Minimally Invasive Characterization of Lithium Iron Phosphate Battery Electrochemical and Health Models using Fisher Information and Optimal Experimental Design

by

Joel C. Forman

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2012

Doctoral Committee:

Assistant Professor Hosam K. Fathy, Co-Chair, Pennsylvania State University
Professor Jeffrey L. Stein, Co-Chair
Professor Dennis S. Bernstein
Professor Huei Peng

But “the least integer not nameable in fewer than nineteen syllables” is itself a name consisting of eighteen syllables; hence the least integer not nameable in fewer than nineteen syllables can be named in eighteen syllables, which is a contradiction.

— Bertrand Russell (The Berry Paradox [1])

This thesis is dedicated to my parents.

Acknowledgments

I would like to thank Professor Jeffrey Stein and Professor Hosam Fathy, my advisors. Both of whom were kind enough to give me a long leash in exploring my thesis and related work. Professor Stein has been tremendously helpful with big picture details and has a knack for seeing what is missing in both my work and my writing. Professor Fathy's hands-on help at the start of my time at Michigan and later his guidance when the road was either blocked or forked was beyond invaluable. Although I still question Professor Fathy's belief that sentences ideally have exactly 25 words, my writing has greatly benefited from his excellent critiquing and review.

I'm also greatly indebted to my committee members Professor Huei Peng and Dennis Bernstein. Prof. Peng was kind enough to let me order and work on battery testing equipment from one of his grants. He also protected me from my own ambition, suggesting that I focus more on battery experiment techniques for the fourth chapter rather than my original plan of focusing on microgrid communication and programming. This undoubtedly lead to a much more cohesive thesis and likely saved myself copious amounts of stress. Additionally, Prof. Peng's willingness to ask questions has been invaluable in polishing my work. I began working with Prof. Bernstein on a proposal relating to battery health modeling work. Prof. Bernstein's email response time and attention to detail are both second to none, as I learned writing a proposal with him. This proposal originally outlined a plan to connect a submodel estimation algorithm with the battery model in the proposal. Then he thought we should have a numeric example, which had seemed very ambitious to me given the deadline. But as a group we sat down and organized exactly what would need to happen to do this and a week later we had a working numeric example making the proposal even stronger. I continue to be impressed with how all the member of my committee continually strive for excellence.

Of course, working with my colleagues within the lab has been both invaluable and a pleasure. Dr. Scott Moura and Dr. Saeid Bashash were both tremendously helpful in their collaborations. As a whole the lab and its meetings were a great place to learn and sharpen skills. So much thanks to Rahul Ahlawat, Ben Pence, Rakesh Patil, and Tulga Ersal. As with many Autolab students the Meetings of the Miller House Congress proved an invaluable

place to meet people and exchange information, and I aspire to return as a keynote speaker.

As for my funding, I am grateful for my first year of funding from Rackham and the remainder from the NSF EFRI program. I am also grateful to live in an age where nearly any journal article can be found quickly, computer processing power is obscenely fast and cheap, and online tutorials can provide a quick introduction to nearly anything.

On a more personal note, I've met so many wonderful people in Ann Arbor that have been supportive, helpful, and insightful. It would be both impossible and silly to try listing all of them, although I'd be remiss to not specifically mention and thank Josh, Jess, Dan, Mike ("Haf"), and Amy. Additionally, while not in Ann Arbor, the group that I regularly played video games with (Chris, Tony, and Matt) either deserves credit for keeping me sane enough to graduate or grief for perhaps making the degree take longer than it should have.

None of this would have been possible without my parents. Brenda and Steve have always been supportive and encouraging of me without fail. From an early age they cultivated a questioning and self reliant attitude in me, for which I am forever grateful for. They have also bought me countless plane tickets, although I fear that may be at an end.

Preface

As with most research, this thesis and related work did not occur in a vacuum. When I first arrived at the University of Michigan my initial work was on the model reduction of an electrochemical battery model created by Doyle, Fuller, and Newman [2, 3]. Broadly, the lab's interest was in the design of PHEVs and our focus was largely on the battery pack. In this work I used infinite dimensional Padé approximation to reduce a spherical diffusion submodel within the DFN model. My colleague, Dr. Saied Bashash, reduced and simplified the algebraic constraints in the model with quasi-linearization. This work resulted in a conference paper, a journal article, and my Master's Thesis [4–6]. It is not included in this document although it enables a great deal of the computational work presented. It also has been used in several collaborations [7–9].

After this it became clear that efficient simulation of the DFN model was not enough, we would also need to experimentally characterize the $LiFePO_4$ cells. To this end we decided to focus on noninvasive and nondestructive experiments, for various reasons as are discussed in later this document. Another colleague, Dr. Scott Moura, built a battery test setup capable of running a wide variety of tests, including Hardware In the Loop (HIL). We then used this setup to collect data that I processed with an evolutionary algorithm. This algorithm estimated the DFN model parameters by matching the resulting input-output behavior. This set of parameters has been leveraged in several other works that I have been involved with [7, 9–11]. The parameter identification work resulted in a conference and journal paper and is the basis of Chapter 2 [12, 13]. Additionally it lead me to learn about Fisher information and the underlying need for optimal experimental design, which greatly influenced the remainder of my thesis work.

At this point the importance of modeling battery health became very clear within the lab. With very little $LiFePO_4$ battery health modeling in the literature we decided to setup our own battery aging experiment. The goal of this experiment was to create an experimentally derived map similar to that used in [8] which would be ideal for both control and design optimization. Dr. Scott Moura priced and ordered the long term battery cycling equipment, and I'm grateful for Professor Huei Peng's cooperation, patience, and funding with regard

to acquiring and using said equipment. For this work I performed optimal experimental design using the previously validated DFN model. This resulted in a set of 14 different Constant Current Constant Voltage (CCCV) cycling experiments, 4 of which were used solely for validation. This experiment is still ongoing, but a model fit to 429 days of cycling experiments matches the validation data very well. This work has been submitted and accepted to 2012 Dynamic Systems and Control Conference and is the work presented in Chapter 3 [10].

Based on the initial parameter identification work and a desire to make a battery State-of-Health (SoH) diagnostic, Chapter 4 focuses on design of open loop current trajectories that optimally gather information while attempting to minimize battery damage. The nucleation point of this work occured when Dr. Scott Moura and I were trying different battery cycles to aid in my identification work in Chapter 2. I found the trial and error necessary to generate information rich cycles frustrating and wondered why there was not a systematic way to design these cycles. Chapter 4 presents such a systematic method, optimizing for information content while minimizing battery damage. Battery damage is assesed using both the model in Chapter 3 and the work of Whitacre *et al.* [14]. Chapter 4 is intended for submission to the 2013 American Control Conference.

The common thread running through this thesis is a focus on minimally invasive battery characterization. I attempt to bring the science of optimal experimental design and information theory to the art of experimental battery testing. This is not to say that the art is not valuable - it is just that when combined with the science greater things become possible. I believe the work in this thesis along with my other work at the University of Michigan greatly improves our ability to efficiently characterize and simulate $LiFePO_4$ batteries.

Table of Contents

List of Tables

Table

List of Figures

Figure

List of Appendices

Appendix

Abstract

This dissertation bridges an important gap from optimal experimental design and information theory to battery modeling and experimentation. In doing so it creates methods to efficiently model and characterize batteries intended for electrified vehicles. This modeling and characterization focuses on both estimating parameters for a pseudo 2D electrochemical model and the determination of a battery aging model. One of the major goals within the work is to be minimally invasive. For the initial parameter identification and health modeling work this means that the cells are not being disassembled. Later in the dissertation this idea is taken further by using battery health models to minimize the damage caused by experiments. Another theme throughout this work is the use of Fisher information, which is a measure of how much information a set of data contains for estimating model parameters. This is used for both *a posteriori* analysis of parameter estimation accuracy and as an *a priori* goal for the two optimal experimental design problems within this dissertation.

This dissertation has three major parts. The first is a parameter identification exercise for the Doyle-Fuller-Newman model, a pseudo 2D electrochemical battery model. In this several experiments based on Plug-in Hybrid Electric Vehicle drive cycles are conducted and the resulting data sets are used to estimate a set of 88 model parameters. This estimation is accomplished using a genetic algorithm. The input-output model results are matched very well and Fisher information is used to quantify the parameter estimation accuracy. This estimation demonstrates not only that the model is appropriate for $LiFePO_4$ but also how to quickly parameterize batteries for this model.

The second part of the dissertation focuses on battery health modeling. Initially it uses the previously parameterized electrochemical model to create a set of possible constant current constant voltage battery experiments. This set is then acted on by the DETMAX algorithm which creates a locally optimal subset of experimental trials. These cycles are then run repeatedly to age the batteries with battery health tests conducted intermittently. Results from the first 429 days of this experiment yield a battery aging model that accurately fits our validation data. This model is designed specifically to be useful for control and optimization work.

The third major part of this dissertation is focused on designing battery experiments that maximize parameter information gathered while minimizing battery damage. The structure of the experiment is to design a current trajectory that maximizes parameter information carried by the output voltage trajectory. The NSGAII-DE evolutionary algorithm is used with the previously fit electrochemical model and the battery aging model to generate Pareto fronts in the form of Fisher information versus battery damage. The resulting improvements are verified by running an estimator on simulated data with noise.

Together this dissertation's contributions provide methods and algorithms for efficient and accurate battery characterization. These contributions will only become more relevant with the increasing prevalence of vehicle electrification and continued creation of new battery types.

Chapter 1
Introduction

1.1 Motivation and Objectives

As Plug-in Hybrid Electric Vehicles (PHEVs) increase in market share they create great potential to reduce dependence on petroleum, potentially decreasing green house gases, avoiding environmental disasters, and decreasing foreign resource dependence. However, PHEVs may have an Achilles heal in one major component - the battery pack. Battery packs today are typically a sizable portion of the cost of a PHEV, to the extent that they frequently must be subsidized to be economically competitive. Batteries must be well modeled to ensure that on vehicle controllers can effectively manage available power and capacity effectively. Additionally, since battery's health degrades with use, it is important to understand battery State-of-Health (SoH). SoH affects available power and capacity which, in turn, effects available power and remaining trip length.

Fortunately the dynamic battery modeling literature is relatively mature for Li-ion batteries (which are typical in today's PHEVs). However these models still require appropriate calibration through parameter identification. Unfortunately the Li-ion battery health modeling literature is not nearly as mature. The battery health literature has a solid understanding of potential degradation mechanisms, but is still struggling to produce health models suitable for controllers.

Experiments are necessary to identify parameters for electrochemical battery models and to correctly deduce health models. While the literature is rich with battery experiments few of these make use of information theory or Optimal Experimental Design (OED). This is important because information theory and OED have great promise for both reducing time and money spent on experiments while at the same time improving parameter identification results from experiments. Information theory allows one to evaluate how well successful one can hope to be from given data. OED optimizes information, specifically Fisher information, for experiments by designing them through either selection of trials (for static experiments) or input trajectory design (for dynamic experiments).

This dissertation applies, creates and studies several information theory and OED techniques and uses them on three relevant battery identification problems. The first problem serves as a motivating example. In this the author attempts to identity 88 parameters related to scalar and parametric function control points in the Doyle-Fuller-Newman (DFN) electrochemical model. This uses a Genetic Algorithm (GA) to fit the model parameters to noninvasively collected current and voltage data. The noninvasiveness of the experiment is important because measuring invasive parameters within the battery often requires special equipment and manufactures frequently consider these parameter values proprietary. The data sets used to drive the experiment are based largely on PHEV drive cycles as opposed to experimental design. Despite the GA being able to accurately fit the model's input-output behavior an information study reveals that some parameters are completely unidentifiable or nearly so given the selected data. To improve this, increasing identifiability through input trajectory design is revisited in the third problem.

The second problem serves to show how static OED can make battery health tests more effective. Battery health is critical to PHEV operation: degraded batteries provide less power, shorten all electric range, and decrease hybrid mode fuel efficiency. Unfortunately, battery health experiments are both costly and time consuming, taking on the order of weeks to months - potentially years. OED can help alleviate this by selecting experimental trials that are best suited for health model identification. This decreases the number of iterations needed for a successful experiment and improves the model identified at the end of the experiment. Since this model can be used to predict and understand how PHEV use affects battery SoH its accuracy has both economic and vehicle design ramifications. Additionally this strategy can serve as a template for identifying health models for other types of batteries.

The third serves as a demonstration of how to design input trajectories to make the DFN model parameters most identifiable. It uses NSGAII-DE to design input trajectories that maximize Fisher information while minimizing the battery's exposure to damage. We demonstrate this with two different health metrics and generates Pareto fronts of Fisher information versus each. In both cases Fisher information is based on the simultaneous estimation of two parameters: d_2, solution diffusivity and R_{SEI}, the resistance due to anode film thickness. Each of these relates to battery health, and d_2 is chosen specifically due to previous difficulties in its estimation [13]. This method can be used for SoH estimation by applying it to parameters relating to health (such as as R_{SEI}). It can also be used to design more efficient experiments to improve battery characterization.

The remainder of the chapter is organized as follows. Next §1.2 outlines the three major contributions for this dissertation. §1.3 explains how these contributions improve the literature and then summarizes various technical areas of the literature for the reader. §1.4

then provides technical background on electrochemical battery modeling. The final section in this chapter §1.5 explains how the remainder of this document is organized.

1.2 Intellectual Contributions

This dissertation makes three major intellectual contributions which are summarized in this section.

1.2.1 Noninvasive Electrochemical Model Identification using Genetic Algorithms

While the DFN model has applicability to a large class of battery types one major hurdle in using this model is the identification of parameters and parameterized functions [2, 3]. A GA identification method is use to noninvasively identify the parameters and parameterized functions of the DFN model. This GA optimizes using noninvasively collected voltage and current data to fit the model parameters. As a demonstration, laboratory data is collected for a $LiFePO_4$ cell and the identification is carried out successfully. Some of this laboratory data is based on driving cycles simulated for a Prius with a 5 kWh PHEV conversion kit. A validation study of model fit is conducted and a Fisher information study has been conducted to assess parameter accuracy.

1.2.2 Optimal Experimental Design for Battery Health Modeling

This dissertation uses traditional OED to design battery health degradation experiments for determining a voltage and current driven health model [15–17]. The OED selects a set of experimental procedures that minimize the geometric sum of Cramér-Rao variance bounds for estimating the model's parameters [18]. This minimization creates the best possible set of data for performing parameter identification. The designed experiments are conducted in a laboratory to obtain this data-driven health model. This sequence serves as a prototype for conducting battery health experiments on other types of batteries. Additionally the specific health model can be used for health-conscious controllers, system level design decisions and warrantying decisions [8, 19].

1.2.3 Maximizing Fisher Information through NSGAII-DE for Battery Health Diagnostics

This dissertation investigates using NSGAII-DE to maximize Fisher information to design input trajectories for experiments that minimize Cramér-Rao lower bound on variance for identifying selected parameters. As a test case the author considers designing inputs to the DFN model. This input trajectory minimizes the variance in identifying the resistance of the solid electrolyte interface layer, R_{SEI} which is a major factor in battery health [20–34]. The solution diffusivity, d_2, is also simultaneously identified with R_{SEI}. One specific application of this technique is a diagnostic procedure for determining battery SoH in an efficient manner. The improvements of the optimized trajectories are verified using an estimator on simulated data with noise.

1.2.4 An OED Battery Modeling and Characterization Toolbox

Together these contributions provide a toolbox for battery system engineers. This toolbox allows one to characterize the battery for the DFN model, determine an empirical data driven health model, and then design diagnostics for determining specific battery parameters. This toolbox incorporates methods from OED to ensure that the experiments are efficient and result in appropriate models and parameters. In addition the author demonstrates specific pairings of method and numerical algorithm, showing the reader how to perform the appropriate computations for the experiments. This toolbox should be very helpful to engineers work on the design, control, and optimization of battery systems as it provides efficient and systematic ways to determine the necessary battery models.

1.3 Literature Review and Background

The issue of battery parameter identification has been addressed in the literature using a number of different models, identification methods, and parameter sets. Santhagopalan *et al.*, for instance, successfully identify a subset of the DFN model parameters using extended Kalman filtering [35]. Speltino *et al.* successfully identify the parameters of a Single Particle Model (SPM) of battery dynamics by splitting these parameters into two sets and identifying these sets sequentially [36]. Schmidt *et al.* also successfully identify an SPM battery model, with several extensions that incorporate temperature information and relate solid diffusion to State of Charge (SoC) [37]. The study by Schmidt *et al.* also examines parameter uncertainty using Fisher information [37]. Finally, Hu *et al.* successfully identify the parameters of

an equivalent circuit battery model using a GA [38]. In contrast to the above literature a noninvasive technique using GAs is proposed for identifying 88 parameters (scalars and function control points). This is believed to be the first attempt at trying to identify more than five parameters simultaneously in the DFN model [35]. This technique serves as a template for identifying parameters for other batteries and makes use of Fisher information to evaluate parameter identification accuracy.

There has been much work performed on studying battery health. For the $LiFePO_4$ battery being considered in this dissertation two long-term health studies have already been conducted (and the literature is rich with health experiments for many types of batteries, see §1.3.6). The first develops a control oriented model based entirely on charge processed [14]. The second develops a non-control oriented model based on a wider range of experiments [39]. However virtually no experiments in the battery literature make use of OED. In contrast, this dissertation uses OED to create battery health experiments that attempt to minimize parameter estimation error for a given model. Additionally a diagnostic experiment is designed using input trajectories to maximize the Fisher information for determining a parameter related to SoH. By designing these experiments to maximize information gathered one can minimize uncertainties in identified parameters and by extension model inaccuracies. Well designed experiments can also decrease time, man-hours and cost by minimizing the number of experiments and experimental batches needed. As battery applications, such as PHEVs, become more dependent on SoH the importance of being able to conduct battery health experiments in an effective and economic matter continues to increase.

The remainder of this section briefly reviews the relevant areas of the literature. Technical background for many of these areas is presented later sections.

1.3.1 Fisher Information

Fisher information measures how easily one can observe parameters in a probability density function from collected data. Due to the Cramér-Rao bound an unbiased parameter estimator must have a variance greater than or equal to the inverse of the Fisher information. When estimating multiple parameters one calculates a Fisher Information Matrix (FIM) and inverts it giving a lower bound on the best possible covariance obtainable when estimating parameters [16, 17, 40].

The concept of Fisher information has been applied to estimating parameters of dynamic models. This case assumes that the dynamic model has a normally distributed measurement error. Using this one can create a FIM that is related to the sensitivity functions of the

dynamic model. This FIM is again inverted, obtaining the best possible covariance matrix for an unbiased estimator. This is similar to parameter identifiability studies but provides a quantitative answer rather than a qualitative one. Computationally efficient methods exist to numerically calculate the FIM for dynamic models. This FIM can be used to both analyze the quality of data collected from an experiment and as a goal in designing experiments [16, 17, 37, 41–44].

1.3.2 Optimal Experimental Design

OED is the study of selecting experiments to maximize the Fisher information gained for parameters within a given model. It can also be used to select which model structures fit data the best. Frequently one looks at a set of possible experiments to run and then attempts to select the best subset of these given various objectives and constraints. For static experiments the DETerminate MAXimizing (DETMAX) algorithm lets one create (locally) optimal subsets from the library of possible experiments [15–17].

Limited work has been conducted on OED of dynamic experiments. Time domain and frequency design for linear systems has been considered, as has optimal sampling times. The use of NSGAII-DE or trajectory design should be broadly applicable and easy to implement, adding to the work already in this area [15–17].

1.3.3 Genetic Algorithms

GAs work by creating a population where each member represents a possible set of decision variable values. This is often represented by a binary string that maps bijectively to these values. This population is evolved using breeding and mutation guided by a fitness function. Due to their convergence properties GAs are frequently used in optimizations with large numbers of decision variables. Parallel computing can be used to help decrease computational expenses in large GA optimizations [18, 45].

1.3.4 The NSGAII-DE

An excellent evolutionary algorithm for multi-objective optimization is the Nondominated Sorting Genetic Algorithm II (NSGAII) which evolves populations to directly create Pareto fronts [46]. Another useful evolutionary algorithm is Differential Evolution (DE), which works directly on real encoded problems and has been shown to be very effective in a variety

of applications [47–49]. The advantages of each of these evolutionary methods are married by Kwan, Yang, and Chen who created the NSGAII-DE algorithm [50]. This algorithm combines the performance of DE with the advantage of direct Pareto front creation, and thus it is ideal for our problem.

1.3.5 Li-ion Battery Modeling

There has been a variety of work done on making electrochemical models of Li-ion batteries. Doyle, Fuller, and Newman wrote two papers creating the DFN model [2, 3]. The DFN model is an electrochemical battery model that models concentration and potential distributions across the width of the cell as well as concentration profiles in the porous electrodes of the anode and cathode. In their original form these models are very computationally inefficient so the Single Particle Model (SPM) was introduced, this simplifies some of the distributed dynamics [30, 36]. These models are more computationally efficient but suffer in accuracy at higher C-rates. SPM models have been augmented with thermal models as well [37]. A variety of work has been conducted on making the DFN model computationally tractable using model reduction techniques [4, 51–54]. The author specifically has a paper that uses infinite dimensional Padé approximation and quasi-linearization to reduce the DFN model (see §1.4) [4].

1.3.6 Li-ion Battery Degradation

Li-ion batteries have many degradation mechanisms depending on which combination of anode, cathode, electrolyte and dopant chemistries are used.

Anode Resistive SEI Film Formation: This degradation mechanism coats anode particles at the Solid Electrolyte Interface (SEI) creating a resistive film which increases impendence. Additionally this film is made in part of Li-ions decreasing the quantity of cyclable Li, negatively affecting maximum capacity [20–34].

Cathode Resistive SEI Film Formation: Similar to anode resistive film formation this increases impedance and decreases capacity. However in this case the film forms on cathode particles. This is reported for $Li_xNi_{0.8}Co_{0.15}Al_{0.05}O_2$, $Li_xNi_{0.8}Co_{0.2}O_2$ and $Li_xV_2O_5$ [32, 55–57].

Cathode Dissolution Causing Anode SEI Film: Here SEI film is formed in the anode from metallic atoms that have been dissolved in the electrolyte and act as Li when forming the film [20–22, 33, 58–60].

Loss of Carbon Dopant: In $LiFePO_4$ batteries, cathodes are often coated with a carbon dopant to increase conductivity. However this carbon may migrate during cycling decreasing conductivity [32, 55, 60–62].

Crystal Structure Distortion: When crystal structures have either too much or too little Li they may experience mechanical stresses, causing crystal structure distortion. This distortion can negatively affect how easily Li can transfer into and out of the crystal structure [63–65].

Electrolyte Degradation: The electrolyte degrades through a complicated multi step chemical reaction. Specific chemical mechanisms are discussed in [66].

Gas Evolution: As cells are cycled pressure inside them can build up due to gas evolution. This is reported for cells with $LiNi_xCo_yAl_zO_2$ cathodes [58].

Dendrite Formation: In many battery chemistries metallic dendrites can from creating short circuits inside the battery. In $LiFePO_4$ batteries the dendrites are made of Fe [67].

For the $LiFePO_4$ batteries under consideration in this dissertation two groups have already conducted health experiments. One group has created a control oriented health model based solely on charge processed [14]. The other has created a non-control oriented model and claim that the major degradation mechanism is anode SEI film formation [39]. This dissertation creates a control oriented health model that takes into account voltage and current. Experimental results in Chapter 3 show that terminal voltage is an important factor in battery health.

1.4 The Doyle-Fuller-Newman Electrochemical Model

The DFN model is an electrochemical battery model that models concentration and potential distributions across the width of the cell as well as concentration profiles in the porous electrodes of the anode and cathode. Spatial distributions across the width of the cell play an important role in high-rate charge and discharge dynamics, typical of PHEV cycles. The model is described thoroughly in [2, 3, 28]. The remainder of this section provides a brief overview of the mathematics involved in this model.

The diffusion of Li-ions within the electrolyte is governed by Fick's law of linear diffusion combined with an intercalation current density term, J, transferring Li-ions between

the solution and solid:

$$\varepsilon_2 \frac{\partial c_2}{\partial t} = \nabla(d_2^{eff} \nabla c_2) + \frac{1-t^+}{F} J. \tag{1.1}$$

The above intercalation current density, J, also acts as an input to the dynamics of Li-ion diffusion within the solid. This diffusion occurs at every point in the anode and cathode and can be modeled using a spherical, radially symmetric diffusion law as follows:

$$\frac{\partial c_{1,j}}{\partial t} = \frac{D_{1,j}}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial c_{1,j}}{\partial r} \right). \tag{1.2}$$

The total intercalation current density, J, equals the main intercalation reaction current density, J_1, plus any additional intercalation current density J_{sd} representing side reactions in the battery. The main intercalation reaction current density, J_1, is driven by potential differences between the solid and electrolyte solution, and governed by the Butler-Volmer equation:

$$J_1 = a_j i_{0,j} \left(e^{\frac{\alpha_{a,j} F}{RT} \eta_j} - e^{-\frac{\alpha_{a,j} F}{RT} \eta_j} \right), \tag{1.3}$$

$$i_{0,j} = k_j \left(c_{1,j}^{max} - c_{1,j}^{S} \right)^{\alpha_{a,j}} \left(c_{1,j}^{S} \right)^{\alpha_{c,j}} (c_2)^{\alpha_{a,j}}. \tag{1.4}$$

The over-potentials in the above equations equal the differences between the solid and solution potentials minus the reference potentials for the main intercalation reaction, which in turn depend on the local SoC, see Eq.1.5 and Eq.1.6.

$$\eta_p = \phi_1 - \phi_2 - u_{pref}, \tag{1.5}$$

$$\eta_n = \phi_1 - \phi_2 - u_{nref} - \frac{J}{a_n} R_{SEI}. \tag{1.6}$$

Since the above potentials/over-potentials can change much faster than the Li-ion concentrations, they are assumed to respond instantaneously. The solid potential is governed by Ohm's law with a term governing the charge transfer due to intercalation:

$$\nabla \left(\sigma_j^{eff} \nabla \phi_{1,j} \right) - J = 0. \tag{1.7}$$

Similarly, the solution potential is governed by Ohm's law, intercalation current density, and the charge carried by the ions in solution:

$$\nabla \left(\kappa^{eff} \nabla \phi_2 \right) + J + \nabla \left(\kappa_D \nabla \ln(c_2) \right) = 0. \tag{1.8}$$

The above system of equations governs the dynamics of charging and discharging in the Li-ion cell. From a dynamic systems perspective, it is a system of Differential Algebraic Equations (DAEs), where the differential equations govern the diffusion dynamics and the algebraic equations constrain the potentials and intercalation current accordingly [68]. Fortunatly there are efficient methods to numerically simulate the DFN model, see §A.1.

1.5 Document Organization

The remainder of this document is organized as follows. While the next three chapters of this dissertation have a logical progression they are self contained enough that they may be read independently. The first of these, Chapter 2 uses an experimental setup along with a genetic algorithm to identify parameters to match the input output behavoir of the DFN model. The last part of this chapter discusses parameter accuracy using Fisher information. This then transistions to the ideas of Chapter 3 where optimal experimental design is used to create battery aging experiments which optimize Fisher information. These experiments are then conducted and a control/optimization oriented model is created. Then, in Chapter 4 we revisit the parameter identification problem but use the NSGAII-DE algorithm to design current trajectories that maximally gather parameter Fisher information while minimizing battery damage (which is computed from the model in Chapter 3). The final chapter in this dissertation is Chapter 5 which is the conclusion.

Chapter 2

Genetic Identication and Fisher identifiability analysis of the Doyle-Fuller-Newman model from experimental cycling of a $LiFePO_4$ cell

2.1 Introduction

This chapter examines the problem of identifying the parameters of the electrochemical battery model developed by Doyle, Fuller, and Newman (DFN) [2, 3] using noninvasive voltage-current cycling experiments. The chapter presents a framework for solving this problem, consisting of a Genetic Algorithm (GA) for parameter identification combined with Fisher information-based estimation of parameter identifiability and identification errors. We apply this framework to $LiFePO_4$ battery cells intended for plug-in hybrid electric vehicles (PHEVs). The ultimate goal is to obtain an experimentally-validated, electrochemistry-based model of these batteries that can enable the optimization of PHEV design and control for objectives such as reducing PHEV fuel consumption and greenhouse gas emissions [69].

The DFN model is well-suited for this study because it is a first-principles electrochemical model that can capture high-rate transient effects. In contrast to equivalent circuit models, first-principles models make it easier to relate model parameters back to physical quantities (such as diffusivity and porosity). This is important because one of our goals is to investigate the accuracy with which one can estimate these physical quantities from noninvasive voltage and current data. The DFN model also captures high-rate transient effects typical of PHEV applications. In particular, a recent study by Santhagopalan *et al.* shows that the DFN model fits battery behavior above 1C current rates better than a single particle model (SPM) [35]. The DFN model achieves these advantages over the SPM in part by modeling spatial distributions of lithium across the width of the anode, separator, and cathode. These effects are ignored in SPMs.

A significant body of literature already exists on battery parameter identification using different models and identification methods for different parameter sets. Here we focus exclusively on identification methods that, like our own work, are intended for offline use. Santhagopalan *et al.*, for instance, successfully identify a subset of five parameter values under constant charge and discharge conditions for both the DFN and SPM models [35]. In both cases, the Levenberg-Marquardt optimization algorithm is used to obtain parameters that minimize model error. Speltino *et al.* successfully identify the parameters of a single-particle model of battery dynamics using a two-step process: they first identify the cathode equilibrium potential function from open circuit voltage measurements, assuming a known anode equilibrium potential function from the literature. They subsequently perform dynamic tests to estimate the remaining model parameters [36]. Schmidt *et al.* also successfully identify a single-particle battery model, with several extensions that incorporate temperature information and relate solid diffusion to State-of-Charge (SoC) [37]. The study by Schmidt *et al.* also examines parameter uncertainty using Fisher information. Finally, Hu *et al.* successfully identify the parameters of an equivalent circuit battery model using a GA [38]. They focus on two types of batteries, one of which is the A123 Systems 26650 cell examined in this chapter. Their equivalent circuit formulation is a set of n parallel resistor-capacitor pairs connected to each other in series.

The above literature provides a rich background for this chapter. In contrast to that background, this work achieves a combination of five important goals never pursued simultaneously in the previous literature to the best of the author's knowledge. First, we focus on identifying parameters of the DFN model: a choice justified by this model's first-principles nature and suitability for high-rate transient battery operation. Second, we identify the full set of parameters (88 scalars and function control points) of the DFN model using a GA, as opposed to a subset of these parameters. Third, we perform this identification using multiple battery cycles derived from vehicle drive cycles. (Note, altogether, these three choices of battery model, identification parameters, and cycling data represent an overarching goal of obtaining a DFN parameter set suitable for PHEV simulation and design/control optimization). Fourth, we use Fisher information to assess the accuracy of all 88 DFN model parameters, as opposed to computing Fisher information for a subset of these parameters. This is extremely important, because a parameter deemed identifiable based on Fisher information computation for a small parameter set may lose identifiability when one computes Fisher information for a larger parameter set. Fisher information provides a minimum variance bound for the estimated parameters via the Cramér-Rao inequality [40–44]. In addition, this chapter provides original insights about the dominant dynamics in the $LiFePO_4$ battery under the tests considered herein, and relates these insights to parameter

uncertainties.

The remainder of this chapter is organized as follows. §2.2 describes the experiments used for identifying the DFN model. This includes a discussion of the experimental setup along with the various PHEV drive cycle inputs. §2.3 summarizes the DFN model. §2.4 and §2.5 describe the unknown parameter set and genetic optimization algorithm, respectively. §2.5 also briefly describes model reduction methods used to simulate the DFN model, including quasi-linearization and modal decomposition [5, 54]. §2.6 presents validation studies for the identified model. This includes voltage and power trajectories of validation data along with probability density plots summarizing the errors. Finally, §2.7 presents the Fisher information results on parameter accuracy analysis. §2.8 summarizes and concludes the chapter.

2.2 Experimental Setup

The battery cells examined in this chapter are A123 Systems ANR26650M1 cells with $LiFePO_4$ cathodes. These cells have a 2.3Ah nominal capacity when fresh, a nominal voltage of 3.3V, and a maximum continuous discharge current of 70A (30.4 C-rate). The cells are intended for transient high-power applications including commercial PHEVs, PHEV conversion kits, and portable power tools. Experimental cycling data sets have been collected for these cells using a custom-built battery tester. This tester is capable of highly transient current/voltage profiles and can switch quickly between charging and discharging. These characteristics make it ideal for testing batteries under conditions similar to those experienced in PHEV battery packs. Additionally, this setup is capable of battery-in-the-loop studies, which will be advantageous for future battery control and estimation research [70].

This battery tester combines three major hardware components: an electric load (Sorenson SLH-60-120-1200), a power supply (Sorenson DSC20-50E), and a Real-Time (RT) controller and I/O board (dSpace DS1104). Figure 2.1 is a photograph of the battery tester, and Figure 2.2 is a schematic of the setup where all signal lines are connected to the I/O board. The power supply and electric load handle battery charging and discharging, respectively. The RT I/O board coordinates the electric load, power supply, and switching board. In addition, the RT I/O board records sensor signals including voltage and current. These signals are exchanged among the setup's various components in a variety of formats, including the analog, digital, PWM, SMBus, RS-232, and TTL formats. The switch board swaps the setup between charging and discharging by swapping the battery's connection between the power supply and load. The Schottky diode protects the power supply from

absorbing battery energy. The battery sensor board measures battery voltage through a voltage-isolating differential op-amp, and measures battery current via a bi-directional 20A Hall effect sensor (Allegro Microsystems ACS714). Finally, all the battery interface electronics are implemented on custom-build Printed Circuit Boards (PCBs) to maximize overall setup reliability, which is critical for long-term tests.

