [59, 60].

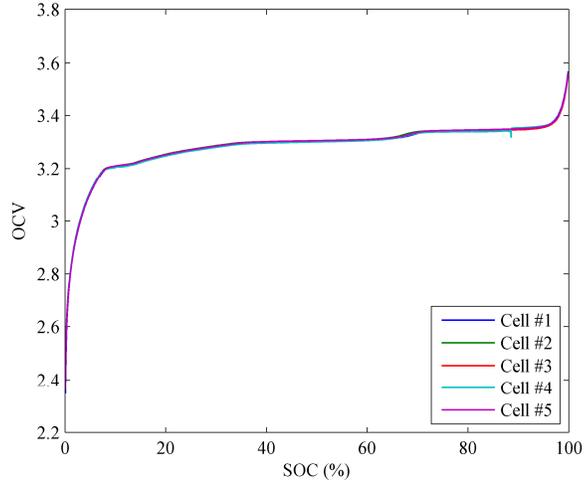


Figure 18: OCV-SOC curve of 5 LiFePO4 cells

Assume that a Gaussian random variable \mathbf{x} (dimension L) has mean $\bar{\mathbf{x}}$ and covariance \mathbf{P}_x . Consider propagating \mathbf{x} through the nonlinear function $\mathbf{y} = g(\mathbf{x})$. To calculate the statistics of \mathbf{y} , we first find a matrix \mathcal{X} of $2L+1$ sigma vectors \mathcal{X}_i with corresponding weights W_i , according to the following equations [54]:

$$\left\{ \begin{array}{ll} \mathcal{X}_0 = \bar{\mathbf{x}} & i = 0 \\ \mathcal{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(L+\lambda) \mathbf{P}_x} \right)_i & i = 1, L, L+1, \dots, 2L \\ \mathcal{X}_i = \bar{\mathbf{x}} - \left(\sqrt{(L+\lambda) \mathbf{P}_x} \right)_{i-L} & i = L+1, L+2, \dots, 2L \\ W_0^{(m)} = \lambda / (L+\lambda) & i = 0 \\ W_0^{(c)} = \lambda / (L+\lambda) + 1 - \alpha^2 + \beta \\ W_i^{(m)} = W_i^{(c)} = 1 / \{2(L+\lambda)\} & i = 1, L, L+1, \dots, 2L \end{array} \right. \quad (43)$$

where $\lambda = \alpha^2(L + \kappa) - L$ is a scaling parameter, α determines the spread of the sigma points around $\bar{\mathbf{x}}$, κ is another scaling parameter which is set to $3 - L$, and β is used to incorporate prior knowledge of the distribution of \mathbf{x} . For Gaussian distributions, $\beta = 2$

is optimal [54]. $\left(\sqrt{(L+\lambda)\mathbf{P}_x}\right)_i$ is the i -th column of the matrix square root of $(L+\lambda)\mathbf{P}_x$.

Then, each sigma point is propagated through the nonlinear function $y_i = g(\mathcal{X}_i)$, $i=0, \dots, 2L$. The estimated mean and covariance of y are computed by the weighted sample mean and covariance as follows:

$$\bar{\mathbf{y}} = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{y}_i \quad (44)$$

$$\mathbf{P}_y = \sum_{i=0}^{2L} W_i^{(c)} (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T \quad (45)$$

The UKF is a straightforward application of the UT for state estimation. The main steps of UKF are summarized below:

(1) Initialize with $\hat{\mathbf{X}}_0 = E[\mathbf{X}_0]$ and $\mathbf{P}_0 = E[(\mathbf{X}_0 - \hat{\mathbf{X}}_0)(\mathbf{X}_0 - \hat{\mathbf{X}}_0)^T]$

(2) For $k \in \{1, \dots, \infty\}$, calculate sigma points:

$$\boldsymbol{\chi}_{k-1} = \begin{bmatrix} \hat{\mathbf{X}}_{k-1} & \hat{\mathbf{X}}_{k-1} + \gamma\sqrt{\mathbf{P}_{k-1}} & \hat{\mathbf{X}}_{k-1} - \gamma\sqrt{\mathbf{P}_{k-1}} \end{bmatrix} \quad (46)$$

(3) State prediction:

Propagate the sigma points through the state model:

$$\boldsymbol{\chi}_{k|k-1} = \mathbf{H}[\boldsymbol{\chi}_{k|k-1}] \quad (47)$$

Calculate the propagated mean:

$$\bar{\mathbf{X}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\chi}_{i,k|k-1} \quad (48)$$

Calculate the propagated covariance:

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\boldsymbol{\chi}_{i,k|k-1} - \hat{\mathbf{X}}_k^-][\boldsymbol{\chi}_{i,k|k-1} - \hat{\mathbf{X}}_k^-]^T \quad (49)$$

(4) Measurement update:

Propagate sigma points through the measurement function:

$$\mathbf{y}_{k|k-1} = \mathbf{H}[\boldsymbol{\chi}_{k|k-1}] \quad (50)$$

Calculate the propagated mean:

$$\bar{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{y}_{i,k|k-1} \quad (51)$$

Calculate the estimated covariance

$$\begin{aligned} \mathbf{P}_{\bar{\mathbf{y}}_k^- \bar{\mathbf{y}}_k^-} &= \sum_{i=0}^{2L} W_i^{(c)} [y_{i,k|k-1} - \bar{y}_k^-][y_{i,k|k-1} - \bar{y}_k^-]^T \\ \mathbf{P}_{x_k y_k} &= \sum_{i=0}^{2L} W_i^{(c)} [\boldsymbol{\chi}_{i,k|k-1} - \bar{\mathbf{x}}_k^-][y_{i,k|k-1} - \bar{y}_k^-]^T \end{aligned} \quad (52)$$

Calculate the Kalman gain \mathbf{K} and update the state estimation and covariance:

$$\begin{aligned} \mathbf{K} &= \mathbf{P}_{x_k y_k} \mathbf{P}_{\bar{\mathbf{y}}_k^- \bar{\mathbf{y}}_k^-}^{-1} \\ \bar{\mathbf{X}}_k &= \bar{\mathbf{X}}_k^- + \mathbf{K}(\mathbf{y}_k - \bar{\mathbf{y}}_k^-) \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K} \mathbf{P}_{\bar{\mathbf{y}}_k^- \bar{\mathbf{y}}_k^-} \mathbf{K}^T \end{aligned} \quad (53)$$

4.3 Moving Window Filter

Conventional state filtering methods are sequential estimation algorithms, which utilize the measurements at a single time point to provide the state estimation. Figure 19 shows the a schematic plot for conventional sequential state filters, e.g. Kalman filter. This approach is computational efficient. However, as we discussed before, it can converge slow and in addition, it may not robust to outliers in the measurement. Therefore, in this thesis, we are trying to develop an approach to improve the state filters. One method is to use more data points in the estimation. For example, as shown in Figure 20, we can use a moving window of N data points (from $T-N+1$ to T) to provide state estimation at $T-N+1$. The estimation accuracy and robustness should be improved as more data points are involved in the estimations. The key question is how to calculate filter gain in this method.

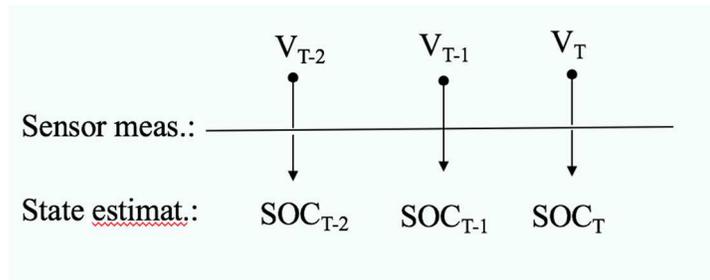


Figure 19: Conventional sequential estimation scheme

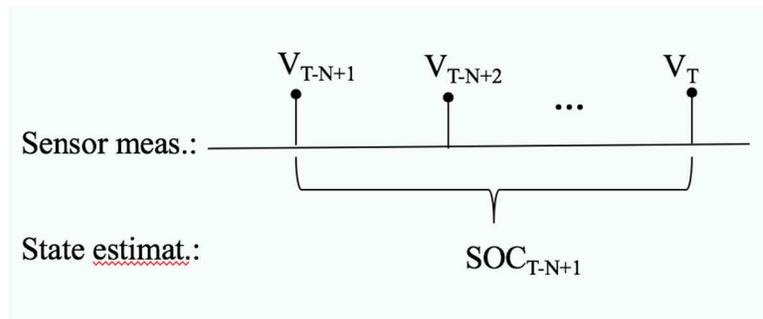


Figure 20: Moving window state filter

Let's consider a maximum likelihood estimation problem:

$$\underset{x_{T-N+1}}{\operatorname{argmax}} P(x_{T-N+1} | y_0, y_1, \dots, y_T) \quad (54)$$

Based on Bayes' theorem [61, 62], we have the following:

$$\begin{aligned} & \underset{x_{T-N+1}}{\operatorname{argmax}} \frac{P(y_0, y_1, \dots, y_T | x_{T-N+1}) P(x_{T-N+1})}{P(y_0, y_1, \dots, y_T)} \\ &= \underset{x_{T-N+1}}{\operatorname{argmax}} \frac{P(y_0, y_1, \dots, y_{T-N} | x_{T-N+1}) \prod_{k=T-N+1}^T P(y_k | x_k) P(x_{T-N+1})}{P(y_0, y_1, \dots, y_T)} \\ &= \underset{x_{T-N+1}}{\operatorname{argmax}} \frac{P(x_{T-N+1} | y_0, y_1, \dots, y_{T-N}) P(y_0, y_1, \dots, y_{T-N}) \prod_{k=T-N+1}^T P(y_k | x_k) P(x_{T-N+1})}{P(y_0, y_1, \dots, y_T) P(x_{T-N+1})} \\ &= \underset{x_{T-N+1}}{\operatorname{argmax}} P(x_{T-N+1} | y_0, y_1, \dots, y_{T-N}) \prod_{k=T-N+1}^T P(y_k | x_k) \end{aligned} \quad (55)$$

Eq. (55) is equivalent to a quadratic estimation problem

$$\begin{aligned} \underset{x_{T-N+1}}{\operatorname{argmin}} \Phi(x_{T-N+1}) &= (\mathbf{y} - \mathbf{H}(\hat{x}_{T-N+1}))^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{H}(\hat{x}_{T-N+1})) \\ &+ (x_{T-N+1} - \hat{x}_{T-N+1})^T \prod_{T-N+1}^{-1} (x_{T-N+1} - \hat{x}_{T-N+1}) \end{aligned} \quad (56)$$

Base on first order optimality:

$$\frac{\partial \Phi(x_{T-N+1})}{\partial x_{T-N+1}} = 0 \quad (57)$$

We have:

$$x_{T-N+1} = \hat{x}_{T-N+1} + \prod_{T-N+1} \mathbf{C}^T (\mathbf{R}^T + \mathbf{C}^T \prod_{T-N+1} \mathbf{C})^{-1} (\mathbf{y} - \mathbf{H}(\hat{x}_{T-N+1})) \quad (58)$$

where

$$\mathbf{y} = [y_{T-N+1}, y_1, \dots, y_T]^T \quad (59)$$

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} h(\mathbf{x}) \\ h(f(\mathbf{x})) \\ h(f(f(\mathbf{x}))) \\ \dots \\ h(f(f(\dots f(\mathbf{x})))) \end{bmatrix}_{N \times 1} \quad (60)$$

$$\mathbf{C} = \frac{\partial \mathbf{H}(x_{T-N+1})}{\partial x_{T-N+1}} \quad (61)$$

$$\hat{x}_{T-N+1} : \text{mean of } P(x_{T-N+1} | y_0, y_1, \dots, y_{T-N}) \quad (62)$$

$$\mathbf{\Pi}_{T-N+1} : \text{cov. of } P(x_{T-N+1} | y_0, y_1, \dots, y_{T-N}) \quad (63)$$

In Eq. (58), $\mathbf{\Pi}_{T-N+1} \mathbf{C}^T (\mathbf{R}^T + \mathbf{C}^T \mathbf{\Pi}_{T-N+1} \mathbf{C})^{-1}$ is the gain of the filter. The gain has the similar structure as Kalman filter. However, the definition of the matrices in the gain is different from Kalman filter. Figure 21 shows the implementation of the moving window filter (MWF) for SOC estimation. At each time step T , N data points will be used to update the states.

