AN AGING MODEL FOR LITHIUM-ION CELLS

A Dissertation
Presented to
The Graduate Faculty of The University of Akron

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Richard L. Hartmann II
December, 2008
This dissertation presents a methodology for using cycling data collected from several similar electrochemical cells to generate an aging model that predicts how the parameters in a first-principles dynamic model of a cell will change as the cell ages. Nine standard 18650 lithium-ion cells were cycled in three sets. Aging models were applied to the identified parameters of the dynamic models. These aging models were then validated by comparing their predictions with the original cycle data resulting in RMS voltage errors of less than 5% over the entire life of the cells. These aging models provide an accurate means of predicting the parameters for the dynamic cell model based on the life fraction of the cell and the maximum charging voltage. Unlike other aging models presented in the literature, the aging models presented here address the external performance of the cells. The aging model containing first-order temperature correction terms for the charge diffusion and current polarization term produced the smallest errors when compared with the original data. Incorporation of the aging model into a battery management system (BMS) will allow the BMS to better track capacity and remaining life of a cell. The methodology presented here could be applied to other cell chemistries.
ACKNOWLEDGEMENTS

I would like to thank my wife, Celia; my daughters, Elizabeth and Victoria; my parents, Elaine and Scooter; and my sister, Jesse for their love and support over these many years. I would also like to thank my advisor, Dr. Tom Hartley, and my committee, Dr. Alex De Abreu-Garcia, Dr. Robert Veillette, Dr. Helen Qammar and Dr. Jerry Young as well as Dr. James Grover for their guidance and support during my entire time at The University of Akron. I would also like to give special thanks to Dr. Mark Tausig of the graduate school for his understanding and support. Finally, I would like to thank everyone at the Ohio Northern Bankruptcy Court and Stark State College for their generous support.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>xix</td>
</tr>
</tbody>
</table>

## CHAPTER

### I. INTRODUCTION

1

### II. BACKGROUND INFORMATION

6

#### 2.1 Electrochemical Cell: Voltage

- 2.1.1 Chemical Reactions
- 2.1.2 Free Energy and Cell Voltage
- 2.1.3 Energy Cycle
- 2.1.4 Voltage Measurements
- 2.1.5 Cell Voltage Summary

#### 2.2 Electrochemical Cell: Current

- 2.2.1 Charge Acceptance
- 2.2.2 State of Charge
- 2.2.3 Capacity
- 2.2.4 Diffusion
- 2.2.5 Cell Current Summary
2.3 Electrochemical Cell: Dynamic Models .......................................................34
  2.3.1 Electric Circuit Models ........................................................................35
  2.3.2 Plett's Model .......................................................................................41
  2.3.3 First-Principles Model .........................................................................42
  2.3.4 Modeling Temperature Effects .............................................................43
  2.3.5 Chosen Dynamic Models .....................................................................44
2.4 Electrochemical Cell: Aging .....................................................................46
  2.4.1 Cycle Life and Life Fraction ..................................................................47
  2.4.2 Maximum Charging Voltage ..................................................................47
  2.4.3 Calendar Aging .....................................................................................49
  2.4.4 Cycle Aging .........................................................................................50
  2.4.5 Aging Models ......................................................................................53
  2.4.6 Cell Aging Summary ............................................................................54
2.5 Moving Average Filter ...............................................................................55
2.6 Batch Least Squares Identification .............................................................56
2.7 Chapter Summary .......................................................................................59

III. THE CYCLING PROCESS .............................................................................61
  3.1 Cycling Circuit ..........................................................................................61
  3.2 Data Storage ..............................................................................................70
  3.3 Data Processing ........................................................................................72
  3.4 Sample Cycles ..........................................................................................75
  3.5 Cell Behavior .............................................................................................90
    3.5.1 Capacity ..............................................................................................90
3.5.2 Open-Circuit Voltage: Charged Cell ............................................94
3.5.3 Open-Circuit Voltage: Discharged Cell.................................95
3.5.4 Cell Temperature .................................................................96
3.6 Chapter Summary .................................................................................101

IV. DYNAMIC MODEL IDENTIFICATION .........................................................103

4.1 Current Response .................................................................103
4.2 Resistance and Diffusion .......................................................105
4.3 Charge Vectors ........................................................................106
4.4 Dynamic Model Identification ..................................................106
4.5 Identified Parameters for Dynamic Models DMT⁰ and DMT¹ ......119
  4.5.1 Capacity Versus Life Fraction ...........................................125
  4.5.2 Identified Parameters for model DMT⁰ Versus Life Fraction ....126
  4.5.3 Identified Parameters for model DMT¹ Versus Life Fraction ....131
  4.5.4 Individual Temperature Correction Terms Versus Life
       Fraction ..................................................................................141
4.6 Chapter Summary ..........................................................................143