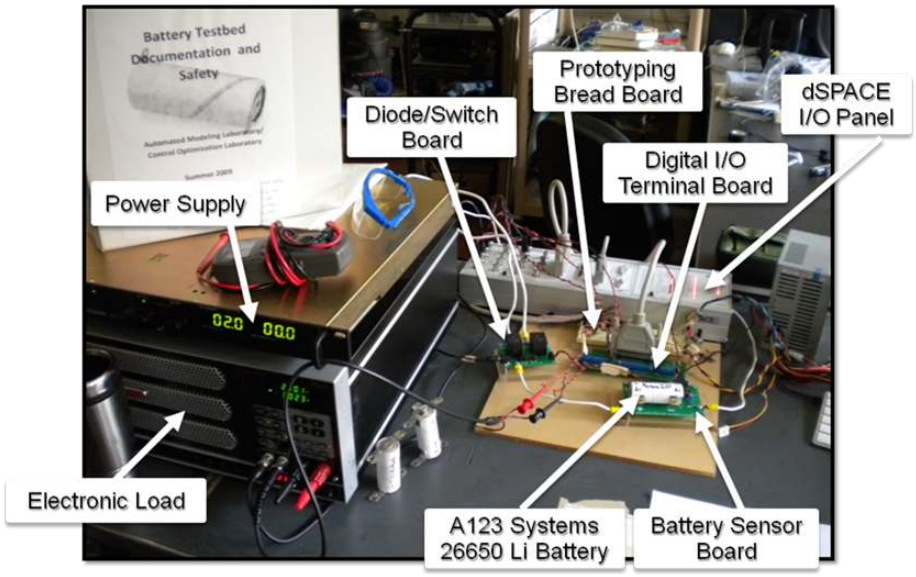

Figure 2.1 Photograph of experimental battery tester

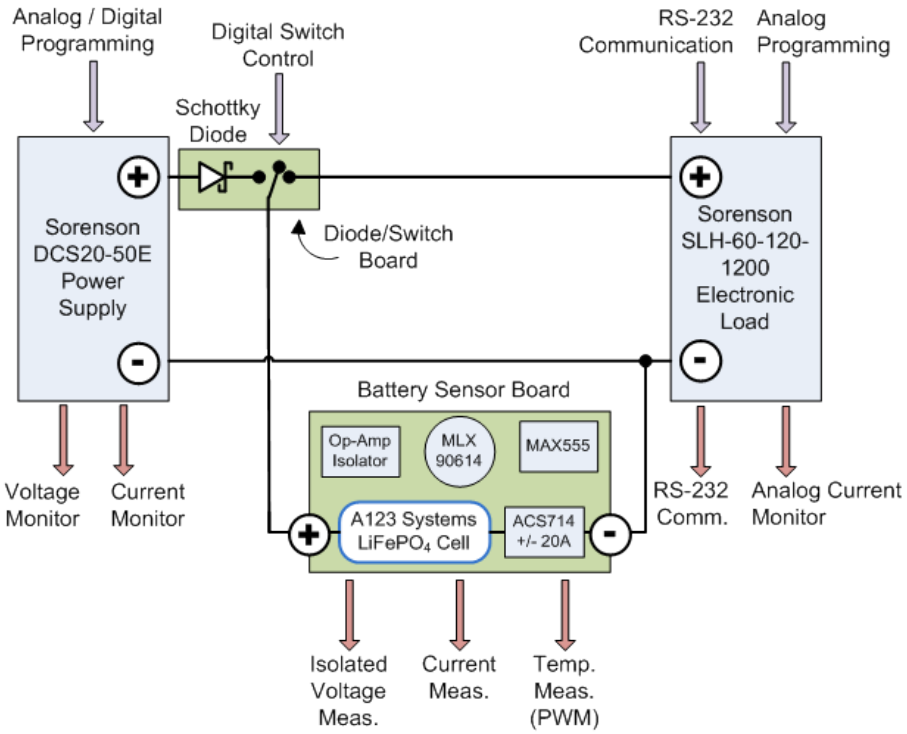

Figure 2.2 Schematic of experimental battery tester

Seven battery cycling tests have been conducted using this battery tester: two for model identification and five for validation. All of these tests initialize the battery SoC to 90% (3.35V relaxed), then subject the battery to a given current profile and measure the resulting battery voltage. In the first identification data set, the current profile consists of a Chirp sequence of three CCCV charge/discharge patterns between 2.0V and 3.6V, with charge/discharge rates of 5C, 2.5C, and 1C. In the remaining tests, the current profiles are generated by simulating a PHEV powertrain for a given vehicle drive cycle (*i.e.*, velocity-versus-time profile). Two of these vehicle drive cycles corresponding to the morning and evening commutes of a real human driver in a naturalistic driving study conducted by the University of Michigan Transportation Research Institute (UMTRI) [71]. These drive cycles are exact recordings of driver behavior using mid-sized sedans, these two specific cycles correspond to the same sedan on the same day. These battery tests are denoted as *Naturalistic1* and *Naturalistic2*, respectively. The four remaining battery tests correspond to multiple repetitions of standard vehicle certification drive cycles. These battery tests are denoted by *UDDSx2*, *US06x3*, *SC03x4*, and *LA92x2*, where the number in"*x#*" refers to the number of drive cycle repetitions [72]. For each of these drive cycle-based battery tests, a mid-size power-split sedan PHEV is simulated with a previously-optimized power management algorithm [73] to map the vehicle drive cycles to battery current profiles. This PHEV has a 5 kWh battery pack consistent with existing Toyota Prius PHEV conversion kits. Due to sensor limitations, drive cycles that produce current magnitudes greater than 20A (namely, *US06x3*, *SC03x2*, and *LA92x2*) are scaled down such that their maximum amplitude over time is 20A [72]. Specifically, this scaling divides the current trajectory by its maximum current and then multiplies the trajectory by 20. Finally, the resulting current profiles are applied to the battery cell to obtain data sets for identification and validation.

2.3 The Doyle-Fuller-Newman Battery Model

The DFN model is an electrochemical battery model that describes the dynamics of concentration and potential distributions across the width of the cell as well as concentration profiles in the porous electrodes of the anode and cathode. Spatial distributions across the width of the cell play an important role in high-rate charge and discharge dynamics, typical of PHEV cycles. The model is described thoroughly in [2, 3, 28]. This section summarizes the model equations, which constitute a nonlinear partial differential algebraic equation system. Appendix C contains the model's boundary conditions.

As seen in Fig. 2.3, a Li-ion battery cell consists of an anode, separator, and cathode

sandwiched between current collectors. Both the anode and cathode are made of porous solid material immersed in an electrolyte solution. When the battery is fully charged, Li-ions occupy interstitial sites in the anode-side solid material. As the battery discharges, the Li-ions leave these interstitial sites, entering the electrolyte solution. The Li-ions then migrate through the solution from the anode to the separator, and eventually the cathode. The discharging process concludes with the lithium ions coming to rest at interstitial sites in the cathode-side solid material. When a Li-ion leaves its interstitial site in the anode an electron is freed to flow through the external circuit, producing useful work. When this electron reaches the cathode it causes a Li-ion to bond with a cathode interstitial site. Charging the battery is the same process in reverse, with the external circuit providing rather than consuming energy.

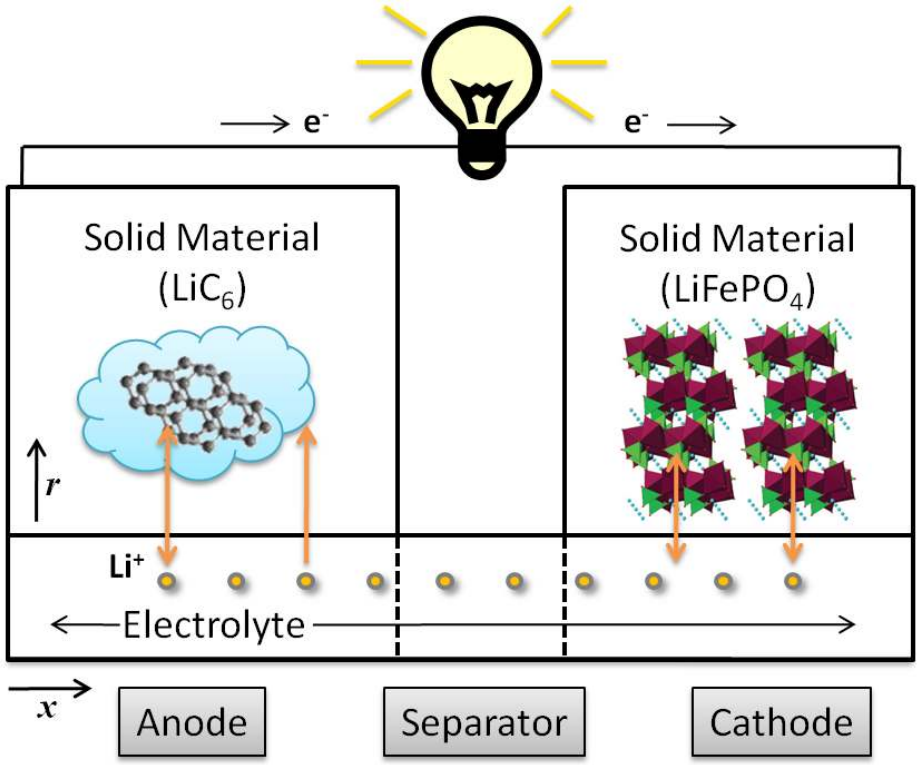

Figure 2.3 Li-ion cell schematic

The DFN model captures local Li-ion concentrations and potentials using coupled partial differential equations (PDEs). These PDEs account for the linear diffusion of Li-ions in the electrolyte, spherical diffusion of Li-ions in the solid, and the spatially distributed electrochemical reactions driving them to transfer between the solution and the solid. The remainder of this section briefly outlines these equations. The parameters of these equations are summarized in Table 2.1 and all of the boundary conditions for these equations are summarized in the appendix.

Table 2.1 Unknown parameters

Name	Unit	Description
L_n	m	Anode Thickness
L_s	m	Separator Thickness
L_p	m	Cathode Thickness
R_n	m	Anode Particle Radius
R_p	m	Cathode Particle Radius
t^+	-	Transference Number
b	-	Brugman Number
d_2	$m^2 s^{-1}$	Solution Diffusivity
ε_{2n}	-	Anode Solution Volume Fraction
ε_{2s}	-	Separator Solution Volume Fraction
ε_{2p}	-	Cathode Solution Volume Fraction
d_{1n}	$m^2 s^{-1}$	Anode Solid Diffusivity
d_{1p}	$m^2 s^{-1}$	Cathode Solid Diffusivity
k_n	$\left(Am^{-2}\right)\left(mol\ m^3\right)^{1+\alpha}$	Anode Reaction Rate
k_p	$\left(Am^{-2}\right)\left(mol\ m^3\right)^{1+\alpha}$	Cathode Reaction Rate
R_{SEI}	Ωm^2	Anode Film Thickness
c_2	$mol\ m^{-3}$	Initial Solution Concentration
$u_{nref}i$	V	Anode Equilibrium Potential Function: Control Point i
$u_{pref}i$	V	Cathode Equilibrium Potential Function: Control Point i
κi	$\Omega^{-1}m^{-1}$	Solution Conductivity Function: Control Point i

The concentration of Li-ions within the electrolyte $c_2(x,t)$ is governed by Fick's law of linear diffusion combined with an intercalation current density term, J, transferring Li-ions between the solution and solid:

$$\varepsilon_2 \frac{\partial c_2}{\partial t}(x,t) = \frac{\partial}{\partial x}\left(d_2^{eff}\frac{\partial c_2}{\partial x}(x,t)\right) + \frac{1-t^+}{F}J(x,t). \tag{2.1}$$

The above intercalation reaction current density, J, also acts as an input to the dynamics of Li-ion diffusion within the solid. This diffusion occurs at every point in the anode and cathode and can be modeled using a spherical, radially symmetric diffusion law as follows:

$$\frac{\partial c_{1,j}}{\partial t}(r,t) = \frac{d_{1,j}}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c_{1,j}}{\partial r}(r,t)\right). \tag{2.2}$$

where we note that while radial spherical diffusion is an appropriate model for the anode, it is only an approximation for the cathode. We refer the interested reader to papers on both

understanding the behavior of the $LiFePO_4$ cathode [74–76] and agglomerate type models that capture various aspects of the electrode's behavior [77–82]. The intercalation reaction current density, J, is driven by potential differences between the solid and electrolyte solution, as governed by the Butler-Volmer equation:

$$J(x,t) = a_j i_{0,j} \left[\exp\left(\frac{\alpha_{a,j} F}{\bar{R}T} \eta_j(x,t) \right) - \exp\left(-\frac{\alpha_{a,j} F}{\bar{R}T} \eta_j(x,t) \right) \right], \tag{2.3}$$

$$i_{0,j} = k_j \left(c_{1,j}^{\max} - c_{1,j}^{S} \right)^{\alpha_{a,j}} \left(c_{1,j}^{S} \right)^{\alpha_{c,j}} (c_2)^{\alpha_{a,j}}, j = n, p. \tag{2.4}$$

The overpotentials in the above equations, η_j, equal the differences between the solid and solution potentials minus the reference potentials for the main intercalation reaction, which in turn depend on the local SoCs. Mathematically the overpotentials are given by:

$$\eta_p(x,t) = \phi_1(x,t) - \phi_2(x,t) - u_{pref}(x,t), \tag{2.5}$$

$$\eta_n(x,t) = \phi_1(x,t) - \phi_2(x,t) - u_{nref}(x,t) - \frac{J(x,t)}{a_n} R_{SEI}. \tag{2.6}$$

Since potentials and overpotentials described above have dynamics orders of magnitude faster than the Li-ion concentrations, they are assumed to respond instantaneously. The solid potential is governed by Ohm's law with a source term governing the charge transfer due to intercalation:

$$\frac{\partial}{\partial x} \left(\sigma_j^{eff} \frac{\partial \phi_{1,j}}{\partial x}(x,t) \right) - J(x,t) = 0. \tag{2.7}$$

Similarly, the solution potential is governed by Ohm's law, intercalation current density, and the charge carried by the ions in solution:

$$\frac{\partial}{\partial x} \left(\kappa^{eff} \frac{\partial \phi_2}{\partial x}(x,t) \right) + J(x,t) + \frac{\partial}{\partial x} \left(\kappa_D \frac{\partial}{\partial x} \ln(c_2(x,t)) \right) = 0. \tag{2.8}$$

The above system of equations are the DFN model that represent the dynamics of charging and discharging in the Li-ion cell. The boundary conditions for this model are given in the Appendix. When the DFN model is discretized it becomes a system of Differential Algebraic Equations (DAEs), where the differential equations govern the diffusion dynamics and the algebraic equations constrain the potentials and intercalation current accordingly.

2.4 Parameter Set

This section describes the DFN model parameters identified in this chapter, and explains some of the constraints placed on these parameters during identification. The parameters are summarized in Table 2.1. Altogether, 88 parameters are optimized by the GA. Five of these parameters pertain to cell geometry, namely, the anode thickness L_n, separator thickness L_s, cathode thickness L_p, anode particle radius R_n, and cathode particle radius R_p. One may directly measure these quantities by disassembling the cell. However, our aim is to use non-destructive methods for identifying the parameters. Three parameters characterize ion diffusion rates. They include the solid diffusivity d_{1n} in the anode, solid diffusivity d_{1p} in the cathode, and solution diffusivity d_2. One parameter governs the fraction of the intercalation current carried by Li-ions, namely, the transference number t^+. Two parameters govern rate kinetics, namely, the k-rates k_n in the anode and k_p in the cathode. These multiplicatively affect the current densities generated by the electrochemical reactions. One parameter scales the solution conductivity and diffusivity to their effective values, namely, the Brugman number b. Three parameters summarize the cell's porosity, namely, the solution volume fractions ε_{2n} for the anode, ε_{2s} for the separator, and ε_{2p} for the cathode. One parameter captures the effective impedance of the anode-side solid electrolyte interphase layer, namely, R_{SEI}. The last scalar parameter is the initial concentration of the solution, c_2, which we assume to be uniformly constant in space. This variable reflects the amount by which the battery electrolyte is initially lithiated.

In addition to the above 17 scalar parameters, the GA also optimizes three parametric functions in the DFN model. Two of these functions are the equilibrium potential functions, u_{nref} and u_{pref}, of the anode and cathode, respectively. We parameterize these functions using 33 control points each, and use monotonic splines to interpolate between these points [83]. The third function is $\kappa^{eff}(c_2)$, which determines the effective conductivity of the solution as a function of solution concentration. We parameterize this function using five control points spaced linearly from 0 $mol\ m^{-3}$ to 4000 $mol\ m^{-3}$, and interpolate between these control points using conventional cubic splines with natural end conditions [84].

Several constraints are placed on the above parameters in the GA. All of these constraints are related to underlying identifiability issues within the model - each of them improves parameter identifiability by first removing parameters from the optimization problem and then algebraically relating them to parameters remaining within the optimization problem. First, we constrain the capacity of each electrode to equal exactly 2.7 Ah. This constraint provides two key benefits. It creates two 0.2 Ah buffers in each electrode, which improves the numerical stability of the DFN model. These buffers add 0.2 Ah of capacity to the

maximum and minimum values of the electrodes. This allows the GA to tolerate minor local over and under filling of electrodes as it searches for the correct parameter values. Finally, it eliminates the interplay between changes in electrode charge capacity and changes in equilibrium potential functions versus capacity. The second optimization constraint forces the three electrode widths (L_n, L_s, L_p) and the area of the sheet rolled up inside the battery to collectively fit within the volume of the battery cell. Constraining the sheet area is particularly important because it acts as a multiplicative scale factor relating applied current to internal current density. The third constraint sets the volume fractions ε_1 and ε_2 in the anode and cathode to sum to exactly one. The final constraint sets the solid conductivities σ_{1n} and σ_{1p} to equal 100, consistent with [28]. This is justified since both conductivities have absolutely zero effect on the voltage trajectory (so long as they are both positive). Not all of these constraints are fully physically justified: a fact that reflects the presence of underlying identifiability issues. This motivates the Fisher information study in Section 2.7.

2.5 Parameter Optimization Scheme

To identify the DFN model's parameters, we first choose: (i) an optimization objective representing the model's accuracy, and (ii) experimental data sets for which this metric is optimized. The parameter identification objective we use in this chapter is to minimize the L^2 error between the experimentally measured voltage $V(t)$ and DFN-simulated voltage trajectories $\hat{V}(t;\theta)$, for a given battery current trajectory, with respect to the DFN model parameter vector $\hat{\theta}$, *i.e.*,

$$\min_{\vec{\theta}} \int_0^T \left(V(t) - \hat{V}\left(t;\vec{\theta}\right)\right)^2 dt. \tag{2.9}$$

We optimize the above objective using only two of the seven cycles previously mentioned, *Chirp* and *Naturalistic1*, leaving the remaining 5 cycles for model validation. The *Chirp* cycle makes SoC-dependent and rate-dependent parameters easier to identify by sweeping through the full range of battery states of charge at different charge/discharge rates. Furthermore, the *Naturalistic1* cycle makes parameters associated with battery transients easier to identify due to rich frequency content resulting from PHEV drive cycle dynamics.

The GA optimizes the above L^2 error over the course of the *Chirp* and *Naturalistic1* cycles by varying 88 of the DFN model's parameters. GAs are well-suited for such large-scale optimization, especially when gradient information is difficult to obtain analytically or numerically. Figure 2.4 provides a high-level snapshot of this chapter's GA-based DFN parameter identification scheme. The optimization process starts with the selection of inputs

to the DFN model, in this case the *Chirp* and *Naturalistic1* current profiles versus time. We apply these current profiles experimentally to the battery, and measure the resulting voltage output. Next, we initialize the GA to a randomized population, where each population member is a DFN model parameter set. The DFN model is simulated for each population member, and a comparison of the resulting simulated voltage versus experimental data furnishes a fitness value based on the inverse of the L^2 voltage fitting error. Population members are selected for removal at random by a fitness-weighted roulette game. This selection process is elitist, in the sense that the fittest population member is excluded from removal. Once the fitness-based selection is complete, we use binary mutation and crossover operators to create new population members. Parents are chosen randomly for mutation, with a selection probability weighted by their fitness. Mutations occur in a purely random manner, and are not weighted by fitness. The DFN model is then used once more to assign fitness values to the new population members, and the process repeats until convergence to a minimal model fitting error level. The final parameter values are obtained from the fittest member of the population. For further background on GA-based optimization, the reader is referred to [85].

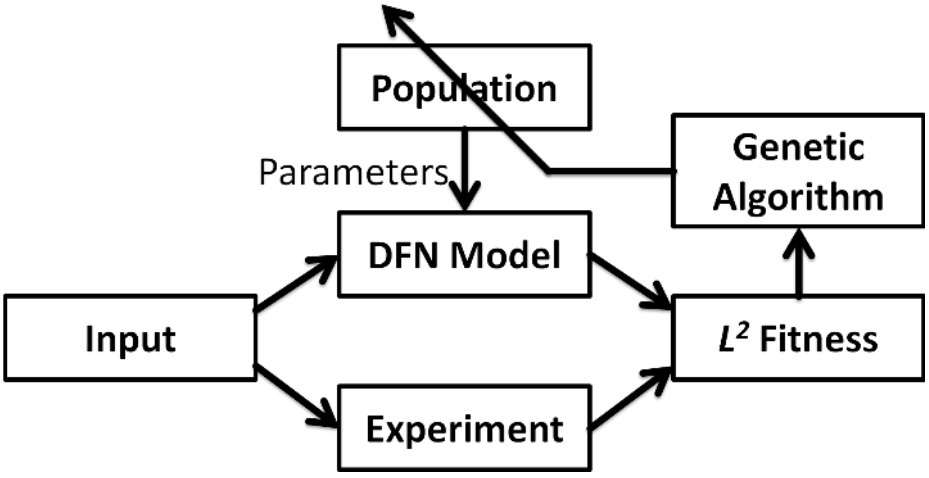

Figure 2.4 Optimizing model parameters via a genetic algorithm

To ensure the convergence of the GA an additional optimization was conducted (henceforth we will refer to this as the tuning optimization and the first optimization as the base optimization). The tuning optimization started with a population centered about the base optimization's fittest population member. Additionally only parameters that were in the identifiable set were taken as variables in this optimization. Recall that to determine this set one needs to already be close to the optimum as the identifiable set is based on local identifiability properties (which is why it could not be computed *a priori* for the base optimization). The tuning optimization converged and slightly improved on the base optimization's result.

The base (tuning) optimization process occurs in the $\mathbb{R}^{88}$ ($\mathbb{R}^{43}$) Euclidean space, with each parameter quantized at 16 bits. This is a very large optimization space, comprising 7.083×10^{423} (1.284×10^{207}) possible parameter sets. We employ two main tools to render these optimizations numerically tractable. First, we use model reduction to accelerate the speed with which the DFN model is simulated, with minimal loss of accuracy. Specifically, we use a Legendre modal coordinate expansion similar to [54], together with algebraic constraint quasi-linearization similar to [12], to improve the DFN model's simulation speed. We apply quasi-linearization directly to the Legendre modal coordinates, allowing for efficient solution of the algebraic constraints imposed by the coupled ϕ_1 and ϕ_2 boundary values problems. This improves computational speed to the point where we are able to simulate the DFN model for each new set of parameters in up to 63 seconds of computation time. Second, we parallelize the GA at the level of simulation function calls, with one server program coordinating multiple quad-core computers, which is a typical server-client arrangement. Custom Java computer code handles server-client information exchange over a TCP/IP network within a MATLAB implementation of the GA and DFN model. Altogether, this use of model reduction in conjunction with parallel processing makes it possible for five quad-core computers (Intel Q8200) to complete the optimization in approximately three weeks.

2.6 Validation Results

One of the major results of this chapter is a set of GA-fitted parameter values that match all five validation cycles; see Fig. 2.5 and Table 2.6. These parameters are based on the tuning optimization which offers slight improvements over a related set of values identified previously [12], which have already been used in two studies of PHEVs: one on power management and one on charge pattern optimization [19, 86]. The values of the fitted parameters are given in Tables 2.3, 2.4, and 2.5. Additional parameters necessary to run the DFN model but not explicitly optimized are listed in Table 2.2. These parameters are implicitly related to the optimization process, in the sense that they are functions of the optimally identified parameters; see Section 2.5 for details. Relative error in voltage and consequently power never exceeds 5% for any of the validation cycles. As shown in Table 2.6, the 50th percentile of voltage error is 15.8mV and the 90th percentile of voltage error is still only 50.5mV.

To examine the accuracy of the optimal parameter fit further, consider the results for the *Naturalistic2* and *LA92x2* validation cycles, which are representative of the set of five cycles.

Table 2.2 Parameters not directly involved in genetic algorithm

Name	Value	Unit
c_{1n}	2.479E+04	$mol\ m^{-3}$
c_{1p}	1.649E+03	$mol\ m^{-3}$
$c_{1n\max}$	2.948E+04	$mol\ m^{-3}$
$c_{1p\max}$	1.035E+04	$mol\ m^{-3}$
T	2.982E+02	K
α	5.000E-01	-
ε_{1n}	3.812E-01	-
ε_{1p}	4.794E-01	-
σ_n	1.000E+02	$m^{-1}\Omega^{-1}$
σ_p	1.000E+02	$m^{-1}\Omega^{-1}$
$Area$	3.108E-01	m^2

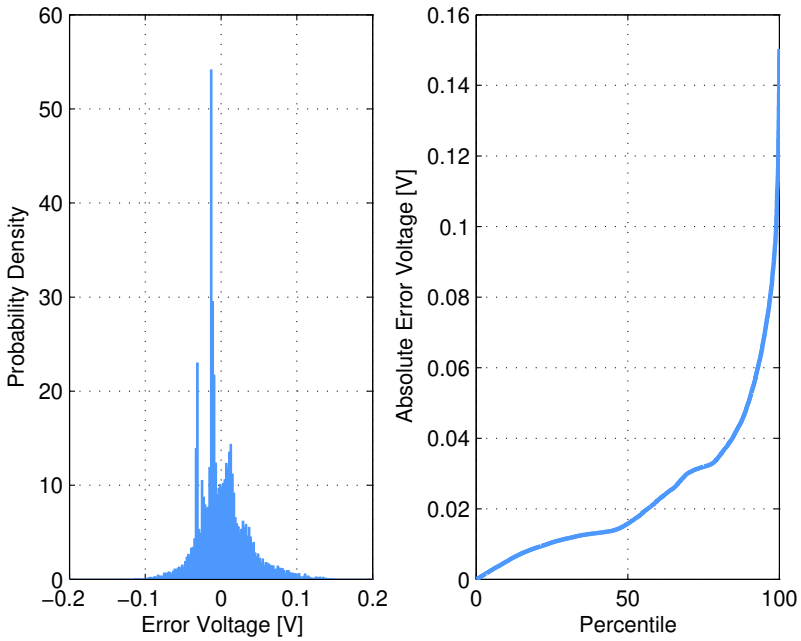

Figure 2.5 Probability density plot of voltage error and the percentiles of absolute volteage error for all five of the validation cycles

Naturalistic2 is based on recorded data from a real driver's evening commute, as opposed to *Naturalistic1*, which is used for fitting and represents a morning commute. Figure 2.6 shows traces of voltage error and Fig. 2.7 shows traces of power error for *Naturalistic2*. The voltage error never exceeds 118.9mV and the 50th percentile of voltage error is 12.5mV. Figure 2.8 presents a probability density plot and a percentile plot of this error.

Table 2.3 First third of optimized parameters

Name	Value	Unit	Variance	95% Confidence interval		
				Min	Max	Rel%
L_n	2.880E-05	m	1.30E-15	2.87E-05	2.89E-05	0.25%
L_s	1.697E-05	m	U	U	U	U
L_p	6.508E-05	m	3.45E-14	6.47E-05	6.54E-05	0.57%
R_n	3.600E-06	m	2.98E-18	3.60E-06	3.60E-06	0.10%
R_p	1.637E-07	m	U	U	U	U
t^+	2.495E-01	-	U	U	U	U
b	1.439E+00	-	1.11E-02	1.23E+00	1.65E+00	14.63%
d_2	6.930E-10	m^2s^{-1}	9.55E-19	-1.26E-09	2.65E-09	281.98%
ε_{2n}	6.188E-01	-	2.62E-02	2.95E-01	9.43E-01	52.33%
ε_{2s}	3.041E-01	-	U	U	U	U
ε_{2p}	5.206E-01	-	4.49E-03	3.87E-01	6.55E-01	25.73%
d_{1n}	8.275E-14	m^2s^{-1}	1.44E-26	-1.57E-13	3.23E-13	289.99%
d_{1p}	1.736E-14	m^2s^{-1}	U	U	U	U
k_n	8.692E-07	$\left(Am^2\right)\left(molm^3\right)^{1+}$	6.38E-20	8.69E-07	8.70E-07	0.06%
k_p	1.127E-07	$\left(Am^2\right)\left(molm^3\right)^{1+}$	U	U	U	U
R_{SEI}	3.697E-03	Ωm^2	6.49E-10	3.65E-03	3.75E-03	1.38%
c_2	1.040E+03	$molm^{-3}$	8.42E+00	1.03E+03	1.05E+03	0.56%
$u_{nref}1$	3.959E+00	V	U	U	U	U
$u_{nref}2$	3.400E+00	V	U	U	U	U
$u_{nref}3$	1.874E+00	V	U	U	U	U
$u_{nref}4$	9.233E-01	V	5.15E-02	4.70E-01	1.38E+00	49.14%
$u_{nref}5$	9.074E-01	V	2.54E-05	8.97E-01	9.17E-01	1.11%
$u_{nref}6$	6.693E-01	V	3.27E-04	6.33E-01	7.06E-01	5.40%
$u_{nref}7$	2.481E-03	V	U	U	U	U
$u_{nref}8$	1.050E-03	V	U	U	U	U
$u_{nref}9$	1.025E-03	V	U	U	U	U
$u_{nref}10$	8.051E-04	V	U	U	U	U
$u_{nref}11$	5.813E-04	V	U	U	U	U
$u_{nref}12$	2.567E-04	V	U	U	U	U
$u_{nref}13$	2.196E-04	V	U	U	U	U

The results for *LA92x2* are similar to those for *Naturalistic2*. Figures 2.9 and 2.10 give the voltage and power trajectories along with their relative and absolute errors. Voltage error never exceeds 150.3mV and the 50th percentile of voltage error is 28.0mV. Figure 2.11 presents probability density and percentile plots of this error.