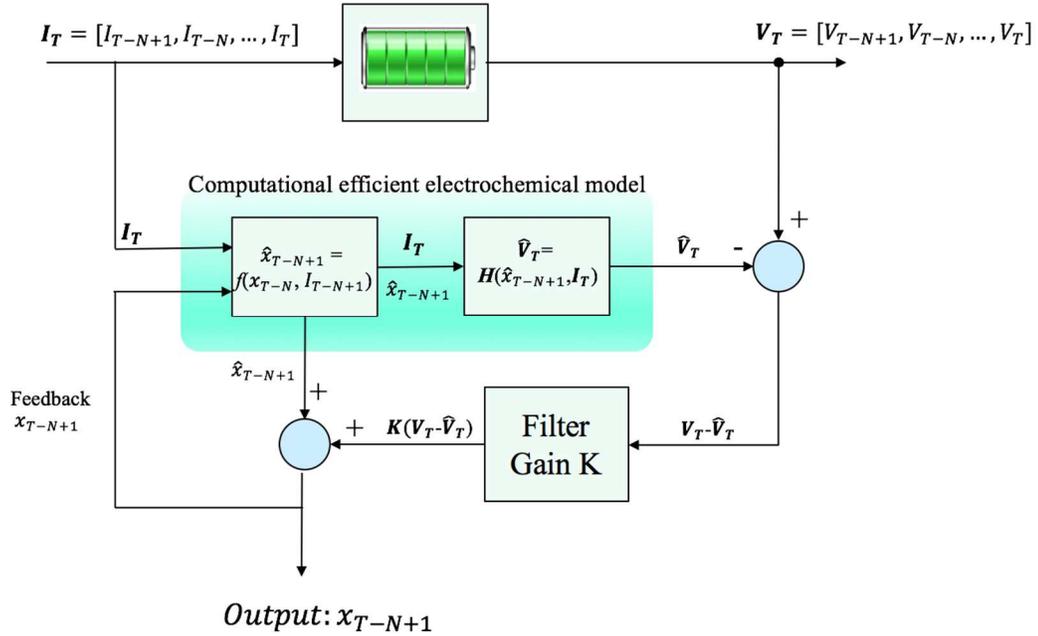


Figure 21: Moving window filter implementation

4.4 Experimental Validation

To validate the proposed approach, experiments were conducted to collect battery data. Figure 22 shows the experimental setup. 5 LiFePO₄ batteries were tested in this research. Arbin BT 2000 was used to control the battery charging discharge, and the voltage, current, and temperature data of the battery were collected. Figure 23 shows the current profile used in the test, which is converted from the speed profile of federal driving schedule (FDS) based on an EV model. This is in order to simulate what the battery can experience in the actual driving condition.

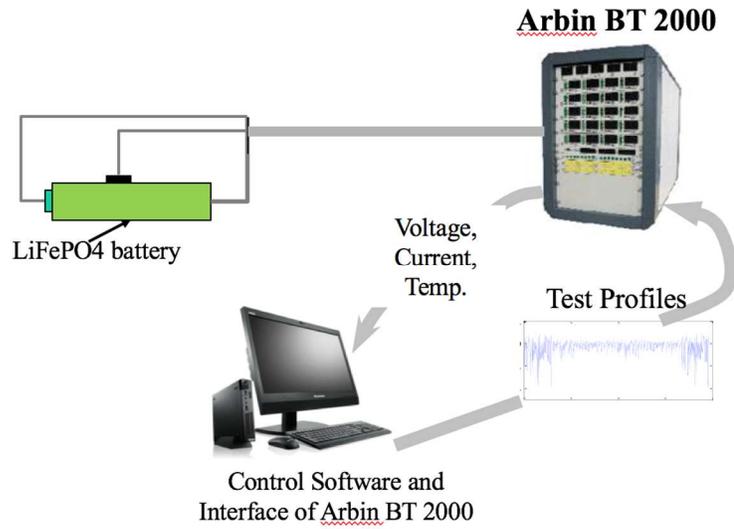


Figure 22: Experimental Setup

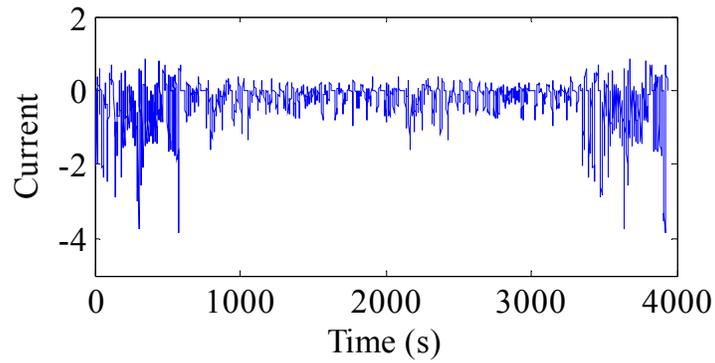


Figure 23: Federal driving schedule (FDS) current profile

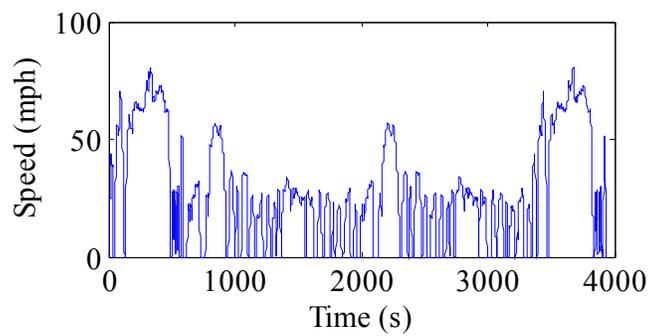


Figure 24: Federal driving schedule (FDS) velocity profile

The collected data is used to test the performance moving window filter with the electrochemical model. The electrochemical model parameters were obtained from [63]. First, we examined the effects of the window size on the SOC accuracy. Figure 25 shows the results. We can see that the SOC estimation error reduces with the increasing of window size, especially when the window size increases from 2 to 4. After 4, there is not much improvement. Hence, in next paragraphs, we will use a window size of 4 for the state estimation and compare the results with unscented Kalman filter (UKF).

Figure 26 shows the comparison results of Cell #1. Black line is actual SOC, which obtained by Coulomb counting as we know the initial condition and the sensor is accurate in the lab. The red line is the MWF, which converges to true value in less than 2 mins. The UKF result is in blue line. It takes about 30mins for the UKF to converge. Therefore, MWF outperforms UKF. Figure 27 to Figure 30 show the comparison results for Cell #2 to Cell #5. MWF shows consistent improvements over UKF.