V. AGING RESULTS ..................................................................................145

5.1 Cell Performance ........................................................................145
  5.1.1 Capacity and Cycle Life.....................................................146
  5.1.2 Capacity Fade .................................................................147
5.2 Identified Parameters .......................................................................149
  5.2.1 Aging Comparison: No Temperature Correction ...............150
  5.2.2 Aging Comparison: With Temperature Correction .........152
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Description of the non-temperature corrected coefficients of the voltage Equations 2.75-2.77</td>
<td>45</td>
</tr>
<tr>
<td>2.2 Summary of aging studies presented in the literature</td>
<td>48</td>
</tr>
<tr>
<td>3.1 List of maximum charging voltages</td>
<td>62</td>
</tr>
<tr>
<td>3.2 List of circuit elements for the cycling circuit</td>
<td>62</td>
</tr>
<tr>
<td>3.3 State of LEDs on circuit board</td>
<td>65</td>
</tr>
<tr>
<td>3.4 List of voltage or current with corresponding integer values</td>
<td>69</td>
</tr>
<tr>
<td>3.5 Cycle at which the cell reached minimum capacity</td>
<td>94</td>
</tr>
<tr>
<td>4.1 List of identified parameters for linear approximation to $R_\Omega$</td>
<td>109</td>
</tr>
<tr>
<td>4.2 Identification error values for the different temperature models</td>
<td>142</td>
</tr>
<tr>
<td>5.1 Description of how identified parameters change with age for model DMT$^0$</td>
<td>151</td>
</tr>
<tr>
<td>5.2 Description of how identified parameters change with age for modified models DMT$^{m1-m7}$ (Each $T_l$ term was identified individually)</td>
<td>156</td>
</tr>
<tr>
<td>5.3 Parameters for the capacity aging model QAM</td>
<td>164</td>
</tr>
<tr>
<td>5.4 Average RMS error values for aging models</td>
<td>165</td>
</tr>
<tr>
<td>5.5 Coefficients for the aging model LAM$^0$</td>
<td>167</td>
</tr>
<tr>
<td>5.6 Coefficients for the aging model LAM$^{m8}$</td>
<td>175</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>2.1</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>23</td>
</tr>
<tr>
<td>2.5</td>
<td>23</td>
</tr>
<tr>
<td>2.6</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>29</td>
</tr>
<tr>
<td>2.8</td>
<td>29</td>
</tr>
<tr>
<td>2.9</td>
<td>32</td>
</tr>
<tr>
<td>2.10</td>
<td>36</td>
</tr>
<tr>
<td>2.11</td>
<td>36</td>
</tr>
<tr>
<td>2.12</td>
<td>36</td>
</tr>
<tr>
<td>2.13</td>
<td>38</td>
</tr>
<tr>
<td>2.14</td>
<td>38</td>
</tr>
<tr>
<td>2.15</td>
<td>39</td>
</tr>
</tbody>
</table>
2.16 Circuit model including self-discharge resistor .........................................................40
2.17 Circuit model including Warburg impedance.............................................................40
2.18 Circuit model including constant phase element .......................................................41
2.19 Cell capacity and cycle life versus maximum charging voltage (Bergveld et al., 2002) ...................................................................................................................51
2.20 General examples of cell aging (Spotniz, 2003).......................................................51
2.21 Sample waveform to illustrate the moving average filter ........................................56
3.1 Control circuit used for cycling cells ........................................................................63
3.2 Flowchart of the Cycle.bas PICBASIC program..........................................................66
3.3 Sample dataline from raw data file ...........................................................................70
3.4 Flowchart of the Logger.pl PERL script ....................................................................71
3.5 Flowchart of the Createdaily.pl PERL script ..............................................................73
3.6 Flowchart of the Convertdata.pl PERL script ............................................................74
3.7 Flowchart of the Createcycles.pl PERL script ..........................................................74
3.8 Voltage, current, and temperature of cell A, cycle 1 ................................................76
3.9 Voltage, current, and temperature of cell A, cycle 230 ............................................77
3.10 Voltage, current, and temperature of cell A, cycle 460 ............................................77
3.11 Voltage, current, and temperature of cell B, cycle 1 ................................................78
3.12 Voltage, current, and temperature of cell B, cycle 120 ............................................78
3.13 Voltage, current, and temperature of cell B, cycle 240 ............................................79
3.14 Voltage, current, and temperature of cell C, cycle 1 ................................................79
3.15 Voltage, current, and temperature of cell C, cycle 225 ............................................80
3.16 Voltage, current, and temperature of cell C, cycle 450 ............................................80
3.17 Voltage, current, and temperature of cell D, cycle 1 ................................................81
3.18 Voltage, current, and temperature of cell D, cycle 120 ............................................81
3.19 Voltage, current, and temperature of cell D, cycle 250 ...........................................82
3.20 Voltage, current, and temperature of cell E, cycle 1 ...............................................82
3.21 Voltage, current, and temperature of cell E, cycle 125 ...........................................83
3.22 Voltage, current, and temperature of cell E, cycle 250 ...........................................83
3.23 Voltage, current, and temperature of cell F, cycle 2 ...............................................84
3.24 Voltage, current, and temperature of cell F, cycle 145 ...........................................84
3.25 Voltage, current, and temperature of cell F, cycle 290 ...........................................85
3.26 Voltage, current, and temperature of cell G, cycle 1 ...............................................85
3.27 Voltage, current, and temperature of cell G, cycle 100 ...........................................86
3.28 Voltage, current, and temperature of cell G, cycle 200 ...........................................86
3.29 Voltage, current, and temperature of cell H, cycle 1 ...............................................87
3.30 Voltage, current, and temperature of cell H, cycle 100 ...........................................87
3.31 Voltage, current, and temperature of cell H, cycle 200 ...........................................88
3.32 Voltage, current, and temperature of cell I, cycle 1 ...............................................88
3.33 Voltage, current, and temperature of cell I, cycle 95 ...............................................89
3.34 Voltage, current, and temperature of cell I, cycle 180 ...........................................89
3.35 Capacity plots of cells with maximum charging voltage of 4.2 V .........................92
3.36 Capacity plots of cells with maximum charging voltage of 4.3 V .........................93
3.37 Capacity plots of cells with maximum charging voltage of 4.4 V .........................93
3.38 Filtered open-circuit voltage of fully charged cells versus cycle .......................95
3.39 Open-circuit voltage of fully discharged cells with MCV of 4.2 V .......................97
3.40 Open-circuit voltage of fully discharged cells with MCV of 4.3 V.................97
3.41 Open-circuit voltage of fully discharged cells with MCV of 4.4 V..................98
3.42 Open-circuit voltage of fully discharged cells along with scaled capacity for cells with MCV of 4.2V.................................................................98
3.43 Open-circuit voltage of fully discharged cells along with scaled capacity for cells with MCV of 4.3V.................................................................99
3.44 Open-circuit voltage of fully discharged cells along with scaled capacity for cells with MCV of 4.4V.................................................................99
3.45 Filtered maximum, minimum, and average temperatures for cells with MCV of 4.2 V .................................................................................................100
3.46 Filtered maximum, minimum, and average temperatures for cells with MCV of 4.3 V .................................................................................................100
3.47 Filtered maximum, minimum, and average temperatures for cells with MCV of 4.4 V .................................................................................................101
4.1 Current decay after the discontinuation of discharge of a lithium-ion cell.......104
4.2 Voltage recovery after the discontinuation of discharge of a lithium-ion cell ....106
4.3 Identified values for cell resistance, \( R_\Omega \) .........................................................107
4.4 Identified values for diffusion resistance, \( R_d \) .....................................................107
4.5 Identified values for diffusion resistance, \( C_d \) ......................................................108
4.6 Linear approximation of identified values of \( R_\Omega \), with filtered values of \( R_\Omega \) from Figure 4.3 shown as dashed lines .........................................................109
4.7 Stored charge vector for cell D, cycle 120.........................................................110
4.8 Diffused charge vector for cell D, cycle 120.......................................................111
4.9 Sample cycles for cell A showing identified dynamic model DMT\(^0\).................113
4.10 Sample cycles for cell B showing identified dynamic model DMT\(^0\).................114
4.11 Sample cycles for cell C showing identified dynamic model DMT\(^0\).................114
4.12 Sample cycles for cell D showing identified dynamic model DMT$^0$ .....................115
4.13 Sample cycles for cell E showing identified dynamic model DMT$^0$ .....................115
4.14 Sample cycles for cell F showing identified dynamic model DMT$^0$ .....................116
4.15 Sample cycles for cell G showing identified dynamic model DMT$^0$ .....................116
4.16 Sample cycles for cell H showing identified dynamic model DMT$^0$ .....................117
4.17 Sample cycles for cell I showing identified dynamic model DMT$^0$ .....................117
4.18 Close up of discharge cycle showing identified dynamic models. .........................120
4.19 Sample cycles for cell A showing identified dynamic model DMT$^1$ ....................120
4.20 Sample cycles for cell B showing identified dynamic model DMT$^1$ .....................121
4.21 Sample cycles for cell C showing identified dynamic model DMT$^1$ .....................121
4.22 Sample cycles for cell D showing identified dynamic model DMT$^1$ .....................122
4.23 Sample cycles for cell E showing identified dynamic model DMT$^1$ .....................122
4.24 Sample cycles for cell F showing identified dynamic model DMT$^1$ .....................123
4.25 Sample cycles for cell G showing identified dynamic model DMT$^1$ .....................123
4.26 Sample cycles for cell H showing identified dynamic model DMT$^1$ .....................124
4.27 Sample cycles for cell I showing identified dynamic model DMT$^1$ .....................124
4.28 Plots of capacity versus life fraction.......................................................................126
4.29 Identified values for the parameter $k_{E0T0}$ for model DMT$^0$ .................................127
4.30 Identified values for the parameter $k_{R0T0}$ for model DMT$^0$ .................................128
4.31 Identified values for the parameter $k_{l0T0}$ for model DMT$^0$ .................................128
4.32 Identified values for the parameter $k_{q0T0}$ for model DMT$^0$ .................................129
4.33 Identified values for the parameter $k_{q0T0}$ for model DMT$^0$ .................................129
4.34 Identified values for the parameter $k_{qNET0}$ for model DMT$^0$ ........................................130
4.35 Identified values for the parameter $k_{qSNET0}$ for model DMT$^0$ ........................................130
4.36 Magnitude of model error for model DMT$^0$ ......................................................................131
4.37 Identified values for the parameter $k_{E0T0}$ for model DMT$^1$ ........................................133
4.38 Identified values for the parameter $k_{E0T1}$ for model DMT$^1$ ........................................133
4.39 Identified values for the parameter $k_{R0T1}$ for model DMT$^1$ ........................................134
4.40 Identified values for the parameter $k_{liT0}$ for model DMT$^1$ ..........................................134
4.41 Identified values for the parameter $k_{liT1}$ for model DMT$^1$ ..........................................135
4.42 Identified values for the parameter $k_{qdT0}$ for model DMT$^1$ ..........................................135
4.43 Identified values for the parameter $k_{qdT1}$ for model DMT$^1$ ..........................................136
4.44 Identified values for the parameter $k_{qT0}$ for model DMT$^1$ ...........................................136
4.45 Identified values for the parameter $k_{qT1}$ for model DMT$^1$ ...........................................137
4.46 Identified values for the parameter $k_{qSNET0}$ for model DMT$^1$ ......................................137
4.47 Identified values for the parameter $k_{qSNET1}$ for model DMT$^1$ ......................................138
4.48 Identified values for the parameter $k_{qSNET0}$ for model DMT$^1$ ......................................138
4.49 Identified values for the parameter $k_{qSNET1}$ for model DMT$^1$ ......................................139
4.50 Magnitude of model error for model DMT$^1$ .....................................................................139
4.51 Combination of terms $k_{E0T0}$ and $k_{E0T1}$ for DMT$^1$ using the average cell temperature for an MCV of 4.2 V .................................................................140
4.52 Combination of terms $k_{E0T0}$ and $k_{E0T1}$ for DMT$^1$ using the average cell temperature for an MCV of 4.3 V .................................................................140
4.53 Combination of terms $k_{E0T0}$ and $k_{E0T1}$ for DMT$^1$ using the average cell temperature for an MCV of 4.4 V .................................................................141

5.1 Cell capacity and cycle life versus maximum charging voltage (Bergveld et al., 2002) including experimental results .................................................................147
5.2 Capacity of cell F shown with general fade points presented by Spotnitz (2003).....................................................................................................................148