As a final validation check, we examine whether the voltage errors for the five validation

Table 2.4 Second third of optimized parameters

Name	Value	Unit	Variance	95% Confidence interval		
				Min	Max	Rel%
$u_{nref}14$	1.104E-04	V	U	U	U	U
$u_{nref}15$	3.133E-06	V	U	U	U	U
$u_{nref}16$	1.662E-06	V	U	U	U	U
$u_{nref}17$	9.867E-07	V	U	U	U	U
$u_{nref}18$	3.307E-07	V	U	U	U	U
$u_{nref}19$	1.570E-07	V	U	U	U	U
$u_{nref}20$	9.715E-08	V	U	U	U	U
$u_{nref}21$	5.274E-09	V	U	U	U	U
$u_{nref}22$	2.459E-09	V	U	U	U	U
$u_{nref}23$	7.563E-11	V	U	U	U	U
$u_{nref}24$	2.165E-12	V	U	U	U	U
$u_{nref}25$	1.609E-12	V	U	U	U	U
$u_{nref}26$	1.594E-12	V	U	U	U	U
$u_{nref}27$	1.109E-12	V	U	U	U	U
$u_{nref}28$	4.499E-13	V	U	U	U	U
$u_{nref}29$	2.250E-14	V	U	U	U	U
$u_{nref}30$	1.335E-14	V	U	U	U	U
$u_{nref}31$	1.019E-14	V	U	U	U	U
$u_{nref}32$	2.548E-16	V	U	U	U	U
$u_{nref}33$	1.654E-16	V	U	U	U	U
$u_{pref}1$	5.502E+00	V	U	U	U	U
$u_{pref}2$	4.353E+00	V	1.79E-02	4.09E+00	4.62E+00	6.15%
$u_{pref}3$	3.683E+00	V	1.36E-05	3.68E+00	3.69E+00	0.20%
$u_{pref}4$	3.554E+00	V	1.64E-06	3.55E+00	3.56E+00	0.07%
$u_{pref}5$	3.493E+00	V	9.58E-06	3.49E+00	3.50E+00	0.18%
$u_{pref}6$	3.400E+00	V	8.66E-06	3.39E+00	3.41E+00	0.17%
$u_{pref}7$	3.377E+00	V	8.03E-06	3.37E+00	3.38E+00	0.17%
$u_{pref}8$	3.364E+00	V	8.51E-06	3.36E+00	3.37E+00	0.17%
$u_{pref}9$	3.363E+00	V	1.28E-05	3.36E+00	3.37E+00	0.21%

cycles are correlated with either input current or SoC. Such correlation would suggest failure to accurately represent internal battery resistance or open-circuit potential as a function of SoC, respectively. Table 2.7 presents the R^2 correlation values between voltage error on the one hand and battery current and SoC on the other hand, for each of the validation cycles. None of the validation cycles have voltage errors linearly correlated with input current, which implies that the identified model captures at least internal battery resistance

Table 2.5 Third third of optimized parameters

Name	Value	Unit	Variance	95% Confidence interval		
				Min	Max	Rel%
$u_{pref}10$	3.326E+00	V	1.09E-05	3.32E+00	3.33E+00	0.20%
$u_{pref}11$	3.324E+00	V	1.27E-05	3.32E+00	3.33E+00	0.21%
$u_{pref}12$	3.322E+00	V	1.13E-05	3.32E+00	3.33E+00	0.20%
$u_{pref}13$	3.321E+00	V	1.57E-05	3.31E+00	3.33E+00	0.24%
$u_{pref}14$	3.316E+00	V	1.54E-05	3.31E+00	3.32E+00	0.24%
$u_{pref}15$	3.313E+00	V	1.40E-05	3.31E+00	3.32E+00	0.23%
$u_{pref}16$	3.304E+00	V	1.64E-05	3.30E+00	3.31E+00	0.25%
$u_{pref}17$	3.295E+00	V	1.20E-05	3.29E+00	3.30E+00	0.21%
$u_{pref}18$	3.293E+00	V	6.76E-06	3.29E+00	3.30E+00	0.16%
$u_{pref}19$	3.290E+00	V	1.11E-05	3.28E+00	3.30E+00	0.20%
$u_{pref}20$	3.279E+00	V	1.22E-05	3.27E+00	3.29E+00	0.21%
$u_{pref}21$	3.264E+00	V	1.19E-05	3.26E+00	3.27E+00	0.21%
$u_{pref}22$	3.261E+00	V	1.04E-05	3.25E+00	3.27E+00	0.20%
$u_{pref}23$	3.253E+00	V	6.13E-06	3.25E+00	3.26E+00	0.15%
$u_{pref}24$	3.245E+00	V	7.85E-06	3.24E+00	3.25E+00	0.17%
$u_{pref}25$	3.238E+00	V	1.59E-05	3.23E+00	3.25E+00	0.25%
$u_{pref}26$	3.225E+00	V	1.09E-05	3.22E+00	3.23E+00	0.20%
$u_{pref}27$	3.207E+00	V	5.81E-05	3.19E+00	3.22E+00	0.48%
$u_{pref}28$	2.937E+00	V	1.64E-04	2.91E+00	2.96E+00	0.87%
$u_{pref}29$	2.855E+00	V	1.09E-04	2.83E+00	2.88E+00	0.73%
$u_{pref}30$	2.852E+00	V	1.13E-04	2.83E+00	2.87E+00	0.74%
$u_{pref}31$	1.026E+00	V	U	U	U	U
$u_{pref}32$	-1.120E+00	V	U	U	U	U
$u_{pref}33$	-1.742E+00	V	U	U	U	U
$\kappa 1$	1.050E-01	$\Omega^{-1}m^{-1}$	U	U	U	U
$\kappa 2$	1.760E-01	$\Omega^{-1}m^{-1}$	U	U	U	U
$\kappa 3$	2.190E-01	$\Omega^{-1}m^{-1}$	U	U	U	U
$\kappa 4$	8.166E-02	$\Omega^{-1}m^{-1}$	U	U	U	U
$\kappa 5$	3.014E-02	$\Omega^{-1}m^{-1}$	U	U	U	U

very well. The upper bound on the correlation between model error and SoC is $R^2 = 0.433$ which is the case for the *Naturalistic2* drive cycle. As a point of comparison the correlation between predicted and measured voltage for *Naturalistic2* is R^2 of 0.871, implying that the identified model captures the dependence of battery dynamics on SoC quite well. This SoC is the "system" SoC (as opposed to the chemical SoC which would be calculated based on the quantity of Li in the anode) and is calculated for the battery by integrating and scaling

Table 2.6 Percentile errorrs of voltage [mV]

Drive cycle	Percentile of error [mV]			
	25%	50%	75%	100%
Naturalistic2	9.8	12.5	13.6	118.9
LA92x2	11.7	28.0	41.0	150.3
US06x3	10.6	23.4	41.5	140.0
SC03x4	9.7	21.0	32.7	146.3
UDDSx2	12.0	28.3	33.2	140.9
All Val Cycles	10.4	15.8	31.9	150.3

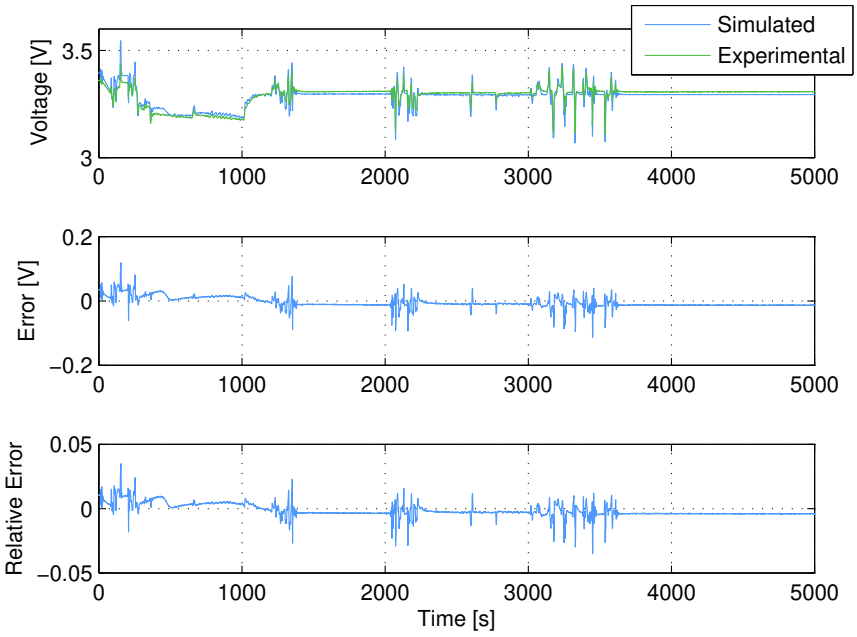

Figure 2.6 Voltage response for *Naturalistic2*

current, knowing that each experiment was initialized at 90% SoC, *i.e.*,

$$SoC_{System} = \frac{\int_0^t I(\tau)\,d\tau}{NamePlateCapacity} + 0.9. \tag{2.10}$$

In summary, this section shows that the DFN model, together with the parameter values identified in this chapter, accurately simulates battery cells under the loading characteristics of PHEVs. This accuracy is evident from the small errors in the voltage and consequently, power traces of the DFN model compared to experimental data. The parameter values in this chapter make it possible to accurately simulate the ANR26650M1A cell for PHEV

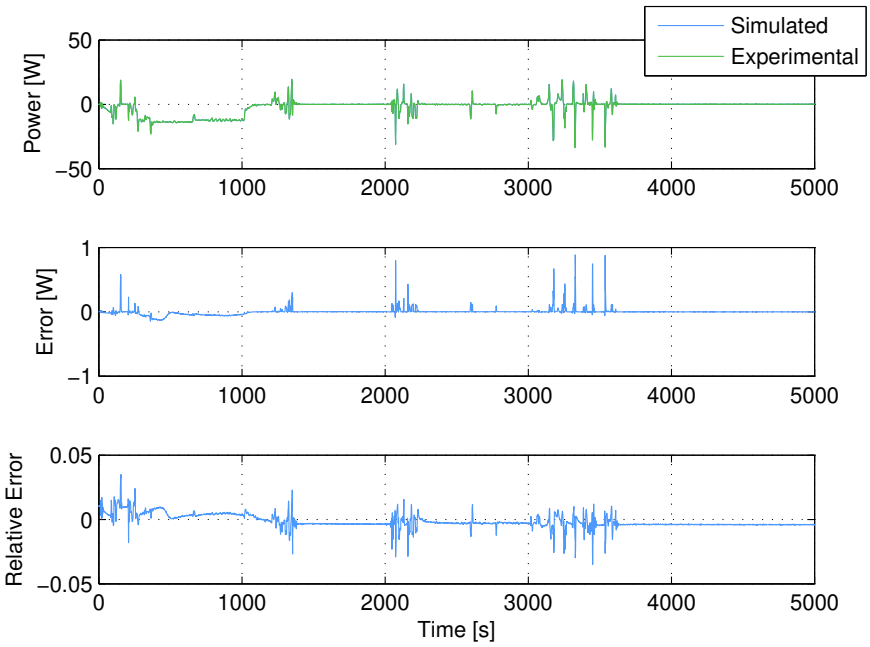

Figure 2.7 Power response for *Naturalistic2*

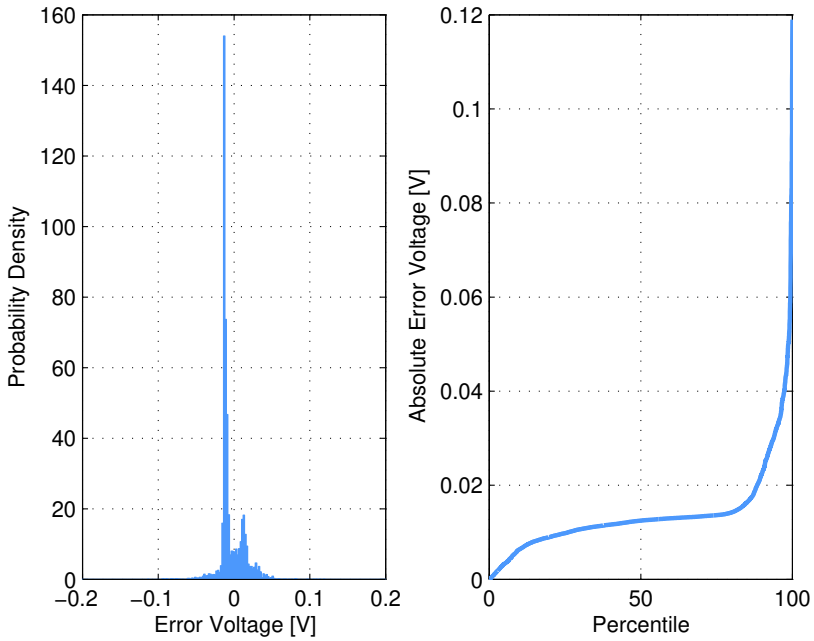

Figure 2.8 Probability density plot of voltage error and the percentiles of absolute voltage error for *Naturalistic2*

applications.

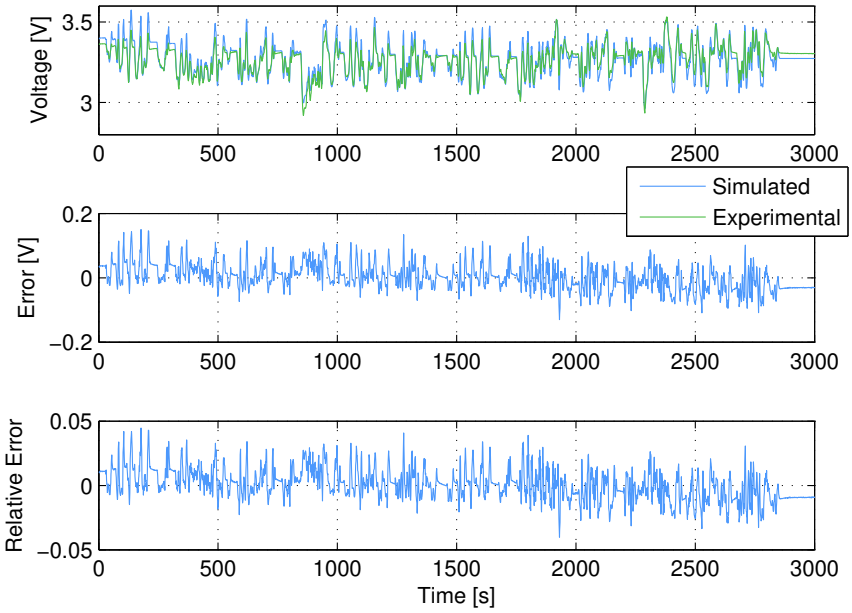

Figure 2.9 Voltage response for *LA92x2*

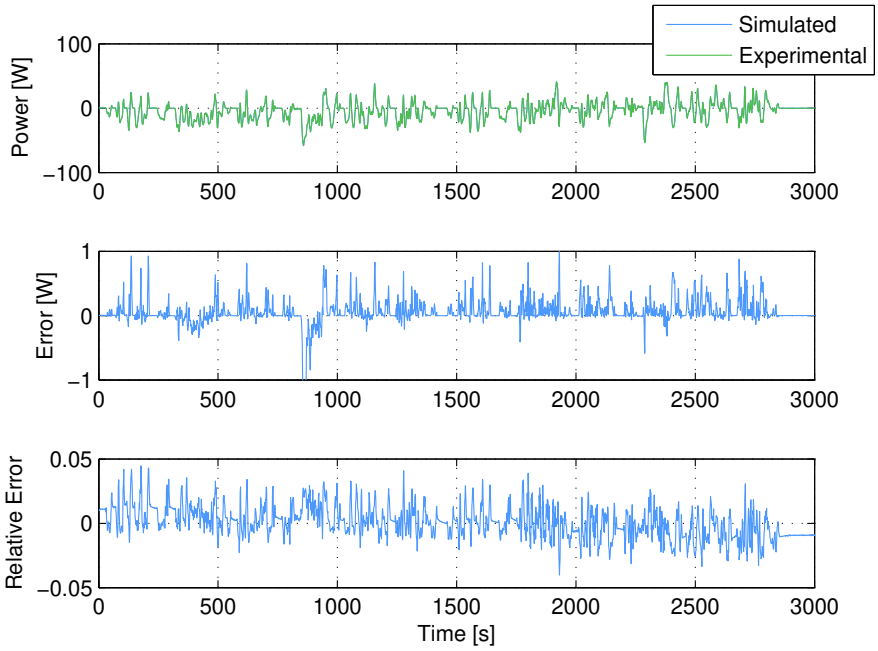

Figure 2.10 Power response for *LA92x2*

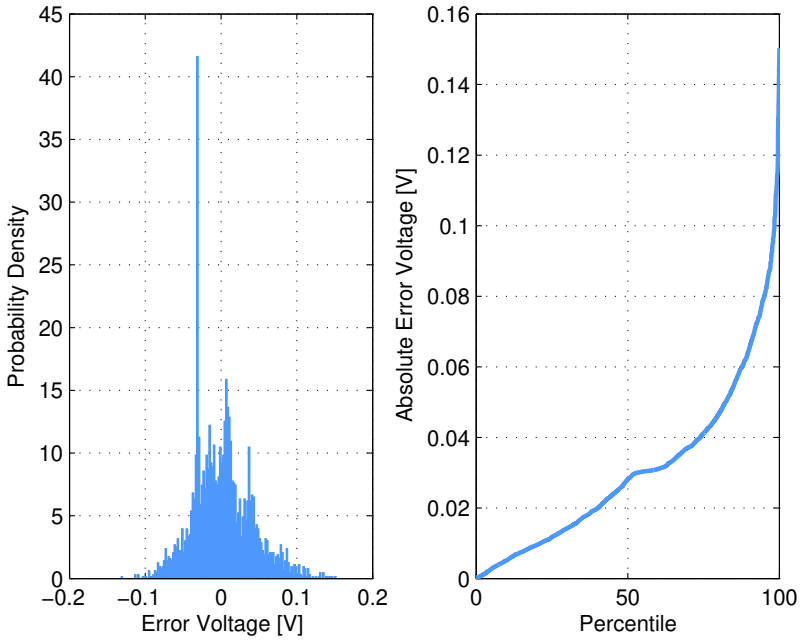

Figure 2.11 Probability density plot of voltage error and the percentiles of absolute voltage error for *LA92x2*

Table 2.7 R^2 coefficients of correlation with voltage estimation error

Drive cycle	I_{app}	SoC
Naturalistic2	0.023	0.433
LA92x2	0.127	0.190
US06x3	0.153	0.151
SC03x4	0.179	0.204
UDDSx2	0.100	0.246

2.7 Fisher Information and Parameter Variance

Section 2.6 of this chapter assesses the degree to which the identified DFN model is able to replicate input-output voltage/current battery cycling behavior. The overarching goal of this section, in contrast, is to evaluate the quality of the model parameter estimates. Previous work by the authors pursues this goal using the identifiability matrix, and shows that while the identified DFN model fits input-output voltage/current data very well, certain model parameters are unidentifiable [12]. This chapter enhances this analysis by quantifying the parameter estimation variance via Fisher information techniques. Fisher information provides the minimum variance for parameter estimation via the Cramér-Rao inequality

[40, 41]. The Cramér-Rao inequality applies to the GA algorithm used herein since we use this algorithm as a maximum likelihood estimator (the GA chooses parameters values to minimize L2). Thus the inverse of the Fisher information matrix is the covariance of estimating the model parameters. Since there is only one output, Fisher information can be calculated by multiplying the identifiability matrix by the voltage sensor's variance. This variance was computed from the voltage error between simulation and experiment for the fitting data sets.

The variance of the estimated parameters is presented in Tables 2.3, 2.4, and 2.5 along with 95% (two standard deviations) confidence bounds and relative error. For each parameter, this relative error is the upper 95% confidence bound minus the parameter's estimated value, divided by this estimated value. Not all of the parameters are identifiable, and those that are unidentifiable are marked with a U. Unidentifiability was determined using the method in [12] where the minimum condition number for the identifiability matrix was taken as 10^{-10}. Lower condition numbers caused unreasonable numerical errors in the inversion of the Fisher information matrix.

The process of partitioning a given parameter set into identifiable versus unidentifiable parameters makes it possible to make quantitative statements regarding these parameters accuracy. Unidentifiable parameters cannot be estimated from experimental measurements. One can only, therefore, estimate their accuracy by comparing their estimated values with the published literature. Identifiable parameters can, in contrast, be estimated from experimental data. Furthermore, the accuracy of these identifiable parameters can itself be methodically estimated from the Fisher information matrix, provided one can associate *a priori* assumed levels of error with the unidentifiable parameters. The remainder of this chapter demonstrates the process of methodically calculating the estimation errors associated with the identifiable parameters, under the optimistic assumption that the unidentifiable parameters are known *a priori*. Our goal, here, is to demonstrate the value of identifiability analysis for the DFN model, rather than to quantify DFN parameter estimation errors exactly. It is very important to note, here, that the choice of which parameters are identifiable vs. unidentifiable has a significant impact on the identification errors computed by this process. One must therefore be very vigilant when making this choice/partitioning.

Of the eleven identifiable scalar parameters six are estimated with good accuracy (relative error $<2\%$). These are associated with geometry: L_n the anode width, L_p the cathode width, R_n the spherical radius in the anode, k_n the k-rate in the anode, R_{SEI} the solid electrolyte interface resistance, and c_2 the initial concentration of Li in solution. The remaining five scalar parameters all have relative errors greater than 25%. It is important to note that these variances correspond to the case where one attempts to identify all model parameters

simultaneously, with the unidentifiable values fixed. If one knows some parameters with certainty and can therefore estimate a smaller subset of the DFN model parameters, the variance in these parameters will be lower (or at least the same). For example, if one is designing a state of health estimator whose sole goal is to estimate R_{SEI} assuming all other DFN model parameters to be known, the variance in estimation would be 4.387E-014, and the relative error in estimating R_{SEI} would decrease from 1.38% to 0.0113%. These results are important, because they: (i) quantify the errors in the parameters identified herein, (ii) highlight the difficulties in estimating specific parameters solely through voltage and current time traces, and (iii) underscore the importance of examining the identifiability of all DFN model parameters, not just a subset of those parameters.

To provide further insights into the DFN model's parameter identifiability, Figures 2.12 and 2.13 present the estimates of the cathode-side and anode-side equilibrium potentials versus SoC, along with their 95% confidence bounds. For plotting purposes, the unidentifiable parameters have confidence bounds at $\pm 2\sigma$. In the model these equilibrium potential functions are represented by monotonic cubic splines in terms of SoC. Here we plot these functions as piecewise linear since the confidence bounds are only for the control points. Between the two equilibrium potential functions, u_{pref} has much less variance than u_{nref}. Specifically, the estimation of u_{pref} exhibits low variance, where the confidence bounds correspond to relative errors less than 1% for 0% SoC to 95% SoC (where this SoC does not include the buffers). In contrast, most of u_{nref} is unidentifiable, and even the identifiable control points still have very high variances. The equilibrium potential functions provide a good example of the effects of assuming that the unidentifiable values are correct. Since u_{pref} is very dependent on u_{nref}, and u_{nref} is largely incorrect, u_{pref} is mostly identifiable but to make the u_{pref} values accurate one needs to plug in accurate values for the u_{nref} first. The conductivity function is completely unidentifiable indicating that the interplay between solution concentration and conductivity could not be determined through these experiments.

Parameter identifiability and variance in estimation are the function of several important factors. First, they are a function of which parameters are being identified and which are already assumed known. This leads to cases where parameter estimation errors can be improved dramatically by changing the number of known parameters versus unknown ones (as shown previously in the case of R_{SEI}). Parameter identifiability also depends on the values of the parameters after they have been fit to the data. This is due to Fisher information being a local quantity in the parameter space. In general, this makes it impossible to determine which parameters will be identifiable *a priori*. Finally we note that the experiments themselves can greatly affect parameter identifiability. Both the structure of the battery experiment (including placement of sensors and actuators) and the experiment's

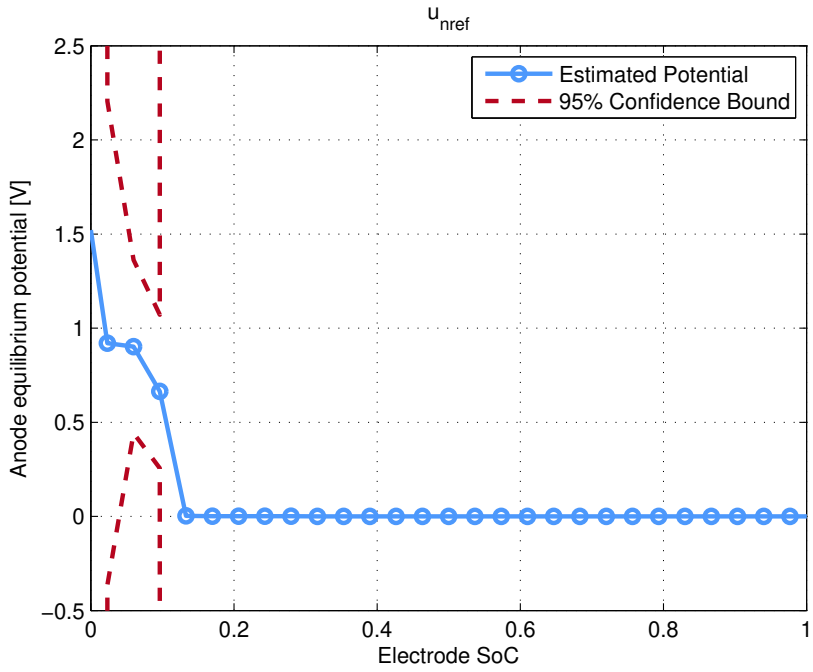

Figure 2.12 Estimated anode equilibrium potential u_{nref} with 95% confidence bounds.

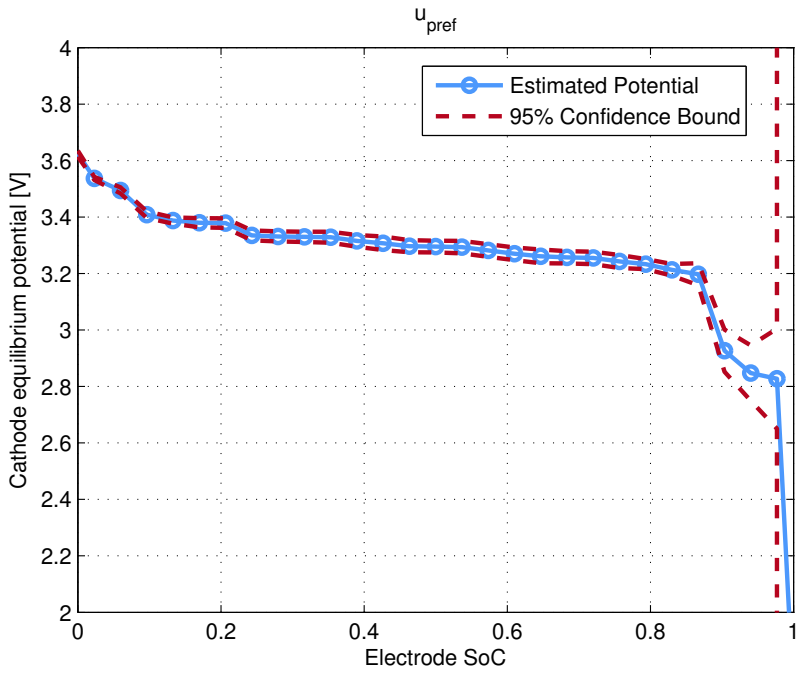

Figure 2.13 Estimated cathode equilibrium potential u_{pref} with 95% confidence bounds.

trajectory can affect parameter identifiability. As an example of structure, if our cell had a third electrode then we would have been able to measure two voltages likely improving our ability to identify u_{nref} and u_{pref} simultaneously. As an example of the experimental

trajectory's impact, the *Natuarlistic2* and *Chirp* cycles are different in terms of the battery dynamics they excite, and therefore different in their impact on parameter identifiability. All of these factors underscore that matching input-output data is not enough to guarantee that the model parameter values are physically accurate.

2.8 Summary and Conclusion

This chapter uses a GA to match the Doyle-Fuller-Newman battery model's voltage predictions to experimental measurements, for given input current profiles. We optimize 88 parameters of the DFN model, including parameterizations of the anode and cathode equilibrium potential functions and the solution conductivity function. The end result is a set of parameter values for the DFN model that predicts cell voltage and power with 5% relative error for all of the validation data sets examined in this work. All of these validation data sets are based on simulated Plug-in Hybrid Electric Vehicle battery pack currents that exhibit high charge/discharge rates and are highly transient in nature. For all of the validation cycles aggregated together, the 50th percentile of voltage error 15.8mV, and the 90th percentile of voltage error is still only 50.5mV. This high level of accuracy justifies the use of the DFN model for the lithium-iron-phosphate ($LiFePO_4$) cathode chemistry examined in this work. In fact, the parameters values identified herein have already been used in two studies involving PHEVs [19, 86]. Additionally, this chapter presents some of the computational logistics involved in using a GA for parameter identification.

The identification procedure used herein makes it possible to find a set of parameter values for the DFN model noninvasively. Unfortunately, this noninvasiveness causes some parameters to be unidentifiable or have a large estimation uncertainty. While this does not affect the accuracy of the model response, it does mean that one must be careful when using these parameters in other contexts.

Chapter 3

Optimal Experimental Design for Modeling Battery Degradation

3.1 Introduction

In recent years there has been increasing interest in vehicle electrification. Electrification has the potential to both decrease green house gas production and improve energy independence. However, to realize these benefits, Plug-in Hybrid Electric Vehicle (PHEVs) must have battery packs that do not rapidly degrade. Accurate modeling of battery health can help both manage and mitigate battery degradation in a variety of ways. For example, battery health modeling allows for more intelligent PHEV design decisions regarding battery size and drive train topology, helping to minimize cost and improve reliability. Additionally, battery health modeling enables health conscious control, extending battery life through judicious use of real-time battery and engine management [8]. Finally, battery health modeling makes feed-forward State-of-Health (SoH) estimation possible - improving online estimation of available power and capacity. Clearly, accurate battery health modeling is a critical tool for the design and control of PHEV battery packs. However, battery health modeling for PHEVs is a challenge due to the aggressive nature of battery pack use.

This challenge emphasizes the importance of experiments for both obtaining and validating battery health models. Unfortunately, conducting battery health experiments can be extremely costly in terms of time, person-hours, and equipment; efficient use of experimental resources is critical. This chapter bridges an important gap between the Optimal Experimental Design (OED) and battery experimental/modeling literatures allowing us to maximize information gained from experiments subject to constraints on time and expense.

To this end, we investigate a specific case of designing and conducting a battery health modeling experiment using 14 $LiFePO_4$ cells. This experiment is optimally designed to identify model parameters under given experimental resource constraints. Results include: a model of battery capacity fade based solely on voltage and current data, demonstration of

battery health dependence on voltage, and a lack of power fade (at 100% State-of-Charge, SoC) under the cycling conditions. The success of this case suggests a generalization: the Optimal Battery Health Model Experiment (OBHME) framework. This framework allows one to apply static OED methods to a large class of battery health modeling experiments. OBHME uses Fisher information and the Cramér-Rao bound to optimally guide the selection of experimental trials. We believe this case and associated framework show the broad applicability and utility of OED when considering battery health modeling.

The remainder of this chapter is organized as follows. A literature review follows the introduction, giving a focused review of both $LiFePO_4$ battery health experiments and a general overview of OED. This is followed by a section which applies OED to health modeling for PHEV type batteries. This case involves both a theoretical OED part regarding applying static OED to dynamic health models and an experimental part in which the batteries undergo laboratory cycling. Experimental results follow, beginning with model fitting and continuing to general observations regarding the degradation of these battery cells. The OBHME framework follows this, generalizing the procedure and suggesting alternative design choices. A discussion section follows, focusing on specific details we would modify in future experiments based on experience gained. The final section presents several conclusions drawn from this work.

3.2 Literature Review

3.2.1 $LiFePO_4$ Battery Health Experiments

While many groups have conducted battery health experiments, we focus on two that use the same $LiFePO_4$ battery type as in this chapters's case study [14, 39]. In [39] the batteries undergo CCCV cycling with varying current rates, depths of discharge, and temperatures. They fit a health model to the experimental data, dependent on charge processed, temperature, and maximum C-rate. This model provides useful insights (especially regarding temperature effects), but is not control oriented due to the inclusion of maximum C-rate as a parameter. A second group cycles these batteries under conditions related to PHEV drive cycles [14]. Five different cycles with drive and Vehicle-to-Grid (V2G) portions are taken as a set of representative behavior. Their analysis of the experiments results in a control oriented model that is a function of energy processed and scalar that depends on if the cell is undergoing driving or V2G. Our chapter improves on the existing experimental methods by demonstrating the applicability of OED to battery health experiments.

3.2.2 Overview of Optimal Experimental Design

The OED literature is broadly divided into experiment design for static and dynamic models. This chapter focuses on the static case and here we introduce the reader to relevant literature. A classic OED text is that of Fedorov [87]. An excellent first introduction to OED is presented by Atkinson, Donev, and Tobias which covers nearly all that the practitioner would need to design an experiment [15]. For a more in depth study of Fisher information and the Cramér-Rao bound one can examine Cover and Thomas, which does an excellent job presenting these mathematical concepts [40]. A thorough discussion of optimization of the nonlinear case is presented by Walter and Pronzato [16]. OED provides a variety of mathematical tools for improving results and decreasing expense of model based experiments.

3.3 Battery Health Modeling Experiment

This section presents an optimal battery health modeling experiment for $LiFePO_4$ cells. A later section discusses the OBHME framework - a generalization of this specific case. This case and the general framework share the same major steps: model selection, experiment design, and realization. In model selection one chooses a health metric (output), input(s), and form. This model guides the experiment design, which selects trials based on optimizing information gathered regarding the model parameters. Experiment design involves considering what trials are possible, generating and regressing their input trajectories, and using an optimization to select an experimental trial set. In realization, the laboratory experiment collects data which is then used to estimate the model parameters. The remainder of this section discusses the experiment and introduces a variety of concepts related to the OBHME framework.

3.3.1 Model Selection

Battery Health Metric

For this experiment two different battery health metrics are considered: capacity fade and power fade. For PHEV applications capacity fade is related to how far the vehicle can drive without charging and power fade is related to maximum available electric power. In this chapter's experimental section we discuss measuring these metrics in a laboratory. For the

modeling work later in the chapter we focus exclusively on the capacity fade health metric as conclusive power fade has not been observed in our experimental work.

Choice of Model Inputs

Battery health can depend on a wide variety of factors. However, for this work our goal is to create a control oriented battery health model. Thus, we focus solely on two factors readily accessible to battery management systems: voltage and current. This choice of inputs ensures that the model will be applicable to many real-time control applications. Our interest in voltage is largely due to it being a proxy for battery SoC. We would have used SoC directly, however, it is difficult to accurately and robustly regulate battery SoC during long term health experiments. As an alternative we use voltage as an input and add longer time holds (1800 sec) to the float charge/discharge segments of the CCCV cycles. These time holds allow the battery to better equilibrate towards the relaxed open circuit voltage, which is closely related to SoC. As will be shown later, the use of voltage and current as inputs allows one to make reasonable predictions about battery degradation.

Battery Health Model Form

The inputs and output are connected by the model form. An effective model form considers *a priori* knowledge (if available) in addition to the model's intended application. In this work a black box model is used due to uncertainty in the underlying degradation process, leading to a model form based on regression. This regression form is based on our specific interest in the battery health dependence on voltage and current polarity. This potential voltage dependence is interesting because it affects optimal PHEV charge scheduling (scheduling charging at different times changes the PHEV battery rest voltage). The effect of current polarity is important because the battery degradation rate may change during charging and discharging.

Based on these interests and a desire to keep the number of model parameters small we arrived at the following model form, referred to as the "Asymmetric" model:

$$\begin{aligned}\dot{h}(I_+, I_-, V) =& \beta_1 + \beta_2 I_+ + \beta_3 I_- + \beta_4 V + \beta_5 I_+^2 + \beta_6 I_-^2 \\ &+ \beta_7 V^2 + \beta_8 I_+ V + \beta_9 I_- V + \beta_{10} V^3.\end{aligned} \tag{3.1}$$

Where I_+ is the current charging the cell, I_- is the current discharging the cell, V is the cell terminal voltage, and h is the battery health. This model has different behavior in

charging versus discharging, a cubic dependence on voltage, and can age under zero current conditions. To keep our exposition more abstract we focus on a generalized form of the model in Eq. 3.1:

$$\dot{h}(\vec{x}) = \sum_{i=1}^{m} \beta_i f_i(\vec{x}). \tag{3.2}$$

Where f_i maps the model inputs $\vec{x}$ to real numbers. This model form is Linear in its Parameters (LP) and has no autoregressive component. An additional model of this form, referred to as the "Symmetric" model, is:

$$\begin{aligned} \dot{h}(I,V) =& \beta_1 + \beta_2 \|I\| + \beta_3 V + \beta_4 \|I\|^2 + \beta_5 V^2 \\ &+ \beta_6 \|I\| V + \beta_7 V^3, \end{aligned} \tag{3.3}$$

Where, unlike Eq. 3.1, this model does not distinguish between current polarities. With the model in Eq. 3.1 selected, the experiment is now ready to be optimized.

3.3.2 Experiment Design

Possible Trial Set

Trial selection is the key to optimizing experiments. This selection begins with the set of all possible trials, Ξ. For battery health experiments members of Ξ are typically rules for cycling an individual battery for a period of time. We use CCCV cycling to generate a wide variety of robust inputs for long term cycling. These CCCV cycles are described by the archetype cycle in Alg. 1.

Three tunable parameters are used to construct Ξ: V_{min}, V_{max}, and I_{max}, with values:

$$V_{min} \in \{2.0V, 2.1V, \cdots, 3.5V\} \tag{3.4a}$$

$$V_{max} \in \{2.1V, 2.2V, \cdots, 3.6V\} \tag{3.4b}$$

$$I_{max} \in \{0.5C, 1.0C, \cdots, 2.5C\}. \tag{3.4c}$$

This results in a total of 680 unique cycles. The next step of experiment design uses these rules to generate (approximate) input trajectories.

Algorithm 1 CCCV Cycle

Require: $V_{min}, V_{max}, I_{max}, I_{trickle}, t_{hold}$
  loop
    while $(V < V_{max})$ **do**
      Constant Current Charge at I_{max}
    end while
    $t = Time$
    while $(\|I\| > I_{trickle})$ And $((Time - t) < t_{hold})$ **do**
      Constant Voltage Float at V_{max}
    end while
    while $(V > V_{min})$ **do**
      Constant Current Discharge at I_{max}
    end while
    $T = Time$
    while $(\|I\| > I_{trickle})$ And $((Time - t) < t_{hold})$ **do**
      Constant Voltage Float at V_{min}
    end while
  end loop

Input Trajectory Generation

Input trajectories are generated using electrochemical battery simulation. Specifically, we simulate voltage and current trajectories with the Doyle-Fuller-Newman (DFN) cell model [2–4]. The parameter values of this model are obtained from a previous study that fit the DFN model to a $LiFePO_4$ battery [12]. To better approximate quasi steady-state conditions, the third cycle simulated is used. In spite of identical charge and discharge current limits, the battery exhibits asymmetric behavior in charging and discharging, due to its electrochemical nature. Next, we regress these input trajectories, encapsulating their data for the optimization algorithm.