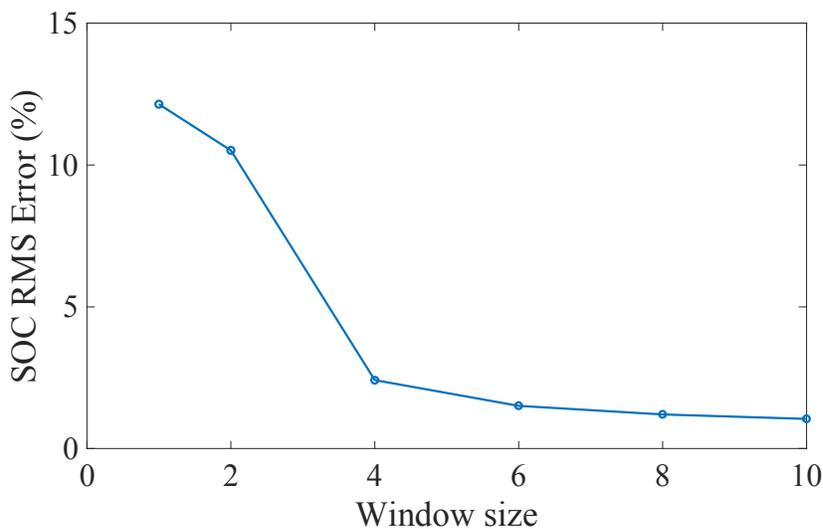


Figure 25: Effect of window size on the filter accuracy

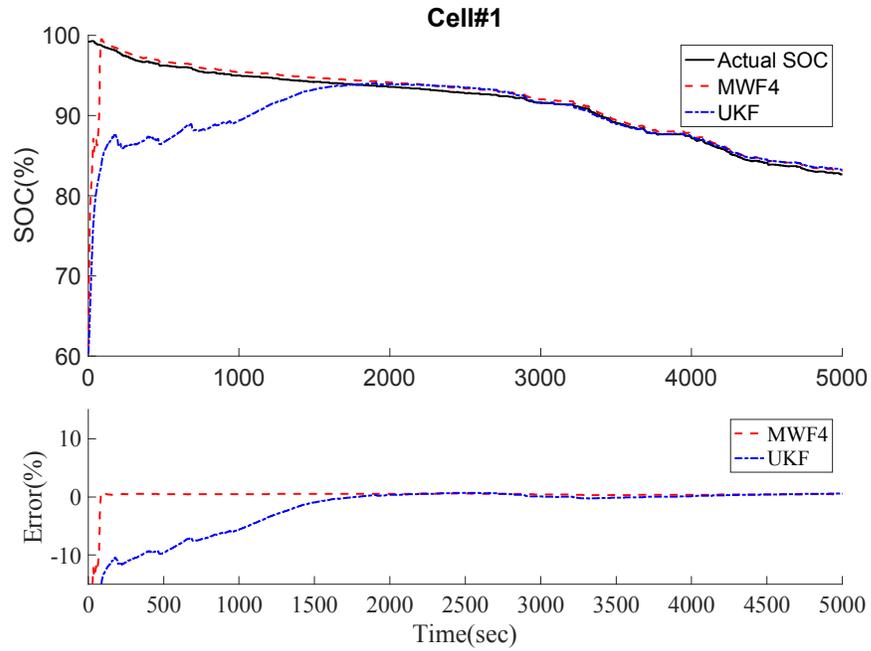


Figure 26: SOC estimation comparison of cell 1

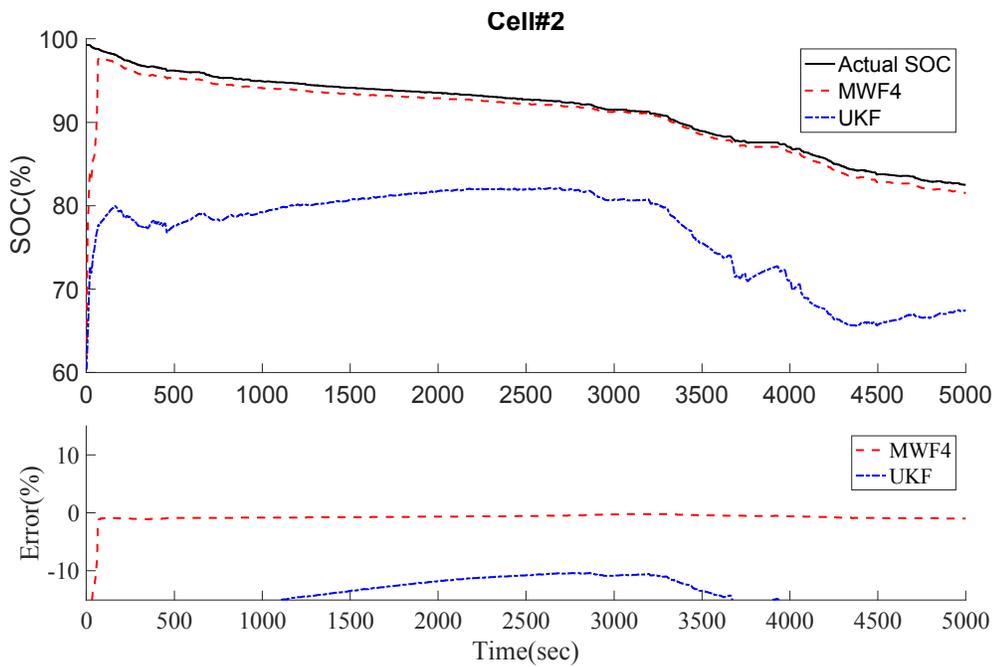


Figure 27: SOC estimation of cell 2

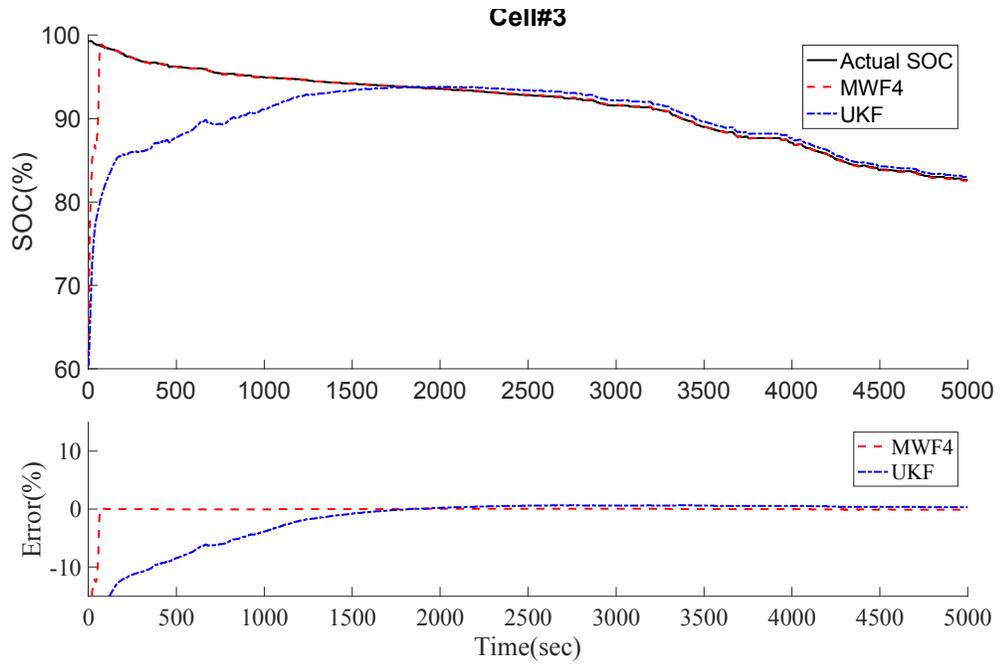


Figure 28: SOC estimation of cell 3

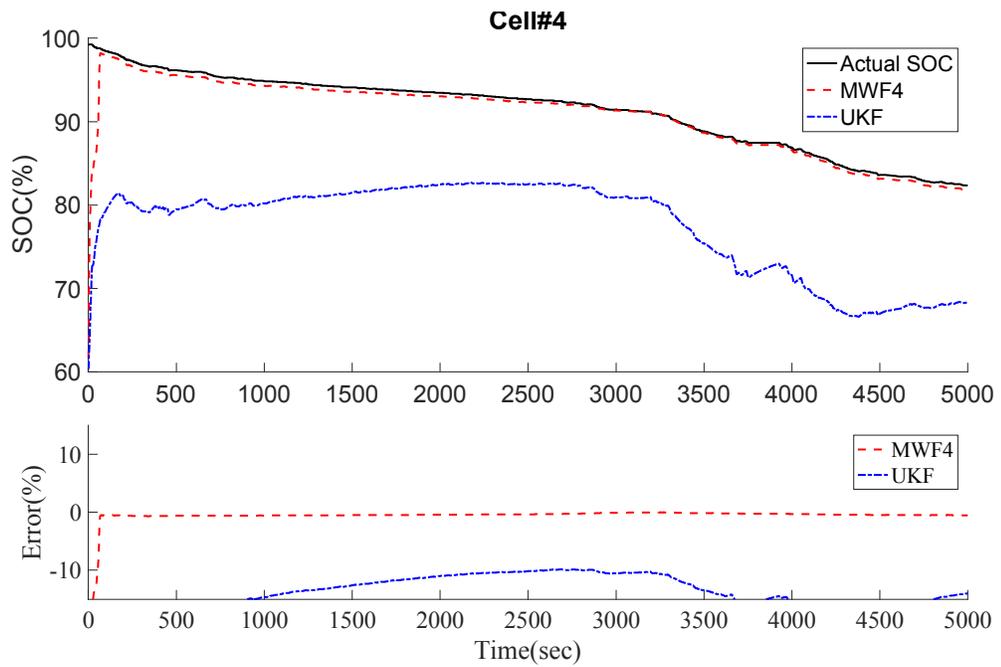


Figure 29: SOC estimation of cell 4

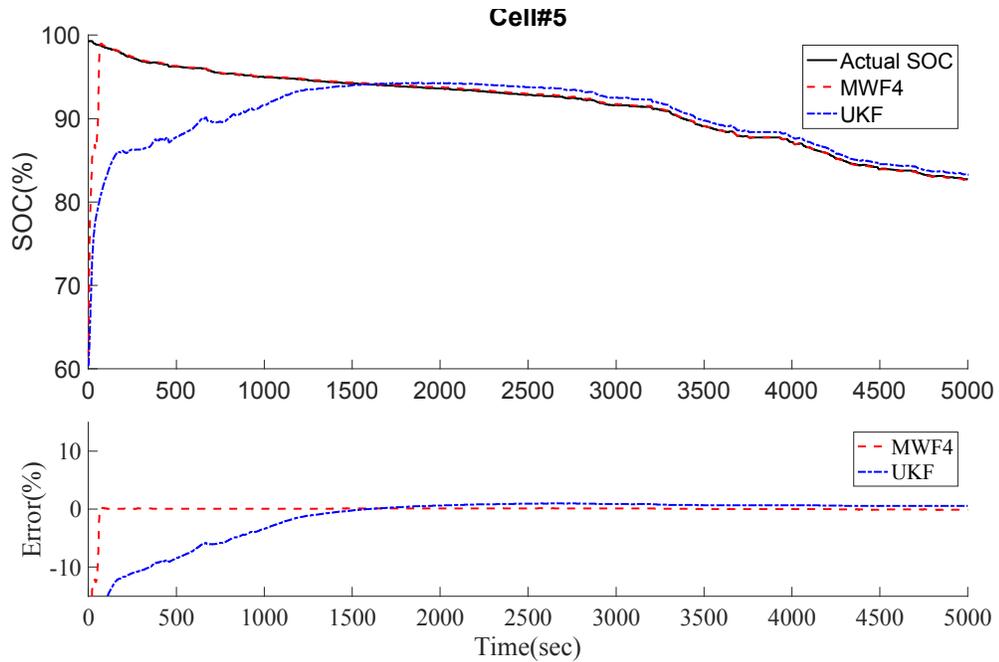


Figure 30: SOC estimation of cell 5

Table 2 summarizes the results of all 5 cells. For convergence analysis, only the first a few minutes are important as we want to see which filter converge faster. Therefore, in Table 2, only the first 10mins error were estimated. We can see the MWF's improvements over UKF is significant as the SOC error decreases from >10% to 3%.

The computational efficient electrochemical model is used here with state filters for SOC estimation. The MWF not only converges fast, but also demonstrates good capability to handle unit to unit variations, because here the model parameters are learned from Cell #1 and then the same parameter set is applied to different cells.

**Table 2: Summary of SOC estimation results based on electrochemical model
and MWF**

Cell ID	Root Mean Square Error (first 10mins)	
	UKF	MWF
Cell 1	12.1%	2.4%
Cell 2	19.7%	2.5%
Cell 3	12.4%	1.5%
Cell 4	18.1%	2.1%
Cell 5	12.1%	1.6%

Chapter 5: Improved Data-driven Approach for SOC Estimation

5.1 Introduction

Data-driven approaches do not require the understanding of battery physics, and are easier to be implemented compared with physics-based approaches, especially for new battery chemistries, where the battery physics are not fully understood and material properties are not available.

This section investigates a data-driven approach for battery SOC estimation based on neural networks. The neural networks are trained by using voltage and current as the inputs, and SOC as the output. There are fairly few assumptions behind the neural network application, as it does not require the data to follow a certain distribution or structure. The only assumption of the application is that there exist function mappings between voltage, current and SOC, which should be always true based on battery physics.

Data-driven approaches are highly dependent on the training data. If the model is trained based on one cell or one loading condition, when it is applied to other cells and other loading condition, it may generate larger errors. That is the generalization problem of the data-driven approach. Therefore, in this section, we also develop a method to improve the generalization capability of the data-driven models by using moving window filters.

5.2 Theory of Neural Network

Neural networks (NNs) are computational intelligence tools that have been widely used for system modeling [54, 61, 62, 64-71], anomaly detection [72, 73], prognostics [74, 75], and classification [76, 77]. An NN comprises a set of interconnected simple processing elements called neurons that mimic the information processing and knowledge acquisition capabilities of the human brain. There are several characteristics of NNs that make them an attractive choice for system modeling: firstly, NNs can fit any nonlinear function with sufficient neurons and layers to make them suitable for complex system modeling; secondly, NNs can learn and update their internal structure to adapt to a changing environment; in addition, NNs are efficient in data processing because of their parallelism in computation; moreover, NNs are data-driven in nature and able to build an accurate system model without detailed physical knowledge of a system [78].

Many types of NNs have been proposed, including Hopfield networks, feed-forward (FF) networks, recurrent networks, and radial basis function (RBF) networks. The feed-forward network is the most widely used type of NN. It consists of an input layer with nodes to represent the input variables, one or more hidden layers with nodes to mimic the nonlinearity between the system input and output, and an output layer to represent the system output variable. Figure 31 shows the structure of an FF network for SOC estimation. The inputs to the neural network are current (I), voltage (V), and temperature (T), and the output is the battery SOC. The nodes between two adjacent

layers are interconnected. The input layer passes on the inputs with weights; no processing takes place in this layer. The hidden layers and output layers are processing layers with the activation function at each node. The hyperbolic tangent sigmoid function is often used in the hidden layer as an activation function. It is defined as follows:

$$\mathbf{F}_{tansig}(u) = \frac{2}{1 + e^{-2u}} - 1 \quad (64)$$

In the output layer, the linear transfer function is used as an activation function for regression and fitting problems, as shown below:

$$\mathbf{F}_{lin}(u) = u \quad (65)$$

The output of a processing node j in the hidden or output layer is given by:

$$y_j = \mathbf{F}(u_j) = \mathbf{F}\left(\sum_i \omega_{ij} x_i + b_j\right) \quad (66)$$

where x_i is the output from the i -th node at the previous layer, ω_{ij} is the weight of the interconnection from the i -th node of the previous layer to j -th node of the present layer, and b_j is the bias. The net weights ω_{ij} and biases b_j need to be determined based on training data.

The back propagation algorithm is used in this study to determine these parameters. Back propagation means that an error during network training can propagate from the output layer to a hidden layer and then to the input layer to estimate the optimal neural weights of each node. To learn more about the BP algorithm for neural network weight estimation, see [79].

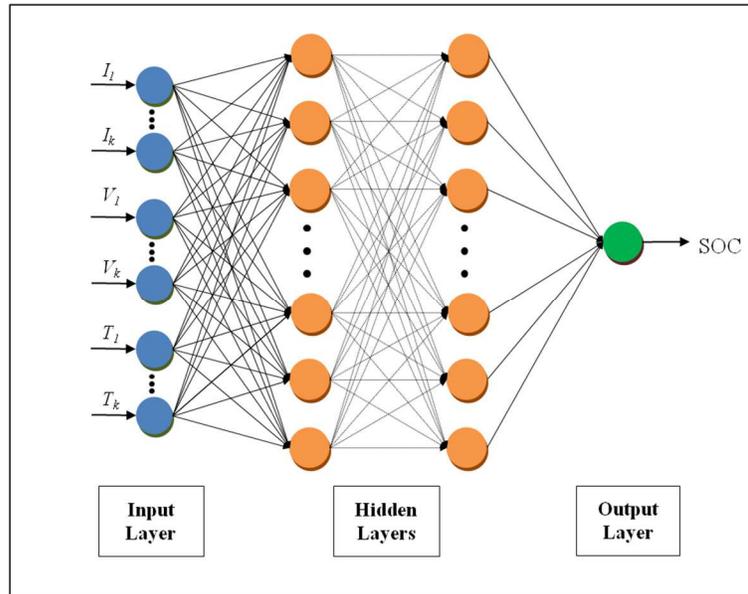


Figure 31: The structure of a multilayer feed-forward neural network

5.3 Training and Testing Data

A practical concern for training the NN is that the real-life loading conditions of batteries are complex and uncertain. For example, for EVs, the loading condition may vary with road conditions, speeds, and driving styles. As a result, the training data should cover the real-life loading conditions as much as possible in terms of SOC span, current and voltage range, and loading changing rate. For example, the training database can be constructed from the battery tests using simulated driving cycles. Also, the data collected during the field application of EVs can be used to improve the performance of the NN. Furthermore, the NN should have a generalization capability in case a loading condition is not included in the database.