5.3 Capacity of all nine cells during the first 45 cycles...........................................148

5.4 Average capacity of cells versus life fraction for different MCVs...............149

5.5 Averaged parameter $k_{E0T0}$ for model DMT$^0$.............................................153

5.6 Averaged parameter $k_{RQT0}$ for model DMT$^0$.............................................153

5.7 Averaged parameter $k_{IIT0}$ for model DMT$^0$................................................153

5.8 Averaged parameter $k_{qdT0}$ for model DMT$^0$.............................................154

5.9 Averaged parameter $k_{qST0}$ for model DMT$^0$.............................................154

5.10 Averaged parameter $k_{qNET0}$ for model DMT$^0$..........................................154

5.11 Averaged parameter $k_{qNFT0}$ for model DMT$^0$..........................................155

5.12 Combination of averaged terms $k_{E0T0}$ and $k_{E1T1}$ of model DMT$^{m2}$..........158

5.13 Combination of averaged terms $k_{RQT0}$ and $k_{R1T1}$ of model DMT$^{m5}$........158

5.14 Combination of averaged terms $k_{IIT0}$ and $k_{I1T1}$ of model DMT$^{m7}$.........159

5.15 Combination of averaged terms $k_{qdT0}$ and $k_{qd1T1}$ of model DMT$^{m6}$.......159

5.16 Combination of averaged terms $k_{qST0}$ and $k_{qST1}$ of model DMT$^{m3}$..........160

5.17 Combination of averaged terms $k_{qNET0}$ and $k_{qNET1}$ of model DMT$^{m1}$......160

5.18 Combination of averaged terms $k_{qNFT0}$ and $k_{qNFT1}$ of model DMT$^{m4}$......161

5.19 Capacity plots showing comparison between linear and quadratic aging models..................................................................................................................165

5.20 Averaged parameter $k_{E0T0}$ for model DMT$^0$ shown with aging model LAM$^0$....167

5.21 Averaged parameter $k_{RQT0}$ for model DMT$^0$ shown with aging model LAM$^0$....168

5.22 Averaged parameter $k_{IIT0}$ for model DMT$^0$ shown with aging model LAM$^0$....168
5.23 Averaged parameter \( k_{qd}\) for model DMT\(^0\) shown with aging model LAM\(^0\)........168
5.24 Averaged parameter \( k_{qs}\) for model DMT\(^0\) shown with aging model LAM\(^0\)........169
5.25 Averaged parameter \( k_{qs,NET}\) for model DMT\(^0\) shown with aging model LAM\(^0\).....169
5.26 Averaged parameter \( k_{qs,NFT}\) for model DMT\(^0\) shown with aging model LAM\(^0\).....169
5.27 Sample cycles for cell A showing aging model LAM\(^0\)...........................................170
5.28 Sample cycles for cell B showing aging model LAM\(^0\)...........................................170
5.29 Sample cycles for cell C showing aging model LAM\(^0\)...........................................171
5.30 Sample cycles for cell D showing aging model LAM\(^0\)...........................................171
5.31 Sample cycles for cell E showing aging model LAM\(^0\)...........................................172
5.32 Sample cycles for cell F showing aging model LAM\(^0\)...........................................172
5.33 Sample cycles for cell G showing aging model LAM\(^0\)...........................................173
5.34 Sample cycles for cell H showing aging model LAM\(^0\)...........................................173
5.35 Sample cycles for cell I showing aging model LAM\(^0\)...........................................174
5.36 Errors between aging model LAM\(^0\) and actual voltages for all cycles...................174
5.37 Averaged parameter \( k_{E0}\) for model DMT\(^m8\) shown with aging model
LAM\(^m8\)....................................................................................................................176
5.38 Averaged parameter \( k_{R0}\) for model DMT\(^m8\) shown with aging model
LAM\(^m8\)....................................................................................................................177
5.39 Averaged parameter \( k_{li}\) for model DMT\(^m8\) shown with aging model
LAM\(^m8\)....................................................................................................................177
5.40 Averaged parameter \( k_{li,1}\) for model DMT\(^m8\) shown with aging model
LAM\(^m8\)....................................................................................................................177
5.41 Averaged parameter \( k_{qd}\) for model DMT\(^m8\) shown with aging model
LAM\(^m8\)....................................................................................................................178
5.42 Averaged parameter \( k_{qd,1}\) for model DMT\(^m8\) shown with aging model
LAM\(^m8\)....................................................................................................................178
5.43 Averaged parameter $k_{qsT0}$ for model DMT$^{m8}$ shown with aging model LAM$^{m8}$ .................................................................................................178

5.44 Averaged parameter $k_{qsNET0}$ for model DMT$^{m8}$ shown with aging model LAM$^{m8}$ .................................................................................................179

5.45 Averaged parameter $k_{qsNFT0}$ for model DMT$^{m8}$ shown with aging model LAM$^{m8}$ .................................................................................................179

5.46 Sample cycles for cell A showing aging model LAM$^{m8}$ ..............................................180

5.47 Sample cycles for cell B showing aging model LAM$^{m8}$ ..............................................180

5.48 Sample cycles for cell C showing aging model LAM$^{m8}$ ..............................................181

5.49 Sample cycles for cell D showing aging model LAM$^{m8}$ ..............................................181

5.50 Sample cycles for cell E showing aging model LAM$^{m8}$ ..............................................182

5.51 Sample cycles for cell F showing aging model LAM$^{m8}$ ..............................................182

5.52 Sample cycles for cell G showing aging model LAM$^{m8}$ ..............................................183

5.53 Sample cycles for cell H showing aging model LAM$^{m8}$ ..............................................183

5.54 Sample cycles for cell I showing aging model LAM$^{m8}$ ..............................................184

5.55 Errors between aging model LAM$^{m8}$ and actual voltages for all cycles..............184
NOMENCLATURE

+, - As a superscript, represents the positive or negative charge of an ion respectively. As a subscript, represents the positive or negative electrode of a cell.

0 Superscript that represents zeroth-order dynamic model, DMT^0, or aging model LAM^0. Subscript that represents zeroth-order temperature correction coefficient, see \( k_{E0T0} \).

1 Superscript that represents first-order dynamic model, DMT^1, or aging model LAM^1. Subscript that represents first-order temperature correction coefficient.

2 Superscript that represents second-order dynamic model, DMT^2. Subscript that represents second-order temperature correction coefficient.

A/D Analog to digital converter.

A, B, C Labels of cells with a maximum charging voltage of 4.2 V.

\( A, B, C, D, E \) Variables representing a species within a chemical equation.

\( a, b, c, d, e \) Variables representing the coefficients of a species within a chemical equation.

\( a, b \) Unknown coefficients to be identified.

\( a(q_s) \) Function describing the charge acceptance.

\( a_p^p \) Represents the activities of species from a chemical reaction, where \( p \) is the reaction coefficient and \( P \) is the chemical species.

BMS Battery management system.

CC Constant current phase of a cycle.

\( C_d \) Diffusion capacitance.

\( C_i \) Capacitance. Numeric superscripts represent capacitors within an electrical circuit.
CPE  Constant phase element.

$C_{PWM}$  Capacitance of PWM circuit.

CR  Rest after charge phase of a cycle.

CV  Constant voltage phase of a cycle.

$1C, C/2$  Charge/discharge rates of a cell. A 2.4 Ah cell charged at a rate at $C/2$ will be charged at a constant current of 1.2 A.

$C_s$  Storage capacitance.

$c_p$  Concentration of the species within the electrolyte measured in mol/m$^3$.

D  Discharge phase of a cycle.

d  Subscript representing diffusion.

D, E, F  Labels of cells with a maximum charging voltage of 4.3 V.

DMT  Dynamic model. Superscripts represent the temperature correction model used.

DR  Rest after discharge phase of a cycle.

$E$  Voltage of a cell.

$E(t)$  Instantaneous voltage of a cell.

$E(s)$  Laplace transform of cell voltage.

$E_+$  Voltage of a cell at the positive electrode.

$E_-$  Voltage of a cell at the negative electrode.

$E^0$  Standard voltage of a cell.

$E^{0+}$  Standard voltage of a cell at the positive electrode.

$E^{0-}$  Standard voltage of a cell at the negative electrode.

$E_a$  Activation energy.

$E_{oc}, E_{ocs}$  Open-circuit voltage and stabilized open-circuit voltage of a cell respectively.

$E^0$  Standard voltage of a cell.

xx
\( e^- \)  
Electron.

\( F \)  
Faraday's constant, 26.8 Ah/mole.

\( f(y_k), g(u_k) \)  
Functions that describe the response of a system.

\( G, H, I \)  
Labels of cells with a maximum charging voltage of 4.4 V.