Input Trajectory Regression

Regressor vectors encapsulate how trial input trajectories affect the estimation information gathered. This encapsulation converts the dynamic form of the model into a static form appropriate for optimization. Since the health measurements happen intermittently we only collect discrete measurements of change in health over a time interval. Consider integrating

the model over the time period between health tests:

$$\int_0^{t_f} \dot{h}\left(\vec{x}\right) dt = \int_0^{t_f} \sum_{i=1}^{m} \beta_i f_i\left(\vec{x}\right) dt \tag{3.5a}$$

$$\Delta h = \sum_{i=1}^{m} \beta_i u_i, \tag{3.5b}$$

where:

$$\Delta h \equiv \int_0^{t_f} \dot{h}\left(\vec{x}\right) dt \tag{3.6a}$$

$$u_i \equiv \int_0^{t_f} f_i\left(\vec{x}\right) dt. \tag{3.6b}$$

Here Δh is the change in battery health and each u_i is a regressor associated with the function f_i. Unique trials can be indexed with j and the model can be rewritten as follows:

$$\Delta h_j = \sum_{i=1}^{m} \beta_i u_{ij} = \vec{\beta} \cdot \vec{u}_j. \tag{3.7}$$

This specific form will be important for the optimization algorithm. For now it provides a formula to compute the $\vec{u}_j$ regressor vectors:

$$\vec{u}_j = \left(\int_0^{t_f} f_1(\vec{x}(t))dt, \cdots, \quad \int_0^{t_f} f_m(\vec{x}(t))dt \right)^T / t_f. \tag{3.8}$$

These regressor vectors will form the rows of the Fisher information matrix which will be optimized through the selection of experimental trials.

Experiment Optimization

Let us consider a natural way to compile the experimental data. One can create a matrix equation by stacking each Δh_j into a vector of health measurements and stacking each $\vec{u}_j^T$ into a corresponding matrix row:

$$\begin{pmatrix} \Delta h_1 \\ \Delta h_2 \\ \vdots \\ \Delta h_n \end{pmatrix} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ u_{n1} & \cdots & \cdots & u_{nm} \end{bmatrix} \vec{\beta}. \tag{3.9}$$

This can be rewritten as:

$$H = U\vec{\beta}, \tag{3.10}$$

where H and U are the appropriate matrices.

The Fisher information matrix F is then defined as:

$$F = \frac{U^T U}{\sigma^2}, \tag{3.11}$$

where the measurement error of experiment is assumed be a normal distribution with zero mean and variance σ^2. The inverse of F is the best possible covariance one can achieve when estimating $\vec{\beta}$ due to the Cramér-Rao bound [40]. Furthermore by using least squares, an unbiased estimator, this bound is achieved. Thus the covariance in estimating $\vec{\beta}$ is:

$$\text{covar}\left(\vec{\beta}\right) = \sigma^2 (U^T U)^{-1}. \tag{3.12}$$

The importance of Eq. 3.12 is twofold. Firstly it shows that the response of the trials does not affect the covariance for estimating $\vec{\beta}$ so it can be ignored for the experimental design (this occurs because of the LP assumption). Second one can influence the covariance directly through the selection of experimental trials. For our work we assume that σ^2 is fixed, but in practice one can influence this value as well, by using better sensors or testing methods. We now describe the DETerminant MAXimizing algorithm (DETMAX), an optimization method for selecting experimental trials.

The objective of OED is to select trials that allow one to estimate $\vec{\beta}$ with minimum covariance (by a given metric). D-optimum is a very common metric in OED and is based on maximizing the determinant of the Fisher information matrix. This in turn, minimizes the determinant of its inverse, the covariance matrix. Minimizing the determinant of the covariance matrix minimizes the product of its eigenvalues. Geometrically if one investigates the confidence region of the $\vec{\beta}$ estimate this results in the smallest possible (by content) hyper ellipsoid [15].

DETMAX attempts to find a subset of fixed size q contained in Ξ that maximizes the value of $\det(F)$. It does this by selecting regressor vectors from Ξ to create the experiment matrix U. Mathematically DETMAX attempts to:

$$\max_{\vec{\alpha}} \det \frac{U^T U}{\sigma^2} \tag{3.13a}$$

$$U = \begin{bmatrix} \vec{u}_{\alpha_1} & \vec{u}_{\alpha_2} & \cdots & \vec{u}_{\alpha_q} \end{bmatrix}^T, \tag{3.13b}$$

subject to:

$$\alpha_i \in \{1, \cdots, p\} \tag{3.14a}$$

$$i \neq j \Rightarrow \alpha_i \neq \alpha_j \tag{3.14b}$$

$$\vec{u}_i \in \Xi. \tag{3.14c}$$

DETMAX attempts to select locally D-optimum trial subsets contained in Ξ from the set of all possible experiments. Here local optimality is in the sense of swapping - one cannot swap any of the experiments in the set with any other possible experiment and increase $\det(F)$. In practice we initialize this algorithm with many random sets to help ensure that the resulting set is closer to the global optimal. DETMAX has a few variations, the version we use is as follows. First DETMAX computes $\det(F)$ for the current set. Then it iterates over all possible swaps of trials between this set and the set of all possible experiments. For each swap a new $\det(F)$ is computed and stored. The swap which improves $\det(F)$ the most is then performed and the process repeats. DETMAX concludes when no swaps improve on $\det(F)$ [15, 16].

DETMAX is ideal because we have a low number of battery cells and do not want to complicate the experiment by switching cycles after each health test. We decided to batch the experiment using 10 cells for the first batch and 4 cells for the second batch (DETMAX can be used to optimize batched experiments) [15, 16]. The second batch is used solely for validation and is not included in the model fitting exercise. With the experimental trials selected our attention now turns to the laboratory experiment.

3.3.3 Realization

Laboratory Experiment

The batteries are repeatedly cycled based on the optimized set of trails (see Tab. 3.1). Thus far, the experiment has been carried out for 429 days with health data being collected approximately every two weeks. For health, we measure capacity fade as discharge capacity in Amp-Hrs during 0.5C CCCV cycling. These cycles repeat four times during each health test and the first value is removed to avoid memory (hysteresis) effects. Power fade is measured by a 2.5C constant current draw for 15 seconds when the batteries are fully charged. All of the tests are conducted at ambient room temperature. We set the cycle parameters $I_{trickle}$ to 50mA and t_{hold} to 1800 seconds. The cycling and health measurements are conducted using an Arbin BT-2000 battery cycler.

Table 3.1 Experimental trials as determined by DETMAX

Batch	V_{min}	V_{max}	I
1	3.0V	3.3V	2.5C
1	2.6V	3.6V	1.5C
1	2.0V	3.5V	0.5C
1	3.0V	3.1V	2.5C
1	3.2V	3.5V	2.5C
1	2.0V	2.1V	0.5C
1	2.4V	2.5V	0.5C
1	2.2V	3.5V	2.5C
1	2.0V	3.2V	2.0C
1	3.5V	3.6V	0.5C
2	3.4V	3.6V	2.5C
2	3.0V	3.4V	2.5C
2	2.0V	3.1V	2.5C
2	2.0V	3.6V	1.5C

This version of the cycler has 32 independent channels that can each source or sink up to 15A. Each of these channels can operate from 0V to 10V (but not negative). 14 of the 32 channels were dedicated to this experiment. All of the cycling was conducted at room temperature.

The cells cycled are manufactured by A123 Systems; the model number is ANR18650M1A. These cells are interesting because they are scaled down versions of those intended for use in PHEVs and are able to continuously discharge at 30C giving them excellent rate capability. They have a rated capacity of 1.1 Amp-Hrs, a nominal voltage of 3.3V, and a voltage range from 2.0V to 3.6V.

Equipment down time is addressed by back filling voltage data with constant values (current is taken as zero during these periods). This is an approximation, but since there is a limited amount of downtime and the batteries relax within a few hours this is considered appropriate.

As the battery experiment is conducted the model input trajectories (voltage and current) are recorded for each battery. This data is mapped to the regressor vectors associated with changes in battery health. These, along with the health data, are then used to calculate the parameters for the battery health model. In the next section we discuss the specifics of parameter estimation.

Parameter Estimation

With all of the experimental data recorded estimating $\vec{\beta}$ is straightforward. One begins by compiling the recorded the health measurements and associated regressor vectors. These can then be arranged to form the matrix equation in Eq. 3.9. This, in turn, can be rewritten as Eq. 3.10, leading to the least squares solution of $\vec{\beta}$ as follows:

$$\vec{\beta} = \left(U^T U\right)^{-1} U^T H. \tag{3.15}$$

The next section discusses the specific results of this model fitting work.

3.4 Experimental Results

3.4.1 Model Fitting Results

Our plan for this model fitting work was to use both power and capacity fade as health metrics. However, as is discussed in the next subsection, power fade was negligible during the experiment (at least for our method of measurement). Thus, we only consider the capacity health metric. As mentioned before this experiment divides the cells into two batches, the first for identification and the second for validation (the batches are conducted concurrently). Only data collected from the first batch is used in the estimation of $\vec{\beta}$.

Initially we attempted to fit the Asymmetric model, given in Eq. 3.1. We then considered the Symmetric model, given in Eq. 3.3, which differs from the Asymmetric model only by not distinguishing between positive and negative current. It was found that the Symmetric model predicts the validation slightly better than the Asymmetric model, which indicates that not all of the parameters in the Asymmetric model are needed for prediction. Figure 3.1 presents the Symmetric model prediction results for the validation cycles. These predictions are initialized with the first set of collected health data and then simulated open-loop. Figure 3.2 summarizes the prediction errors for the validation cycles with a histogram and percentile plot.

To ensure that each of the β parameters is converging in a meaningful manner we observed how they changed after each health data set was collected. Health data was collected approximately every two weeks. Figure 3.3 shows how the β values changed with each iteration of data collection. We see that they appear to converge smoothly, which helps to confirm that they are not being over fitted. Next we consider additional experimental observations that are not directly related to model fitting.

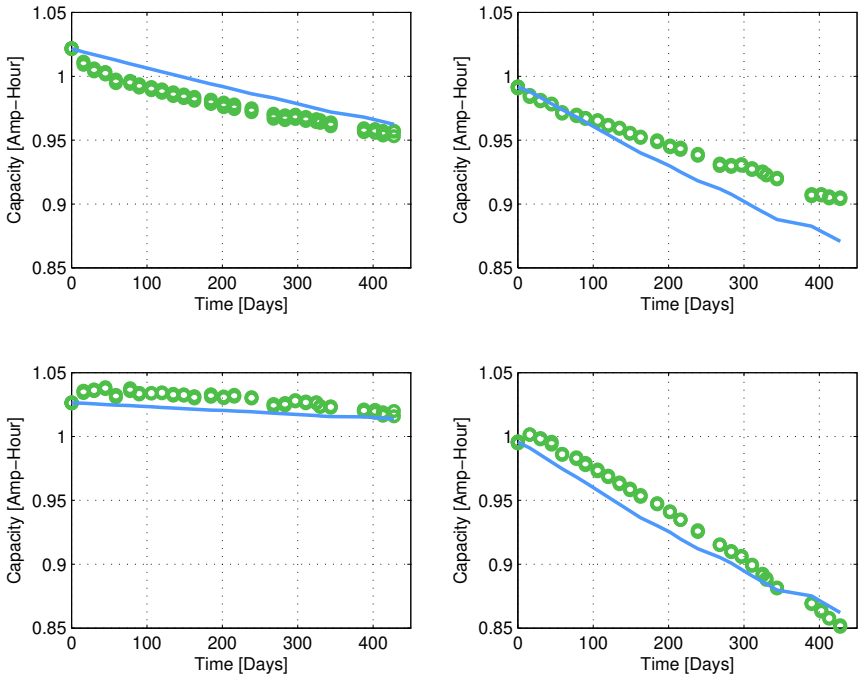

Figure 3.1 Symmetric model predictions of validation cycles. Curve is the Symmetric model and circles represent experimental health measurements

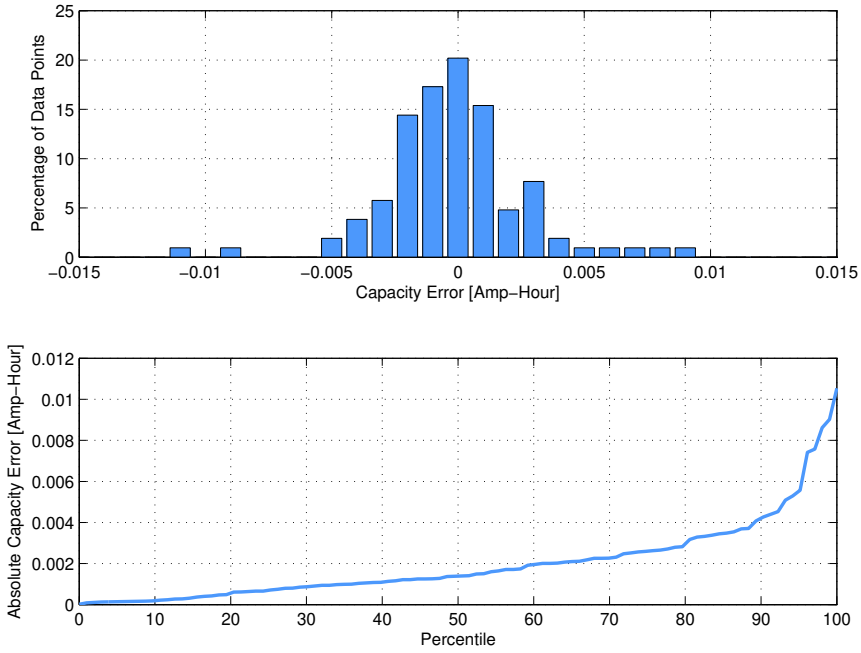

Figure 3.2 Aggregated errors of validation data for the Symmetric model.

3.4.2 Results Independent of Model Fitting

This section presents two experimental results that are independent of model fitting. The first is the dependency of battery aging on voltage, in contrast to [14], where battery degradation

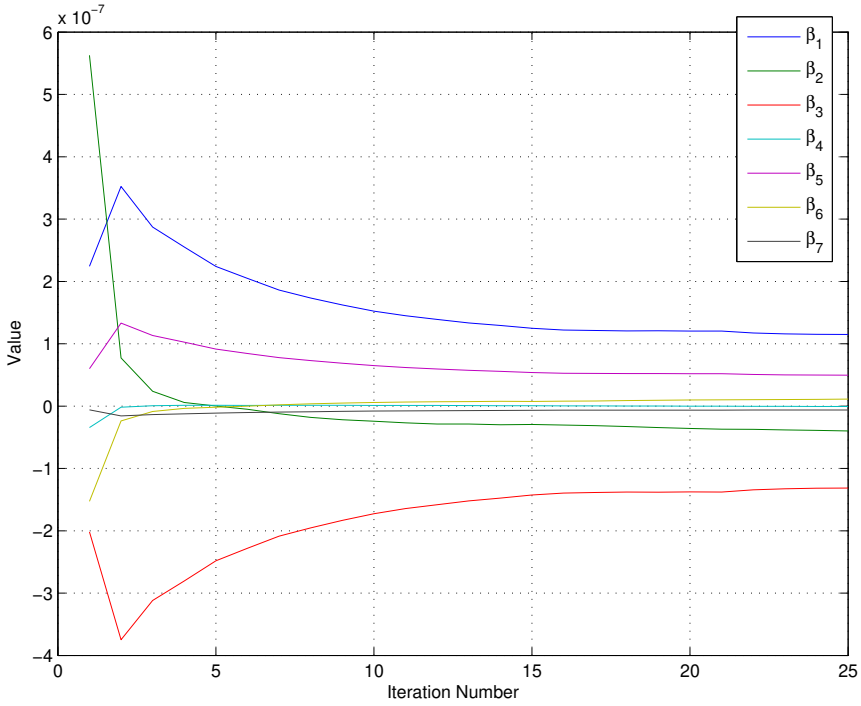

Figure 3.3 Convergence of the 7 β parameters. Each iteration corresponds to a new health test (health tests occur approximately every two weeks).

Table 3.2 Estimated parameters for the Symmetric model

Parameter	Value	Units
β_1	1.1484×10^{-7}	$Amp \times Hour \times Sec^{-1}$
β_2	-3.9984×10^{-8}	$Hour \times Sec^{-1}$
β_3	-1.3158×10^{-7}	$Amp \times Hour \times Sec^{-1} \times Volt^{-1}$
β_4	-5.5487×10^{-10}	$Amp \times Hour \times Sec^{-1} \times Amp^{-1}$
β_5	4.9680×10^{-8}	$Amp \times Hour \times Sec^{-1} \times Volt^{-2}$
β_6	1.1166×10^{-8}	$Hour \times Sec^{-1} \times Volt^{-1}$
β_7	-6.1665×10^{-9}	$Amp \times Hour \times Sec^{-1} \times Volt^{-3}$

is found to be entirely a function of energy processed. Specifically batteries that undergo light duty cycling at voltages at or above 3.4V age much more quickly than those cycled with voltages at 3.1V or below. For the group of batteries that underwent more moderate cycling there is an approximately linear fit between battery health and energy processed - in agreement with [14]. Capacity results for these three groups are in presented Fig. 3.4.

The second major result is that power fade is negligible in all of the batteries, see Fig. 3.5. This could perhaps be an artifact of either our measurement method or simply not

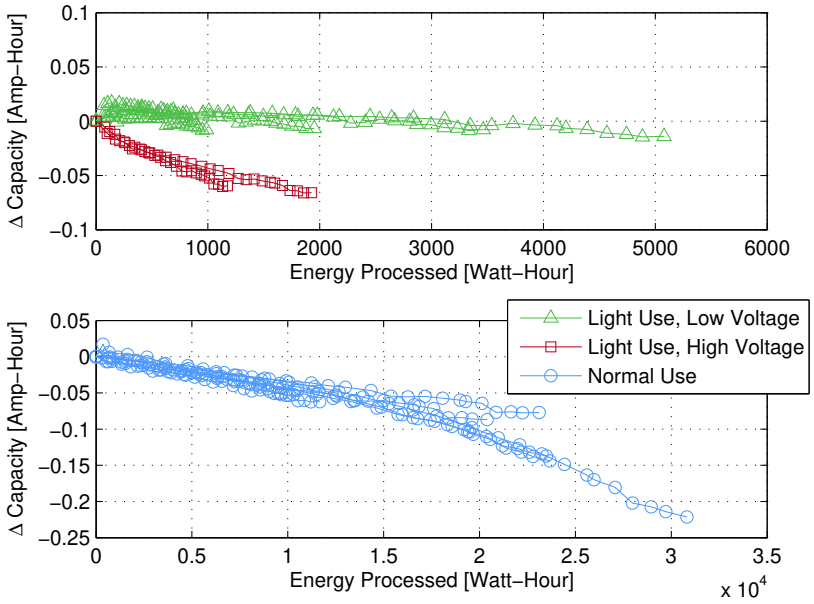

Figure 3.4 Δ Capacity as a function of charge processed

driving the batteries aggressively enough. The batteries are rated for 30C discharge whereas they were tested at 2.5C. The power test was a 15 second constant current draw, starting with the battery at 100% SoC (based on float charging at 3.6V). The recorded voltages and currents during this period were used to compute average power. Experimental design methods likely facilitated the appearance of the voltage dependence related to battery health. In the next section we discuss a generalization of this experimental work: OBHME.

3.5 The OBHME Framework

The work in this chapter suggests a generalized framework for conducting optimal battery health modeling experiments. The OBHME framework has three main steps: model selection, experiment design, and realization; see Fig. 3.6. The first step, model selection, chooses health metrics, model inputs, and an appropriate model form. The second step, experiment design, creates the possible trial set, generates and regresses the associated trials, and optimizes the trial set to maximize Fisher information. The final step, realization, conducts the battery experiment and then estimates the model parameters. The remainder of this section highlights how OBHME provides a method appropriate for optimizing a wide variety of battery health experiments.

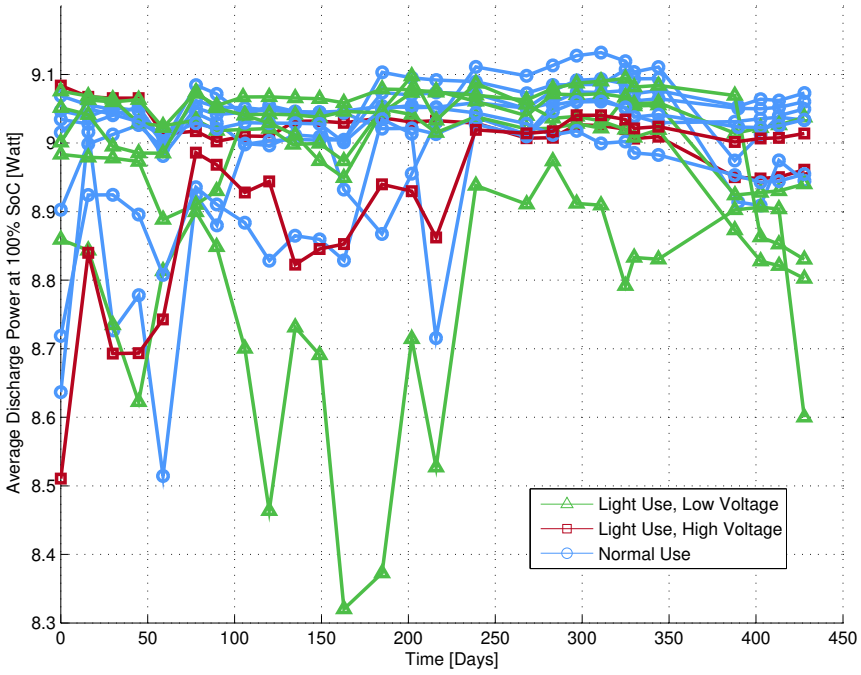

Figure 3.5 Average discharge power as a function of experimental time

3.5.1 Model Selection

The goal of OBHME is to efficiently produce an accurate model of observed physical phenomena. Typically, this model is intended for design, control, and/or optimization applications. When selecting a model one must consider the physical phenomena being investigated and the intended use of said model. The model selection step is critical because the optimal experiment depends directly on the selected model. The remainder of this section focuses on the individual parts of the battery health model: health metric (output), input(s), and model form.

Health Metric

The battery health metric's importance is two-fold; it both measures battery health and quantifies the health model's output. Capacity and power fade are typical choices for health metrics, although a variety of additional metrics exist. For example, in destructive testing, one may consider looking at specific internal features of the battery such as the solid electrolyte interface layer's thickness. One may also use Electric Impedance Spectroscopy (EIS) to investigate the frequency response characteristics related to battery aging. While many features of battery health are of interest, a good metric considers the following factors: intended model application, physical phenomena, and measurement cost. Health metrics play

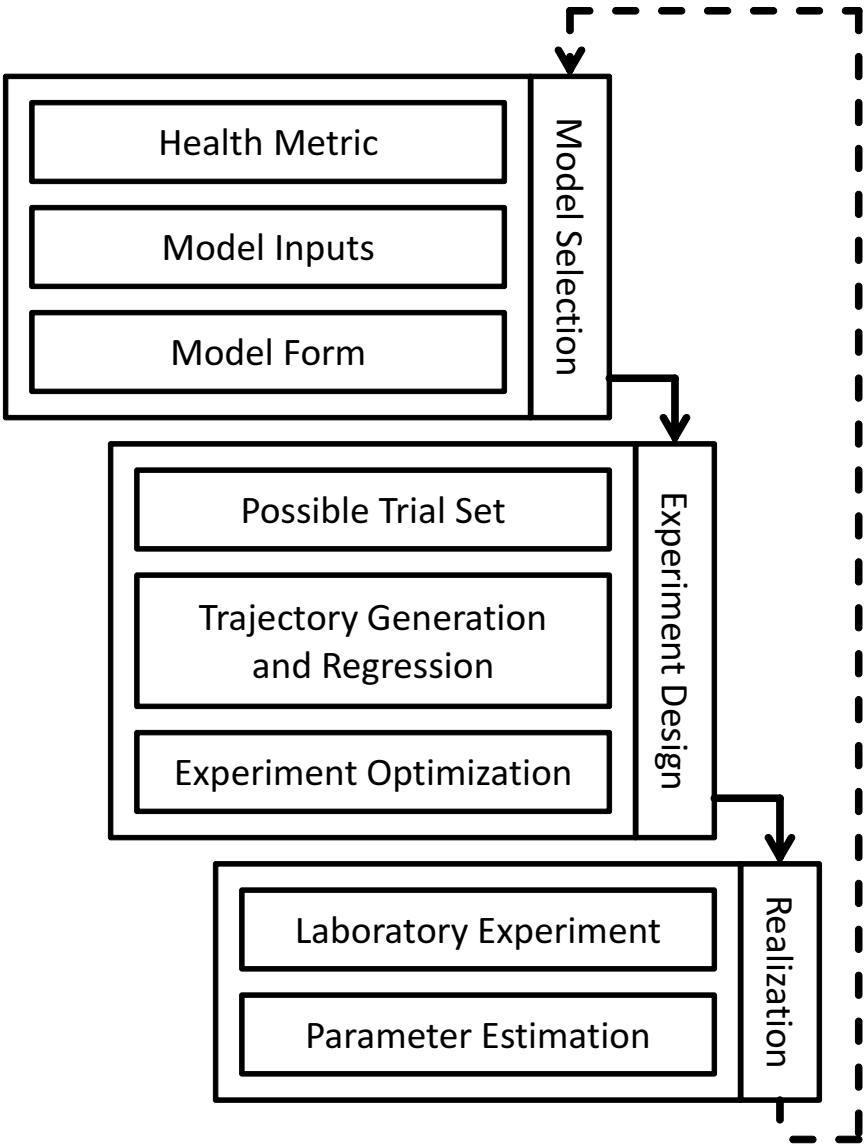

Figure 3.6 The OBHME Framework

key roles in both measuring aging phenomena and interpreting the battery health model's output.

Model Inputs

Similar to model outputs, choice of model inputs is tremendously important for both observing phenomena and ensuring the model's later applicability. One may choose from a variety of inputs, including: voltage, current, SoC, and temperature. Use of dynamic modeling and/or *in situ* measurements allows one to include various internal battery states as inputs. For example, local inputs such as current density, over potential and SoC can be used as

inputs by using the DFN model to simulate them. Internal temperature can be included through the use of *in situ* temperature measurement [88]. The input and output of the battery health model are connected by its mathematical form.

Model Form

The model form is selected based on *a priori* information if available, or an appropriate regression form otherwise. Models that can be cast in the form of Eq. 3.2 can be used without further mathematical complications. However, model's with autoregressive components or that are nonlinear in their parameters can still be accommodated in principle, but the mathematical complication will increase [16]. In the next step OBHME optimizes the experiment's trial set to gather the maximum parameter estimation information.

3.5.2 Experiment Design

The key to OBHME is deciding on a model *before* conducting the experiment and optimizing the selected experimental trials. The experiment design step breaks down into four steps. First one must decide on what set of experimental trials they are willing to consider - the possible trial set. Second one must generate their associated trajectories - the ones that are input into the battery health model. These trajectories are then regressed into a form that makes them appropriate for parameter estimation. Finally an optimization method is used on these regressed forms to either select a subset or duty cycle fraction. The remainder of this section discusses alternative choices one can make when design these experiments.

Possible Trial Set

The possible trial set Ξ is the set of all trials one wishes to consider and can actually conduct for an experiment. Typically this is limited by available equipment and conditions desired for the battery modeling. Earlier we focused entirely on a set of CCCV cycles described by an archetype cycle. Here we present an alternative archetype cycle which would be based on PHEV drive cycles, the Drive Charge Cycle (DCC). The Ξs generated by DCC is interesting because they mimic what PHEV batteries would experience in the field. For driving the battery undergoes a current trajectory (that may involve charging and discharging) mimicking PHEV battery pack behavior with appropriate scaling. One can imagine a probabilistic type cycle parameterized by driver aggressiveness and trip length. Additionally the vehicle battery may be parameterized by size and weight when converting the drive cycles into

current trajectories for the battery. After each drive (or several drives) the battery will need to be charged. This can be done using the charge portion of a CCCV cycle or one could use a Constant Power Constant Voltage (CPCV) cycle in order to better emulate battery pack charging. Additionally one can choose to emulate charging the vehicle with the various classes of PHEV home charging units. Algorithm 2 outlines how one would implement the DCC type cycles in a laboratory environment.

Algorithm 2 Drive Charge Cycle

Require: $I(t)$; P, V_{max}, $I_{trickle}$
  loop
    Track Current Trajectory $I(t)$
    while $(V < V_{max})$ **do**
      Constant Power Charge at P
    end while
    $t = Time$
    while $(\|I\| > I_{trickle})$ And $((Time - t) < t_{hold})$ **do**
      Constant Voltage Float at V_{max}
    end while
  end loop

Of course, DCC is just one of the many possibilities. The possible trial set can be customized for cycles based on cell phones, laptops, and satellites. In the next section we consider how to convert these cycle rule sets into these input trajectories for the battery health model.

Trajectory Generation and Regression

With a set of possible trials decided upon the trials must now be converted from rule sets to input trajectories for the health model. To achieve this one can generate quasi steady-state cycling trajectories through a variety of methods (these are cycles that repeat almost identically). The assumption here is that the input trajectories are not going to change substantially throughout the experiment, so all of the cycling behavior can be approximated in this behavior (alternative methods are needed for models with autoregressive components).

These quasi steady-state input trajectories are obtained in two main ways, both with tradeoffs. The first method is to use simulation. This requires that one has models that generate accurate trajectories of the variables to be used as inputs. An alternative method is to use laboratory equipment and directly measure the quasi steady-state cycles. One advantage of this method is it may result in highly accurate cycles as it eliminates model error. The results of both methods are the same, namely they both produce input trajectories

from steady-state cycles based on trial rule sets. The choice between these methods is largely dependent on the size of Ξ and available resources.

The input trajectories for each trial are then mapped to a trial regressor vector, to create a encapsulated version of the input trajectory's effects on the battery health model. The regression vectors are then collected into a set that describes the information available from various combinations of each of the trials. The procedure is the same as demonstrated earlier, see Eq. 3.8. Now we have a form amenable to optimizing the choice of experimental trials.

Experiment Optimization

OBHME allows for the direct use of either the DETMAX algorithm or a Linear Matrix Inequality Interior Point algorithm (LMIIP) both of which provide optimal experimental designs given slightly differing experimental structures [16, 89]. DETMAX was discussed in detail earlier so we focus on LMIIP here.

Instead of selecting a small subset of trials for the experiment, as DETMAX does, LMIIP allows one to incorporate all the possible trials. Specifically LMIIP finds the global optimum of the following problem:

$$\max_{\vec{\lambda}} \det \frac{U^T U}{\sigma^2} \tag{3.16a}$$

$$U^T U = \sum_{k=1}^{p} \left(\lambda_k \vec{u}_k \vec{u}_k^T \right), \tag{3.16b}$$

subject to:

$$\sum_{k=1}^{p} \lambda_k = 1 \tag{3.17a}$$

$$\lambda_k \geq 0, i = 1, \cdots, p. \tag{3.17b}$$

Where here there are two interpretations. The first is that there are a large number of cycles conducted and the λ_is determine the portion of time that each cycle is being conducted. Alternatively, one can treat this as a probability mass function and use it to generate randomized subset of cycles [89]. This is especially nice for stochastic type cycles (such as the aforementioned DCC) because it allows the incorporation of stochasticity into the experiment. With the trials/time fractions selected we continue onwards to the laboratory experiment.

3.5.3 Realization

The battery experiment is carried out by repeatedly running the cycles associated with the optimized set of trials. This cycling is interrupted at various times to conduct battery health measurements as related to the previously selected metric. For experiments generated with LMIIP there must be a mechanism to change the cycles after every health measurement. The regressors associated with these health measurements are computed by mapping the recorded input trajectories in accordance with the model form (see Eq. 3.8). This data can then be directly compiled into a form amenable for the linear least squares estimation of $\vec{\beta}$ (see Eq. 3.15). At the conclusion of the method one may wish to design a new experiment based on the results, perhaps to fine tune the modeling efforts further.

3.6 Discussion

Based on experience gained from this experiment, we suggest several opportunities for improving future iterations. First, duplicating trials would help to better elucidate underlying statistical properties of the batteries. Second, for reasons mentioned earlier, future modeling efforts may be further improved through nonidentical current limits, increasing the charge and discharge asymmetry. Additionally, there are several areas where increasing experimental scope is also desirable. For example, temperature can become an additional input by using several thermal chambers to regulate multiple ambient temperatures. Also, more aggressive cycling is possible by increasing the current limits, leading to a health model with broader applicability (although internal temperature may need to be considered). Further, replacing SoC as an input for voltage could provide additional insights into battery degradation, at the expense of requiring a robust implementation strategy. Finally, power testing could be augmented by conducting it at different SoCs, higher rates, and/or incorporating EIS. Clearly, the experience gained from this experiment suggests several advantageous improvements and extensions.

3.7 Conclusions

This chapter demonstrates how to apply OED to a specific battery case and then introduces a general framework for handling a wide variety of battery health experiments. This specific case designs and conducts a battery health modeling experiment for $LiFePO_4$ battery cells. The experiment results in a control oriented model useful for design, control, and

optimization of PHEVs. This initial work suggests the OBHME framework, which bridges an important gap between the OED literature and the battery health experimental/modeling literature. It is hoped that the OBHME framework case study in this chapter provides a guide for making battery health experiments less costly in terms of time, effort, and equipment and more profitable in terms of information and model accuracy.