In this paper, the training data are collected using a dynamical stress testing (DST) profile, as specified in the US Advanced Battery Consortium (USABC) testing

procedures [80]. The current profile of the DST is shown in Figure 32. Although DST consists of a variety of current steps with different amplitudes and lengths and takes into account regenerative charging (as shown by the negative amplitudes in Figure 32), it is still much simpler than the real-life loading conditions of batteries. DST will be used in this paper as training data to examine the NN's accuracy and generalization capability for complex real-life loading conditions for SOC estimation. Figure 34 shows the test setup of the experiments. 5 LiFePO₄ batteries which are typically used for EVs, were tested. The maximum capacity of the battery is 2.3 Ah. The battery was put in the temperature chamber, and the temperature of the cell was measured. The Arbin BT2000 was used to control the charging and discharging of the battery.

The testing data of the NN should be different from the training data. In this study, the testing data were collected using Federal Driving Schedule (FDS) [81], which is shown in Figure 33. FDS are much more complex than the DST in terms of the changing rate of the current. This profile was used to test the robustness and generalization of the NN.

The NN training can be made more efficient and robust through proper normalization of the data. Therefore, before training, the inputs are normalized to the range [-1, 1] by the following equation:

$$x = \frac{2(x - x_{\min})}{x_{\max} - x_{\min}} - 1 \quad (67)$$

where x_{\min} and x_{\max} are the minimum and maximum in the input vector x of the NN. In the testing step, the testing data should also be scaled using the same x_{\min} and x_{\max} used in the training data.

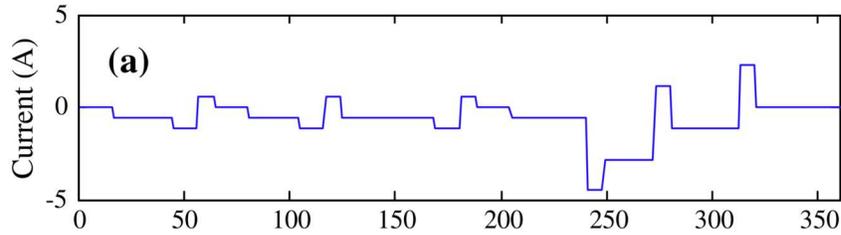


Figure 32: Dynamical Stress Testing (DST) profile, which will be used as training data

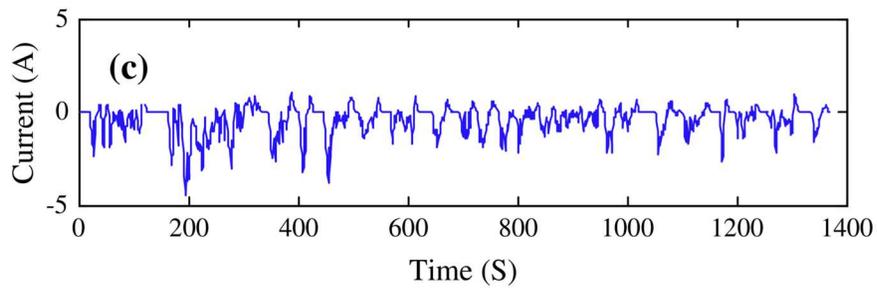


Figure 33: Federal Driving Schedule (FDS), which will be used as testing data

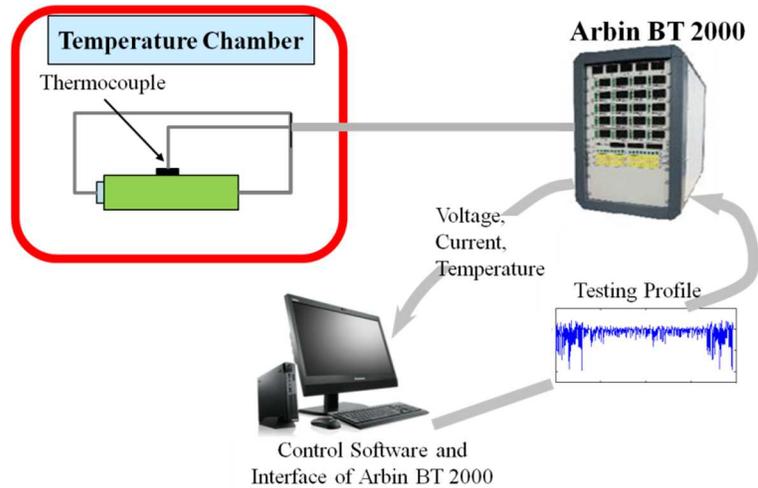


Figure 34: Experimental setup for battery tests

5.4 Determination of the Neural Network Structure

Several equivalent circuit models (ECMs) have been developed for SOC estimation. Based on an ECM, the OCV of a battery can be estimated using the measured voltage and current. Then, the SOC can be inferred using a predetermined OCV–SOC lookup table. Similarly, an NN-based model tries to determine a relationship between the SOC and the measured voltage and current. The difference between NN and ECM is that NN will infer the SOC directly from the voltage and current measurement without an OCV–SOC lookup table. Thus, this approach eliminates the time-consuming tests necessary to establish an OCV–SOC lookup table. The adaptation of the NN-battery model to different chemistries is convenient and efficient.

In this study, the inputs of the NN were the current and voltage measurements. Because of the capacitive resistance in the battery, the current and voltage in the previous samples will affect the present battery state. Therefore, the measurements of previous samples were also input into the NN model. In addition, to avoid over-training of the

NN, 1 sample was selected out of every 4 samples to train the NN. The inputs of the NN at time i were $[I(i), I(i-4), \dots, I(i-4k), V(i), V(i-4), \dots, V(i-4k), T(i), T(i-4), \dots, T(i-4k)]$ and the output was $SOC(i)$, where k was a constant that depended on the time constant of the battery system response and was determined before training the NN.

Another practical problem with the application of NN is that the number of hidden layers and the number of neurons at each layer need to be determined. There are some “rules of thumb” for determining an optimal NN structure. First, there is no theoretical reason to use an NN with more than 2 hidden layers since it has been proven that a 2-layer NN can fit to a function with any kind of shape [32-34]. Second, the presence of too many neurons will present a risk of over-fitting, while too few neurons will under-fit the data. We adopted a constructive method to determine these parameters, as well as the parameter k of the NN inputs. The NN was constructed starting from the simplest version with low-dimension inputs to a complex structure with high-dimension inputs. If the training error is not acceptable, then additional nodes will be added and the dimension of the inputs will be increased. Once the training error is acceptable, then the optimization is terminated. The following algorithm provides detailed procedures to determine the NN structure and the parameter k :

Step 1: Initialization: $k = 1$, $n = 1$, where k is a parameter used to determine the dimension of the input vector, and n is the number of neurons in the hidden layers.

Step 2: Train the neural network with $[I(i), I(i-4), \dots, I(i-4k), V(i), V(i-4), \dots, V(i-4k), T(i), T(i-4), \dots, T(i-4k)]$ as input and SOC (i) as output.

Step 3: Calculate the root mean square (RMS) error between the NN estimated output and the actual SOC.

Step 4: If the RMS error $< 1\%$, then the searching is ended; else if $k < 2n$, $k = k+1$ and go to the step 2; else $n = n+1$, $k = n$ and go to Step 2.

Algorithm 1 A constructive approach for the structure determination of neural networks for SOC estimation

Using Algorithm 1, the parameters of the NN were determined to be $k = 30$ and $n = 5$.

5.5 Training and Testing Results

Figure 35 shows the NN training results for Cell #1 with 5 neurons in each layer based on the DST data. The solid curve is the SOC calculated by Coulomb counting. Since the battery was discharged from 100% SOC and the integration error was negligible because the current sensor was well calibrated, the solid curve is regarded as the actual SOC for the comparison. The neural network can fit the DST data pretty well, as the RMS error is only 0.3%. However, when applying the trained NN to FDS data of cell #1, the RMS error of SOC is increased to 3.3% and max error is higher than 10%. The errors mainly present in the middle SOC range (30~80%) due to the flat plateau of the LiFePO₄ batteries' discharge characteristics. The problem cannot be simply solved by adding more neurons to get better fit. As an illustration example, we trained the neural network with 15 neurons in each hidden layer based on the DST data, and then use it to estimate the SOC of Cell #1's FDS data again. Figure 38 shows the training results

and we can see the fitting is better with more neurons, and RMS error is reduced to 0.2%, however, the testing results with the FDS data actually gets worse and the error increases to 3.7%. Hence, if the neural network is trained to an RMS that is too low, it may lose its generalization ability. To prevent over-fitting, a simple structure neural network is preferred. To prevent under-fitting, we trained the neural network to a reasonable RMS error that is sufficient to capture the dynamics of the battery.

Figure 39 to Figure 42 present the NN SOC estimation results for Cell #2 to Cell #5. The results are worse than that of Cell #1, and there are lots of sudden jumps or drops in the estimation, that's because the model is trained from Cell #1, and there are unit to unit differences between cells. For applications, such as electric vehicles (EVs), this will confuse the user, as the residual range prediction of the EV may change drastically from time to time. The model estimation error is potentially due to the unit to unit variations, since our model is trained to Cell #1 only. We could mitigate this issue by training the model to each individual cell. However, this is not practical in many actual applications. For example, for companies that ship millions of product each year, it is infeasible to develop a model for each individual cells. Therefore, the generalization capability of the model is extremely important. Approaches to improve the generalization capability model to address the variations due to unit to unit differences and loading condition changes must be developed for the practical use of data-driven models, and this is the main research objective this chapter.

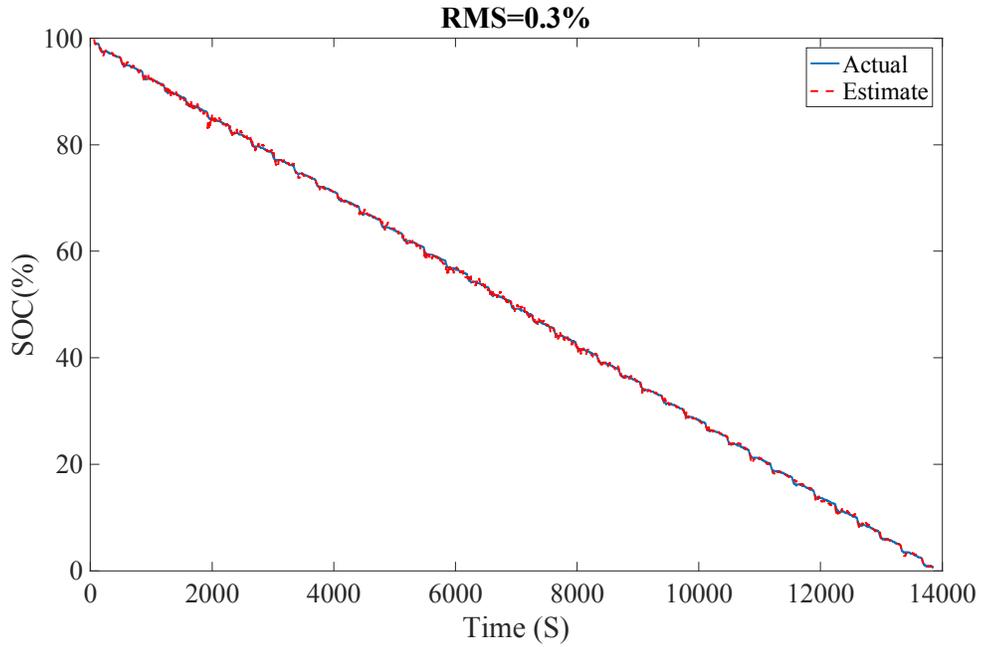


Figure 35: Neural network training results of cell #1 with 5 neurons for each layer based on DST data