\( \Delta G^0 \)  
Gibbs free energy.

\( \Delta H \)  
Enthalpy of a chemical reaction.

\( H \)  
Size of spatial lump.

\( h \)  
Activity layer thickness.

\( I(s) \)  
Laplace transform of cell current.

\( I_{d0} \)  
Initial discharge current of a cell.

\( I_d \)  
Constant discharge current.

\( i, j \)  
As a subscript or superscript, represents the number of a constant or spatial lump.

\( i(t) \)  
Instantaneous current.

\( J_0(s), J_2(s) \)  
Laplace transform of current flux or charge flux.

\( j_{d}(x,t) \)  
Current or charge flux at position \( x \) within the cell at time \( t \).

\( j_{d0}(t), j_{dL}(t) \)  
Current or charge flux at the electrodes at time \( t \).

\( K_x, K_b, k_x, k_i \)  
Constants or undetermined coefficients designated by subscripts.

\( k_{Arr} \)  
Arrhenius constant.

\( k^0_{rate} \)  
Pre-exponential factor based on the chemistry of a reaction.

\( k_{E0T0} \)  
Base voltage of a cell when the cell current, diffused charge, and stored charge all zero.

\( k_{R0T0} \)  
Primary resistance of the cell. Caused primarily by the SEI film and the diffusion through the electrolyte.

\( k_{liT0} \)  
Polarization adjustment caused by the current flowing in the cell.

\( k_{qdT0} \)  
Potential adjustment caused by the diffused charge within the cell.
\( k_{qsT0} \)  Linear change in potential due to the stored charge within the electrode.

\( k_{qsNET0} \)  Potential of the cell when it is near empty (very little charged material remaining), resulting in a majority of uncharged material available within the electrode.

\( k_{qsNFT0} \)  Potential of the cell when it is near full (very little uncharged material remaining), resulting in a majority of charged material available within the electrode.

\( k_{scale} \)  Scaling factor chosen empirically to be 8 V/Ah.

\( L \)  Distance of right boundary of the electrolyte.

LAM  Linear aging model. Superscripts represent the temperature correction model used.

\( LF \)  Life fraction.

M  Metal alloy of NiMH cell.

\( M \)  Number of data points used for moving average filter.

MCV  Maximum charge voltage.

\( m, n, N \)  Integers.

\( m_P \)  Molar amount of the species within the electrode.

\( m1-m9 \)  Superscript representing modified versions of the first-order temperature correction models of the dynamic model, DMT and the linear aging model, LAM.

PWM  Pulse width modulation.

\( Q, Q_R \)  Capacity of a cell and rated capacity of a cell respectively.

QAM  Quadratic aging model.

\( Q_i \)  The \( i^{th} \) transistor in a circuit.

\( q_d(t) \)  Instantaneous diffused charge within a cell.

\( q_d(x, t) \)  Instantaneous diffused charge density at position \( x \) within the electrolyte.

\( q_s(t) \)  Instantaneous stored charge within a cell.

\( \dot{q}_d(t), \dot{q}_s(t) \)  Instantaneous rate in diffused and stored charge within a cell.
\( q_{bulk} \)  
Equilibrium linear charge density within the electrolyte.

\( R_f \)  
SEI film resistance.

\( R_g \)  
Gas constant, 8.314472 J/K/mole.

\( R_i \)  
Resistance. Numeric superscripts represent resistors within an electrical circuit.

\( R_{PWM} \)  
Resistance of PWM circuit.

RMS  
Root mean square.

\( R_{sd} \)  
Self discharge resistance of a cell.

\( R_{G} \)  
Resistance of a cell.

SEI  
Solid electrolyte interface.

SOC  
State of charge. Usually expressed as a fraction or percentage of the capacity.

\( s \)  
Laplace transform variable. As a subscript, represents stored charge.

\( s_{sd} \)  
Subscript representing self discharge.

\( T \)  
Temperature.

\( T_0 \)  
As a subscript, represents the coefficients to the dynamic model voltage equation with no temperature correction.

\( T_1 \)  
As a subscript, represents the coefficients to the dynamic model voltage equation with first-order temperature correction.

\( T_2 \)  
As a subscript, represents the coefficients to the dynamic model voltage equation with second-order temperature correction.

\( T\Delta S \)  
Heat released or absorbed during a chemical reaction.

\( t \)  
Time.

\( t_s \)  
Time of the discharge.

\( U \)  
Input vector.

\( U_1 \)  
Temperature sensor, LM61.

\( u_k \)  
Input at time step \( k \).
\begin{align*}
\textit{Volume} & \quad \text{Volume of an electrode.} \\
v_d(t) & \quad \text{Voltage dropped across the diffusion capacitor, } C_d. \\
v_s(t) & \quad \text{Voltage dropped across the storage capacitor, } C_s. \\
V_{\text{capacity}} & \quad \text{Scaled value of cell capacity for comparison to cell voltage.} \\
V_{\text{ref}} & \quad \text{Empirically chosen reference voltage.} \\
V_{\text{rms}} & \quad \text{RMS voltage error.} \\
V_T & \quad \text{Voltage of the temperature sensor, } U_1. \\
x[n] & \quad \text{Unfiltered data point.} \\
y[n] & \quad \text{Filtered data point.} \\
y_k & \quad \text{Output at time step } k. \\
\hat{y}_k & \quad \text{Model output of system at time step } k. \\
Z_1 & \quad \text{Zener reference diode, LM336.} \\
x & \quad \text{Distance from the reference electrode within a cell. When } x \text{ is used as a}
\text{subscript, it indicates a general subscript for a collection or range of}
\text{variables.} \\
\alpha & \quad \text{Constant representing the reaction symmetry.} \\
\beta & \quad \text{Constant } 1/k_d, \text{ where } k_d \text{ is the diffusion constant in Fick's first law.} \\
\gamma & \quad \text{Constant dependent on the type of cell. Referred to as the Peukert number.} \\
\delta & \quad \text{Thickness of the SEI.} \\
\varepsilon_k & \quad \text{Error between the true system output and the model output at step } k. \\
\zeta & \quad \text{Dimensionless activity coefficient often assumed to be unity.} \\
\eta & \quad \text{Charge efficiency.} \\
\Theta & \quad \text{Vector of unknown coefficients to be identified.} \\
\Phi & \quad \text{Matrix consisting of data points for identification of the elements of } \Theta.
\end{align*}
\( \tau \)  Integral variable.

\( \tau_{PWM} \)  Time constant of RC PWM circuit.

\( \kappa \)  Conductivity of the SEI.

\( \Lambda \)  Input matrix created by applying a central difference to Fick’s second law using Fick’s first law.

\( \chi \)  Diffusion vector.

\( \Psi \)  System matrix created by applying a central difference to Fick’s second law.

\( \Omega \)  Subscript representing the primary resistance in a cell.
CHAPTER I
INTRODUCTION

A battery is a group of electrochemical cells that are used to store energy. In this paper, a cell refers to one electrochemical cell. A battery or battery pack is a group of cells connected in series, usually about three to six cells packaged together. A battery stack is a large group of individual cells or batteries connected together in series to obtain large voltages, often in the range of 100 V to 300 V. Since the cells in a battery are connected in series, the current through one cell or battery is assumed to be the same as the current through any other cell or battery, unless the controller contains bypass mechanisms. Voltage, current, temperature and pressure measurements may be obtained from the whole stack, individual batteries, or individual cells.

Batteries are often separated into three categories: primary, secondary and reserve (Linden, 1995). Primary batteries are made of chemical ingredients that cannot be electrically recharged. Primary batteries are discarded after their stored energy has been depleted. Alternatively, secondary batteries can be discharged and then recharged many times. Reserve batteries are designed for long-term storage. In reserve batteries, the electrolyte is usually stored separately from the electrodes and inserted when needed. Another approach for reserve batteries is to store the electrolyte as a solid that melts when heated, allowing ion flow within the cell. This paper focuses on secondary
batteries. Rechargeable cell chemistries of interest include NiH2, NiCd, NiMH, Li-Ion, and lead acid. This paper will focus on Li-Ion cells.

A battery has three main usage phases: charging, discharging, and resting. During the charging phase, electrical energy is supplied from an outside source and is stored as charged chemical species within the electrodes of the cell. During the discharge phase, the storage reaction is reversed, and energy is transferred from the chemical species to an external circuit. When the cell is at rest, no energy is transferred in or out of the cell; however, a small amount can be lost through self-discharge.