Chapter 4

Electrochemical Model Parameter Characterization via Optimal Design of Maximally Informative and Minimally Invasive $LiFePO_4$ Cell Experiments

4.1 Introduction

Understanding, estimating, and managing battery health for ever increasing fleets of electrified vehicles is clearly of great financial and engineering importance. Financially, the quantity of electric vehicles continues to increase. For many of these vehicles, the battery will be a significant portion of their manufacturing cost. From an engineering perspective, battery health affects usability, including vehicle performance and range. This chapter improves on understanding, estimating, and managing battery health by creating a systematic method for designing battery experiments and diagnostics based on tools from optimal experimental design and evolutionary optimization. These experiments and diagnostics gather the greatest amount of battery parameter estimation information possible while minimizing damage to the battery. We demonstrate this method by using it to create Pareto fronts of current trajectories that maximize Fisher information (in the D-optimal sense) while minimizing battery damage. To assess damage we use two metrics: energy processed (following Whitacre *et al.* [14]) and the health model created in Chapter 3. The generated trajectories are simulated with artificial measurement noise which is fed to an estimator for validation. This creates sets of parameter estimations whose statistics can be related to Fisher information. Since this method improves parameter estimation accuracy while minimizing battery damage, it can improve both battery State-of-Health (SoH) estimation and battery health modeling experiments.

The trajectories designed by this method have a variety of practical applications. Firstly, the diagnostics based on these trajectories can be implemented to run while the vehicle

charges overnight. The estimation algorithm can then be computed on-site with a micro processor or off-site via cloud computing. After this computation the vehicle's estimate of SoH would be updated, allowing the vehicle controllers to make better decisions. This is different from traditional SoH estimation methods in that it is both offline and the trajectory the battery undergoes is prescribed [90–94]. We believe conducting estimation offline is not only appropriate but actually beneficial. It is appropriate, as the SoH changes very slowly and beneficial because it does not increase the online computational burden. Additionally, the trajectory is custom designed rather than imposed by the driver and vehicle which helps to increase information content. Another interesting application is the improvement of long term battery health experiments. An ideal battery health test for long term experiments would be short in duration (to minimize the time the battery does not spend cycling), cause very little damage (so that observing health has a minimal impact on the measured results), and maximally informative about the parameters of interest. The trajectories created by this method help with all three of these goals: one can specify the duration and then use the Pareto front to choose an appropriate trade-off between information and damage. This has the potential to further improve the methods described in Chapter 3.

The major difference between our approach and that typically used for battery experiments is the scope of possible trajectories considered. Typically one works with a small finite set of trajectories and the problem is treated as a subset selection problem. This is the case in the health experiments in [10] as well as by Schmidt *et al.*'s work fitting a single particle model [37]. Rather than limiting ourselves to preselected trajectories, we use an algorithm that explores the function space (up to a sampling rate of 5 Hz). This has a variety of advantages. Firstly, by being dramatically less restrictive, it enables the optimization to achieve greater improvements. Second, it enables us to gain insight into what types of trajectories are best for this identification work. Finally, it allows one to approach situations where it is unclear which trajectories might be appropriate. The cost of these advantages is a dramatic increase in the computational complexity of the optimization. As computers continue to become faster this will continue to become less disadvantageous. Presently the authors overcome this difficulty by making use of parallel computing clusters. This allows us to simulate 10 minutes of the Doyle-Fuller-Newman (DFN) model roughly 1.5 million times per optimization.

For this work we focus on estimating parameter's of the DFN model. Intuitively, ease of estimating parameters depends on how slight perturbations of parameters affect the measured outputs of the model. This work designs input current trajectories that maximize the parameter estimation information carried by the voltage output. This idea of estimation information is formalized by Fisher information, which uses the output parameter sensi-

tivities to compute information content. In the linear case (the DFN model is nonlinear) Fisher information provides a bound on best possible parameter estimation behavior [40]. In the nonlinear case it is a useful guide for improving estimator performance. Each current trajectory has an associated Fisher information value, and this value is used as one of the two optimization multi-objectives.

An excellent evolutionary algorithm for multi-objective optimization is the Nondominated Sorting Genetic Algorithm II (NSGAII) which evolves populations to directly create Pareto fronts [46]. Another useful evolutionary algorithm is Differential Evolution (DE), which works directly on real encoded problems and has been shown to be very effective in a variety of applications [47–49]. The advantages of each of these evolutionary methods are married by Kwan, Yang, and Chen who created the NSGAII-DE algorithm [50]. This algorithm combines the performance of DE with the advantage of direct Pareto front creation, and thus it is ideal for our problem.

By pairing both health metrics with Fisher information we create optimization problems for which NSGAII-DE generates Pareto fronts. In both cases Fisher information is based on the simultaneous estimation of two parameters: d_2, solution diffusivity and R_{SEI}, the resistance due to anode film thickness. Each of these relates to battery health, and d_2 is chosen specifically due to previous difficulties in its estimation [13].

Both optimizations result in similar Pareto fronts. Specifically, at high *energy* and *damage* ranges there is little to no improvement in Fisher information. However, at mid to low *energy* and *damage* ranges the Fisher information improves, frequently by orders of magnitude. The practical nature of these improvements are validated through the repeated use of an estimator on simulated data. This validation finds that Fisher information is an excellent qualitative guide for the estimator's performance. However, it is a poor quantitative predictor as the estimator dramatically out performs the predicted Fisher information. This is likely due to Fisher information being based on linear assumptions whereas the DFN model is highly nonlinear - this idea is further considered in the discussion section.

The remainder of this chapter focuses on the specifics of optimal trajectory generation, estimation and results. Specifically, §4.3 presents the problem in a mathematical context and discusses how the application is optimized. §4.4 presents the results of the optimization along with estimator results based on simulated noise (as a validation). A discussion follows in §4.5 where assumptions, computational advantages, and extensions to the method are discussed. Finally §4.6 contains concluding remarks and a summary of this chapter's contributions.

4.2 Batterys and Electrochemical Model

4.2.1 Lithium Iron Phosphate

As with the rest of this thesis we focus on the $LiFePO_4$ battery chemistry. Specifically we focus on cells ANR26650M1 cells from A123 systems. These cells are used in several PHEV and EV applications. They are capable of very high rates, including a continuous 30 C-rate discharge. They have a nominal voltage of 3.3V and an operating region of 2.0V to 3.6V. The name plate capacity of these batteries is 2.3 Amp-Hrs. These batteries have been used in several PHEVs, including a plug-in Prius upgrade kit. Of course, the methods within this paper could be applied to other battery types, provided the correct parameters are available for accurate simulation of the DFN model.

4.2.2 The Doyle-Fuller-Newman Model

The DFN model originates from work by Doyle, Fuller, and Newman published in 1993 and 1994 [2, 3]. Much has been written about this model and it has been used in a variety of applications regarding batteries. Of specific interest is [5], which fits many parameters of the DFN model to experimental data for the $LiFePO_4$ cells that are used within this work. The salient features of the DFN model are diffusion in both the solution and the solid, where the distribution in the solid is distributed in a pseudo 2D manner. These are connected to the electrical equations within the battery by two highly coupled nonlinear boundary value problems that control the flow of current and voltage distributions within side the battery. This creates a nonlinear DAE which can be very computationally intensive to solve. We use a combination of model reduction methods to simulate the model, see §A.1 and article [5]. The DFN model excels in areas where there are high rates that are highly transient as the model stems from first principles (it is based on binary concentrated solution theory).

The importance of the model in this application is two-fold. A model is obviously needed for simulation. However, the model also plays an important role in how the Fisher information is defined. Fisher information is related to the ease of estimating various parameters, but these parameters must be from a model. In fact, different Fisher information values can result for a physical parameter if the model used in calculation is changed. In addition, for Fisher information to be useful the model used must be reasonably accurate. For all of these reasons we use the DFN model solely throughout this chapter.

4.2.3 Parameters of Interest

In this work we focus on two parameters of interest in the DFN model. The first, R_{SEI}, is the resistance of the solid electrolyte interface layer on the anode and it is related to battery health in two ways. First, creation of this film consumes the batteries cyclable Li-ions which decreases the maximum capacity. Second, the film resistance can cause power fade (at least within the DFN model) as the resistance decreases the batteries ability to provide power. The second parameter d_2, the diffusivity of the electrolyte, is slightly related to battery health as its decrease represents the electrolyte clogging within the battery. However our major interest in this parameter is as a challenge to the method - this parameter has been shown to be very difficult to identify in a previous work by the author [5]. Of course any combination of model parameters could have been used, and these two were chosen for demonstration. It is worth noting the that method is not limited to the two parameter case - Fisher information can be readily computed for larger sets of parameters.

4.3 Trajectory Optimization Method

Our goal is to design dynamic experiments to maximize the Fisher information gained regarding relevant parameters while minimizing battery damage. To this end we design open-loop current trajectories and measure the voltage response of the cell. These current and voltage trajectories are then fed to an offline estimator which attempts to minimize voltage error (L^2) by assigning the model appropriate parameter values. This section explains the mathematical and computational specifics of this procedure. The following section discusses the results for the trajectory optimization and the estimator's performance.

4.3.1 Optimization Formulation

The result of this optimization is a set of current trajectories that form a Pareto front for identifying these two specific parameters, although the problem could easily be reformulated for different parameters or more parameters. Solving this problem enables more efficient experiments for battery characterization by better designing input trajectories.

This problem is formulated as a multi-objective open-loop trajectory optimization. Our first objective is to optimize Fisher information. For multiple parameters different versions of Fisher information exist. We use D-optimal as this minimizes the area (or volume/content for dimensions greater than two) of the the estimation ellipses. This makes for more accurate estimations, provided one is using an appropriate estimator. The second objective is to

minimize the the damage caused by the trajectory (invasiveness). To this end we use two different measures, each for one optimization. The first, energy processed, follows [14] which treat this as proportional to loss of maximum cell capacity (with different multiplicative constants depending on the battery cycling mode). In addition we use the health model from chapter 3. For the signal we allow any currents that are of magnitude less than 2.5 C-rate. Current trajectories that result in the voltage being less than 2.0V or greater than 3.6V are considered invalid and not used (as these are the outside the batteries' design limits). The sample rate on the signal is 5 Hz and all of the points in the current are considered independent of each other in the optimization. There is no splining or similar method to decrease the number of points by design - this would filter the signal and as is shown later the higher frequency content is important in this application. The author originally tried both cubic splines and Legendre-Gauss-Lobotto points to simplify the optimization but the signal filtering caused by these methods was substantial and unacceptable.

4.3.2 Fisher Information

For a dynamic model one can compute the Fisher information as follows. We note that we are dealing with a SISO model (current in, voltage out). Fisher information is not additive in the two (or higher) parameter case, preventing the use of common dynamic programming procedures (as the suboptimallity assumption is not applicable) [95]. It may be theoretically feasible to use suboptimality in the one parameter case, however the large number of states in the DFN model (even after our modeling reduction techniques) makes this nearly impossible computationally (save cases with both very short time periods and large numbers of cores).

We now formally introduce Fisher information for dynamic systems and show how it applies to this problem. There is quite a bit of literature on both Fisher information in general and Fisher information as applied to dynamic systems [16, 17, 37, 40–44].

Consider the standard model formulation:

$$\dot{x} = f\left(x; \vec{\theta}\right) \tag{4.1a}$$

$$y = g\left(x; \vec{\theta}\right) + error \tag{4.1b}$$

$$error \sim Normal\left(0, \sigma^2\right). \tag{4.1c}$$

Where x is the state and y is the output and f and g are the appropriate functions. $\vec{\theta}$ is the parameter vector, that we are computing the Fisher information for. The astute reader may notice that this formulation is a differential equation rather than DAE - the DFN model

has a low enough index that it can be treated as a differential equation for this section [68].

Of particular interest is the sensitives at a set of instances in time, one can form a sensitivity matrix as follows [44]:

$$S_{i,j} = \left[\frac{\partial y(t)}{\partial \theta_i}\right]\Bigg|_{t=t_j}. \tag{4.2}$$

Where the entry $S_{i,j}$ is the output sensitivity to parameter θ_i sampled at time instant t_j. The Fisher information is then given by [16, 17, 37, 40–43]:

$$Fisher = \frac{S^T S}{\sigma^2}. \tag{4.3}$$

Where $Fisher \in \mathbb{R}^{p\times p}$. For optimization we use the D-optimal metric which is as follows [15]:

$$F_D = \det(Fisher). \tag{4.4}$$

We compute the Fisher information by perturbing $\vec{\theta}$ in each of its direction and simulating the DFN model. We use forward differencing (instead of central differencing) to decrease the number of necessary simulation calls. This allows us to compute the two sensitivities that we need with only three runs of the simulations (or $p+1$ simulations for p parameters).

We note that the Fisher information is a local measure of information. This can be seen as it solely depends on the partial derivatives related to the output. For linear systems this can be used to accurately bound an estimator's performance via the Cramér-Rao bound. For nonlinear systems this is only an approximation that we use as a guide for our optimizations. We continue further with this guide notion in §4.5, where we discuss the results of the optimization and simulated estimator.

4.3.3 Energy and Health

We consider two measures of invasiveness. The first is based entirely on energy processed following Whitacre *et al.* [14]. The second is based on Chapter 3 where a battery health model is developed.

Energy processed is straightforward, we compute and integrate the battery's power:

$$energy = \int_{t=0}^{t=T_f} I(t)V(t)dt. \tag{4.5}$$

Where I and V are current and voltage respectively.

Table 4.1 Estimated parameters for the Symmetric model (repeated)

Parameter	Value	Units
β_1	1.1484×10^{-7}	$Amp \times Hour \times Sec^{-1}$
β_2	-3.9984×10^{-8}	$Hour \times Sec^{-1}$
β_3	-1.3158×10^{-7}	$Amp \times Hour \times Sec^{-1} \times Volt^{-1}$
β_4	-5.5487×10^{-10}	$Amp \times Hour \times Sec^{-1} \times Amp^{-1}$
β_5	4.9680×10^{-8}	$Amp \times Hour \times Sec^{-1} \times Volt^{-2}$
β_6	1.1166×10^{-8}	$Hour \times Sec^{-1} \times Volt^{-1}$
β_7	-6.1665×10^{-9}	$Amp \times Hour \times Sec^{-1} \times Volt^{-3}$

The battery health model is as follows:

$$\dot{h}(I,V) = \beta_1 + \beta_2\|I\| + \beta_3 V + \beta_4\|I\|^2 + \beta_5 V^2 + \beta_6\|I\|V + \beta_7 V^3. \tag{4.6}$$

Where the β parameters are given in Table 4.1. The objective then is to minimize:

$$damage = -\int_{t=0}^{t=T_f} \dot{h}(t)dt. \tag{4.7}$$

Where this model has been developed with long term cycling of multiple batteries undergoing different CCCV cycles. For both of these objectives (*energy* and *damage*) the goal is minimization of the values.

4.3.4 Trajectory Constraints

The current trajectories in these specific optimizations are constrained to be exactly 10 minutes long. The sampling rate is at 5 Hz so each trajectory has 3001 control points. One could use splines and other methods to decrease the number of control points (indeed, the author has in the past - cubic and Gauss-Legendre-Lobotto, see [96]). However decreasing the number of control points introduces filtering due to the choice of splines. Typically, when one expects a smooth trajectory as a result of an optimization this is not a problem. However, OED trajectories tend to have very high frequency characteristics. Due to this we opt to make every point in the trajectory a control point. As can be seen in §4.4 the optimized trajectories do have a high frequency component.

Additionally the voltage and current are both bound. The current is bound between $\pm$5.75A ($\pm$2.5C-rate). This bounding is easy to implement - the optimization algorithm

will not try trajectories outside of this. The voltage is bound between 2.0V and 3.6V which are the design specifications for these specific cells. This voltage constraint is enforced by rejecting population members that violate it.

4.3.5 NSGA-II with DE

There are a multitude of optimization methods available in the literature. Evolutionary algorithms are typically very useful in environments with large numbers of variables and cases where models are noisy. In this case we happen to be dealing with both of these so these methods are ideal. Additionally there are a variety of ways to generate Pareto fronts. One can change the weights on each objective and reoptimize repeatedly. Or one can use an evolutionary method where one searches for the entire Pareto front at once by optimizing an entire population. NSGAII is such an algorithm [46].

NSGAII is the Nondominated Sorting Genetic Algorithm II and was created by Deb, Pratap *et al.* [46]. This algorithm starts as a typical genetic algorithm. However, rather than using a single objective related to fitness, it can use several. It does this by sorting the population into fronts, where members of each front are dominated by exactly i members. Here, a member is dominated if another member has better values for all of the multi-objective functions. The best front is the front of members which are not dominated by any member of the population and this front forms the Pareto front. As the population evolves NSGAII selects members from the least dominated fronts and has a mechanism to avoid overcrowding sections of the fronts. Because of this, it can compute a Pareto front in one optimization, rather than having to reoptimize with different weighting factors on each of the objectives. This frequently saves large amounts of computational time, as the evolutionary algorithm does not have to repeat similar optimizations.

Originally NSGA-II was designed to use a genetic algorithm for cross breeding and mutations - the mechanisms that create a new population from an old one. However, one can use other mechanisms to build new populations. We choose to use DE because it is naturally real encoded (as is our problem) and has been shown to be very effective on a variety of difficult optimization problems [49]. DE was introduced by Storn and Price in [47] and has performed very well as a optimization/search algorithm. The basic DE algorithm works as follows. An initial population is generated randomly. Each population member is then perturbed based on vector directions with in the population. Not all of the elements are written to the new vector, this is controlled by a weighted coin flip (with probability C_R). Fitness values for the new population are then computed, and if the perturbed members improve on their associated nonpertubred member they are used to replace them. One can refer to a

variety of the literature on the subject for further specifics on the algorithm [47–49, 97]. One of the major advantages of DE is it changes step size and direction based on the spread of the population members. When the spread is large DE creates large perturbations for searching the space. When the population members are close, the step size is proportionally reduced. Additionally, the optimization variable sensitivity is automatically taken into account with the population spread. All of this happens with a very low number of tuning parameters for the algorithm, making it easier to robustly apply to a variety of optimization problems.

Ideally one would like to combine the benefits of both NSGAII and DE. NSGAII-DE is such an algorithm - it uses NSGAII to directly evolve Pareto fronts and DE to improve its search and optimization [50]. NSGAII-DE uses an unmodified version NSGAII's selection mechanism and a slightly modified form of DE's member generation. This modification creates new members based solely off perturbations about the members of the nondominated front. Each member of the nondominated front is equally likely to be used in each perturbation. These perturbations are then combined with the base population and culled by NSGAII to produce the next generation. This combination of NSGAII with DE is highly effective, as can be seen in the results section. The next subsection discusses the computational specifics of running NSGAII-DE for the trajectory optimization and plain DE for estimation.

4.3.6 Computation

There are two main computational problems within this work. The first is the NSGAII-DE optimization which evolves the current trajectories. This requires the Fisher information to be computed for each trajectory considered. The second examines the effects of these created trajectories through the repeated use of DE to estimate parameters based on simulated data with measurement noise. These DE optimizations adjust the estimation parameter values by computing and comparing voltage responses. Much of the computation between these two problems is similar, we discuss these similarities first and then follow with their differences.

In both of these problems efficient computation of the DFN model is critical. To this end we use the modal Legendre form of the DFN model with quasi-linearization, see Appendix A.1. We use a 5^{th} order polynomial representation across the width of each cell region: anode, separator, and cathode. In the radial direction we use a 10^{th} order polynomial representation, where the polynomials of odd degree are omitted due to symmetry. By using Legendre modes we reduce the number of differential equations to 56 and the number of algebraic equations to 30. In addition, we use quasi-linearization nearly identical to that in [5] except it is applied to the modal coordinates. We implement our simulation code in MATLAB although other environments may provide better computational speed (MATLAB

Table 4.2 NSGAII-DE optimization values

Parameter	Value
Initial Population Members	4000
Population Members	1000
C_r	0.9
Number of Generations	550

is excellent for agile development).

Even with these computational reductions we still need to make use of parallel computing. Specifically we use a server-client architecture to parallelize the optimization function calls. The NSGAII-DE and normal DE run on the server, and then evaluation requests are sent out to the clients. This communication uses custom Java code to connect the server and clients over TCP/IP. We link the DFN model simulations to this code by calling it within the MATLAB environment. A combination of our in lab cluster and the University of Michigan's Center for Advanced Computing cluster were used to conduct these computations. Maximally we had 56 cores running to compute the trajectory optimizations.

Another commonality between these two problems is that they both feature bounds on their optimization variables. To gracefully handle perturbations that find members outside these bounds we only halve their distance to the boundary when they violate it [48]. This performs better than placing them on the boundary, which would discourage population diversity.

Despite all these similarities the two problems have a few important differences. For the trajectory optimization we have a 3001 point current trajectory for 10 minute runs. To compute Fisher information we repeat the runs with the parameters perturbed. This allows us to compute an optimization evaluation with three DFN model runs. The specifics of the trajectory optimization are summarized in Table 4.2. Where C_r is the probability that a specific value mutates and the number of generations was decided upon by stopping the optimization after it appeared to have converged for a substantial number of steps.

It is important to note how the optimization population is initialized. We use stochastic Algorithm 3 to create the initial trajectories. In this algorithm $r_{i,j}$ is the i^{th} trajectory's j^{th} value. This gives us a set of trajectories similar to white noise, but with nonzero means that allow the State-of-Charge (SoC) to drift. Originally we had only used white noise, but we discovered this useful heuristic through experimentation with the population initializations.

While the estimation work is very similar there are some key differences. The estimator was run 50 times for each of the 26 cases (7 optimized and 7 initial for *energy* and another

Algorithm 3 Initial Population Generation

Require: $NumberOfTrajectories$; $NumberOfControlPoints$; $I_{\max}$
  for $i = 1 \rightarrow NumberOfTrajectories$ **do**
    $p \leftarrow (i-1)/(NumberofTrajectories-1)$
    for $j = 1 \rightarrow NumberOfControlPoints$ **do**
      $randomReal \leftarrow$ RandomUniform$[0,1]$
      $randomSign \leftarrow$ RandomUniform$\{-1,1\}$
      if $p \leq 0.5$ **then**
        $r_{i,j} \leftarrow 2 \times I_{\max} \times randomSign \times randomReal$
      else
        $r_{i,j} \leftarrow 2 \times I_{\max} \times randomSign \times (1-(2-p)\, randomReal)$
      end if
    end for
  end for

6 each for *damage*). For the estimator we only need to run the DFN model once to get an evaluation (the trajectory optimization has to be run three times to compute Fisher information). There are only two inputs to the estimator's optimization: the values of R_{SEI} and d_2. The optimizations are cut off after 500 generations - this only occurs for the lowest *energy* case (where the estimator consistently fails to converge due to lack of information). The stopping criteria is based on the spread of the top 50% (Stopping Criteria Value 1) of the population, and stops when the maximum distance between these selected population members is less than 10^{-4} (Stopping Criteria Value 2). This is the *MaxDistQuick* criteria from [98]. We base this not on absolute distance in the parameter space (as the parameters are scaled differently) but base it on the maximum and minimum values that the parameters are bound (artificially) by. This distance between the population members is computed as follows:

$$popDistance(x,y) = \left(\left(\frac{x_{d_2} - y_{d_2}}{\max(d_2) - \min(d_2)} \right)^2 + \left(\frac{x_{R_{SEI}} - y_{R_{SEI}}}{\max(R_{SEI}) - \min(R_{SEI})} \right)^2 \right)^{\frac{1}{2}}. \tag{4.8}$$

We note that using DE for this estimation works very well, as is does an excellent job quickly searching the space and the population members always come within close agreement.

We use the estimation sets to compute a Fisher information based on estimator performance. In a linear system this would be in near agreement with the trajectory optimization Fisher information. However in a nonlinear system these can vary from each other by quite a bit (as is shown later). To compute the Fisher information based on estimator performance we compute the covariance matrix of the estimations, invert this and compute the

Table 4.3 DE estimator values

Parameter	Value
Population Members	40
C_r	0.9
Stopping Criteria Value 1	0.5
Stopping Criteria Value 2	1×10^{-4}
$\max(d_2)$	$1 \times 10^{-8} m^2 s^{-1}$
$\min(d_2)$	$0\, m^2 s^{-1}$
$\max(R_{SEI})$	$1 \times 0.01\Omega$
$\min(R_{SEI})$	$0\,\Omega$

determinant:

$$\hat{F}_D = \det\left(Covariance\,(Estimations)^{-1}\right). \tag{4.9}$$

The agreement and disagreement between the predicted and actual Fisher information is discussed further in the results and discussion sections.

4.4 Trajectory Optimization Results

This section discusses the trajectory optimizations and their related estimation results. These optimizations yield Pareto fronts of trajectories that increase the Fisher information content. However - since the DFN model is nonlinear - this improvement cannot be directly quantified through the Cramér-Rao bound. Instead, we quantify this informational improvement by applying an estimator to sets of simulated data containing measurement noise. This allows us to generate sets of estimations whose statistics quantify the trajectories' practical estimation improvements. Some of the more interesting estimation sets are presented. These are followed by plots similar to Pareto fronts showing the improvements offered by the optimized trajectories as quantified by the estimator results. Finally we present graphs of the optimized trajectories for the interested reader. Together, these results demonstrate that using Fisher information as a multi-objective creates dramatic improvements in estimator performance.

4.4.1 Optimization Results

We performed two trajectory optimizations, both using Fisher information as a multi-objective (d_2 and R_{SEI} are taken as the parameters of interest). These optimizations differ in their other multi-objective: the first focuses on energy processed and the second uses the battery health model from Chapter 3. Both cases yield similar results, making large gains in Fisher information at low to mid *energy*/*damage* ranges and at the high range the differences become inconsequential. Figures 4.1 and 4.2 present the *energy* and *damage* results respectively, with Fisher information presented on a logarithmic scale. However, Fisher information is only a guide for this nonlinear model and next we present a more accurate quantification metric based on simulating estimator performance. Unfortunately, simulating estimator performance is not computationally tractable for use as a multi-objective in optimization (see the discussion section).

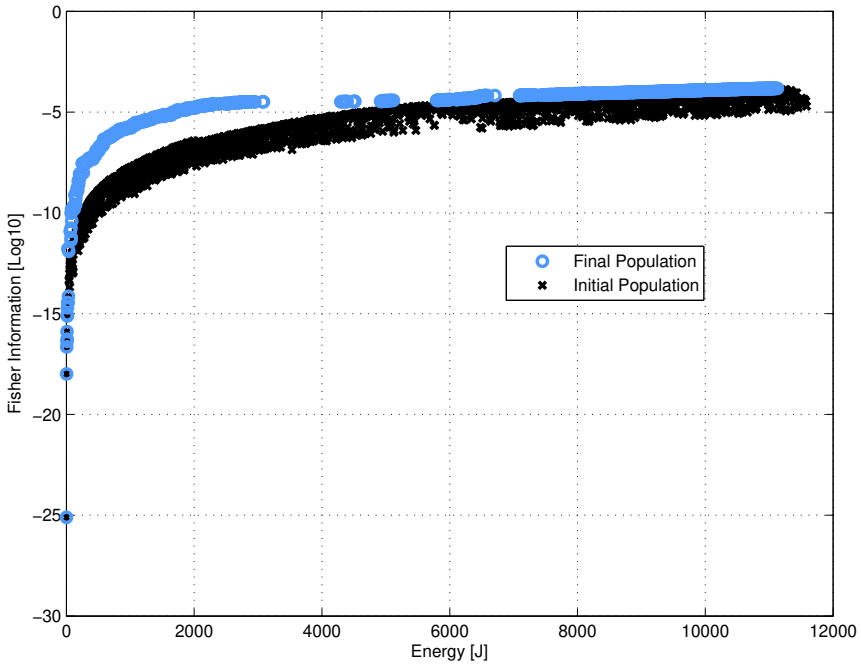

Figure 4.1 Trajectory optimization Pareto front: Fisher information and energy

4.4.2 Estimation Results

By using an estimator on simulated data we are better able to quantify the improvements of using these optimized trajectories. Unfortunately it was not computationally tractable to run an estimator on every single trajectory. Instead we selected subsets of interesting trajectories from each Pareto front. Then, to compare with the initial population, we took the trajectories

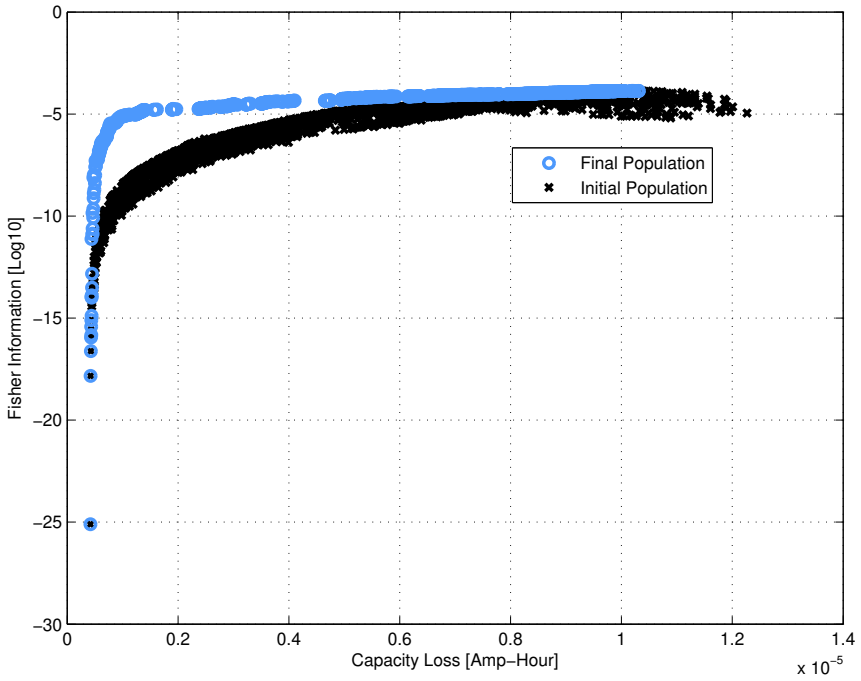

Figure 4.2 Trajectory optimization Pareto front: Fisher information and health

Table 4.4 *energy* values for the compared trajectory pairs

Trajectory Number	Energy [J]		
	Value	Optimal	Best Initial
1e	0	0	0
2e	1000	1014	1013
3e	2000	2015	1992
4e	5000	4996	4970
5e	8000	8004	8011
6e	10000	10002	9994
7e	11000	11130	11048

from the best initial front that were closest in value to the *energy* and *damage* levels of interest. Tables 4.4 and 4.5 summarize our selection of *energy* and *damage* values.

To create a statistically meaningful set of estimations we ran the estimator 50 times on for each trajectory on simulated data. This data took the voltage measurement error to be zero mean Gaussian with $\sigma = 1mV$ which is well within what is possible with laboratory equipment. The results of these estimator runs are a cloud each of 50 estimations of the R_{SEI} and d_2 pair. The statistics of these clouds allow us to compute an estimator based Fisher information, which is used for comparisons both between estimators and from the initial predicted Fisher information from the optimization.

Table 4.5 *damage* values for the compared trajectory pairs

Trajectory Number	Damage [μ Amp-Hr] Value	Optimal	Best Initial
1h	1	1.022	1.011
2h	2	1.910	1.925
3h	4	4.035	4.014
4h	6	5.880	5.861
5h	8	7.976	7.980
6h	10	9.978	10.00

The results for the *energy* based optimization are shown in Figures 4.3, 4.4, and 4.5. These are the trajectories 2e to 4e, which show improvements over the best trajectories in the initial set with similar *energy* values (see Table 4.4). The estimator fails to converge when acting on trajectory pair 1e's simulation data due to these trajectories' having low information content. Trajectory pairs 5e to 7e perform similarly in both the initial population and the optimized set, yielding no meaningful improvements in estimator performance. The *damage* case is similar and estimator results of 1h to 3h are shown in Figures 4.6, 4.7, and 4.8. All of these yield excellent improvements over the best trajectories in the initial set with similar *damage* values (see Table 4.5). We choose not to run the lowest possible *damage* case as it was likely to have the same results as 1e. 4h yields some improvements (plot not shown) and then 5h and 6h yield no meaningful improvements over their associated initial trajectories. For both the *energy* and *damage* cases substantial improvements are made at the low to mid ranges and then no meaningful improvements are made at the high range.

Now we use the Fisher informations calculated from the estimator results to compute the improvement of the optimizations. Figures 4.9 and 4.10 show these results for the *energy* and *damage* optimizations respectability. These have been normalized so that $\sigma = 1V$ to be consistent with the predicted Fisher information. The estimator based Fisher information is much higher than originally predicted by the optimization. However, the qualitative trends are similar when compared to figure s 4.1 and 4.2. Thus we have found Fisher information to be an excellent guide for these optimizations.