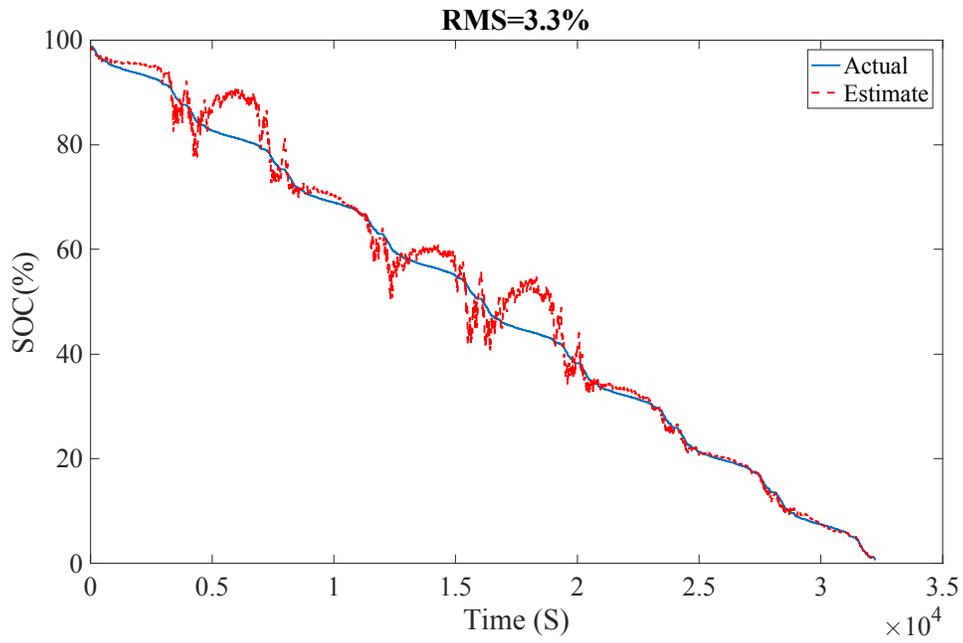


Figure 36: Testing results of neural network with 5 neurons in each layer for Cell #1 FDS data

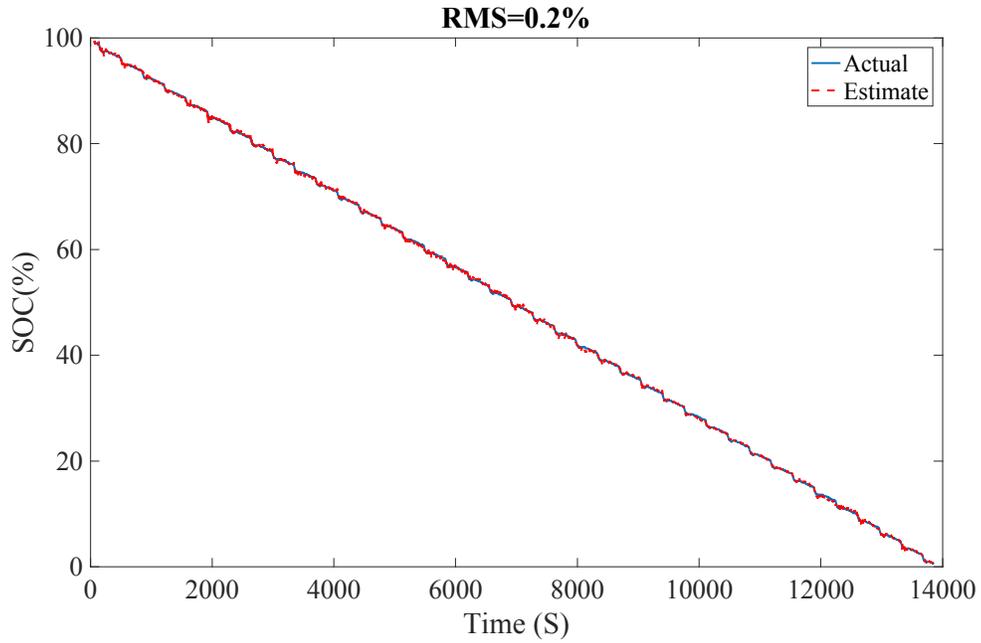


Figure 37: Training result of neural networks with 15 neurons in each player based on Cell #1 DST data

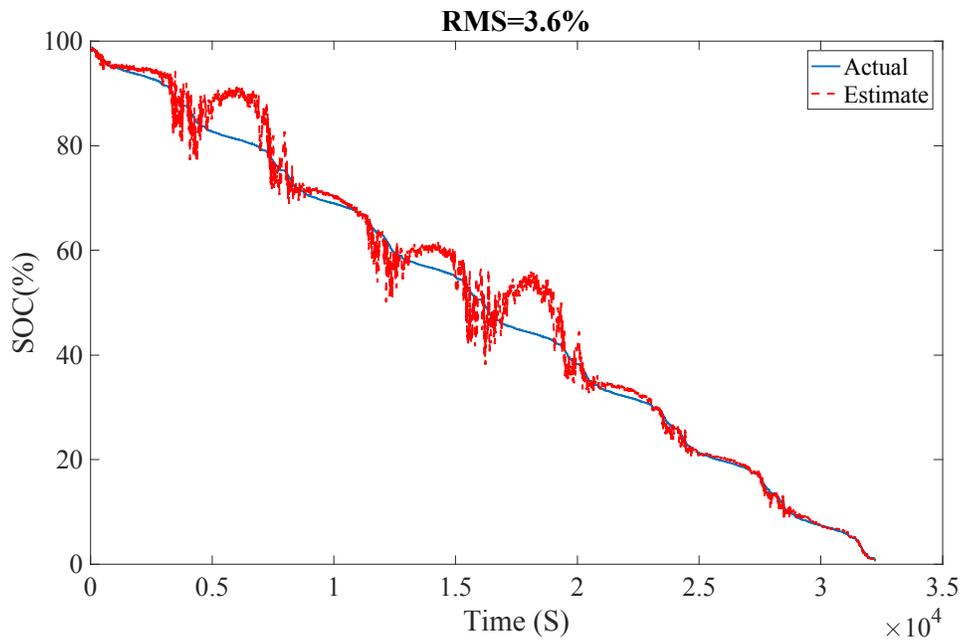


Figure 38: Testing results of neural network with 15 neurons in each layer for Cell #1 FDS data

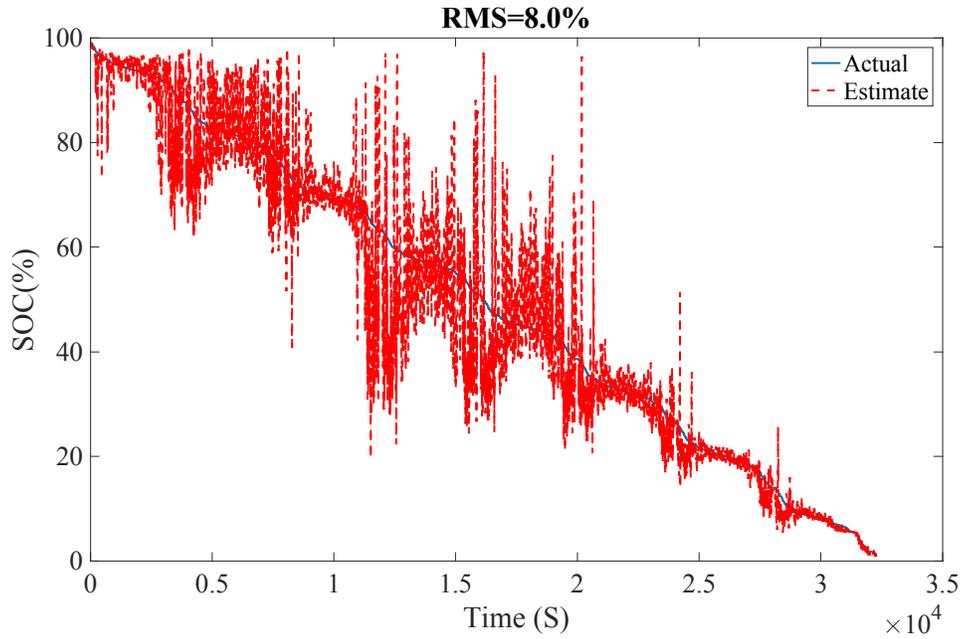


Figure 39: NN SOC estimation results for Cell#2 FDS data, where NN is trained based on Cell#1 DST data

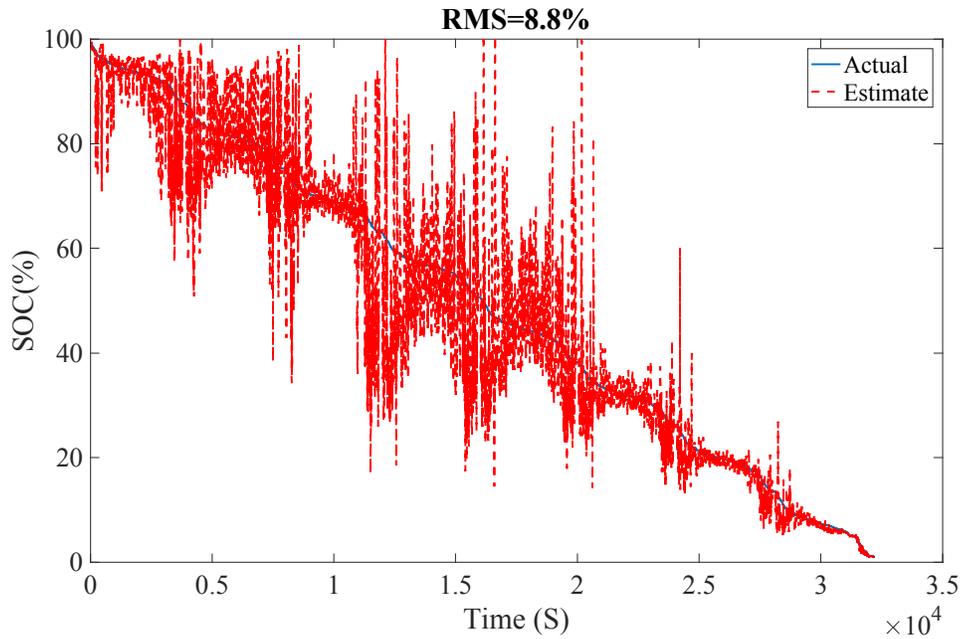


Figure 40: NN SOC estimation results for Cell#3 FDS data, where NN is trained based on Cell#1 DST data

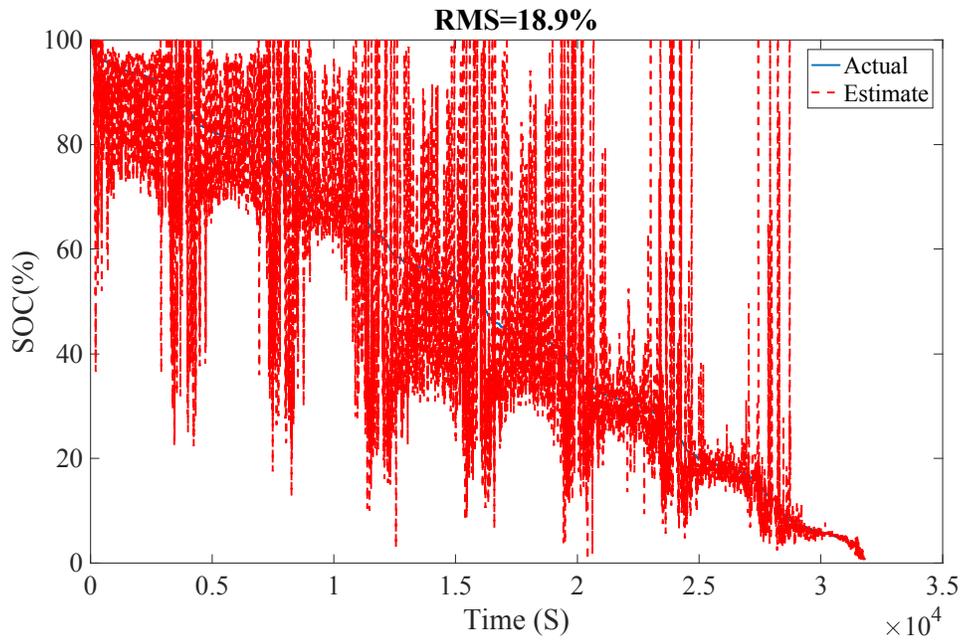


Figure 41: NN SOC estimation results for Cell#4 FDS data, where NN is trained based on Cell#1 DST data