Charging is usually done in several phases. If the battery is at a low state of charge (SOC), then it can accept a relatively large charging current without harm. The battery is typically charged at a constant current (CC), where the limiting factor is usually the maximum amount of current the charging circuit can supply. After the battery reaches a certain voltage, continuous high currents will damage it. Therefore, charging is done at a constant voltage (CV) to prevent overvoltage conditions. As the battery approaches full charge, it is sometimes given current pulses, which are designed to allow the cells to equalize for some chemistries. Once the battery has reached full charge, it is often trickle charged, which means that a small pulsed or constant current is applied to the battery. Trickle charging is often used to ensure the cell is at 100% charge capacity and to counteract the self-discharge within the cells. There are other charge strategies that can also be used (Linden, 1995).

A typical Battery Management System (BMS) consists of a battery, sensors to monitor the state of the battery, a battery model or charge estimator, and a controller. A BMS must monitor the battery to ensure that the battery is operating within safe limits. Typically, the important sensed variables are voltage, current, and
temperature. For some battery types, such as NiH2, the internal cell pressure is also measured. In a simple BMS, voltage could be monitored using a simple comparator. This comparator circuit could be used to disable the circuit when the voltage goes outside the operational limits specified by the manufacturer. In a more advanced BMS, the voltage, current, and temperature can be monitored by a microprocessor system and used to correct the state of charge estimate or be recorded for long-term trending of battery performance.

The primary purpose of the BMS is to control the battery to ensure that its operation stays within the design limits to minimize damage (Plett, 2004). During charging, damage can occur to the battery if one of several conditions takes place. For example, if the temperature of the battery gets too high, gassing will occur, resulting in a loss of electrolyte and shorter life of the battery. If charging is continued after the battery has reached full charge, electrolysis of the electrolyte occurs which also shortens the battery’s life. In cases of extreme overcharging, high temperature, electrolysis, and gassing can lead to explosion of the cell.

As previously mentioned, a dynamic cell model can be included as part of a BMS. This dynamic model is used primarily to estimate the state of charge of the cell. The dynamic model can be based on an analysis of the physical properties of the cell (Hartley & Jannette, 2005) or simply by defining the model as a set of basis functions (Stamps et al., 2005). Regardless of how the cell model is chosen, the parameters of the model are normally determined using some form of least-squares identification.

As the cell ages, its capacity decreases (fades) and its internal resistance increases. Capacity fade is dependent on many factors including the maximum charging voltage (Linden 1995) (Ning et al., 2006), the temperature (Santhanagopalan et al., 2008), and...
the discharge rate (Ning et al., 2003) (Abraham et al., 2007). To maintain accuracy, the dynamic model parameters must be adjusted in order to account for these aging effects.

Many aging studies have been presented in the literature and will be discussed in the next chapter. These studies usually concentrate on how the internal physics of the cell changes with cell age. Aging models presented in these studies address the loss of active lithium and the thickness of the solid electrolyte interface (SEI) layer. Many of the models are based on destructive testing of a cell. This paper presents the development of an aging model that does not require the physical opening of a cell and that can be used in a BMS.

This paper presents an aging study of lithium-ion cells, followed by the development of an aging model which uses the maximum charging voltage (MCV) and cell life fraction (LF) to accurately predict the coefficients of a dynamic cell model as the cell ages. To generate this model, nine lithium-ion cells are cycled until they reach their end of life. Cells are cycled in groups of three. Each group has a different MCV. These voltages are 4.2 V, 4.3 V, and 4.4 V. Cycle data collected from the cells is used to determine the coefficients of dynamic models which are based on the physics of the cells. An aging model is then constructed to track the coefficients of the dynamic model of the cells throughout their life. Simulations using the aging model are compared with the original cycle data to validate the aging model. This aging model can be used in battery management systems to provide increased accuracy in predicting the remaining life of the cells.

The remainder of this paper is organized as follows: Chapter II provides background information about electrochemical cells, dynamic cell models, and cell aging. Chapter II also discusses moving average filters and batch least squares identification,
which is used to identify the coefficients of the cell dynamic models. Chapter III presents the method of cycling used to perform the aging study and presents sample results from the cycling. Chapter IV discusses the process of identification of the unknown parameters for the dynamic models. Chapter V presents cell aging results and presents the cell aging model. Chapter V also provides recommendations on using the aging model. Finally, Chapter VI provides some conclusions and suggestions for future work.
CHAPTER II
BACKGROUND INFORMATION

This chapter presents the background information necessary to understand electrochemical cell dynamics and cell aging. First, the physical principles describing a cell are discussed. Dynamic models of cells are then presented, followed by studies of how the cell dynamics change during the aging process. Finally, data filtering and parameter identification methods are discussed.

2.1 Electrochemical Cell: Voltage

This section develops the relationship between the chemical reactions of an electrochemical cell and the voltage that it produces. First, the chemical reactions of several common secondary battery chemistries are presented. Next, the relationship between the Gibbs free energy and the voltage of the cell is developed. Then, a general equation for modeling the cell voltage is given. Also, common methods of measuring the actual cell voltage are discussed.

2.1.1 Chemical Reactions

Fundamentally, an electrochemical cell consists of two electrodes and an electrolyte inside a container (Linden, 1995). When a passive electrical circuit is attached to the two electrodes, an electrical current is generated and the cell is discharged.
Chemically, this current is created by a reduction and oxidation (redox) reaction. During discharge, electrons are accepted from the external circuit at the positive electrode, and the electrode is chemically reduced. Similarly, the negative electrode is oxidized as it releases electrons to the external circuit. Internally, the circuit is completed by ions flowing through the electrolyte. To charge the cell, a circuit with a higher voltage than the cell’s terminal voltage is attached. This allows current to flow through the cell in the opposite direction, which reverses the redox reactions.

Several types of battery chemistries use nickel hydroxide as the positive electrode. The simplest of these is the NiH2 cell. To charge the cell, a positive current is applied to the nickel hydroxide electrode, oxidizing it to nickel oxyhydroxide (Hartley & Lorenzo, 2002) as,

$$\text{Ni(OH)}_2 + \text{OH}^- \rightarrow \text{NiOOH} + \text{H}_2\text{O} + e^-.$$ \hspace{1cm} (2.1)

The negative electrode consists of platinum, which acts as a catalyst for the hydrogen reaction

$$\text{H}_2\text{O} + e^- \rightarrow \frac{1}{2}\text{H}_2 + \text{OH}^-.$$ \hspace{1cm} (2.2)

When discharging, the nickel oxyhydroxide electrode is reduced as

$$\text{NiOOH} + \text{H}_2\text{O} + e^- \rightarrow \text{Ni(OH)}_2 + \text{OH}^-,$$ \hspace{1cm} (2.3)

while the hydrogen is oxidized as

$$\frac{1}{2}\text{H}_2 + \text{OH}^- \rightarrow \text{H}_2\text{O} + e^-.$$ \hspace{1cm} (2.4)

Combining and balancing the reactions give

$$2\text{Ni(OH)}_2 \leftrightarrow 2\text{NiOOH} + \text{H}_2.$$ \hspace{1cm} (2.5)

Since NiH2 cells store hydrogen as a gas, this type of cell is considered impractical for many applications.
An alternative to storing hydrogen gas is to use cadmium hydroxide for the negative electrode. The chemical reactions of a NiCd cell are presented below (Linden, 1995). Since the positive electrode is still nickel hydroxide, during charging the chemical reaction is the same as Equation 2.1, repeated here as

\[
\text{Ni(OH)}_2 + \text{OH}^- \rightarrow \text{NiOOH} + \text{H}_2\text{O} + e^-, \quad (2.6)
\]

while the cadmium hydroxide electrode is reduced as

\[
\text{Cd(OH)}_2 + 2e^- \rightarrow \text{Cd} + 2\text{OH}^- . \quad (2.7)
\]

Recognizing that the discharge equations are the reverse of Equations 2.6 and 2.7, the combined and balanced overall reactions give

\[
2\text{Ni(OH)}_2 + \text{Cd(OH)}_2 \rightleftharpoons 2\text{NiOOH} + 2\text{H}_2\text{O} + \text{Cd}. \quad (2.8)
\]

The NiMH cell is similar to the NiCd cell. The primary difference is that the cadmium hydroxide electrode is replaced with a metal alloy. Gu et al. (1999) describe the equations of a NiMH cell during charge as shown below. Again, the reaction at the nickel hydroxide electrode of the NiMH cell is

\[
\text{Ni(OH)}_2 + \text{OH}^- \rightarrow \text{NiOOH} + \text{H}_2\text{O} + e^-. \quad (2.9)
\]

At the negative electrode, the metal alloy, M, combines with a hydrogen atom to become metal hydride as

\[
\text{M} + \text{H}_2\text{O} + e^- \rightarrow \text{MH} + \text{OH}^- . \quad (2.10)
\]

Combining Equations 2.9 and 2.10 above gives the overall reaction within the cell as

\[
\text{Ni(OH)}_2 + \text{M} \rightleftharpoons \text{NiOOH} + \text{MH}. \quad (2.11)
\]

Li-Ion cells use lithium metal oxides as the positive electrode. These metallic compounds allow the lithium ions to be inserted and extracted out of the host lattice without structural modification of the electrode. Common compounds include lithium
cobalt oxide, LiCoO$_2$, or lithium manganese oxide, LiMn$_2$O$_4$ (Linden, 1995). The negative electrode of the Li-Ion cell is typically made of a graphite carbon material.