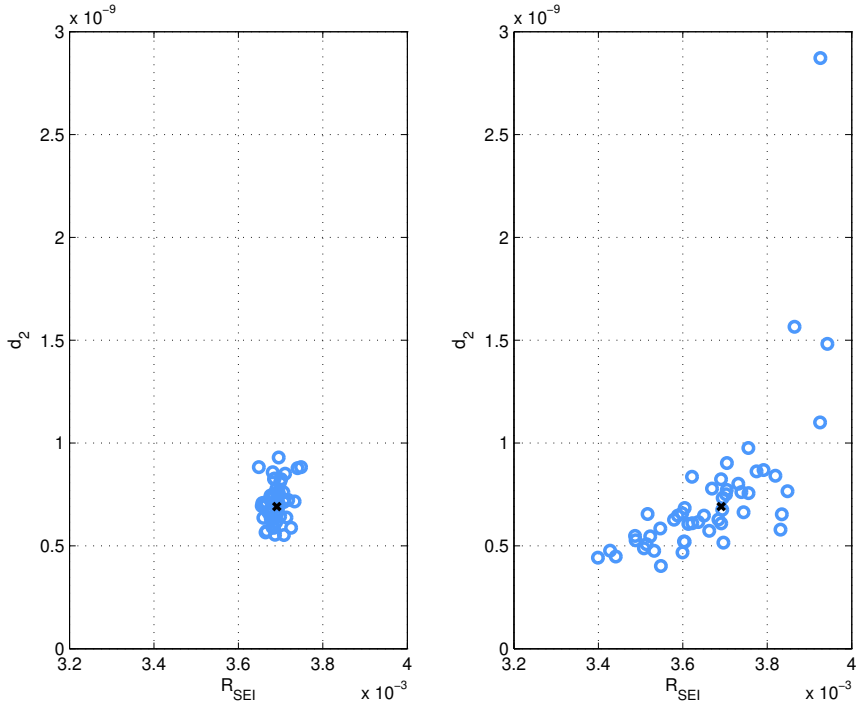

Figure 4.3 Energy trajectory pair 2e approximately 1000 Joules

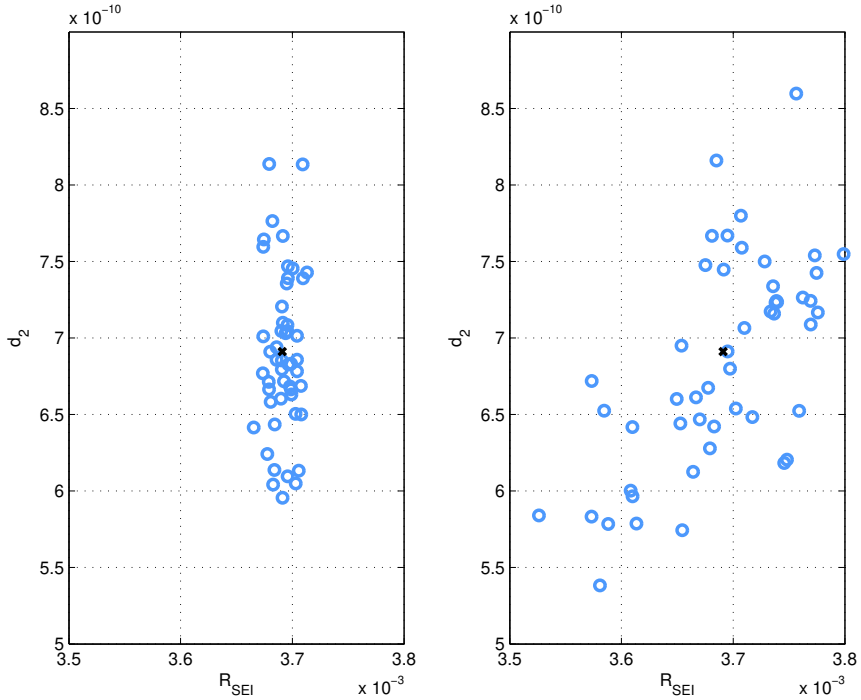

Figure 4.4 Energy trajectory pair 3e approximately 2000 Joules

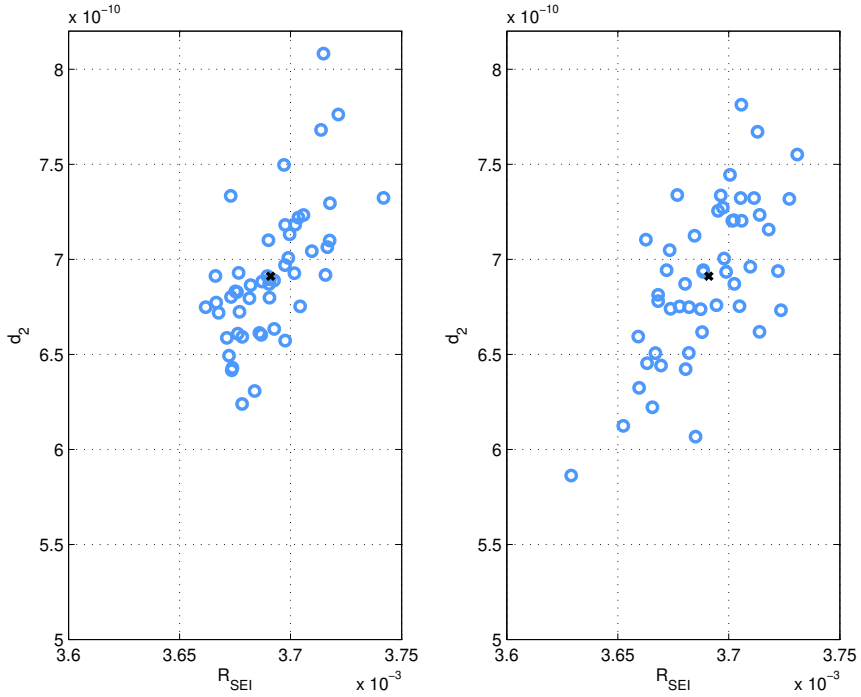

Figure 4.5 Energy trajectory pair 4e approximately 5000 Joules

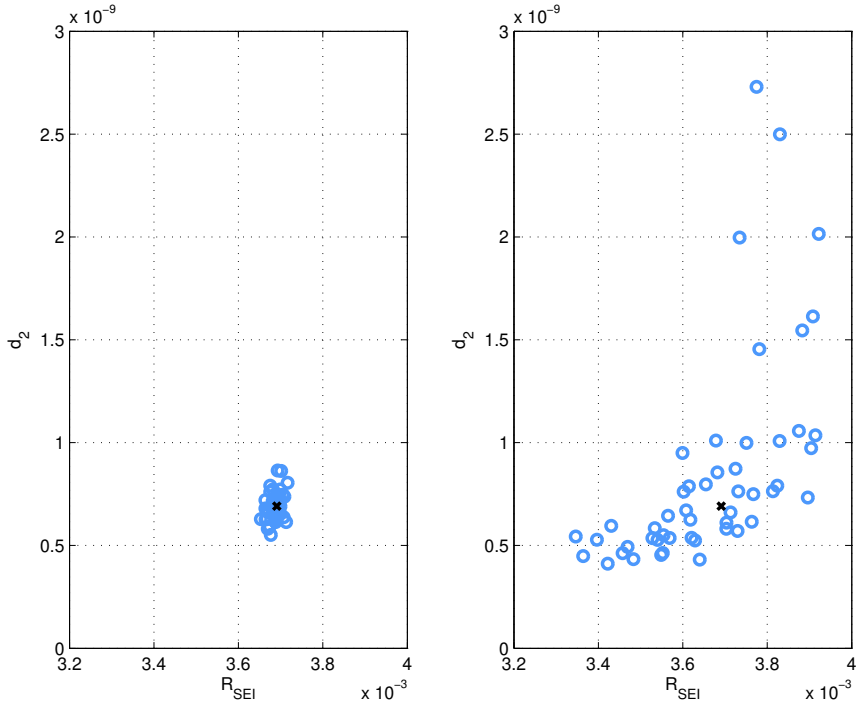

Figure 4.6 Health trajectory pair 1h approximately 1e-6 Amp-Hr

4.4.3 Optimized Trajectories

Of course, one may wonder about the shape, character, and salient features of these optimized trajectories. The optimized trajectories for 2e to 4e are shown in Figures 4.11,4.12,

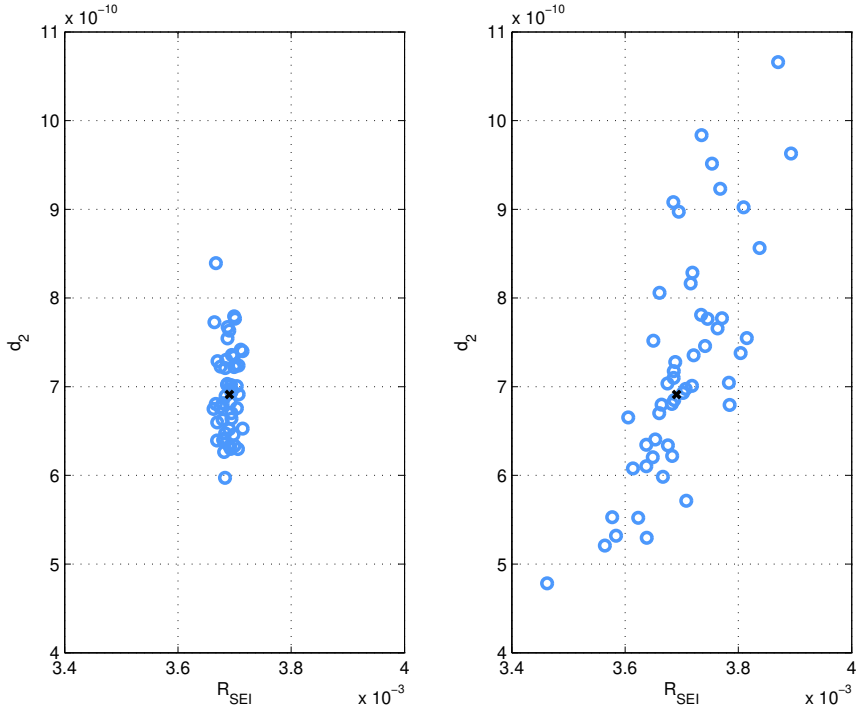

Figure 4.7 Health trajectory pair 2h approximately 2e-6 Amp-Hr

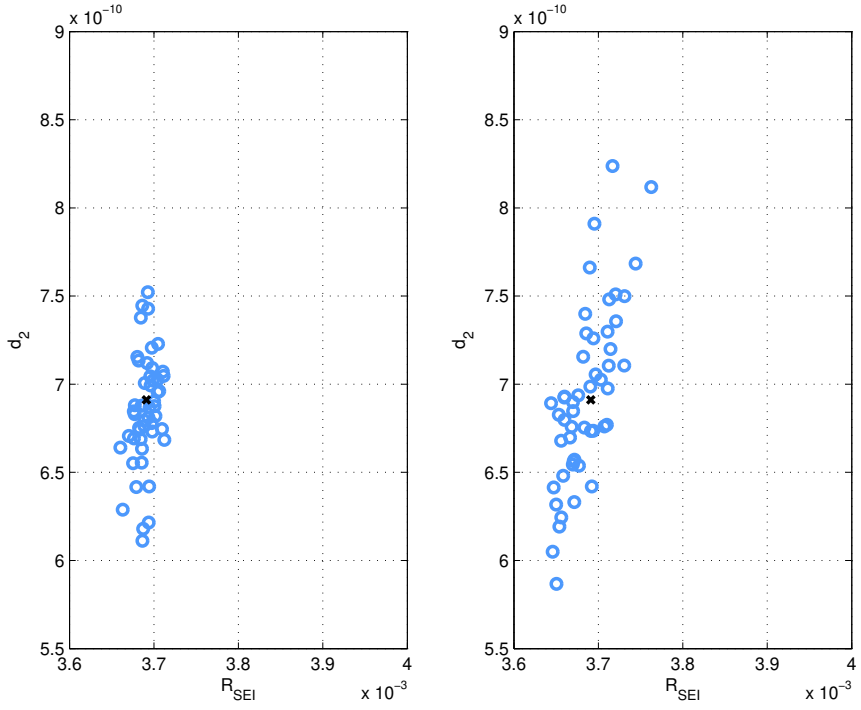

Figure 4.8 Health trajectory pair 3h approximately 4e-6 Amp-Hr

and 4.13. The cases for 1h to 3h are shown in Figures 4.14,4.15, and 4.16. We note that in all cases the trajectories have high frequency components (with respect to the sample rate)

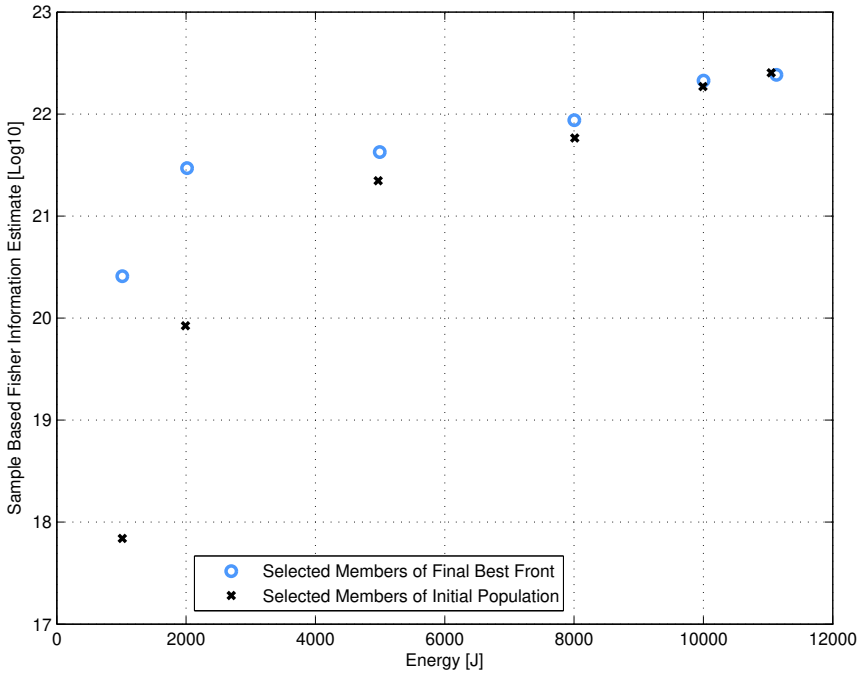

Figure 4.9 Energy estimator Fisher information

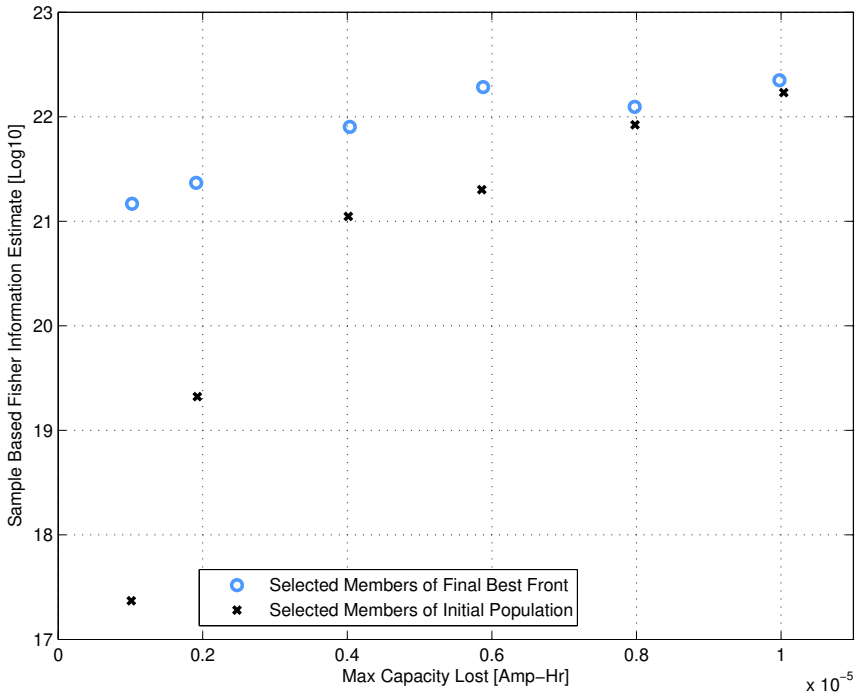

Figure 4.10 Health estimator Fisher information

and that they result in SoC trajectories that look similar to weighted random walks. The lack of smoothness in the trajectories shows that not using splines was appropriate as they would not have been able to accurately capture all of the salient features of the trajectories.

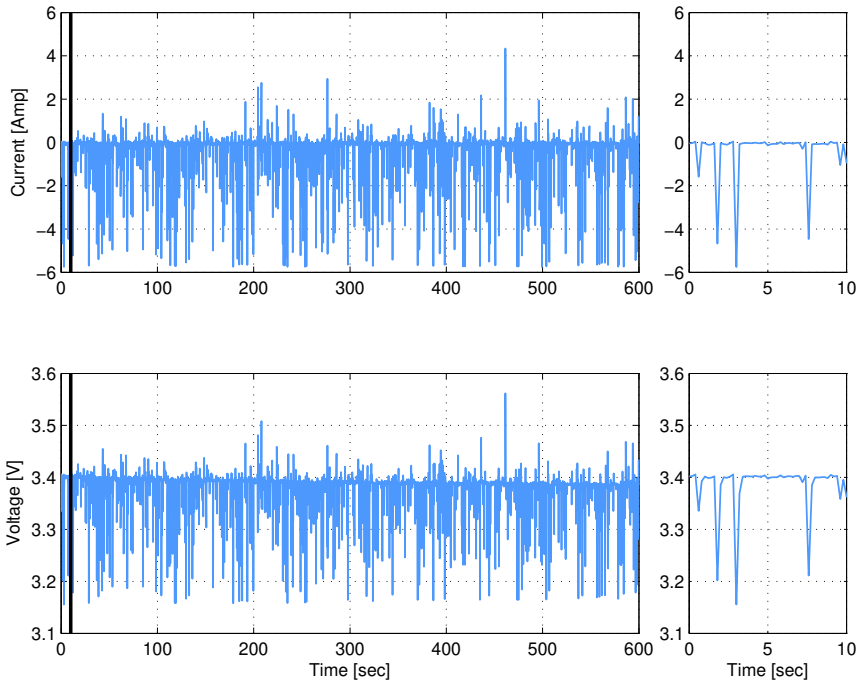

Figure 4.11 Optimized energy trajectory 2e

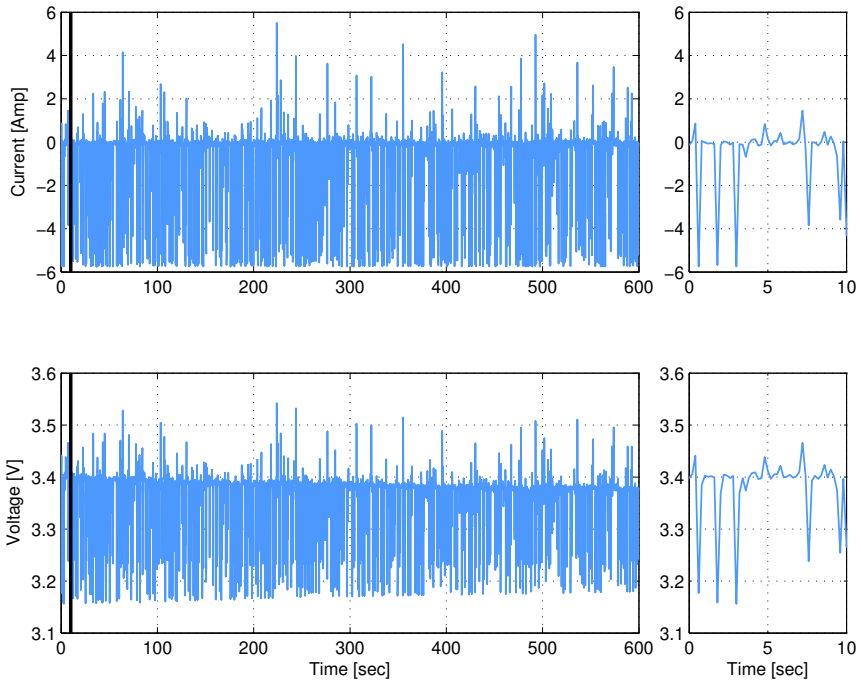

Figure 4.12 Optimized energy trajectory 3e

4.5 Discussion

This paper investigates two specific cases of designing current trajectories to gather information using NSGAII-DE. This can be thought of as demonstrating a general method, in which

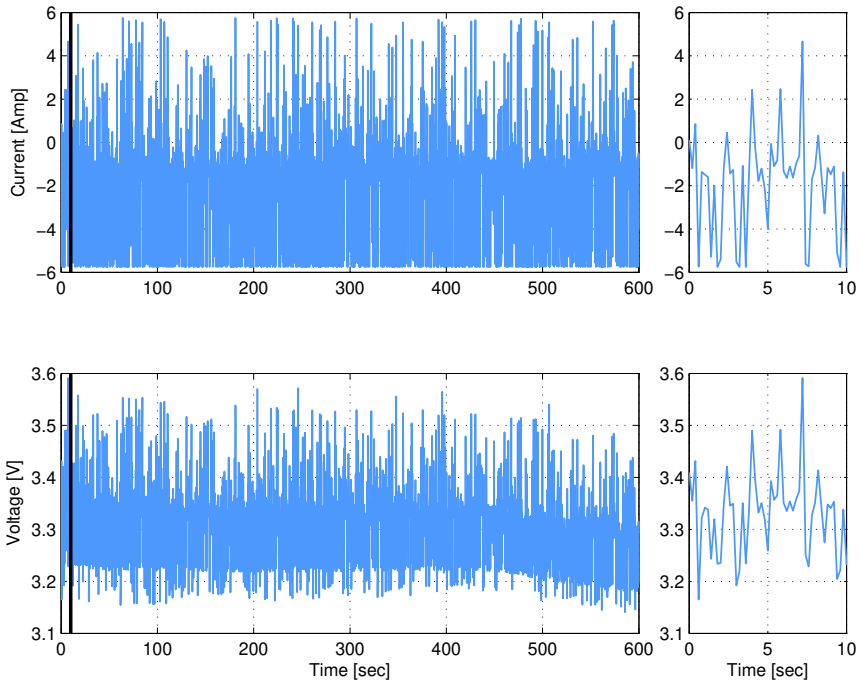

Figure 4.13 Optimized energy trajectory 4e

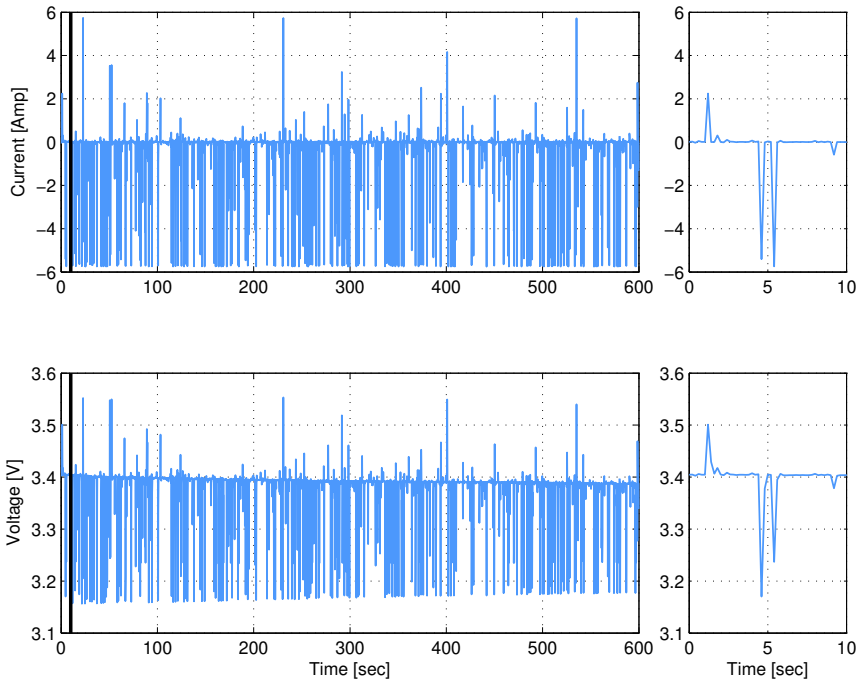

Figure 4.14 Optimized health trajectory 1h

many variations are possible. The most obvious variation is that one could easily change the parameters used in estimation. Additionally, while the DFN model is a very interesting and important model, it is by no means the only model that could be used. For example,

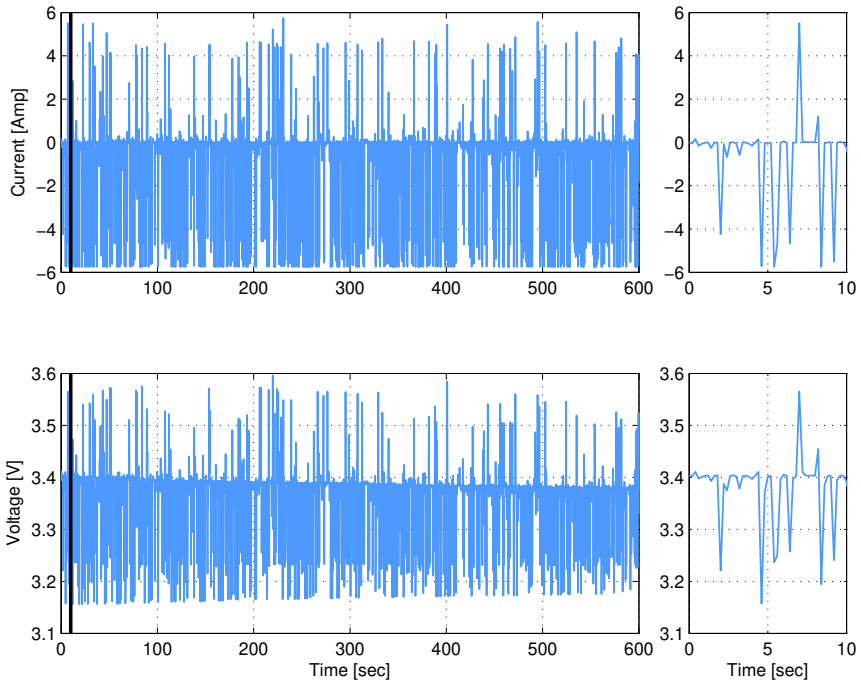

Figure 4.15 Optimized health trajectory 2h

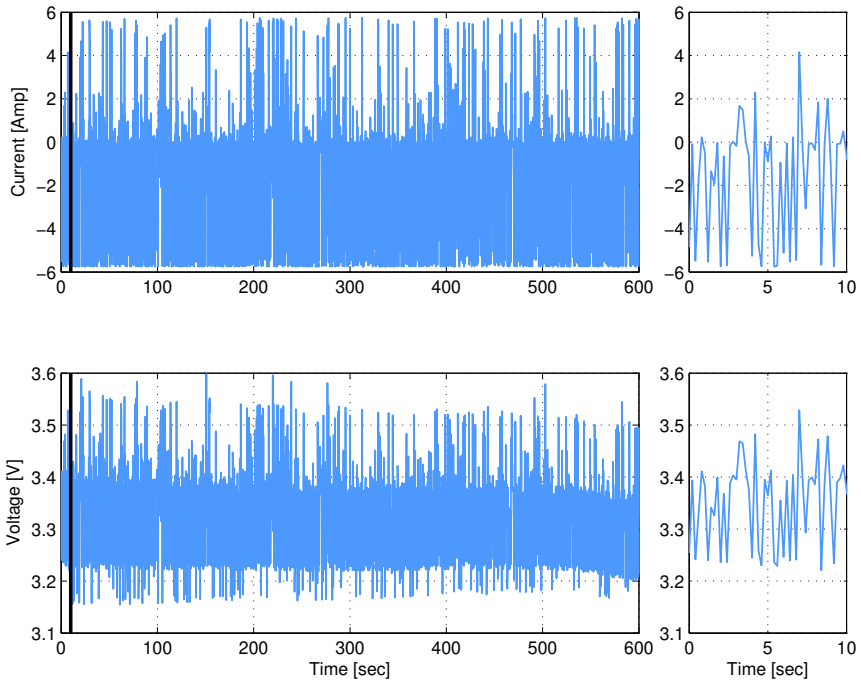

Figure 4.16 Optimized health tsrajectory 3h

if one wanted to further improve the modeling work, the spherical diffusion submodel in the cathode could be replaced with an appropriate submodel that accurately captures its path dependence behavior. For more information on the physics of this see [74–76] and

for modeling see [77–82]. If one is more focused on less complicated models (perhaps for control or optimization applications) then one could replace the DFN model with a single particle models [35–37]. Of course, in addition to changing the parameters or underlying model one can also change some of the inputs and outputs to the model. For example, nuclear magnetic resonance imaging can be used to measure the neutron transmission, which is related to Li concentration, leading to additional outputs [99]. This can be used to observe spatial concentrations of Li throughout a sample of the cell. In this case Fisher information's direct use of sample times may be very beneficial as imaging speed can be very limited. Another, more readily available example of an additional output is that of a third electrode [100]. This would yield two different voltages allowing one to collect greater quantities of information about the cell's parameters.

While the method presented in this chapter works well in simulation the discrepancies between the predicted Fisher information and the Fisher information observed by the estimator require some addressing. To reiterate, we observed similar trends for both, but the estimator based information is orders of magnitude higher than that predicted (this is after normalizing for the signal noise). Thus we suggest that the predicted information is a guide and computational tool. For example, to use the estimator based Fisher information in an optimization one would need a dramatically larger number of simulations. This is because while the predicted information can be computed with 3 simulations (2 perturbations and a center) the estimated information requires running the estimator enough times to create a statistically meaningful cloud of estimations. For the health estimation we needed 104.5 simulations on average per estimation point. Assuming 50 points are needed for estimation (this is a rough number, but in the right order of magnitude), then we would need $50 \times 104.5 = 5225$ simulations for the estimator based Fisher information optimization. Clearly when compared with the 3 simulations needed to get the predicted Fisher information this becomes computationally intractable. With unlimited computing power using the estimator based method would be preferable, but given that these optimizations already take approximately 3 days when using 56 cores the use of predicted Fisher information has great practical value. An interested researcher with (even greater) resources could scale this up and see what the true results would be. There are some practical obstacles to implementing this method. The main one is that it assumes all of the parameters not being estimated are know. This can be overcome by measuring them. One could do this for the battery parameters that should be similar between all of the batteries manufactured and then use this method to estimate health parameters that will drift. An additional question with this method is robustness - how optimal do these trajectories remain if the underlying values change substantially? This is a question of Bayesian estimation which is a more computationally intensive problem [101].

All things considered, the work in this paper is an excellent starting point for further work combining dynamic optimal experimental design with battery modeling and experiments.

4.6 Conclusions

This chapter creates and demonstrates a method for designing maximally informative and minimally invasive input trajectories for battery experiments. This method bridges an important gap between the dynamic optimal experimental design literature and the battery modeling literature. By using NSGAII-DE the method allows one to algorithmically design these experiments making it more efficient to accurately characterize battery parameters. We note that substantial improvements at equal levels of invasiveness were possible in both the predicted information and the information observed by simulating the estimator on data with measurement noise. We believe this method is useful for both directly designing battery experiments and as a foundation for future dynamic optimal experimental design work for battery modeling experiments.

Chapter 5
The Conclusion

5.1 Intellectual Contributions

This dissertation makes three major intellectual contributions which are summarized in this section.

5.1.1 Noninvasive Electrochemical Model Identification using Genetic Algorithms

While the DFN model has applicability to a large class of battery types one major hurdle in using this model is the identification of parameters and parameterized functions [2, 3]. A GA identification method is use to noninvasively identify the parameters and parameterized functions of the DFN model. This GA optimizes using noninvasively collected voltage and current data to fit the model parameters. As a demonstration, laboratory data is collected for a $LiFePO_4$ cell and the identification is carried out successfully. Some of this laboratory data is based on driving cycles simulated for a Prius with a 5 kWh PHEV conversion kit. A validation study of model fit is conducted and a Fisher information study has been conducted to assess parameter accuracy.

5.1.2 Optimal Experimental Design for Battery Health Modeling

This dissertation uses traditional OED to design battery health degradation experiments for determining a voltage and current driven health model [15–17]. The OED selects a set of experimental procedures that minimize the geometric sum of Cramér-Rao variance bounds for estimating the model's parameters [18]. This minimization creates the best possible set of data for performing parameter identification. The designed experiments are conducted in a laboratory to obtain this data-driven health model. This sequence serves as a prototype for

conducting battery health experiments on other types of batteries. Additionally the specific health model can be used for health-conscious controllers, system level design decisions and warrantying decisions [8, 19].

5.1.3 Maximizing Fisher Information through NSGAII-DE for Battery Health Diagnostics

This dissertation investigates using NSGAII-DE to maximize Fisher information to design input trajectories for experiments that minimize Cramér-Rao lower bound on variance for identifying selected parameters. As a test case the author considers designing inputs to the DFN model. This input trajectory minimizes the variance in identifying the resistance of the solid electrolyte interface layer, R_{SEI} which is a major factor in battery health [20–34]. The solution diffusivity, d_2, is also simultaneously identified with R_{SEI}. One specific application of this technique is a diagnostic procedure for determining battery SoH in an efficient manner. The improvements of the optimized trajectories are verified using an estimator on simulated data with noise.

5.1.4 An OED Battery Modeling and Characterization Toolbox

Together these contributions provide a toolbox for battery system engineers. This toolbox allows one to characterize the battery for the DFN model, determine an empirical data driven health model, and then design diagnostics for determining specific battery parameters. This toolbox incorporates methods from OED to ensure that the experiments are efficient and result in appropriate models and parameters. In addition the author demonstrates specific pairings of method and numerical algorithm, showing the reader how to perform the appropriate computations for the experiments. This toolbox should be very helpful to engineers work on the design, control, and optimization of battery systems as it provides efficient and systematic ways to determine the necessary battery models.

5.2 Possible Future Extensions

5.2.1 Future Battery Aging Experiments

There are a few possible extensions for the battery aging experiment discussed in Chapter 3. First, one open question with this work is that perhaps the positive and negative current

effects on aging could have been separated by using differing charge and discharge current limits. This could be explored by designing an experiment where these rates were allowed to differ and then conducting the experiment. Additionally higher charge and discharge rates could be investigated. Another improvement would be to design an experiment for dealing with temperature effects. An OED exercise could be conducted based on the Arrhenius equation for this case.

Another area of interest would be to use cycles that did not have the CCCV form. Specifically it would be interesting to use PHEV or EV based drive cycles with overnight charging patterns. If this was based on a large set of drive cycles stochasticity could be incorporated by using the Linear Matrix Inequality Interior Point algorithm. This would be useful for vehicle management as there would be no logical jump from the type of cycle used.

5.2.2 Trajectory Optimization Experiments

The trajectory optimization work in Chapter 4 suggests a few possible variations and more extensive follow ons to the work. The most obvious variation is that one could easily change the parameters used in estimation. Additionally, while the DFN model is a very interesting and important model, it is by no means the only model that could be used. For example, if one wanted to further improve the modeling work, the spherical diffusion submodel in the cathode could be replaced with an appropriate submodel that accurately captures its path dependence behavior. If one is more focused on less complicated models (perhaps for control or optimization applications) then one could replace the DFN model with a single particle models [35–37].

The most obvious piece of follow on work would be to conduct the experiment in a laboratory (rather than simulation) environment. To do this one will need accurate parameter values for the DFN model, which will require tearing down the battery or communicating with manufactures. Then one could program the trajectories into appropriate experimental equipment and see how well the method works in a laboratory environment. This would likely result in some valuable insights about the method and promote further fine tuning.

Of course, in addition to changing the parameters or underlying model one can also change some of the inputs and outputs to the model. For example, nuclear magnetic resonance imaging can be used to measure the neutron transmission, which is related to Li concentration, leading to additional outputs [99]. This can be used to observe spatial concentrations of Li throughout a sample of the cell. In this case Fisher information's direct use of sample times may be very beneficial as imaging speed can be very limited. Another, more readily available example of an additional output is that of a third electrode [100]. This

would yield two different voltages allowing one to collect greater quantities of information about the cell's parameters.