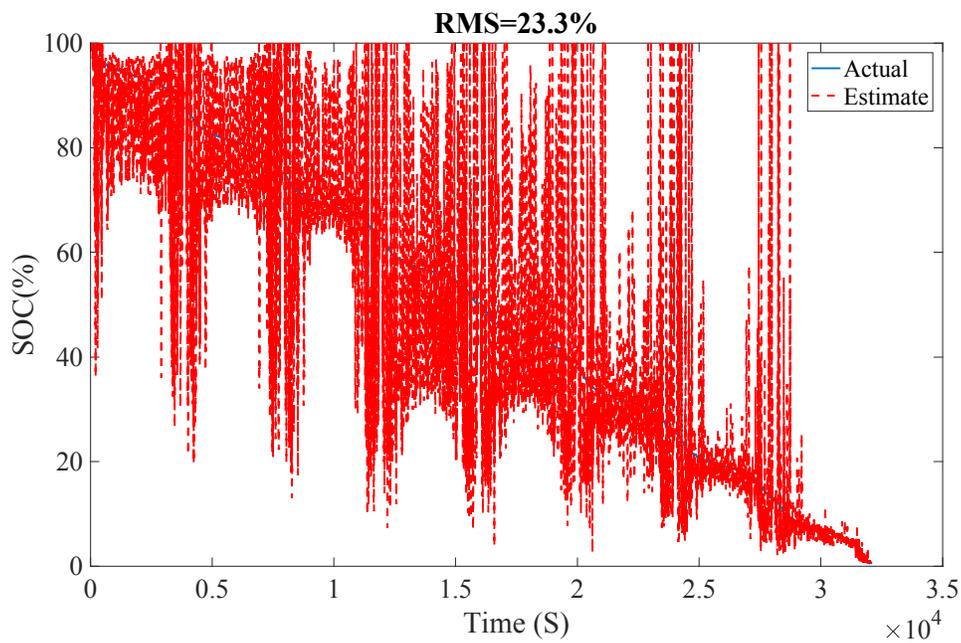


Figure 42: NN SOC estimation results for Cell#5 FDS data, where NN is trained based on Cell#1 DST data

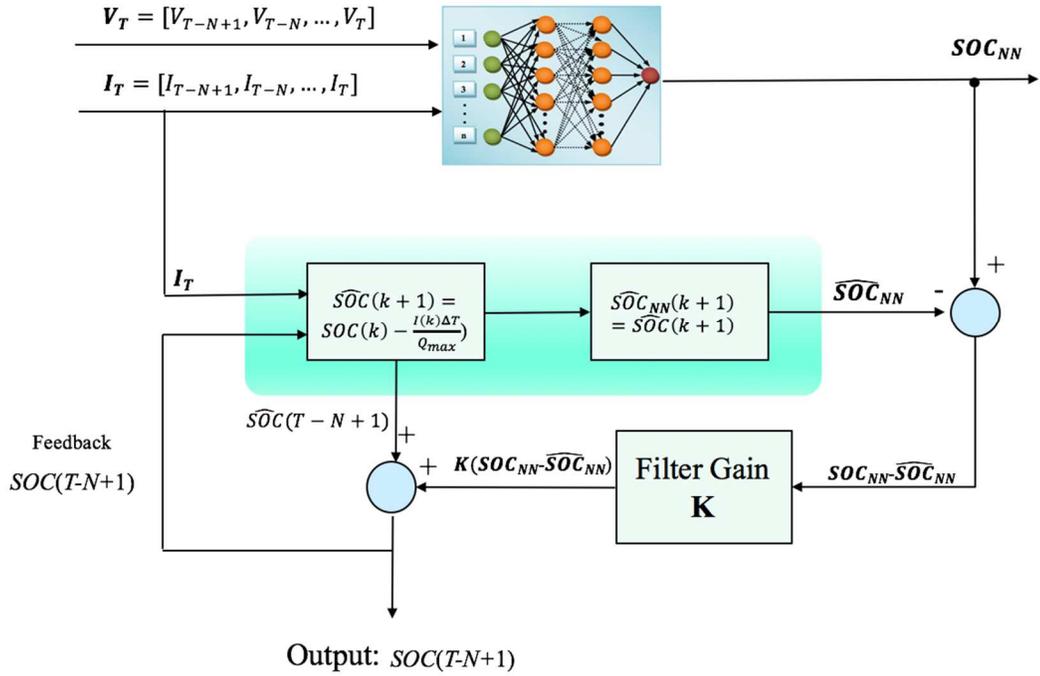


Figure 43: Flowchart of the combined approach of neural network and moving window filter

5.6 Developed Solution by Combing NN with State Filters

We have demonstrated in last chapter that state filter can effectively handle unit to unit variations, and it can also uncover the underlying states from noisy measurements. The neural network outputs have high variations and jump up and down around the true values. We could consider the neural network outputs as the noisy measurements, and our objective is to remove the noise and uncover the true SOC values. Based on experience from previous chapter, the state filter can be used to address this problem.

To use the state filter, we have to reformulate the neural network to fit in the structure of the state filtering frame work.

In state filter, we need a state-space model including state functions and measurement functions. We propose to use the physics of the SOC evolution, i.e. Coulomb counting rule as the state function, which can ensure the smooth transitions of the SOC and avoid the sudden jumps. Since we consider NN outputs as noisy measurements of SOC, the measurement function would be the neural network SOC output equal to state function output plus noise as shown in (69).

$$\textit{state function: } SOC(k + 1) = SOC(k) - \frac{I(k)\Delta T}{Q_{max}} + \omega \quad (68)$$

$$\textit{Measurement function: } NN(k + 1) = SOC(k + 1) + \varepsilon \quad (69)$$

Based on the state-space model, we can use the moving window filter (MWF) to filter out the noise from the NN outputs. The implementation flowchart of the combined NN and MWF approach is shown in Figure 43. Figure 44 to Figure 48 shows the SOC estimation results for Cell#1 to Cell #5. Again, the model used for different cells is the same and is learned from Cell#1's DST data, which is a different loading condition. Using the combined approach, the accuracy is improved consistently for all cells as you can see from Table 3, which summarizes the estimation errors. For example, the SOC estimation error of Cell #2 is reduced from 80% to 1.8% by using the new approach compared with original NN model. In addition, there are no sudden jumps and drops in the SOC any more. Smooth estimation of SOC is guaranteed.