Assuming a LiCoO$_2$ electrode, the chemical reaction of a Li-Ion cell (Bergveld et al., 2002) at the positive electrode during charging is given as

$$\text{LiCoO}_2 \rightarrow \text{CoO}_2 + \text{Li}^+ + e^-,$$  \hspace{1cm} (2.12)

while the reaction at the negative electrode is

$$\text{C}_6 + \text{Li}^+ + e^- \rightarrow \text{LiC}_6.$$  \hspace{1cm} (2.13)

Therefore, the overall reaction of a Li-Ion cell is

$$\text{LiCoO}_2 + \text{C}_6 \leftrightarrow \text{CoO}_2 + \text{LiC}_6.$$  \hspace{1cm} (2.14)

Regardless of actual cell chemistry, when discharging the cell, electrons are released to the external circuit as the negative electrode material is oxidized, while the positive electrode is reduced as it accepts electrons from the external circuit. To charge the cell, the current is reversed, thereby reversing the chemical reactions. The *Handbook of Batteries* (Linden, 1995) generalizes the reduction and oxidation (redox) reactions as

$$aA \leftrightarrow cC + nE^+ + ne^-,$$  \hspace{1cm} (2.15)

for the positive electrode, and

$$bB + nE^+ + ne^- \leftrightarrow dD,$$  \hspace{1cm} (2.16)

for the negative electrode, which give the combined reaction of

$$aA + bB \leftrightarrow cC + dD.$$  \hspace{1cm} (2.17)

Equations 2.15 and 2.16 illustrate that whenever any general redox reaction occurs, electrons are either released or absorbed. Equation 2.15 describes the reaction at the positive electrode. When the cell is charged, the species $A$ within the electrode is oxidized and becomes the energized species $C$, releasing a positive ion into the
electrolyte. Similarly, Equation 2.16 shows how the negative electrode combines the species $B$ with a positive ion and an electron to produce the energized species $D$. The reactant coefficients $a$, $b$, $c$, $d$, and $n$ represent the numbers of moles of the associated species in the reactions. The next section describes how the voltage at the cell terminals is related to these reactions.

2.1.2 Free Energy and Cell Voltage

The terminal voltage of a cell is measured easily and gives an indication of the status of the cell. Figure 2.1 shows an actual discharge curve of a 9.5 Ah NiMH cell being discharged at 9.5 A. The cell starts out at full charge with an open-circuit voltage of approximately 1.365 V. When the discharge current starts, the voltage immediately drops to approximately 1.290 V due to the internal resistance of the cell. A diffusive transient follows and, within minutes, the voltage of the cell drops to around 1.2 V. The voltage of this cell remains around 1.2 V until the cell reaches the end of its charge, then the voltage quickly drops toward zero. The voltage of the cell comes from the Gibbs free energy of the chemical reactions at the electrodes. Figure 2.2 shows how the voltage differences occur at the electrode-electrolyte interface, along with a slight voltage drop due to the resistance of ion flow through the electrolyte (Hartley & Lorenzo). As the figure illustrates, the cell voltage is generated by the energy available in the chemical reactions which occur at the electrodes.

The enthalpy, $\Delta H$, of the reaction at an electrode can be separated into two parts as

$$\Delta H = \Delta G^0 + T \Delta S,$$

(2.18)
Figure 2.1: Voltage discharge curve of 9.5-Ah NiMH cell.

Figure 2.2: Voltage profiles inside an electrochemical cell (Hartley & Lorenzo, 2002).
where $T \Delta S$ is the heat either released or absorbed during the chemical reaction, and $\Delta G^0$ is
the Gibbs free energy (Linden, 1995). The Gibbs free energy is the energy available to
do work. The free energy is related to the electrode voltage by

$$\Delta G^0 = -nF E^0,$$

(2.19)

where $F$ is Faraday’s constant which is equal to 96,487 C/mole or 26.8 Ah/mole, $n$ is
number of moles of electrons in the reaction, and $E^0$ is the standard voltage based on the
materials being used (Hartley & Lorenzo, 2002) (Linden, 1995). Solving Equation 2.19
above for the standard voltage yields

$$E^0 = -\frac{\Delta G^0}{nF}.$$

(2.20)

The open-circuit voltage, $E$, across the terminals of the cell can be written as

$$E = E_+ + E_-,$$

(2.21)

where $E_+$ and $E_-$ are the actual voltages from the positive and negative electrodes
respectively. The open-circuit voltage $E$ deviates from the standard voltage $E^0$ due to the
reactant activities at the electrode-electrolyte interfaces. The dependence of the voltages
on the activities of the species is given by the Nernst equation. The Nernst relationship
for the reaction at the positive electrode of Equation 2.15 is

$$E^* = E^0 + \frac{R_g T}{nF} \ln \left( \frac{a_{P}^c a_{E}^o}{a_{A}^{u}} \right),$$

(2.22)

where $E^0$ is the standard voltage for the reaction, $R_g$ is the universal gas constant, and $T$ is
the absolute temperature. Each parameter $a_p$ in Equation 2.22 represents the activity of
the species $P$ from the reaction in Equation 2.15, and each exponent $p$ is the reaction
coefficient. The activity of a species is often defined as the effective concentration of the species. Since the reactions in an electrochemical cell take place at the surface of an electrode, the activities of the solid-phase species $A$ and $C$ from Equation 2.15 are proportional to the surface areas of these species on the electrode, as these surface areas represent their effective concentrations. The activity of species $E$ is the effective concentration of the diffused species $E$ adjacent to the electrode. Similarly, the open-circuit voltage due to the negative electrode is

$$E_\text{oc} = E_0^- + \frac{R_T}{nF} \ln \left( \frac{a_D^d}{a_A^a a_E^d} \right),$$

where $E_0^-$ is the standard voltage for the reaction and the terms $a_P$ are as described above.

Substituting Equations 2.22 and 2.23 into Equation 2.21 gives

$$E = E_0^+ + E_\text{oc}^- + \frac{R_T}{nF} \ln \left( \frac{a_D^d a_E^c}{a_A^a a_B^a} \right) + \frac{R_T}{nF} \ln \left( \frac{a_D^d}{a_B^b a_E^a} \right).$$

Combining the like terms in the equation above results in the Nernst equation for the overall reaction of Equation 2.17 (Linden, 1995), which is described as

$$E = E_0^+ + \frac{R_T}{nF} \ln \left( \frac{a_D^d a_E^c}{a_A^a a_B^a} \right),$$

where $E_0^+$ is the sum of the standard voltages at each electrode.

In both lithium- and nickel-based electrochemical cells, the negative electrode potential is very small relative to the reference material (Linden, 1995). If the activity effects of the negative electrode are ignored, then Equation 2.25 becomes

$$E = E_0^+ + \frac{R_T}{nF} \ln \left( \frac{a_D^d a_E^c}{a_A^a} \right).$$
The term $a_c^*$ in Equation 2.26 corresponds to the surface activity of the energized species $C$ in Equation 2.15, which is equivalent to the $\text{CoO}_2$ term given in Equation 2.12.

Similarly, the term $a_a^a$ corresponds to the surface activity of the unenergized species $A$ in Equation 2.15 or $\text{LiCoO}_2$ in Equation 2.12, and the term $a_e^a$ represents the surface activity of the diffused charge $\text{Li}^+$ in the electrolyte.

In dealing with battery applications, it is often more convenient to consider the stored and diffusing charges at a surface rather than effective species concentrations or surface activities. Using the one-electrode model of Equation 2.26, the energized species is species $C$, which will now be referred to as instantaneous stored charge. The total quantity of this species on the surface of the positive electrode will be given the symbol $q_s(t)$. Likewise, the diffusing ion species $E$ will hereafter be referred to as instantaneous diffusing charge. The surplus or deficit of the ionic species, relative to its bulk value, in the vicinity of the positive electrode, will be given the symbol $q_d(t)$. If $q_s(t)$ is the instantaneous stored charge, the quantity of the non-energized species $A$ will be written as $Q - q_s(t)$, where $Q$ is the total charge capacity of the cell. Using these charge terms, Equation 2.15 can be rewritten as

$$
(Q - q_s(t)) \leftrightarrow q_s(t) + q_d(t)^+ + e^-.
$$

(2.27)

for the positive electrode, where $a$, $c$, and $n$ are all chosen unity. These assumptions are chosen for the common positive lithium electrode reaction given in Equation 2.12. Equation 2.27 can easily be modified for other cell chemistries such as the positive nickel hydroxide electrode reaction given in Equation 2.1.