Appendices

Appendix A
Model Reduction Techniques

A.1 Numerical Simulation of the Doyle-Fuller-Newman Model

If the DFN model is naively discretized it can easily have a very large numbers of states. As a specific example if one places 100 points across the width of the cell and another 100 between the center and surface of the spherical direction then the model has 6668 states (this can vary slightly on the exact implementation of the discretization). However in practice the actual numerical bottleneck in the model is the algebraic constraint equations. The author has used model reduction to reduce both the number of states and the difficulties encountered with the algebraic constraint equations.

A.1.1 Quasi-Linearization

In the DFN model the electric potentials driving the electrochemical reactions are represented by a coupled nonlinear Boundary Value Problem (BVP). When discretized this results in a set of algebraic constraint equations that creates a numerical bottleneck within the model. To reduce the time spent in this bottleneck the authors quasi-linearize these nonlinear equations. This quasi-linearization is a linearization of the equations about the previous time step and assumes the diffusion dynamics in the DFN model are frozen in time (during the calculation). By using this quasi-linearization one can solve a set of linear equations instead of the previous nonlinear root finding problem. Specific numerical results are presented in [5].

A.1.2 Infinite Dimensional Padé Approximation of Spherical Diffusion

While states in the DFN model are not the major bottleneck, reducing the number of states still has several advantages including: faster computational speeds and making the model easier to work with. The author has approached this problem by reducing a repeating submodel within the DFN model: spherical diffusion of Li. This approach applies Padé approximation to an infinite-dimensional transfer function representing the spherical diffusion submodel. This results in an analytical model reduction that dramatically reduced the number of states in the DFN model by 98% without significant decreases in accuracy [5].

A.1.3 Legendre Modal Decomposition

The author has also used modal decomposition similar to that presented by [54] to reduce the DFN model. This modal decomposition uses orthogonal Legendre polynomials along both the width of the cell and the radial direction of the spheres. Along the width of the cell these modes couple the Partial Differential Equation (PDEs) and BVP in a very natural way. This results in both a large reduction in states but more importantly a large reduction in algebraic constraint equations (the numerical bottleneck) greatly improving computation time.

A.1.4 Quasi-Linearization Combined with Legendre Modal Decomposition

The author goes further than [54] by quasi-linearizing these modal constraints resulting in further time savings. Mathematically this is very similar to the original quasi-linearization but instead of acting on a finite differenced model it acts on the model in a coordinate frame of orthogonal Legendre polynomials. This has allowed the DFN model to be used in various optimizations [8, 12, 19].

A.2 Mathematics of Quasi-Linearization Combined with Legendre Modal Decomposition

The overarching reason that Legendre modal decomposition is useful is that the x direction in the DFN model can be described by them unifying the PDEs and AEs which both naturally

connects them and dramatically decreases the number of states. The remainder of this section is intended for those working on programming the DFN model into this form and as such the discussion is heavy on mathematics and short on words.

A.2.1 The Projection

Throughout this section we use the Legendre modes as a basis. Where $Leg_i(x)$ is the i^{th} mode. A function f can be composed into the basis as follows:

$$f(x) = \sum_{i=0}^{\infty} \langle f(x), Leg_i(x) \rangle \, Leg_i(x). \tag{A.1}$$

For notational reasons we define e_i^x the i^{th} basis over the x direction.

$$e_i^x(x) = Leg_{i+1}(x) \tag{A.2}$$

$$f(x) \approx \sum_{i=1}^{N_x} \langle f(x), e_i^x(x) \rangle \, e_i^x(x) = \begin{pmatrix} \langle f(x), e_1^x(x) \rangle \\ \vdots \\ \langle f(x), e_{N_x}^x(x) \rangle \end{pmatrix} \tag{A.3}$$

Where N_x is the number of basis vectors used in the x direction. We note that the above equations map a function to a vector. For further information on the Legendre polynomials please see [102].

A.2.2 Singularly Perturbed Systems

For the readers familiarity we use the following common method from the theory of linear singularly perturbed systems [68].

Consider the following singularly perturbed system:

$$\begin{pmatrix} \dot{\vec{x}} \\ 0 \end{pmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} \vec{x} \\ \vec{w} \end{pmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \tag{A.4a}$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{pmatrix} \vec{x} \\ \vec{w} \end{pmatrix} D_1 u. \tag{A.4b}$$

Which can be written as (provided A_{22} is invertible):

$$\dot{\vec{x}} = \left[A_{11} - A_{12}A_{22}^{-1}A_{21}\right]x + \left[B_1 - A_{12}A_{22}^{-1}B_2\right]u \tag{A.5a}$$

$$y = \left[C_1 - C_2A_{22}^{-1}A_{21}\right]x + \left[D_1 - C_{12}A_{22}^{-1}B_2\right]. \tag{A.5b}$$

Where x and w can be recovered as follows:

$$\begin{pmatrix}\vec{x}\\ \vec{w}\end{pmatrix} = \begin{bmatrix} I \\ -A_{22}^{-1}A_{21}\end{bmatrix}x + \begin{bmatrix} 0 \\ -A_{22}^{-1}B_2\end{bmatrix}u. \tag{A.6}$$

A.2.3 An Example Problem

Here we demonstrate an example of converting a PDE into a DAE using Legendre modal decomposition. This example is for a linear diffusion model, with a time and space varying input. This model is closely related to several submodels with in the DFN model.

The mathematical model is as follows:

$$\frac{\partial}{\partial t}H(t,z) = \frac{\partial^2}{\partial z^2}H(t,z) + J(t,z). \tag{A.7}$$

With initial condition:

$$H(0,z) = H_0(z). \tag{A.8}$$

With boundary conditions:

$$H(t,-1) = 0, \tag{A.9}$$

$$\frac{\partial H(t,1)}{\partial z} - u(t) = 0. \tag{A.10}$$

With outputs:

$$y_1(t) = \int_{-1}^{1} H(t,z)dz, \tag{A.11}$$

$$y_2(t) = H(t,1). \tag{A.12}$$

We now use modal decomposition to represent the relevant functions.

$$H(t,z) = \sum_{i=1}^{\infty} H_i(t) e_i^z(z) \approx \sum_{i=1}^{N_z} H_i e_i^z(z) \tag{A.13a}$$

$$J(t,z) = \sum_{i=1}^{\infty} J_i(t) e_i^z(z) \approx \sum_{i=1}^{N_z} J_i e_i^z(z) \tag{A.13b}$$

$$e_i^z(z) = Leg_{i+1}(z) \tag{A.13c}$$

The time derivative passes neatly through the LHS of equation A.7:

$$\frac{\partial}{\partial t} H(t,z) \approx \sum_{i=1}^{N_z} \dot{H}_i(t) e_i^z(z). \tag{A.14}$$

The double space derivative in the RHS of equation A.7 is more complicated:

$$A_{i,j} = \left\langle \frac{\partial^2}{\partial z^2} e_i^z(z), e_j^z(z) \right\rangle, \ \forall i \in \{1,\cdots,N_z\}, \ \forall i \in \{1,\cdots,N_z-2\}. \tag{A.15}$$

We note that if j were allowed to go to N_z the last two columns in the A matrix would have entries all identical to zero. For notational purposes we do not add these columns to the matrix, thus $A \in \mathbb{R}^{(N_z)\times(N_z-2)}$.

$$\frac{\partial^2}{\partial z^2} H(t,z) = \sum_{i=1}^{N_z} H_i(t) \frac{\partial^2}{\partial z^2} e_i^z(z) \tag{A.16a}$$

$$= \sum_{i=1}^{N_z} H_i(t) \sum_{j=1}^{N_z-2} \left\langle \frac{\partial^2}{\partial z^2} e_i^z(z), e_j^z(z) \right\rangle e_j^z(z) \tag{A.16b}$$

$$= \sum_{i=1}^{N_z} H_i(t) \sum_{j=1}^{N_z-2} A_{i,j} e_j^z(z) \tag{A.16c}$$

$$= \sum_{j=1}^{N_z-2} \sum_{i=1}^{N_z} A_{j,i} H_j(t) e_j^z(z) \tag{A.16d}$$

Now we use orthogonality to convert equation A.7 from a PDE into a matrix equation. We end short two rows of having a uniquely solvable system. These extra two rows come

from the boundary equations.

$$\left\langle \sum_{i=1}^{N_z} H_i e_i^z(z), e_k^z(z) \right\rangle = \left\langle \sum_{j=1}^{N_z-2} \left(\sum_{i=1}^{N_z} \left(A_{j,i} H_j(t) e_j^z(z) \right) + J_i(t) e_i^z(z) \right), e_k^z(z) \right\rangle \ \forall k \in \{1, \cdots, N_z\} \tag{A.17}$$

$$\dot{H}_i(t) = \sum_{j=1}^{N_z} A_{j,i}^T H_i(t) + J_i(t), \ \forall k \in \{1, \cdots, N_z\} \tag{A.18}$$

Where we note that we've omitted the last two rows of this equation because they are vacuous. In summary we have:

$$\begin{pmatrix} \dot{H}_1(t) \\ \vdots \\ \dot{H}_{N_z-2}(t) \end{pmatrix} = A^T \begin{pmatrix} H_1(t) \\ \vdots \\ H_{N_z}(t) \end{pmatrix} + \begin{pmatrix} J_1(t) \\ \vdots \\ J_{N_z-2}(t) \end{pmatrix}. \tag{A.19}$$

Two additional rows are added to this equation via the boundary conditions. Consider A.9, the Dirichlet (height) constraint.

Define:

$$E_{BC1,i} = e_i^z(-1), \ \forall i \in 1, \cdots, N_z. \tag{A.20}$$

Then,

$$H(t,-1) = \left(\sum_{i=1}^{N_z} H_i(t) e_i^z(z) \right) \Bigg|_{z=-1} \tag{A.21a}$$

$$= \sum_{i=1}^{N_z} H_i(t) e_i^z(-1) \tag{A.21b}$$

$$= E_{BC1} \begin{pmatrix} H_1(t) \\ \vdots \\ H_{N_z}(t) \end{pmatrix}. \tag{A.21c}$$

Thus, the modal form for A.9 is:

$$E_{BC1} \vec{H}(t) = 0. \tag{A.22}$$

Consider A.10, the Neumann (slope) constraint.

$$\frac{\partial H(t,1)}{\partial z} = \frac{\partial}{\partial z}\sum_{i=1}^{N_z} H_i(t)e_i^z(z)\bigg|_{z=1} \tag{A.23a}$$

$$= \left(\sum_{i=1}^{N_z} H_i(t)\frac{\partial}{\partial z}e_i^z(z)\right)\bigg|_{z=1} \tag{A.23b}$$

$$= \left(\sum_{i=1}^{N_z} H_i(t)\sum_{j=1}^{N_z-1}\left\langle \frac{\partial}{\partial z}e_i^z(z), e_j^z(z)\right\rangle e_j^z(z)\right)\bigg|_{z=1} \tag{A.23c}$$

$$= \left(\sum_{i=1}^{N_z-1}\sum_{j=1}^{N_z} H_j(t)\left\langle \frac{\partial}{\partial z}e_j^z(z), e_i^z(z)\right\rangle e_i^z(z)\right)\bigg|_{z=1} \tag{A.23d}$$

$$= \left(\sum_{i=1}^{N_z-1}\sum_{j=1}^{N_z} H_j(t)\left\langle \frac{\partial}{\partial z}e_j^z(z), e_i^z(z)\right\rangle e_i^z(1)\right) \tag{A.23e}$$

$$= \sum_{i=1}^{N_z-1} E_{BC2,j}H_j(t) \tag{A.23f}$$

$$\tag{A.23g}$$

Where:

$$E_{BC2,j} = \sum_{i=1}^{N_z-1}\left\langle \frac{\partial}{\partial z}e_j^z(z), e_i^z(z)\right\rangle e_i^z(1) \;\; \forall j \in \{1,\cdots,N_z\} \tag{A.24}$$

This gives us the modal form:

$$E_{BC2}\vec{H}(t) + B_{BC2}u = 0. \tag{A.25}$$

Where:

$$B_{BC2} = -1. \tag{A.26}$$

Now we consider the output y_1 of this model.

$$y_1(t) = \int_{-1}^{1} e_i^z(z)dz \tag{A.27a}$$

$$= \int_{-1}^{1} \sum_{i=1}^{N_z} H_i(t) e_i^z(z)dz \tag{A.27b}$$

$$= \sum_{i=1}^{N_z} H_i(t) \int_{-1}^{1} e_i^z(z)dz \tag{A.27c}$$

$$= \sum_{i=1}^{N_z} H_i(t) C_{1,i} \tag{A.27d}$$

Where:

$$C_{1,i} = \int_{-1}^{1} e_i^z(z)dz \;\; \forall i \in \{1, \cdots, N_z\} \tag{A.28}$$

Now the modal form for y_1 is:

$$y_1(t) = C_1 \vec{H}(t) \tag{A.29}$$

Now we consider the second output, y_2.

$$y_2(t) = \left(\sum_{i=1}^{N_z} H_i(t) e_i^z(z) \right) \Bigg|_{z=1} \tag{A.30a}$$

$$= \sum_{i=1}^{N_z} H_i(t) e_i^z(1) \tag{A.30b}$$

$$= \sum_{i=1}^{N_z} H_i(t) C_{2,i} \tag{A.30c}$$

Where:

$$C_{2,i} = e_i^z(1) \;\; \forall i \in \{1, \cdots, N_z\}. \tag{A.31}$$

And thus, the modal form is:

$$y_2(t) = C_2 \vec{H}(t). \tag{A.32}$$

This leads us to a modal form for the whole model.

$$\begin{pmatrix} \dot{H}_1(t) \\ \vdots \\ \dot{H}_{N_z-2}(t) \\ 0_{2\times 1} \end{pmatrix} = \begin{bmatrix} A^T \\ E_{BC1} \\ E_{BC2} \end{bmatrix} \vec{H}(t) + \begin{bmatrix} I_{(N_z-2)\times(N_z-2)} & 0_{(N_z-2)\times 1} \\ 0_{1\times(N_z-2)} & 0_{1\times 1} \\ I_{(N_z-2)\times 1} & B_{BC2} \end{bmatrix} \begin{pmatrix} J_1(t) \\ \vdots \\ J_{N_z-2}(t) \\ u(t) \end{pmatrix} \tag{A.33a}$$

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \vec{H}(t) \tag{A.33b}$$

Which can be further reduced using singular perturbation.

This serves as an analogy for many of the pieces within the DFN model. The main piece bearing further explanation is the 2D modal reduction for spherical diffusion. We discuss this next and how the individual pieces connect together later.

A.2.4 Spherical Diffusion

Spherical diffusion is similar to the example except for two major differences. First it occurs in two spatial dimensions: r and x. Second, due to symmetry, only even orders of the Legendre polynomials are relevant. Throughout this section we use the $\dagger$ symbol as $\forall \dagger \in n, p$ in order to work through spherical diffusion in the anode and cathode simultaneously.

The model in our case is:

$$\frac{\partial c_1(t,x,r)}{\partial t} = \frac{d_{1\dagger}}{r^2}\frac{\partial}{\partial r}\left(\frac{r^2 \partial c_1(t,x,r)}{\partial r}\right). \tag{A.34}$$

With boundary condition:

$$\frac{\partial c_1(t,x,R_\dagger)}{\partial r} - \frac{1}{d_{1\dagger}a_\dagger F} J_1(t,x) = 0, \;\; @r = R. \tag{A.35}$$

There is only one boundary condition due to symmetry.

With outputs:

$$\vec{y}(t,x) = \begin{pmatrix} c_S(t,x) \\ c_{SoC}(t,x) \end{pmatrix} = \begin{pmatrix} c_1(t,x,r)|_{r=R_\dagger} \\ \frac{3}{c_{\dagger,\max}} \int_0^{R_\dagger} c_1(t,x,r) r^2 dr \end{pmatrix}. \tag{A.36}$$

These outputs interface directly with the DFN model. For notational convenience the

remainder of this section we use $[\bullet]$ to indicate matrix subscripts.

$$c_1(t,x,r) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_r} C_1[i,j](t)e_i^x(x)e_j^r(r) \tag{A.37}$$

Where:

$$e_j^r(r) = Leg_{2(j-1)}(r), \tag{A.38a}$$
$$e_i^x(x) = Leg_{i+1}(x). \tag{A.38b}$$

As in the example, the time derivative passes neatly into the modal form.

$$\frac{\partial}{\partial t}c_1(t,x,r) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_r} \dot{C}_1[i,j](t)e_i^x(x)e_j^r(r) \tag{A.39}$$

Similar to the example we define a matrix $A_\dagger$.

$$\frac{d_{1\dagger}}{r^2}\frac{\partial}{\partial r}\left(\frac{r^2\partial c_1(t,x,r)}{\partial r}\right) = \left(\frac{d_{1\dagger}}{r^2}\frac{\partial}{\partial r}\frac{r^2\partial}{\partial r}\right)\sum_{i=1}^{N_x}\sum_{j=1}^{N_r} C_1[i,j](t)e_i^x(x)e_j^r(r) \tag{A.40a}$$
$$= \sum_{i=1}^{N_x} C_1[i,j](t)e_i^x(x)\sum_{j=1}^{N_r}\left(\frac{d_{1\dagger}}{r^2}\frac{\partial}{\partial r}\frac{r^2\partial}{\partial r}\right)e_j^r(r) \tag{A.40b}$$
$$= \sum_{i=1}^{N_x} C_1[i,j](t)e_i^x(x)\sum_{j=1}^{N_r}\sum_{k=1}^{N_r-1}\left\langle\left(\frac{d_{1\dagger}}{r^2}\frac{\partial}{\partial r}\frac{r^2\partial}{\partial r}\right)e_j^r(r), e_k^r(r)\right\rangle e_k^r(r) \tag{A.40c}$$
$$= \sum_{i=1}^{N_x} C_1[i,j](t)e_i^x(x)\sum_{j=1}^{N_r}\sum_{k=1}^{N_r-1} A_\dagger[j,k]e_k^r(r) \tag{A.40d}$$
$$= \sum_{i=1}^{N_x}\sum_{j=1}^{N_r-1}\sum_{k=1}^{N_r} C_1[i,k](t)A_\dagger[k,j]e_i^x(x)e_j^r(r) \tag{A.40e}$$

Then by orthogonality in the x and r directions:

$$\dot{C}_1[i,j](t) = \sum_{k=1}^{N_r} C_1[i,k](t)A_\dagger[k,j]. \tag{A.41}$$

Where:

$$A_\dagger[j,k] = \left\langle\left(\frac{d_{1\dagger}}{r^2}\frac{\partial}{\partial r}\frac{r^2\partial}{\partial r}\right)e_j^r(r), e_k^r(r)\right\rangle \;\; \forall j \in \{1,\cdots,N_r\} \textit{ and } k \in \{1,\cdots,N_r-1\}. \tag{A.42}$$

In order to stack this neatly into a matrix equation we remap the indexing of C_1 as follows:

$$C_1[i,j](t) \to C_1\{i+(j-1)(N_r-1)\}(t) \tag{A.43}$$

Where we use $\{\bullet\}$ to indicate the vector indexing. This allows us to stack the equations similar to A.33:

$$\begin{pmatrix} \dot{C}_1\{1\}(t) \\ \vdots \\ \dot{C}_1\{(N_x-2)(N_r-1)\}(t) \end{pmatrix} = \begin{bmatrix} A_\dagger^T & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_\dagger^T \end{bmatrix} \begin{pmatrix} C_1\{1\}(t) \\ \vdots \\ C_1\{(N_x-2)N_r\}(t) \end{pmatrix}. \tag{A.44}$$

Where the N_x-2 term arises due to how the spherical diffusion problem fits with the other problems through the unified basis in the x direction, as is discussed later in this appendix.

Again we have to work with the boundary condition. We work with the first term of A.35 and then the second term.

$$\left.\frac{\partial c_1(t,x,R_\dagger)}{\partial r}\right|_{r=R} = \left.\frac{\partial}{\partial r}\sum_{i=1}^{N_x}\sum_{j=1}^{N_r} C_1[i,j](t)e_i^x(x)e_j^r(r)\right|_{r=R} \tag{A.45a}$$

$$= \left.\sum_{i=1}^{N_x}\sum_{j=1}^{N_r} C_1[i,j](t)e_i^x(x)\frac{\partial}{\partial r}e_j^r(r)\right|_{r=R} \tag{A.45b}$$

$$= \left.\sum_{i=1}^{N_x}\sum_{j=1}^{N_r} C_1[i,j](t)e_i^x(x)\sum_{k=1}^{N_r-1}\left\langle \frac{\partial}{\partial r}e_j^r(r), e_k^r(r)\right\rangle e_k^r(r)\right|_{r=R} \tag{A.45c}$$

$$= \sum_{i=1}^{N_x}\sum_{j=1}^{N_r} C_1[i,j](t)e_i^x(x)\sum_{k=1}^{N_r}\left\langle \frac{\partial}{\partial r}e_j^r(r), e_k^r(r)\right\rangle e_k^r(R) \tag{A.45d}$$

$$= \sum_{i=1}^{N_x}\sum_{j=1}^{N_r} C_1[i,j](t)e_i^x(x)E_\dagger[j] \tag{A.45e}$$

Where:

$$E_\dagger[j] = \sum_{k=1}^{N_r}\left\langle \frac{\partial}{\partial r}e_j^r(r), e_k^r(r)\right\rangle e_k^r(R). \tag{A.46}$$

Now we work through the second term of A.35.

$$-\frac{1}{d_{1\dagger}a_\dagger F}J_1(t,x) \approx -\frac{1}{d_{1\dagger}a_\dagger F}\sum_{i=1}^{N_x} J_1[i](t)e_i^x(x) \tag{A.47a}$$

$$= B_\dagger \sum_{i=1}^{N_x} J_1[i](t)e_i^x(x) \tag{A.47b}$$

$$\tag{A.47c}$$

Where:

$$B_\dagger = -\frac{1}{d_{1\dagger}a_\dagger F}. \tag{A.48}$$

After making use of orthogonality this gives us:

$$\sum_{j=1}^{N_r} C_1[i,j]E[j] + B_\dagger J_1[i] = 0 \;\; \forall i \in \{1,\cdots,N_x-2\}. \tag{A.49}$$

Again with re-indexing this makes for a set of matrix equations amenable to singular perturbation:

$$\begin{pmatrix} \dot{C}_1\{1\}(t) \\ \vdots \\ \dot{C}_1\left\{(N_x-2)(N_r-1)\right\}(t) \\ 0_{(N_x-2)\times 1} \end{pmatrix} = \begin{bmatrix} A_\dagger^T & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_\dagger^T \\ E_\dagger^T & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & E_\dagger^T \end{bmatrix} \begin{pmatrix} C_1\{1\}(t) \\ \vdots \\ C_1\left\{(N_x-2)N_r\right\}(t) \end{pmatrix} \cdots + \begin{bmatrix} 0_{(N_x-2)(N_r-1)} \\ B_\dagger \\ \vdots \\ B_\dagger \end{bmatrix} \begin{pmatrix} J_1[1] \\ \vdots \\ J_1[N_x] \end{pmatrix}. \tag{A.50}$$

A.2.5 Unifying Basis

The key insight in using the Legendre modal decomposition is that the PDEs and AEs within the DFN equation are spatially aligned with each other. This lets one use a common basis

in the x direction for all of the submodels. This means that the models can be connected modally as well. Consider how the functions $c_1(t,x,r)$, $c_2(t,x)$, $\phi_1(t,x)$, $\phi_2(t,x)$, $J_{main}(t,x)$, $J_{side}(t,x)$, $J_(t,x)$, and $\delta_{film}(t,x)$ all have an argument in x. Granted one has to align each of these over the correct anode, separator, or cathode domain. Additionally one has to tie the boundary conditions together for several of the submodels at the anode/separator and separator/cathode interface. However even despite these nuances one is able to dramatically reduce the overall number of states and the number of algebraic equations that have to be solved at each time step.

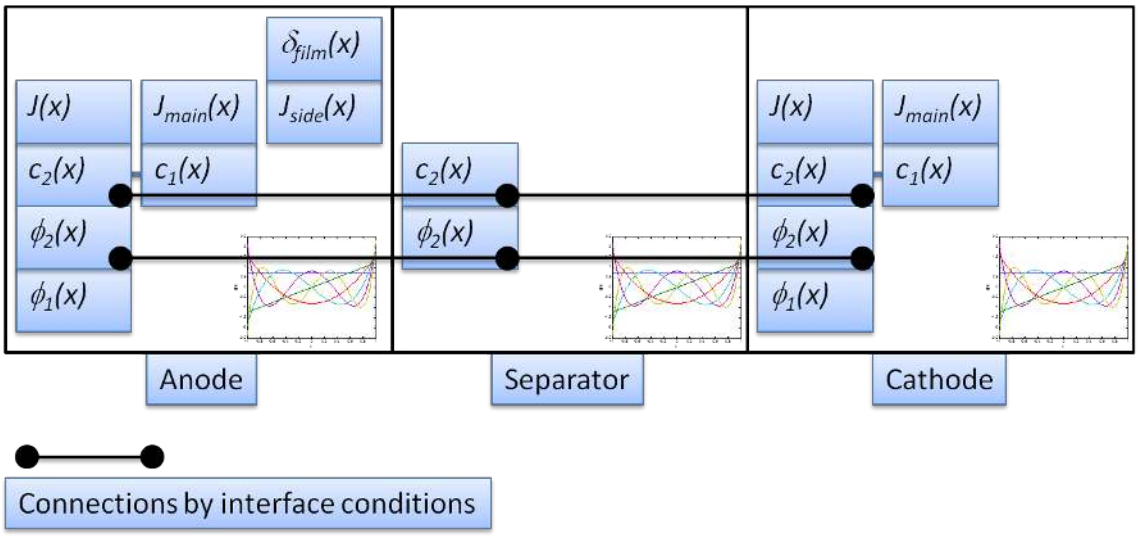

Figure A.1 Basis unification of Legendre modes for the DFN model

A.2.6 Linear Diffusion

This follows the example problem and spherical diffusion problem very closely. The only difference is that each domain (anode, separtor, and cathode) is treated as a separate are for modal decomposition and then they are linked through he boundary conditions.

A.2.7 Quasi-Linearization

This is the same as from [5] except the coordinates are in modal form.

A.2.8 Numerical Solution

Numerical solution is very similar to that in [5]. Namely see figure A.2.

A.2.9 Unifying Basis

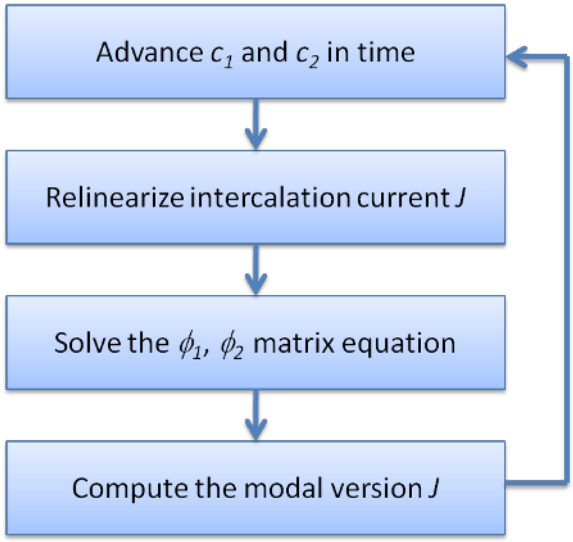

Figure A.2 Numerical solution process for Legendre modal model with quasi-linearization

A.2.10 Advantages

The main advantages are a reduction in the number of states, including a large reduction in the number of states related to the numerical bottle neck that is the constraint equations. In practice one can run the model very accurately with $N_x = N_r = 6$. This leads to 5^{th} order Legendre polynomials in the x direction for the anode, separator, and cathode. In the r direction the polynomials continue to the $10th$ order - this is due to only even order polynomials being used. This results in a total of 56 states and 30 constraint equations. Due to this the model can be simulated in MATLAB at speeds 2 to 3 orders of magnitude faster than real time. This has enabled much of the optimization conducted within this thesis.

Table A.1 Modeling complexity versus N_x and N_r

	N_x	3	4	5	**6**	7	8	9	10	11
N_r		Number of Differential Equations:								
2		6	12	18	24	30	36	42	48	54
3		8	16	24	32	40	48	56	64	72
4		10	20	30	40	50	60	70	80	90
5		12	24	36	48	60	72	84	96	108
6		14	28	42	**56**	70	84	98	112	126
		Number of Constraint Equations:								
		15	20	25	**30**	35	40	45	50	55

Appendix B
Fisher Information for Dynamic Models

Fisher information can be used with dynamic models to determine how accurately model parameters can be estimated [16, 17, 37, 41–44]. This theory is introduced using a two parameter one output dynamic model. Consider:

$$\dot{x} = -p_2 x \tag{B.1a}$$
$$y = x + error; error \sim N\left(0, \sigma^2\right) \tag{B.1b}$$
$$x_0 = p_1; p_2 > 0. \tag{B.1c}$$

This system's dynamics are trivial to solve for:

$$x(t) = p_1 e^{-p_2 t}. \tag{B.2}$$

The two parameter's output sensitivity functions are:

$$s_{p_1}(t) = \frac{\partial}{\partial p_1} x(t) = e^{-p_2 t}, \tag{B.3a}$$
$$s_{p_2}(t) = \frac{\partial}{\partial p_2} x(t) = -p_1 e^{-p_2 t}. \tag{B.3b}$$

These sensitivities are dependent on the parameter value and time. The sensitivities can then be sampled at various points in time $t_i \in \{t_1, \cdots, t_n\}$. This results in a matrix of sampled parameter sensitives:

$$S = \begin{pmatrix} e^{-p_2 t_1} & -p_1 e^{-p_2 t_1} \\ e^{-p_2 t_2} & -p_1 e^{-p_2 t_2} \\ \vdots & \vdots \\ e^{-p_2 t_n} & -p_1 e^{-p_2 t_n} \end{pmatrix}. \tag{B.4}$$

We note $S^T S$'s rank has direct relevance on identifiability. To convert this matrix to the FIM we divide by the variance of measurement error:

$$FIM = \frac{S^T S}{\sigma^2}. \tag{B.5}$$

Then by the Cramér-Rao bound the best possible covariance for a unbiased estimator is:

$$\text{Cov} \geq FIM^{-1}. \tag{B.6}$$

To demonstrate how Fisher information relates to parameter estimation a numerical example is now presented using $p_1 = 2$, $p_2 = 3$, $\sigma = 0.25$. 21 samples at 10Hz are collected for each of 1000 realizations. An example realization is shown in Fig B.1. The identifications for the 1000 realizations are shown in Fig. B.2 overlayed with the standard deviations predicted by the Fisher information. Note that the Fisher information is computed analytically and *a priori*.

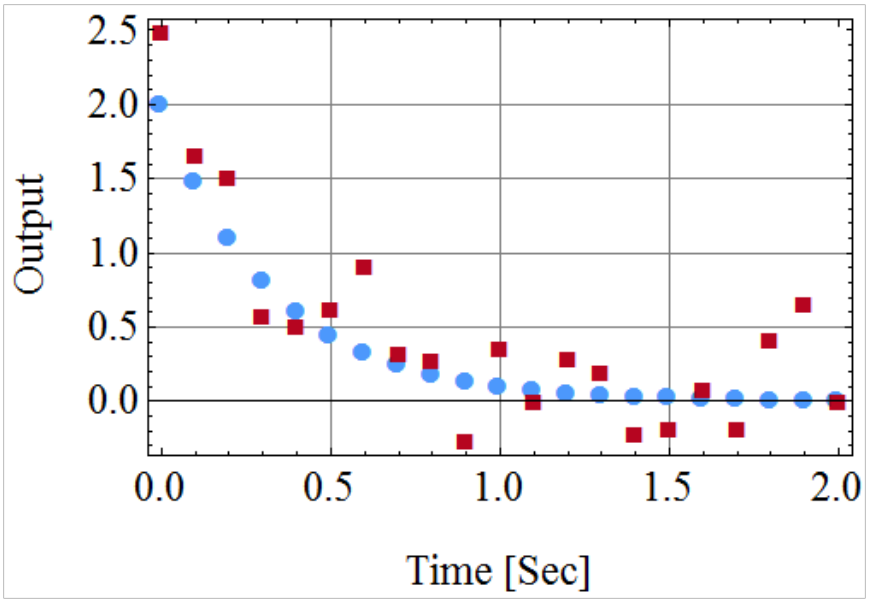

Figure B.1 A realization of a numerical experiment involving Fisher information

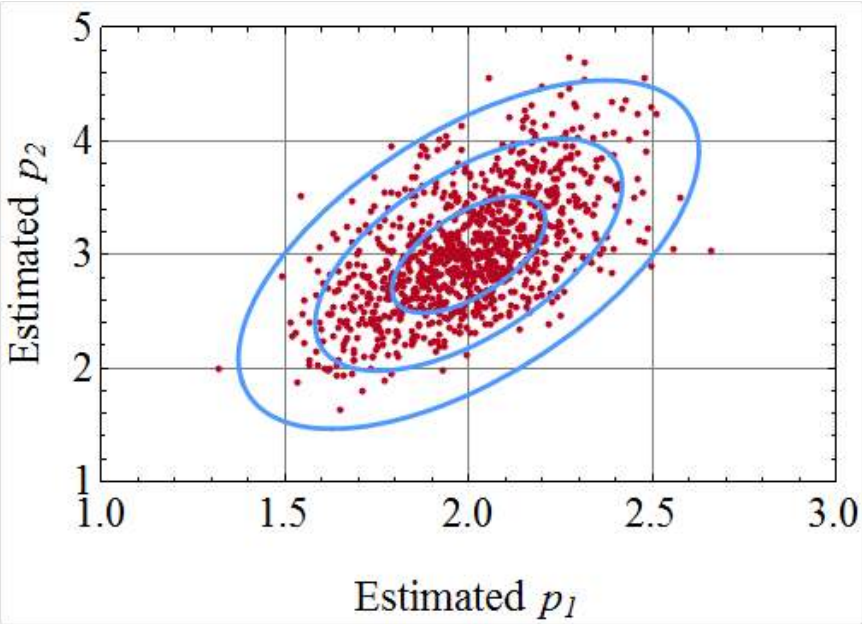

Figure B.2 Predicted and actual results of identifying parameters in a numerical experiment. Ellipses are 1st, 2nd and 3rd standard deviations of predicted estimation

Appendix C

Boundary Conditions for the DFN Model

This appendix summarizes the boundary conditions present in the DFN model. The effective values of various quantites are related to the volume fraction and the Brugman, b, number as follows:

$$d_2^{eff} = d_2 \varepsilon_{2j}^b \;\; \forall j \in \{n,s,p\}, \tag{C.1}$$

$$\sigma_j^{eff} = \sigma \varepsilon 1 j \;\; \forall j \in \{n,p\}, \tag{C.2}$$

$$\kappa^{eff}(c_2) = \kappa(c_2)\, \varepsilon_{2j}^b \;\; \forall j \in \{n,s,p\}, \tag{C.3}$$

$$\kappa_D(c_2) = \kappa^{eff}(c_2) \left(\frac{2\bar{R}T(1-t^+)}{F} \right). \tag{C.4}$$

For solid concentration c_1:

$$\frac{\partial}{\partial r} c_1(r) = 0, \;\; @r = 0, \tag{C.5}$$

$$\frac{\partial}{\partial r} c_1(r) = -\frac{J_j(r)}{d_1 j a_j F}, \;\; @r = R, \;\; \forall j \in \{n,p\}. \tag{C.6}$$

For solution concentration c_2:

$$\frac{\partial}{\partial x} c_2(x) = 0, \;\; @x = 0, x = L_n + L_s + L_p, \tag{C.7}$$

$$d_2^{eff}|_{anode} \frac{\partial}{\partial x} c_2(x) = d_2^{eff}|_{separator} \frac{\partial}{\partial x} c_2(x) \;\; @x = L_n, \tag{C.8}$$

$$d_2^{eff}|_{separator} \frac{\partial}{\partial x} c_2(x) = d_2^{eff}|_{cathode} \frac{\partial}{\partial x} c_2(x) \;\; @x = L_n + L_s. \tag{C.9}$$

In addition c_2 is equated at the anode/separator interface and the separator/cathode interface.