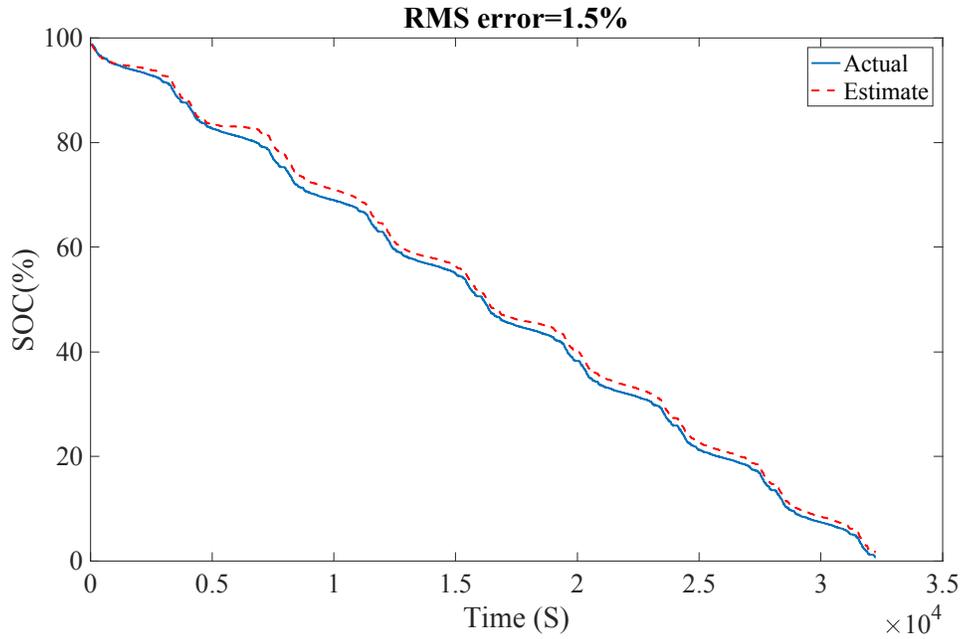


Figure 44: SOC estimation results for Cell #1 FDS data based on the combined

NN&MWF approach

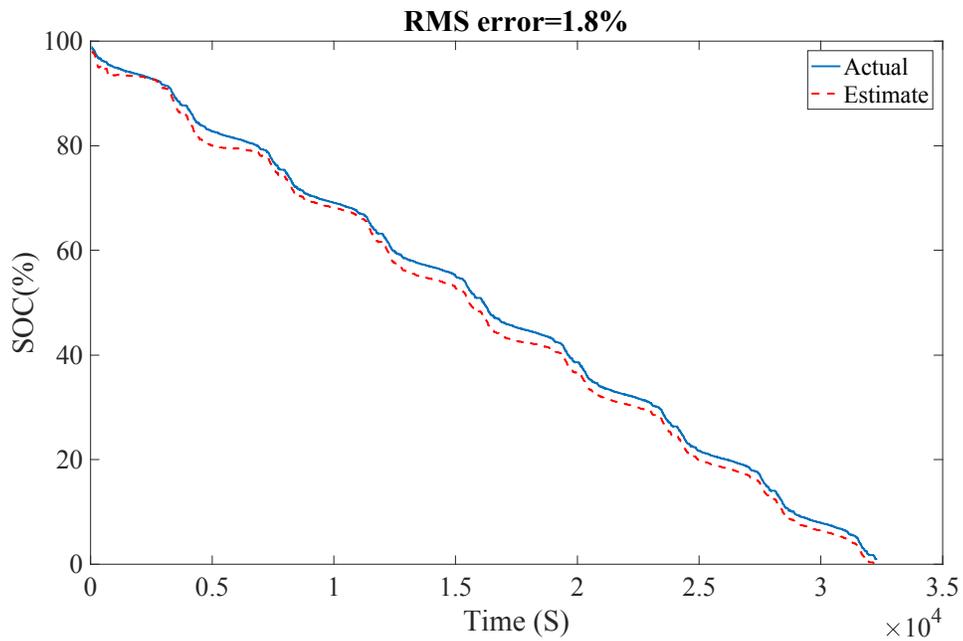


Figure 45: SOC estimation results for Cell #2 FDS data based on the combined

NN&MWF approach

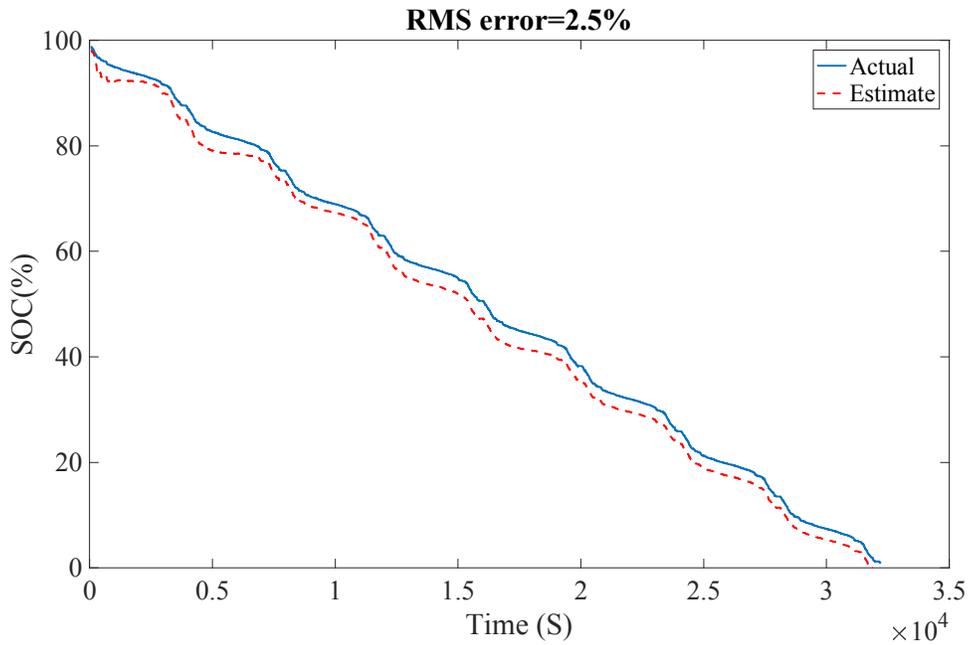


Figure 46: SOC estimation results for Cell #3 FDS data based on the combined

NN&MWF approach

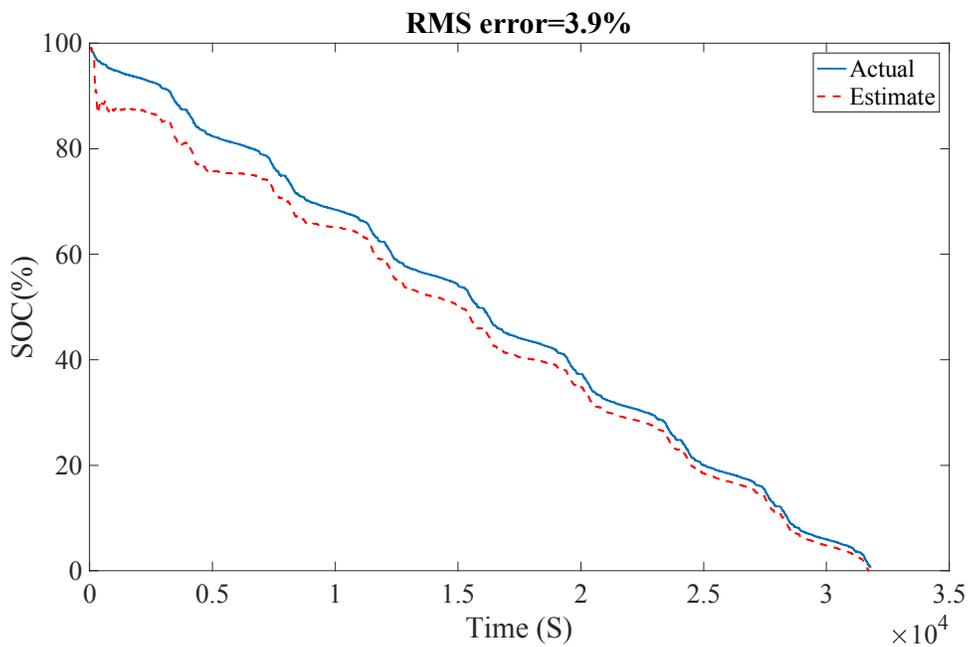


Figure 47: SOC estimation results for Cell #4 FDS data based on the combined

NN&MWF approach

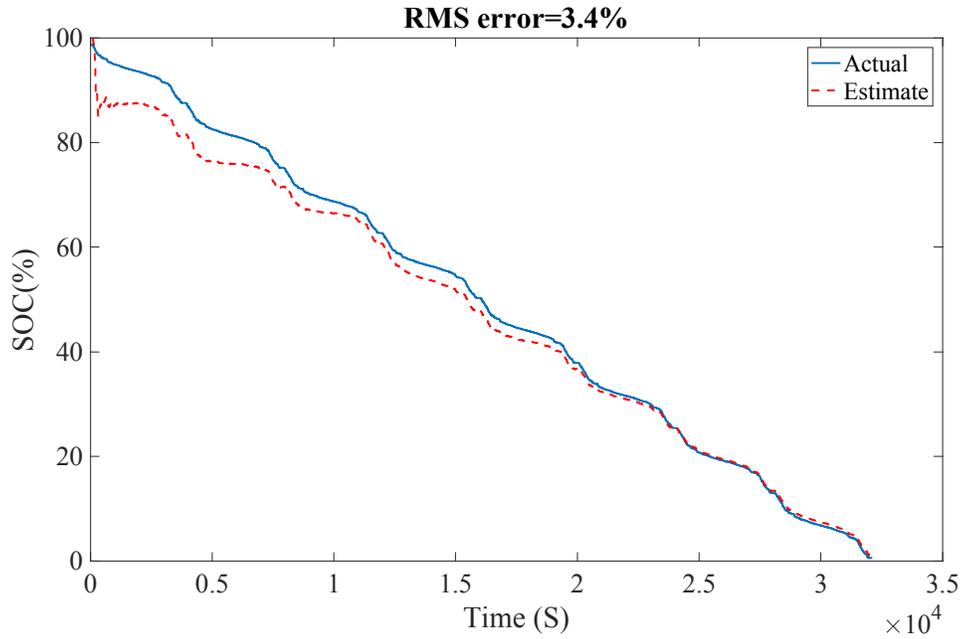


Figure 48: SOC estimation results for Cell #5 FDS data based on the combined NN&MWF approach

Table 3: Summary of SOC estimation results based on the data-driven approach

Cell ID	SOC Root Mean Square Error	
	NN	NN+MWF
Cell 1	3.3%	1.5%
Cell 2	8.0%	1.8%
Cell 3	8.8%	2.5%
Cell 4	18.9%	3.9%
Cell 5	23.3%	3.4%

5.7 Conclusions and Discussions

In this chapter, a state of charge (SOC) estimation approach was developed based on a feed forward neural network and a moving window filter (MWF). Neural network is a data-driven approach which can approximate the input and output relation of a system. Based on this advantage, the nonlinear mapping relationship of the SOC with the current and voltage measurements was modeled by a 2-layer feed-forward neural network. This study adopted a searching method to determine the best number of nodes and the dimension of the neural network inputs. This searching method begins with the simplest network structure with low dimension inputs and goes to a complex network with high dimension inputs until the RMS error of the trained model reaches 1%.

To estimate battery SOC, the inputs of the neural network were the voltage and current measurements and the output was the SOC. The data collected from the dynamical stress testing were used to train the neural network, and the data acquired from the Federal Driving Schedule were used to test the accuracy and generalization capability of the neural network. The overall RMS error of neural network based SOC estimation from 100% to 0% SOC was larger than 8% for most of the cells. High estimation variations were shown at the flat discharge zone of the LiFePO₄ battery, i.e., 30–80% SOC, which is not desirable in EV applications, since it will result in a sudden drop or jump in the residual range prediction and cause inconvenience and anxiety to drivers. Thus, to further improve the estimation accuracy and reduce the estimation variance, a MWF-based approach was implemented to filter out the errors in the neural network

estimation. After the filtering, the RMS error of the overall SOC estimation was reduced to less than 4%

This modeling approach eliminates the need to determine an open circuit voltage–SOC lookup table, unlike the equivalent circuit model based SOC estimation. The data required to train the model are current, voltage, and SOC. Therefore, the field collected data can be used to update the neural network and increase the estimation accuracy.

Chapter 6: Contributions and Suggestions for Future Research

6.1 Contributions of this Dissertation

Contributions of this thesis include the following:

1) Computational complexity of electrochemical models is the major obstacle for the application of electrochemical models to the real-time battery SOC estimation. The diffusion partial differential equations (PDEs) in solid phase particles contributes more than 80% calculations of the full model. Available methods for solving PDEs are either too computationally intensive or numerically instable. This work developed a new computationally efficient numerical solution for the diffusion PDEs in solid phase particles based on the optimized projection method. The new solution guarantees numerical stability and minimizes the computational requirements:

- Developed a method to generate numerical stable equations for the diffusion PDEs in solid phase particles. This was achieved by using orthonormal basis functions and reformulating the generated differential algebraic equations (DAEs) to ordinary differential equations (ODEs). A process to generate orthonormal basis functions from elementary basis functions (e.g. $1, r^2, r^3, \dots, r^{2n}$) was developed based on Gram-Schmidt algorithm.
- Developed an algorithm to optimize the basis function for the diffusion PDEs in solid phase particles, in order to reduce the number of generated equations after the projection of the PDE. The optimized basis functions were formulated by a linear transformation of the high-order orthonormal

basis functions. The transformation matrix was found by minimizing the second-order norm difference between the reduced order model and the high-order model. The optimized basis function generates much less equations compared with the original orthonormal basis functions, while keeps the similar accuracy and preserves numerical stability. The evaluation results showed that the developed method significantly improves efficiency and reduces the memory requirement of diffusion PDEs. For instance, the developed algorithm is 20 times more efficient than conventional finite difference method and 3 times more efficient than the conventional projection method.

Besides being used in solving solid diffusion PDEs in this thesis, the developed optimized projection method can be applied to solve electrolyte diffusion PDEs, and can be extended to solve multi-dimension diffusion PDEs, to enable more efficient and accurate calculations.

- 2) Conventional state filter algorithms, e.g. extended Kalman filter and unscented Kalman filter, converge slowly given large initial SOC estimation errors (e.g. 40% SOC errors). This thesis developed a moving window filter (MWF) that utilizes data from a moving window, which can converge faster than traditional Kalman filters. The filter gain was derived based on maximum likelihood theory. The MWF was implemented with the computationally efficient electrochemical model developed in the first part, and was validated using experimental data based on the federal driving schedule. It showed that the MWF can reduce the convergence time

from 30mins to less than 2mins compared with unscented Kalman filter for the SOC estimation.

- 3) The major disadvantage of data-driven models is their ability to generalize, which means that a trained model can generate large estimation errors for the loading conditions that are not included in the training data set. To solve this problem, this thesis developed a hybrid data-driven method for SOC estimation, which combines the neural network and the moving window filter. The new approach significantly improves the neural network performance, for example it reduces more than half of the SOC estimation error compared with the original neural network approach. The developed approach can effectively improve the generalization capability of the neural network.

6.2 Suggestions for Future Research

The future research can focus on how to take the battery aging into account for SOC estimation. For electrochemical models, it means that the model parameters have to be updated with aging, for example, the cyclable lithium will diminish and film resistance will grow with aging. For data-driven models, like neural networks, the model may have to be re-trained in order to represent the current aging state. Two approaches can be used to solve this problem. The first one is to develop the aging model based on lab or field data off-line. This will require us to identify an aging indicator, and then build a model to describe how the model parameters change as a function of the aging indicator. The second approach is to develop an on-board parameter updating algorithm

to estimate the parameters based on in-situ collected data. Ideally, this approach would be better than the first one, because model can be adapted to particular cells, while the first approach may only be able to capture a median aging trend among a distribution and unit-to-unit variation can be an issue. With these merits, however, the development of the on-line parameter updating algorithm is challenging, as it is difficult to prove the stability of the on-line update algorithm. In addition, to avoid out of range estimations, constraints must be implemented for the optimization, as a result, iterative optimizations may be necessary. How to reduce the computational complexity of the optimization and how to ensure the convergence of the optimization for on-line parameter estimation can be the major topics in this research direction.

Bibliography

1. Palacin, M.R., *Recent advances in rechargeable battery materials: a chemist's perspective*. Chemical Society Reviews, 2009. **38**(9): p. 2565-2575.
2. Taracson, J. and M. Armand, *Issues and challenges facing lithium ion batteries*. nature, 2001. **414**: p. 359-367.
3. Breitfelder, K. and D. Messina, *The Authoritative Dictionary of IEEE Standards Terms*. Institute of Electrical and Electronics Engineers (IEEE), 2000.
4. Ng, K.S., et al., *Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries*. Applied energy, 2009. **86**(9): p. 1506-1511.
5. Yan, J., et al., *Robust state of charge estimation for hybrid electric vehicles: Framework and algorithms*. Energies, 2010. **3**(10): p. 1654-1672.
6. Zhao, L., M. Lin, and Y. Chen, *Least - squares based coulomb counting method and its application for state - of - charge (SOC) estimation in electric vehicles*. International Journal of Energy Research, 2016. **40**(10): p. 1389-1399.
7. Pop, V., et al., *State-of-the-art of battery state-of-charge determination*. Measurement Science and Technology, 2005. **16**(12): p. R93.
8. Xing, Y., et al., *State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures*. Applied Energy, 2014. **113**: p. 106-115.

9. Tanaka, T., et al., *Accurate and versatile simulation of transient voltage profile of lithium-ion secondary battery employing internal equivalent electric circuit*. Applied Energy, 2015. **143**: p. 200-210.
10. Xiong, R., et al., *A novel method to obtain the open circuit voltage for the state of charge of lithium ion batteries in electric vehicles by using H infinity filter*. Applied Energy, 2017. **207**: p. 346-353.
11. He, H., R. Xiong, and J. Fan, *Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach*. Energies, 2011. **4**(4): p. 582-598.
12. Lee, S., et al., *State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge*. Journal of power sources, 2008. **185**(2): p. 1367-1373.
13. Hu, X., S. Li, and H. Peng, *A comparative study of equivalent circuit models for Li-ion batteries*. Journal of Power Sources, 2012. **198**: p. 359-367.
14. Liaw, B.Y., et al., *Modeling of lithium ion cells—A simple equivalent-circuit model approach*. Solid state ionics, 2004. **175**(1-4): p. 835-839.
15. Cho, S., et al., *State-of-charge estimation for lithium-ion batteries under various operating conditions using an equivalent circuit model*. Computers & Chemical Engineering, 2012. **41**: p. 1-9.
16. Feng, T., et al., *Online identification of lithium-ion battery parameters based on an improved equivalent-circuit model and its implementation on battery state-of-power prediction*. Journal of Power Sources, 2015. **281**: p. 192-203.