The charge terms, $q_s(t)$, $q_d(t)$, and $Q - q_s(t)$ are directly proportional to the surface activities of the corresponding chemical species at the electrode-electrolyte interface.
Substituting these time-dependent charge terms of Equation 2.27 into Equation 2.26 gives

$$E(t) = E^0 + \frac{R_g T}{nF} \ln \left( \frac{q_s(t)(q_d(t) + hq_{\text{bulk}})}{Q - q_s(t)} \right), \quad (2.28)$$

where $q_{\text{bulk}}$ is the linear bulk charge density and $h$ is the activity layer thickness. Equation 2.28 agrees with the actual voltage response of the cell shown in Figure 2.1. As the amount of stored charge, $q_s(t)$, increases, the terminal voltage of the cell will increase. Likewise, as the amount of stored charge approaches zero, the terminal voltage quickly drops to zero as the magnitude of the logarithmic term becomes large and negative. Of course, as the logarithmic terms approaches infinity, the Nernst equation is no longer valid.

The Nernst equation given in Equation 2.25 relates species surface activity to terminal voltage, but does not include current effects. It will be shown below that the Nernst equation is an approximation of the cell voltage for small currents, but does not represent larger current conditions.

When an electrical circuit is attached to the electrodes of a cell, electrons can flow through the circuit, allowing the redox reactions to occur. The current available is related to the electrode voltage and the surface activity of the reacting species by the Butler-Volmer equation applied to Equation 2.27 (Hartley & Lorenzo, 2002),

$$i(t) = k_0 \left( Q - q_s(t) \right) e^{\left( \frac{nF}{RT} \right) \left( 1 - \alpha \right) (E(t) - E^0)} - \left( q_s(t)(q_d(t)) \right) e^{\left( \frac{nF}{RT} \right) \left( -\alpha \right) (E(t) - E^0)}, \quad (2.29)$$

where $\alpha$ is a constant representing the reaction symmetry, $q_s(t)$ and $q_d(t)$ represent the stored and diffused charge amounts as discussed above, and $k_0$ is a constant.
The constant $k_0$ is dependent on the geometry of the electrode and the reaction rates of the chemicals given in Equation 2.17 (Bergveld et al., 2002). The rate of the chemical reactions within the electrochemical cell is dependent on temperature. This temperature-dependent reaction rate, $k_{Arr}$, is described by the Arrhenius equation as

$$k_{Arr} = k_0^0 e^{-E_a / R_g T}.$$  

(2.30)

In Equation 2.30, $k_0^0$ is the standard rate constant of the reaction, $E_a$ is the activation energy, $R_g$ is the gas constant, and $T$ is the temperature given in Kelvin. The Arrhenius rate, $k_{Arr}$ is a factor of $k_0$.

Although the Butler-Volmer equation shown in Equation 2.29 accurately describes what occurs at an electrode, it is generally impossible to solve for the voltage in terms of $i(t)$. However, several approximations to the solution are available.

One approximation is to assume the current is near zero. For very small currents, Equation 2.29 becomes

$$(Q - q_s(t)) e^{\left\{ \frac{nF}{R_g T} (1-\alpha)(E(t) - E^0) \right\}} - (q_s(t)q_d(t)) e^{\left\{ \frac{nF}{R_g T} (-\alpha)(E(t) - E^0) \right\}} = 0,$$  

(2.31)

which reduces to the Nernst equation for just the positive electrode shown above as Equation 2.28.

Another approximation to Equation 2.29 assumes the current is very large (in either direction). Choosing the discharge case, where the current is large and negative, gives

$$-|i(t)| = -k_0 \left\{ (q_s(t)q_d(t)) e^{\left\{ \frac{nF}{R_g T} (-\alpha)(E(t) - E^0) \right\}} \right\},$$  

(2.32)

where the $Q-q_s(t)$ term has been driven to zero by the associated exponential. Dividing
through by the term \(k_0(q_s(t)q_d(t))\) and taking the natural logarithm gives

\[
\ln\left( \frac{|i(t)|}{k_0(q_s(t)q_d(t))} \right) = \frac{nF}{R_sT}(-\alpha)(E(t) - E^0).
\] (2.33)

Now solving for \(E(t)\) gives

\[
E(t) = E^0 - \frac{R_sT}{cnF} \ln|i(t)| + \frac{R_sT}{cnF} \ln(k_0(q_s(t)q_d(t)))).
\] (2.34)

If the last term in Equation 2.34 remains fairly constant, then it can be combined with the \(E^0\) term giving

\[
E(t) = E^0 - \frac{R_sT}{cnF} \ln|i(t)|,
\] (2.35)

which is referred to as the Tafel equation (Linden, 1995).

The approximations discussed above have been derived from the Butler-Volmer equation. Other approximations to the cell voltage have been formulated based on empirical observations. One approximation (Plett, 2004) to the cell voltage is the Shepherd model

\[
E(t) = E_0 + R_{\alpha}i(t) + \frac{K_1}{q_s(t)},
\] (2.36)

where \(E_0\) is the initial voltage of the cell, \(R_{\alpha}\) is the resistance of the cell, and \(K_1\) is a constant. Other approximations are the Unnewehr universal model

\[
E(t) = E_0 + R_{\alpha}i(t) + K_2q_s(t)
\] (2.37)

and the Nernst model (not to be confused with the Nernst equation)

\[
E(t) = E_0 + R_{\alpha}i(t) + K_3 \ln(q_s(t)/Q) - K_4 \ln\left(\frac{Q - q_s(t)}{Q}\right).
\] (2.38)
In the approximations shown above, the resistance of the cell, $R$, and the constants, $K_i$, can be determined from experimental data.

A generalized form to approximate the solution of the Butler-Volmer equation is presented by Hartley and Jannette (2005). The approximation combines a resistance term along with the small current equation given by Nernst and the large current equation given by Tafel together into the form

$$E(t) = E_0 + R \frac{q_s(t)}{Q} + k_{i_1} \ln\left(1 + \left|i(t)\right|\right) \text{sign}(i(t)) + k_{qd} \ln\left(1 + \left|q_d(t)\right|\right) \text{sign}(i(t)) +$$

$$k_{qs}q_s(t) - k_{qsNE} \ln\left(q_s(t)\right) + k_{qsNF} \ln\left(\frac{Q - q_s(t)}{Q}\right)$$

where $Q$ refers to the cell capacity, and the constants $E_0$, $R$, and $k_{i_1}$ are dependent on the properties of the cell being used and can be determined from experimental data as described in Section 2.5. Positive current implies cell charging. Equation 2.39 shows that the voltage of the cell is controlled by the current and the amount of energized species within the cell.

2.1.3 Energy Cycle

An illustration of cell voltage versus charge for both charging and discharging at a constant current for NiH2 cells are shown in Figure 2.3 (Hartley & Lorenzo, 2002). These plots illustrate the hysteresis effect of an electrochemical cell’s charge/discharge cycle. The center line of the hysteresis curve assumes no current is flowing. This line represents the Nernst equation where the current is essentially zero. The stored energy in the cell is equivalent to the area below this line. The actual energy required to charge the cell is the area below the upper hysteresis line, and the actual energy that is used is the area below the lower hysteresis line.
Figure 2.3: A typical charge/discharge cycle for a NiH₂ cell (Hartley & Lorenzo, 2002).

The vertical lines indicate the instantaneous heating effects as the cell is either charged or discharged. During charging, the process of storing charge within the electrodes leads to a chemical cooling of the cell. This cooling effect is offset by the electrical heating caused by the internal resistance of the cell. When discharging, the
chemical reaction releases heat. This chemical heating combines with the electrical heating which reduces the actual energy that is available (Hartley & Lorenzo, 2002).

The horizontal line labeled $v_{TN}$ represents the thermoneutral voltage. The thermoneutral voltage comes from the enthalpy of the reaction, which is constant for NiH2 cells. If it were possible to charge the cell at this voltage, no net heating or cooling of the cell would occur. During charging, voltages above the thermoneutral voltage cause heating, while voltages below the line result in net cooling. During discharge, the voltages below the thermoneutral voltage result in both chemical and electrical heating (Hartley & Lorenzo, 2002).