For solid potential ϕ_1:

$$\frac{\partial}{\partial x}\phi_1(x) = 0, \;\; x = L_n, \;\; x = L_n + L_s, \tag{C.10}$$

$$\phi_1(x) = 0, \;\; @x = 0. \tag{C.11}$$

This locates the ground at the negative electrode. At the positive electrode one can choose between a current or voltage input. For the current input:

$$\frac{\partial}{\partial x}\phi_1(x) = \frac{i_{app}}{Area\sigma_n^{eff}}, \;\; @x = L_n + L_s + L_p, \tag{C.12}$$

where *Area* converts from current density to absolute current.

For the voltage input:

$$\phi_1(x) = Volt_{app}, \;\; @x = L_n + L_s + L_p. \tag{C.13}$$

For the solution potential ϕ_2:

$$\frac{\partial}{\partial x}\phi_2(x) = 0, \;\; @x = 0, \;\; x = L_n + L_s + L_p, \tag{C.14}$$

$$\begin{aligned} &\kappa(c_2)|_{anode}\frac{\partial}{\partial x}\phi_2(x) + \kappa_D(c_2)|_{anode}\frac{\partial}{\partial x}\ln(c_2(x)) = \\ &\kappa(c_2)|_{separator}\frac{\partial}{\partial x}\phi_2(x) + \kappa_D(c_2)|_{separator}\frac{\partial}{\partial x}\ln(c_2(x)), \;\; @x = L_n, \end{aligned} \tag{C.15}$$

$$\begin{aligned} &\kappa(c_2)|_{separator}\frac{\partial}{\partial x}\phi_2(x) + \kappa_D(c_2)|_{separator}\frac{\partial}{\partial x}\ln(c_2(x)) = \\ &\kappa(c_2)|_{cathode}\frac{\partial}{\partial x}\phi_2(x) + \kappa_D(c_2)|_{cathode}\frac{\partial}{\partial x}\ln(c_2(x)), \;\; @x = L_n + L_s. \end{aligned} \tag{C.16}$$

In addition ϕ_2 is equated at the anode/separator interface and the separator/cathode interface.

Appendix D

Battery Testing Procedures

D.1 Procedures

This section outlines common procedures that are useful for conducting the battery experiment. They are intended as check lists. These are intended as suggestions and one should not follow them blindly.

D.1.1 Power On

- ☐ In rare instances you may need to plug the equipment back in. This would be typical after a long planned power outage to avoid damage from an electrical surge. Make sure to properly twist the plugs into place.
- ☐ Ensure that the Arbin machine is turned off.
- ☐ Turn on the computer.
- ☐ Start MITS-Pro and manually turn off all of the jobs in the scheduler (you need to do this even though every channel should be in the communcation error mode).
- ☐ Turn the Arbin tester back on. If the tester does not turn on check the breaker switch on the back of the machine.
- ☐ Restart all of the jobs in the scheduler. Individuals should restart their own jobs.

The reason for stopping all of the jobs before turning the Arbin tester on is to avoid cases where a battery starts at a constant voltage. This avoids current spikes caused by step changes in voltage. It is suggested that individuals restart their own jobs as they have the best insight into whether the power outage may have caused problems for their batteries.

D.1.2 Experiment Check In

- ☐ Start remote login and connect to the computer's IP address.
- ☐ Open the scheduler.
- ☐ Stop any unsafe channels and email their owners.

- ☐ Check the step times and ensure that they are reasonable. Occasionally batteries will be stuck in a constant current mode where no current is being applied. If this is the case restart the channel or email the owner.
- ☐ Disconnect from the machine. DO NOT LOG OUT, THIS WILL STOP ALL OF THE EXPERIMENTS.

D.1.3 Experiment Turn Over

This is specific to the experiment I'm conducting for this thesis. One can change this for their own work as needed.

- ☐ Set all the schedules back to JF_CCCVa_120323a.
- ☐ Check the timer in the three places in the file. The last time in the file is the end time. The other two times in the file should be 30 hours sooner than this time (see steps 9 and 21 in the schedule).
- ☐ Start all of the tests. "JF_CCCV_ <Year><Month><Day>_Chan<Channel Number>.
- ☐ Compress the battery data into a .zip file (making sure not to pick files that are currently running).
- ☐ Upload the zip file to Amazon cloud drive.

D.1.4 Data Post Processing

- ☐ Download to steinlab3 from cloud.
- ☐ Reorder files for health if needed (look at 'i' and 'a' files).
- ☐ Access .rec $\rightarrow$ .txt (using visual basic code, module 1).
- ☐ Access update history file (using visual basic code, module 2).
- ☐ Open MATLAB version 2010b go to directory C:\Users\steinlab3\Documents\mat11\batt\
- ☐ Run batt_health_aggregatorRunner.m, save as health <date of test end>.mat.
- ☐ Run datapointProc.m, making sure to update the number of files.
- ☐ Run datapointAgg.m, you need to change endFileNumber and increment the size of datapoints and cycles.
- ☐ Update makeSStruct.m with new health file name.

D.2 Arbin Programming

This section outlines how I programmed the Arbin machine to conduct the experiment. Admittedly, while successful, my programming practices may not be the best for this environment. However being able to use only one schedule file is extremely important for effectively maintaining the experiment.

Table D.1 Summary of schedule file

Step	Label	Purpose	Notes
1	OCV	10 second rest	If voltage $\geq$ F_Vmax goto step 6
2	Charge_Ramp	5 second charge ramp	Start of CCCV charge
3	CC_Charge	CC charge at F_Crate	
4	CV_Charge	CV charge at F_Vmax	1800 sec and $\|current\| \geq 50mA$
5	Charge_Rest	10 second rest	
6	Discharge_Ramp	5 second charge ramp	Start of CCCV discharge
7	CC_Discharge	CC discharge at F_CrateNeg	
8	CV_Discharge	CV discharge at F_Vmin	1800 sec and $\|current\| \geq 50mA$
9	While_Loop	Control logic	goto 1 if time is $\leq Time1$
10	PreCapacityTestRest	10 second rest	
11	CC_Capacity_Charge	CC charge	2.5 C-rate
12	TrickleHigh	CV charge at 3.6V	1800 sec and $\|current\| \geq 50mA$
13	OCV1	10 second rest	Health test cycle starts
14	CC_Dis1	CC discharge at 0.5 C-rate	
15	CV_Dis1	CV discharge at 2.0V	1800 sec and $\|current\| \geq 50mA$
16	OCV2	10 second rest	
17	CC_Dis1	CC charge at 0.5 C-rate	
18	CV_Dis1	CV charge at 3.6V	1800 sec and $\|current\| \geq 50mA$
19	HealthForLoop	goto step 13	Repeat 4 times
20	Dis_15A	2.5C-rate CC discharge	Power measurement, 15 secs
21	OCV_Final	Rest battery until $Time2$	Voltage relaxation

The schedule file has two main parts, a step by step schedule for running the experiment and a set of formulas for importing numbers from the batch file.

Table D.1 shows the purpose and notes for each step, see file Joel_Forman\JF_CCCVa_110323.sdu. This is mainly intended for those who may be editing or building on this file in the future.

It is worth noting that the values in Steps 9 and 21 change depending on how long that section of the experiment is intended to last. Step 9 has two time values that are set to be identical, $Time1$. Step 21 has the final end time value $Time2$ which is always 30 hours later than $Time1$. Typical values are $Time1 = 12:18:00:00$ (12 days 18 hours) and $Time2 = 14:00:00:00$ (14 days). These are always off set by 30 hours.

This next table shows the formulas used. At a first glance this may seem more complicated than it needs to be, but due to the limited number of inputs and functions available we go through some mathematical gymnastics of mapping single input numbers to multiple numbers.

Table D.2 Importing formulas for the schedule file

	Label	Experssion
1	F_Vmin	FLOOR(MV_Mass/100) /10*1
2	F_Vmax	MV_Mass/10 - 10*FLOOR(MV_Mass/100)
3	F_Crate	1.1*MV_SpecificCapacity + 0
4	F_Ramp	0.22*MV_SpecificCapacity + 0
5	F_CrateNeg	-1.1*MV_SpecificCapacity + 0
6	F_RampNeg	-0.22*MV_SpecificCapacity + 0

Specifically the first the MV_Mass input is exactly four digits, where:

$$MV\_Mass = \#_1\#_2\#_3\#_4 \tag{D.1a}$$

$$F\_Vmin = \#_1.\#_2 \tag{D.1b}$$

$$F\_Vmax = \#_3.\#_4 \tag{D.1c}$$

The C-rate limit for the cycles is equal to $MV\_SpecificCapacity$. The Arbin software only provides these two numbers for import, a third number for import would be more convenient.

The imported numbers are set in the batch file. They are a randomly ordered version of 3.1. For all of the cells $VoltageClampLow = 1.9$ and $VoltageClampHigh = 3.7$. These voltage clamps help limit the possible operating region of the cell for safety. $Mass$ and $SpecificCapacity$ are the MV_Mass and $MV\_SpecificCapacity$ values respectively and, of course, have nothing to do with the cells' mass or specific capacity.

As an example, the cell on channel 3 has $V\_min = 3.0V$, $V\_max = 3.3V$, and the $Crate = 2.5$. For future experiments one needs only edit the batch file to produce battery degradation cycles with varying limits.

Table D.3 The system batch file

Channel	Schedule	Mass (g)	Specific Capacity (Ah/g)
1	Joel_Forman\JF_CCCVa_110323.sdu	3436	2.5
2	Joel_Forman\JF_CCCVa_110323.sdu	3033	2.5
3	Joel_Forman\JF_CCCVa_110323.sdu	2636	1.5
4	Joel_Forman\JF_CCCVa_110323.sdu	2035	0.5
5	Joel_Forman\JF_CCCVa_110323.sdu	3031	2.5
6	Joel_Forman\JF_CCCVa_110323.sdu	3235	2.5
7	Joel_Forman\JF_CCCVa_110323.sdu	3034	2.5
8	Joel_Forman\JF_CCCVa_110323.sdu	2021	0.5
9	Joel_Forman\JF_CCCVa_110323.sdu	2425	0.5
10	Joel_Forman\JF_CCCVa_110323.sdu	2235	2.5
11	Joel_Forman\JF_CCCVa_110323.sdu	2032	2
12	Joel_Forman\JF_CCCVa_110323.sdu	2031	2.5
13	Joel_Forman\JF_CCCVa_110323.sdu	2036	1.5
14	Joel_Forman\JF_CCCVa_110323.sdu	3536	0.5

Bibliography

[1] Russell, B., 1908. "Mathematical logic as based on the theory of types". *American Journal of Mathematics,* **30**(3), pp. pp. 222–262.

[2] Doyle, M., Fuller, T., and Newman, J., 1993. "Modeling of galvanostaic charge and discharge of the lithium/polymer/insertion cell". *J. Electrochemical Society,* **140**, June, pp. 1526–1533.

[3] Fuller, T., Doyle, M., and Newman, J., 1994. "Simulation and optimization of the dual lithium ion insertion cell". *J. Electrochemical Society,* **141**, January, pp. 1–10.

[4] Forman, J. C., Bashash, S., Stein, J., and Fathy, H., 2010. "Reduction of an electrochemistry-based Li-ion battery health degradation model via constraint linearization and Padé approximation". *ASME 2010 Dynamic Systems and Control Conference, DSCC2010,* **2**, pp. 173 – 183.

[5] Forman, J., Bashash, S., Stein, J. L., and Fathy, H. K., 2011. "Reduction of an electrochemistry-based Li-ion battery model via quasi-linearization and Padé approximation". *J. Electrochemical Society,* **158**(2), February, pp. A93–A101.

[6] Forman, J. "Reduction of an electrochemistry-based Li-ion battery health degradation model via constraint linearization and Padé approximation". Master's thesis, The University of Michigan.

[7] Moura, S. J., Forman, J. C., Stein, J. L., and Fathy, H. K., 2010. "Control of film growth in lithium ion battery packs via switches". *Proceedings of the ASME Dynamic Systems and Control Conference 2009, DSCC2009*(PART A), pp. 139 – 147.

[8] Moura, S. J., Forman, J. C., Bashash, S., Stein, J. L., and Fathy, H. K., 2011. "Optimal control of film growth in lithium-ion battery packs via relay switches". *Industrial Electronics, IEEE Transactions on,* **58**(8), aug., pp. 3555 –3566.

[9] D'Amato, A. M., Forman, J. C., Tulga, E., Ali, A. A., Stein, J. L., Peng, H., and Bernstein, D. S., 2012. "Noninvasive battery-health diagnostics using retrospective-cost identification of inaccessible subsystems". *Dynamic Systems and Control Conference - DSCC2012*.

[10] Forman, J. C., Moura, S., Stein, J., and Fathy, H., 2012. "Optimal experimental design for modeling battery degradation". *Dynamic Systems and Control Conference - DSCC2012*.

[11] Bashash, S., Moura, S. J., Forman, J. C., and Fathy, H. K., 2011. "Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity". *Journal of Power Sources,* **196**(1), pp. 541 – 549.

[12] Forman, J., Moura, S., Stein, J. L., and Fathy, H. K., 2011. "Genetic parameter identification of the Doyle-Fuller-Newman model from experimental cycling of a LiFePO4 battery". *American Control Conference*.

[13] Forman, J., Moura, S., Stein, J., and Fathy, H., 2012. "Genetic identification and Fisher identifiability analysis of the Doyle-Fuller-Newman model from experimental cycling of a LiFePO4 cell". *Journal of Power Sources,* **210**, pp. 263 – 75.

[14] Peterson, S. B., Apt, J., and Whitacre, J., 2010. "Lithium-ion battery cell degradation resulting from realistic vehicle and vehicle-to-grid utilization". *J. Power Sources,* **195**, pp. 2385–2392.

[15] Atkinson, A., Donev, A., and Tobias, R., 2007. *Optimum Experimental Designs, with SAS*. Oxford statistical science series. Oxford University Press.

[16] Walter, É., and Pronzato, L., 1997. *Identification of Parametric Models from Experimental Data*. Communications and control engineering. Springer.

[17] Goodwin, G., and Payne, R., 1977. *Dynamic System Identification: Experiment Design and Data Analysis*. Mathematics in science and engineering. Academic Press.

[18] Cantù-Paz, E., 2000. *Efficient and Accurate Parallel Genetic Algorithms*. Genetic algorithms and evolutionary computation. Kluwer Academic Publishers.

[19] Bashash, S., Moura, S. J., Forman, J. C., and Fathy, H. K., 2011. "Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity". *Journal of Power Sources,* **196**(1), pp. 541 – 549.

[20] Amine, K., Liu, J., and Belharouak, I., 2005. "High-temperature storage and cycling of C-LiFePO4/graphite Li-ion cells". *Electrochemistry communications,* **7**(7), p. 669.

[21] Amine, K., Liu, J., Kang, S., Belharouak, I., Hyung, Y., Vissers, D., and Henriksen, G., 2004. "Improved lithium manganese oxide spinel/graphite Li-ion cells for high-power applications". *Journal of Power Sources,* **129**(1), p. 14.

[22] Aurbach, D., 2000. "Review of selected electrode-solution interactions which determine the performance of Li and Li ion batteries". *Journal of Power Sources,* **89**(2), p. 206.

[23] Hardwick, L., Marcinek, M., Beer, L., Kerr, J., and Kostecki, R., 2008. "An investigation of the effect of graphite degradation on irreversible capacity in lithium-ion cells". *Journal of the Electrochemical Society,* **155**(6), p. A442.

[24] Kpostecki, R., and McLarnon, F., 2003. "Microprobe study of the effect of Li intercalation on the structure of graphite". *Journal of Power Sources,* **119-121**, p. 550.

[25] Muto, S., Sasano, Y., Tatsumi, K., Sasaki, T., Horibuchi, K., Takeuchi, Y., and Ukyo, Y., 2009. "Capacity-fading mechanisms of LiNiO2-based lithium-ion batteries". *Journal of the Electrochemical Society,* **156**(5), p. A371.

[26] Peled, E., Menachem, C., Bar-Tow, D., and Melman, A., 1996. "Improved graphite anode for lithium-ion batteries. chemically bonded solid electrolyte interface and nanochannel formation". *Journal of the Electrochemical Society,* **143**(1), p. L4.

[27] Peled, E., 1979. "The electrochemical behavior of alkali and alkaline earth metals in nonaqueous battery systems-the solid electrolyte interphase model". *Journal of the Electrochemical Society,* **126**(12), p. 2047.

[28] Ramadass, P., Haran, B., Gomadam, P. M., White, R., and Popov, B. N., 2004. "Development of first principles capacity fade model for Li-ion cells". *J. Electrochemical Society,* **151**(2), January, pp. A196–A203.

[29] Rong, P., and Pedram, M., 2006. "An analytical model for predicting the remaining battery capacity of lithium-ion batteries". *IEEE transactions on very large scale integration (VLSI) systems,* **14**(5), p. 441.

[30] Santhanagopalan, S., Guo, Q., Ramadass, P., and White, R., 2006. "Review of models for predicting the cycling performance of lithium ion batteries". *Journal of Power Sources,* **156**(2), p. 620.

[31] Sasaki, T., Nonaka, T., Oka, H., Okuda, C., Itou, Y., Kondo, Y., Takeuchi, Y., Ukyo, Y., Tatsumi, K., and Muto, S., 2009. "Capacity-fading mechanisms of LiNiO2-based lithium-ion batteries. I. Analysis by electrochemical and spectroscopic examination". *Journal of the Electrochemical Society,* **156**(4), p. A289.

[32] Shim, J., Kostecki, R., Richardson, T., Song, X., and Striebel, K., 2002. "Electrochemical analysis for cycle performance and capacity fading of a lithium-ion battery cycled at elevated temperature". *Journal of Power Sources,* **112**(1), p. 222.

[33] Wu, M., Chiang, P., and Lin, J., 2005. "Electrochemical investigations on capacity fading of advanced lithium-ion batteries after storing at elevated temperature". *Journal of the Electrochemical Society,* **152**(6), p. A1041.

[34] Yoshida, T., Takahashi, M., Morikawa, S., Ihara, C., Katsukawa, H., Shiratsuchi, T., and ichi Yamakic, J., 2006. "Degradation mechanism and life prediction of lithium-ion batteries". *J. Electrochemical Society,* **153**, March, pp. A576–A582.

[35] Santhanagopalan, S., Guo, Q., and White, R. E., 2007. "Parameter estimation and model discrimination for a lithium-ion cell". *J. Electrochemical Society,* **154**, March, pp. A198–A206.

[36] Speltino, C., Domenico, D., Fiengo, G., and Stefanopoulou, A., 2009. "On the experimental identification of an electrochemical model of a lithium-ion battery: Part II". *The European Control Conference, Budapest*.

[37] Schmidt, A. P., Bitzer, M., Imre, A. W., and Guzzella, L., 2010. "Experiment-driven electrochemical modeling and systematic parameterization for a lithium-ion battery cell". *Journal of Power Sources,* **195**(15), pp. 5071 – 5080.

[38] Hu, Y., Yurkovich, S., Guezennec, Y., and Yurkovich, B., 2009. "A technique for dynamic battery model identification in automotive applications using linear parameter varying structures". *Control Engineering Practice,* **17**(10), pp. 1190 – 201.

[39] Wang, J., Liu, P., Hicks-Garner, J., Sherman, E., Soukiazian, S., Verbrugge, M., Tataria, H., Musser, J., and Finamore, P., 2011. "Cycle-life model for graphite-LiFePO4 cells". *J. Power Sources,* **196**, pp. A260–A271.

[40] Cover, T., and Thomas, J., 2004. *Elements of Information Theory*. Wiley series in telecommunications. Wiley-Interscience.

[41] McWhorter, T., and Scharf, L. L., 1993. "Cramér-Rao bounds for deterministic modal analysis". *IEEE Transactions on Signal Processing,* **41**(5), pp. 1847 – 1866.

[42] Wynn, H. P., and Parkin, N., 2001. "Sensitivity analysis and identifiability for differential equation models". Vol. 4, pp. 3116 – 3121.

[43] Jacquez, J., and Perry, T., 1990. "Parameter estimation: Local identifiability of parameters". *AJP - Endocrinology and Metabolism,* **258**, pp. E727–E736.

[44] Khalil, H., 2002. *Nonlinear Systems*. Prentice Hall.

[45] Gen, M., and Cheng, R., 2000. *Genetic Algorithms and Engineering Optimization*. Wiley series in engineering design and automation. Wiley.

[46] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation,* **6**(2), pp. 182 – 197.

[47] Storn, R., and Price, K., 1997. "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces". *Journal of Global Optimization,* **11**, pp. 341–359.

[48] Price, K., Storn, R., and Lampinen, J., 2005. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer.

[49] Das, S., and Suganthan, P. N., 2011. "Differential evolution: A survey of the state-of-the-art". *IEEE Transactions on Evolutionary Computation,* **15**(1), pp. 4 – 31.

[50] Kwan, C., Yang, F., and Chang, C., 2007. "A differential evolution variant of NSGA-II for real world multiobjective optimization". *Progress in Artificial Life. Proceedings Third Australian Conference, ACAL 2007. (Lecture Notes in Artificial Intelligence vol. 4828)*, pp. 345 – 56.

[51] Cai, L., and White, R., 2009. "Reduction of model order based on proper orthogonal decomposition for lithium-ion battery simulations". *Journal of the Electrochemical Society,* **156**(3), pp. 154 – 61.

[52] Wang, C.-Y., and Smith, K., 2006. "Solid-state diffusion limitations on pulse operation of a lithium ion cell for hybrid electric vehicles". *Journal of Power Sources,* **161**(1), pp. 628 – 39.

[53] Subramanian, V., Boovaragavan, V., and Diwakar, V., 2007. "Toward real-time simulation of physics based lithium-ion battery models". *Electrochemical and Solid-State Letters,* **10**(11), pp. 255 – 60.

[54] Subramanian, V. R., Boovaragavan, V., Ramadesigan, V., and Arabandi, M., 2009. "Mathematical model reformulation for lithium-ion battery simulations: Galvanostatic boundary conditions". *J. Electrochemical Society,* **156**, April, pp. A260–A271.

[55] Dubarry, M., Svoboda, V., Hwu, R., and Liaw, B., 2007. "Capacity and power fading mechanism identification from a commercial cell evaluation". *Journal of Power Sources,* **165**(2), p. 566.

[56] Wang, Y., Guo, X., Greenbaum, S., Liu, J., and Amine, K., 2001. "Solid electrolyte interphase formation on lithium-ion electrodes: a 7Li nuclear magnetic resonance study". *Electrochemical and solid-state letters,* **4**(6), p. A68.

[57] Zheng, J., Moss, P., Fu, R., Ma, Z., Xin, Y., Au, G., and Plichta, E., 2005. "Capacity degradation of lithium rechargeable batteries". *Journal of Power Sources,* **146**(1-2), p. 753.

[58] Broussely, M., Biensan, P., Bonhomme, F., Blanchard, P., Herreyreb, S., Nechev, K., and Staniewiczc, R., 2005. "Main aging mechanisms in Li ion batteries". *Journal of Power Sources,* **146**(1-2), p. 90.

[59] Yu, D., Donoue, K., Kadohata, T., Murata, T., Matsuta, S., and Fujitani, S., 2008. "Impurities in LiFePO4 and their influence on material characteristics". *Journal of the Electrochemical Society,* **155**(7), p. A526.

[60] Zhi, X., Liang, G., Wang, L., Ou, X., Zhang, J., and Cui, J., 2009. "The cycling performance of LiFePO4/C cathode materials". *Journal of Power Sources,* **189**(1), p. 779.

[61] Lei, J., McLarnon, F., and Kostecki, R., 2005. "In situ raman microscopy of individual LiNi0.8Co0.15Al0.05O2 particles in a Li-ion battery composite cathode". *The journal of physical chemistry.B,* **109**(2), p. 952.

[62] Nagpure, S., Bhushan, B., Babu, S., and Rizzoni, G., 2009. "Scanning spreading resistance characterization of aged Li-ion batteries using atomic force microscopy". *Scripta Materialia,* **60**(11), p. 933.

[63] Brown, S., Ogawa, K., Kumeuchi, Y., Enomoto, S., Uno, M., Saito, H., Sone, Y., Abraham, D., and Lindbergh, G., 2008. "Cycle life evaluation of 3 Ah LixMn2O4-based lithium-ion secondary cells for low-earth-orbit satellites. Part II. Harvested electrode examination". *Journal of Power Sources,* **185**(2), p. 1454.

[64] Wang, Z., and Chen, L., 2005. "Solvent storage-induced structural degradation of LiCoO2 for lithium ion batteries". *Journal of Power Sources,* **146**(1-2), p. 254.

[65] Zhou, J., and Notten, P., 2008. "Studies on the degradation of Li-ion batteries by the use of microreference electrodes". *Journal of Power Sources,* **177**(2), p. 553.

[66] Gachot, G., Grugeon, S., Armand, M., Pilard, S., Guenot, P., Tarascon, J., and Laruelle, S., 2008. "Deciphering the multi-step degradation mechanisms of carbonate-based electrolyte in Li batteries". *Journal of Power Sources,* **178**(1), p. 409.

[67] Jin, H., Liu, Z., Teng, Y., Gao, J., and Zhao, Y., 2009. "A comparison study of capacity degradation mechanism of LiFePO4-based lithium ion cells". *Journal of Power Sources,* **189**(1), p. 445.

[68] Brenan, K., Campbell, S., and Petzold, L., 1996. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Classics in applied mathematics. Society for Industrial and Applied Mathematics.

[69] Elgowainy, A., Burnham, A., Wang, M., Molburg, J., and Rousseau, A., 2009. Well-to-wheels energy use and greenhouse gas emissions analysis of plug-in hybrid electric vehicles. Tech. rep., Argonne National Laboratory.

[70] Fathy, H., Filipi, Z., Hagena, J., and Stein, J., 2006. "Review of hardware-in-the-loop simulation and its prospects in the automotive area". Vol. 6228, pp. 62280 – 1.

[71] LeBlanc, D., Sayer, J., Winkler, C., Ervin, R., Bogard, S., Devonshire, J., Mefford, M., Hagan, M., Bareket, Z., Goodsell, R., and Gordon, T., 2006. Road departure crash warning system field operational test: Methodology and results. Tech. rep., University of Michigan Transportation Research Institute.

[72] Duoba, M., Carlson, R., Jehlik, F., Smart, J., and White, S., 2009. "Correlating dynamometer testing to in-use fleet results of plug-in hybrid electric vehicles". *World Electric Vehicle Journal,* **3**(1).

[73] Moura, S., Fathy, H., Callaway, D., and Stein, J., 2011. "A stochastic optimal control approach for power management in plug-in hybrid electric vehicles". *IEEE Transactions on Control Systems Technology,* **19**(3), pp. 545 – 55.

[74] Laffont, L., Delacourt, C., Gibot, P., Wu, M., Kooyman, P., Masquelier, C., and Tarascon, J., 2006. "Study of the LiFePO4/FePO4 two-phase system by high-resolution electron energy loss spectroscopy". *Chemistry of materials,* **18**(23), p. 5520.

[75] Delmas, C., Maccario, M., Croguennec, L., Le Cras, F., and Weill, F., 2008. "Lithium deintercalation in LiFePO4 nanoparticles via a domino-cascade model". *Nature Materials,* **7**(8), p. 665.

[76] Brunetti, G., Robert, D., Bayle-Guillemaud, P., Rouviere, J., Rauch, E., Martin, J., Colin, J., Bertin, F., and Cayron, C., 2011. "Confirmation of the domino-cascade model by LiFePO4/FePO 4 precession electron diffraction". *Chemistry of Materials,* **23**(20), pp. 4515 – 4524.

[77] Srinivasan, V., and Newman, J., 2004. "Discharge model for the lithium iron-phosphate electrode". *Journal of the Electrochemical Society,* **151**(10), pp. A1517 – A1529.

[78] Srinivasan, V., and Newman, J., 2006. "Existence of path-dependence in the LiFePO4 electrode". *Electrochemical and Solid-State Letters,* **9**(3), pp. A110 – A114.

[79] Safari, M., and Delacourt, C., 2011. "Mathematical modeling of lithium iron phosphate electrode: Galvanostatic charge/discharge and path dependence". *Journal of the Electrochemical Society,* **158**(2), pp. A63 – A73.

[80] Safari, M., and Delacourt, C., 2011. "Modeling of a commercial graphite/LiFePO4 cell". *Journal of the Electrochemical Society,* **158**(5), pp. A562 – A571.

[81] Liu, F., Siddique, N., and Mukherjee, P., 2011. "Nonequilibrium phase transformation and particle shape effect in LiFePO4 materials for Li-ion batteries". *Electrochemical and Solid-State Letters,* **14**(10), pp. 143 – 7.

[82] Singh, G. K., Ceder, G., and Bazant, M. Z., 2008. "Intercalation dynamics in rechargeable battery materials: General theory and phase-transformation waves in LiFePO4". *Electrochimica Acta,* **53**(26), pp. 7599 – 7613.

[83] Fritsch, F., and Carlson, R., 1980. "Monotone piecewise cubic interpolation". *SIAM Journal on Numerical Analysis,* **17**(2), pp. 238 – 46.

[84] Burden, R., and Faires, J., 2001. *Numerical Analysis*. Brooks/Cole.

[85] Cantú-Paz, 2000. *Efficient and Accurate Parallel Genetic Algorithms*. Communications and control engineering. Kluwer Academic Publishers.

[86] Moura, S., Callaway, D., Fathy, H., and Stein, J., 2010. "Tradeoffs between battery energy capacity and stochastic optimal power management in plug-in hybrid electric vehicles". *Journal of Power Sources,* **195**(9), pp. 2979 – 88.

[87] Fedorov, V., 1972. *Theory of Optimal Experiments*. Probability and mathematical statistics. Academic Press.

[88] Forgez, C., Do, D. V., Friedrich, G., Morcrette, M., and Delacourt, C., 2010. "Thermal modeling of a cylindrical LiFePO4/graphite lithium-ion battery". *Journal of Power Sources,* **195**(9), pp. 2961 – 8.

[89] Vandenberghe, L., Boyd, S., and Wu, S.-P., 1998. "Determinant maximization with linear matrix inequality constraints". *SIAM Journal on Matrix Analysis and Applications,* **19**(2), pp. 499 – 533.

[90] Plett, G., 2004. "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs; Part 1. Background". *J. Power Sources,* **134**, June, pp. 252–261.

[91] Plett, G., 2004. "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs; Part 2. Modeling and identification". *J. Power Sources,* **134**, June, pp. 262–2276.

[92] Plett, G., 2004. "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs; Part 3. State and parameter estimation". *J. Power Sources,* **134**, June, pp. 277–292.

[93] Saha, B., Poll, S., Goebel, K., and Christophersen, J., 2007. "An integrated approach to battery health monitoring using Bayesian regression and state estimation". *AUTOTESTCON (Proceedings)*, pp. 646 – 653.

[94] Verbrugge, M., 2007. "Adaptive, multi-parameter battery state estimator with optimized time-weighting factors". *Journal of Applied Electrochemistry,* **37**, pp. 605–616.

[95] Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E., 2001. *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education.

[96] Yan, H., Fahroo, F., and Ross, I., 2001. "Optimal feedback control laws by Legendre pseudospectral approximations". *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148),* **vol.3**, pp. 2388 – 93.

[97] Saravanan, R., Ramabalan, S., and Balamurugan, C., 2008. "Evolutionary collision-free optimal trajectory planning for intelligent robots". *International Journal of Advanced Manufacturing Technology,* **36**(11-12), pp. 1234 – 51.

[98] Zielinski, K., and Laur, R., 2008. "Stopping criteria for differential evolution in constrained single-objective optimization". In *Advances in Differential Evolution*, U. Chakraborty, ed., Vol. 143 of *Studies in Computational Intelligence*. Springer Berlin / Heidelberg, pp. 111–138.

[99] Siegel, J. B., Lin, X., Stefanopoulou, A. G., Hussey, D. S., Jacobson, D. L., and Gorsich, D., 2011. "Neutron imaging of lithium concentration in LFP pouch cell battery". *Journal of the Electrochemical Society,* **158**(5), pp. A523 – A529.

[100] Wang, J., Liu, P., Hicks-Garner, J., Sherman, E., Soukiazian, S., Verbrugge, M., Tataria, H., Musser, J., and Finamore, P., 2011. "Cycle-life model for graphite-LiFePO4 cells". *Journal of Power Sources,* **196**(8), pp. 3942 – 3948.

[101] Chaloner, K., and Verdinelli, I., 1995. "Bayesian experimental design". *Statistical Science,* **10**(3), pp. 273 – 273.

[102] Spiegel, M., 1998. *Schaum's Mathematical Handbook of Formulas and Tables*. Schaum's Outline Series. McGraw-Hill.