17. Doyle, M. and J. Newman, *The use of mathematical modeling in the design of lithium/polymer battery systems*. *Electrochimica Acta*, 1995. **40**(13-14): p. 2191-2196.
18. Newman, J. and W. Tiedemann, *Porous - electrode theory with battery applications*. *AIChE Journal*, 1975. **21**(1): p. 25-41.
19. Srinivasan, V. and J. Newman, *Discharge model for the lithium iron-phosphate electrode*. *Journal of the Electrochemical Society*, 2004. **151**(10): p. A1517-A1529.
20. Di Domenico, D., A. Stefanopoulou, and G. Fiengo, *Lithium-ion battery state of charge and critical surface charge estimation using an electrochemical model-based extended Kalman filter*. *Journal of dynamic systems, measurement, and control*, 2010. **132**(6): p. 061302.
21. Newman, J. and K.E. Thomas-Alyea, *Electrochemical systems*. 2012: John Wiley & Sons.
22. Rahimian, S.K., S. Rayman, and R.E. White, *Comparison of single particle and equivalent circuit analog models for a lithium-ion cell*. *Journal of Power Sources*, 2011. **196**(20): p. 8450-8462.
23. Rahimian, S.K., S. Rayman, and R.E. White, *State of Charge and Loss of Active Material Estimation of a Lithium Ion Cell under Low Earth Orbit Condition Using Kalman Filtering Approaches*. *Journal of The Electrochemical Society*, 2012. **159**(6): p. A860-A872.

24. Khaleghi Rahimian, S., S. Rayman, and R.E. White, *Extension of physics-based single particle model for higher charge–discharge rates*. Journal of Power Sources, 2013. **224**(0): p. 180-194.
25. Ramadesigan, V., et al., *Modeling and Simulation of Lithium-Ion Batteries from a Systems Engineering Perspective*. Journal of The Electrochemical Society, 2012. **159**(3): p. R31-R45.
26. Subramanian, V.R., V.D. Diwakar, and D. Tapriyal, *Efficient macro-micro scale coupled modeling of batteries*. Journal of The Electrochemical Society, 2005. **152**(10): p. A2002-A2008.
27. Li, J., et al., *New method for parameter estimation of an electrochemical-thermal coupling model for LiCoO₂ battery*. Journal of Power Sources, 2016. **307**: p. 220-230.
28. Mayhew, C., et al. *Investigation of projection-based model-reduction techniques for solid-phase diffusion in Li-ion batteries*. in *American Control Conference (ACC), 2014*. 2014.
29. Chaturvedi, N.A., et al. *Approximations for partial differential equations appearing in li-ion battery models*. in *ASME 2013 Dynamic Systems and Control Conference*. 2013. American Society of Mechanical Engineers.
30. Harvey, A.C., *Forecasting, structural time series models and the Kalman filter*. 1990: Cambridge university press.
31. Han, X., et al., *Simplification of physics-based electrochemical model for lithium ion battery on electric vehicle. Part I: Diffusion simplification and single particle model*. Journal of Power Sources, 2015. **278**: p. 802-813.

32. He, H., et al., *State-of-charge estimation of the lithium-ion battery using an adaptive extended Kalman filter based on an improved Thevenin model*. IEEE Transactions on Vehicular Technology, 2011. **60**(4): p. 1461-1469.
33. Plett, G.L., *Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation*. Journal of Power sources, 2004. **134**(2): p. 277-292.
34. Xiong, R., et al., *Evaluation on state of charge estimation of batteries with adaptive extended Kalman filter by experiment approach*. IEEE Transactions on Vehicular Technology, 2013. **62**(1): p. 108-117.
35. Stetzel, K.D., et al., *Electrochemical state and internal variables estimation using a reduced-order physics-based model of a lithium-ion cell and an extended Kalman filter*. Journal of Power Sources, 2015. **278**: p. 490-505.
36. Han, J., D. Kim, and M. Sunwoo, *State-of-charge estimation of lead-acid batteries using an adaptive extended Kalman filter*. Journal of Power Sources, 2009. **188**(2): p. 606-612.
37. Vasebi, A., M. Partovibakhsh, and S.M.T. Bathaee, *A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended Kalman filter for hybrid electric vehicle applications*. Journal of Power Sources, 2007. **174**(1): p. 30-40.
38. He, W., et al., *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, 2014. **62**: p. 783-791.

39. He, W., et al., *State of charge estimation for electric vehicle batteries using unscented kalman filtering*. Microelectronics Reliability, 2013. **53**(6): p. 840-847.
40. Sun, F., et al., *Adaptive unscented Kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles*. Energy, 2011. **36**(5): p. 3531-3540.
41. Santhanagopalan, S. and R.E. White, *State of charge estimation using an unscented filter for high power lithium ion cells*. International Journal of Energy Research, 2010. **34**(2): p. 152-163.
42. He, H., R. Xiong, and J. Peng, *Real-time estimation of battery state-of-charge with unscented Kalman filter and RTOS μ COS-II platform*. Applied energy, 2016. **162**: p. 1410-1418.
43. Tian, Y., et al., *A modified model based state of charge estimation of power lithium-ion batteries using unscented Kalman filter*. Journal of power sources, 2014. **270**: p. 619-626.
44. Zhang, J. and C. Xia, *State-of-charge estimation of valve regulated lead acid battery based on multi-state Unscented Kalman Filter*. International Journal of Electrical Power & Energy Systems, 2011. **33**(3): p. 472-476.
45. Aung, H., K.S. Low, and S.T. Goh, *State-of-charge estimation of lithium-ion battery using square root spherical unscented Kalman filter (Sqrt-UKFST) in nanosatellite*. IEEE Transactions on Power Electronics, 2015. **30**(9): p. 4774-4783.

46. Partovibakhsh, M. and G. Liu, *An adaptive unscented Kalman filtering approach for online estimation of model parameters and state-of-charge of lithium-ion batteries for autonomous mobile robots*. IEEE Transactions on Control Systems Technology, 2015. **23**(1): p. 357-363.
47. Aung, H. and K.S. Low, *Temperature dependent state-of-charge estimation of lithium ion battery using dual spherical unscented Kalman filter*. IET Power Electronics, 2015. **8**(10): p. 2026-2033.
48. Corno, M., et al., *Electrochemical model-based state of charge estimation for Li-ion cells*. IEEE Transactions on Control Systems Technology, 2015. **23**(1): p. 117-127.
49. Doyle, M., et al., *Comparison of modeling predictions with experimental data from plastic lithium ion cells*. Journal of The Electrochemical Society, 1996. **143**(6): p. 1890-1903.
50. Forman, J.C., et al., *Genetic identification and fisher identifiability analysis of the Doyle–Fuller–Newman model from experimental cycling of a LiFePO₄ cell*. Journal of Power Sources, 2012. **210**(0): p. 263-275.
51. Meier III, L. and D.G. Luenberger, *Approximation of linear constant systems*. Automatic Control, IEEE Transactions on, 1967. **12**(5): p. 585-588.
52. Gugercin, S., A.C. Antoulas, and C. Beattie, *H₂ model reduction for large-scale linear dynamical systems*. SIAM journal on matrix analysis and applications, 2008. **30**(2): p. 609-638.
53. Jacobsen, T. and K. West, *Diffusion impedance in planar, cylindrical and spherical symmetry*. Electrochimica acta, 1995. **40**(2): p. 255-262.

54. Wan, E.A. and R. Van Der Merwe. *The unscented Kalman filter for nonlinear estimation*. in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. 2000. Ieee.
55. Jafarzadeh, S., C. Lascu, and M.S. Fadali, *State estimation of induction motor drives using the unscented Kalman filter*. IEEE Transactions on Industrial Electronics, 2012. **59**(11): p. 4207-4216.
56. He, W., et al. *State of charge estimation for electric vehicle batteries under an adaptive filtering framework*. in *Prognostics and System Health Management (PHM), 2012 IEEE Conference on*. 2012. IEEE.
57. Xiong, K., C. Chan, and H. Zhang, *Detection of satellite attitude sensor faults using the UKF*. IEEE Transactions on Aerospace and Electronic Systems, 2007. **43**(2).
58. Mirzaee, A. and K. Salahshoor, *Fault diagnosis and accommodation of nonlinear systems based on multiple-model adaptive unscented Kalman filter and switched MPC and H-infinity loop-shaping controller*. Journal of Process Control, 2012. **22**(3): p. 626-634.
59. Wang, Y. and W. Wang, *Failure prognostic of systems with hidden degradation process*. Journal of Systems Engineering and Electronics, 2012. **23**(2): p. 314-324.
60. Dai, J., D. Das, and M. Pecht, *Prognostics-based risk mitigation for telecom equipment under free air cooling conditions*. Applied energy, 2012. **99**: p. 423-429.

61. Kruschke, J.K., *Bayesian data analysis*. Wiley Interdisciplinary Reviews: Cognitive Science, 2010. **1**(5): p. 658-676.
62. Lee, P.M., *Bayesian statistics: an introduction*. 2012: John Wiley & Sons.
63. Zhang, L., et al., *Multi-objective optimization of lithium-ion battery model using genetic algorithm approach*. Journal of Power Sources, 2014. **270**: p. 367-378.
64. Antón, J.Á., et al., *Battery state-of-charge estimator using the SVM technique*. Applied Mathematical Modelling, 2013. **37**(9): p. 6244-6253.
65. Anton, J.C.A., et al., *Support vector machines used to estimate the battery state of charge*. IEEE Transactions on power electronics, 2013. **28**(12): p. 5919-5926.
66. Chang, W.-Y., *Estimation of the state of charge for a LFP battery using a hybrid method that combines a RBF neural network, an OLS algorithm and AGA*. International Journal of Electrical Power & Energy Systems, 2013. **53**: p. 603-611.
67. Charkhgard, M. and M. Farrokhi, *State-of-charge estimation for lithium-ion batteries using neural networks and EKF*. IEEE transactions on industrial electronics, 2010. **57**(12): p. 4178-4187.
68. Chen, C.-W., *Modeling and control for nonlinear structural systems via a NN-based approach*. Expert Systems with Applications, 2009. **36**(3): p. 4765-4772.
69. Cheng, B., et al., *Ni-MH batteries state-of-charge prediction based on immune evolutionary network*. Energy Conversion and Management, 2009. **50**(12): p. 3078-3086.

70. Kang, L., X. Zhao, and J. Ma, *A new neural network model for the state-of-charge estimation in the battery degradation process*. Applied Energy, 2014. **121**: p. 20-27.
71. Lee, Y.-S., W.-Y. Wang, and T.-Y. Kuo, *Soft computing for battery state-of-charge (BSOC) estimation in battery string systems*. IEEE Transactions on Industrial Electronics, 2008. **55**(1): p. 229-239.
72. Schlechtingen, M. and I.F. Santos, *Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection*. Mechanical systems and signal processing, 2011. **25**(5): p. 1849-1875.
73. Markou, M. and S. Singh, *Novelty detection: a review—part 2:: neural network based approaches*. Signal processing, 2003. **83**(12): p. 2499-2521.
74. Ali, J.B., et al., *Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network*. Mechanical Systems and Signal Processing, 2015. **56**: p. 150-172.
75. Ali, J.B., et al., *Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals*. Applied Acoustics, 2015. **89**: p. 16-27.
76. Giacinto, G. and F. Roli, *Design of effective neural network ensembles for image classification purposes*. Image and Vision Computing, 2001. **19**(9-10): p. 699-707.

77. Anthimopoulos, M., et al., *Lung pattern classification for interstitial lung diseases using a deep convolutional neural network*. IEEE transactions on medical imaging, 2016. **35**(5): p. 1207-1216.
78. Basheer, I.A. and M. Hajmeer, *Artificial neural networks: fundamentals, computing, design, and application*. Journal of microbiological methods, 2000. **43**(1): p. 3-31.
79. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*. nature, 1986. **323**(6088): p. 533.
80. Hunt, G., *USABC electric vehicle battery test procedures manual*. Washington, DC, USA: United States Department of Energy, 1996.
81. Kruse, R.E. and T.A. Huls, *Development of the federal urban driving schedule*. 1973, SAE Technical Paper.