Most electrochemical cells operate best between 20 and 40 °C (Linden, 1995). At low temperatures, there is less chemical activity, which increases the internal resistance of the cell. This higher resistance reduces the terminal voltage during discharge. Operating the cell at high temperatures will increase the amount of gassing, resulting in the loss of electrolyte and shortening the life of the cell. High temperatures also increase the amount of corrosion. For the remainder of this document, it will be assumed that cells will be operated at approximately constant temperature.

2.1.4 Voltage Measurements

The previous sections have shown that the cell voltage is a function of the cell’s current and temperature as well as the charged species within the cell. These voltages can be measured using either an analog circuit or an analog-to-digital circuit.

A simple operational amplifier comparator circuit is an example of an analog circuit that both measures and reacts to the cell voltage. The comparator can be used to monitor the cell voltage and to switch the charging circuit off if the voltage exceeds the
threshold voltage. Likewise, a comparator circuit can also be used to disconnect a load before the voltage drops below a certain value.

If a more advanced controller is necessary, the voltage can be digitized using an analog-to-digital converter. One example of using the digitized voltage would be in a laptop computer. Software running on the computer can use the output of the analog-to-digital converter to calculate an estimate of the remaining laptop battery charge. In addition, once the voltage falls below a threshold voltage, a proper shutdown of the computer can be performed, thereby saving the system state and preventing file corruption that can occur if power were abruptly lost.

2.1.5 Cell Voltage Summary

This section has addressed the relationship between the internal chemical reactions and the terminal voltage of an electrochemical cell. The next section addresses the behavior of the cell when a current passes through it.

2.2 Electrochemical Cell: Current

When creating a BMS, one of the primary quantities of an electrochemical cell to be monitored is the state of charge (SOC). SOC is often described as being analogous to the amount of gasoline in the tank of a typical automobile. While this analogy is certainly valid, especially when referring to an electric vehicle, determining the amount of gasoline in a tank can be done quite accurately with a simple float device. Determining the state of charge of a battery, however, is more difficult. Charge enters or leaves the cell via the current. Faraday’s law of electrochemistry states that the species reaction rates inside the cell are equal to the current entering or leaving the cell, ignoring
any side reactions (Bergveld et al., 2002). The rate of change in the energized material, often referred to as the stored charge, can be defined as

\[ \dot{q}_s(t) = a(q_s)i(t) - k_{sd}q_s(t), \]  

(2.40)

where \( a(q_s) \) is a function describing the charge acceptance and \( k_{sd} \) is the rate constant of the cell’s self-discharge (Hartley & Jannette, 2005).

### 2.2.1 Charge Acceptance

Charge acceptance is defined by Gu et al. (1998) as "the ratio of partial charge used by the electrochemical reaction [in reversing] the active materials to the total charge applied to the cell." The term \( a(q_s) \) in Equation 2.40 is required to provide an upper limit to the amount of charge that can be stored in the cell. Figure 2.4 illustrates charge acceptance using the analogy of a storage tank. As current is poured into or out of the tank, the amount of stored charge within the tank changes. In this analogy, the charge acceptance term accounts for the spillage of current. In reality, the spillage is lost as side reactions and gassing. Once the tank is completely filled, the tank can no longer accept any more charge. In an ideal world, the battery would transform all the charge applied to the cell into stored charge for later use, as shown in the first plot of Figure 2.5. A more realistic plot of charge acceptance is shown in the second plot of Figure 2.5. This second plot was generated using the equation

\[ a(q_s) = \begin{cases} 1, & i(t) > 0 \\ 1 - 5e^{-40(1-q_s(t)/Q)} + 4e^{-50(1-q_s(t)/Q)}, & i(t) \leq 0 \end{cases}, \]  

(2.41)

which was presented by Hartley and Jannette (2005). The exponential terms in Equation 2.41 provide the upper limit on the total amount of charge that can be stored in the cell, as well as the rounding effects near full SOC.
Figure 2.4: Visualization of charge acceptance (Hartley & Lorenzo, 2002).

Figure 2.5: Plots showing ideal acceptance and non-ideal acceptance realized using the equation provided by Hartley & Jannette (2005).
As stated above, charge acceptance describes how much of the current through an electrochemical cell is actually stored as energized material. As the cell approaches full charge, all of the active material becomes energized. Quite often, the ideal case of acceptance, shown in the first plot of Figure 2.5, is accurate enough for modeling. Assuming the ideal case of acceptance and ignoring self-discharge, the change in stored charge of the cell becomes

$$\dot{q}(t) = i(t).$$

(2.42)

This equation leads to the amp-hour integrator or amp-hour counting method discussed in the next section. If more accuracy of the model near full charge is needed, then an approximation similar to the one shown in the second plot of Figure 2.5 can be used.

2.2.2 State of Charge

Since there is no sensor that will read the state of charge directly, the state of charge must be determined using other methods. For lead acid and lithium-ion batteries, the relationship between the stabilized open circuit voltage, $E_{ocs}$, and the $SOC$ is approximately linear (Wang & Stuart, 2002). By disconnecting the cell from the circuit and allowing the open circuit voltage, $E_{oc}$, to stabilize, the $SOC$ can be directly determined. Unfortunately, the open circuit voltage will drift before stabilizing. For lead acid batteries, the voltage could drift for up to 60 minutes, although after 15 minutes, the drift is considered minor. A Li-Ion cell reaches its $E_{ocs}$ faster than a lead acid cell. Wang suggests using this linear relationship to reset a charge estimator during times of rest; however, the flatness of the $E_{ocs}$ versus $SOC$ curve limits the accuracy of this approach (Wang & Stuart, 2002). For most battery chemistries, the relationship between the $SOC$
and $E_{oc}$ is temperature dependent and varies with age, and using the stabilized open circuit voltage is not often feasible.

Another method used to determine the state of charge requires prior discharge tests at various constant currents with the results stored in a look-up table. During operation, the voltage of the cell is measured, and the present state of charge is approximated using some form of interpolation of the stored data.

A simple method for tracking the state of charge of a cell is referred to as amp-hour counting or Coulomb counting (Hansen & Wang, 2005). From Equation 2.42, the state of charge in an ideal cell would be

$$SOC(t) = SOC(0) + \int_0^t \frac{i(\tau)}{Q} d\tau ,$$

(2.43)

where $Q$ is the capacity of the cell. As Equation 2.43 shows, the state of charge is the integral of current through the cell divided by the capacity of the cell. The $SOC$ is usually given as a fraction or percentage of the capacity $Q$. When performing Coulomb counting, the cell or battery voltage is used to determine the initial $SOC$ and can lead to a 15% error in the state of charge estimate of that charge/discharge cycle (Hansen & Wang, 2005).

2.2.3 Capacity

The capacity, $Q$, of a cell is the approximate amount of charge that can be stored in the cell when it is fully charged (Hansen & Wang, 2005). For example, if a fully-charged cell has a capacity of 10 Ah (Amp-hours) and it is at 50% $SOC$, then the cell is expected to produce a constant current of 10 A for a half hour before reaching full discharge. Assuming the cell is initially fully charged, if $i(t) = -I_d$, then Equation 2.43
gives

\[ SOC(t_s) = 1 - \frac{I_d t_s}{Q}, \]

where \( t_s \) is the discharge time, and \( I_d \) is the value of the constant discharge current (Van Mierlo et al., 2004).

The usable capacity of the cell is affected by the rate at which the cell is discharged. Figure 2.6 shows the discharge plots of a lithium-ion cell rated at 2.4 Ah for three constant currents. When the current is increased, the state of charge available for use decreases due to diffusion effects in the electrolyte. For constant current, an estimate of discharge time based on the magnitude of the current can be obtained using Peukert’s equation. Peukert’s equation is given as

\[ Q = I_d^\gamma t_s, \]

where \( Q \) is the nominal capacity of the cell, \( I_d \) is the constant discharge current, \( t_s \) is the time required for complete discharge, and \( \gamma \) is a constant dependent on the type of cell and is referred to as the Peukert number (Van Mierlo et al., 2004). In the ideal case, the Peukert number would be one. In reality, the Peukert number ranges from 1.0 to 1.4.

Normally, the parameters \( Q \) and \( \gamma \) are found experimentally from discharge data. Then, given a current, the discharge time can be calculated from Equation 2.45. The process for finding the parameters is now demonstrated. Figure 2.6 shows discharge curves for a 2.4 Ah lithium-ion cell. The legend in Figure 2.6 shows the magnitude of the constant discharge current, the discharge time (in minutes), the amount of charge that was added to the cell during the charge phase, and the amount of actual charge removed from the cell during the discharge cycle shown. Taking the natural logarithm of both sides of Equation 2.45 gives
\[
\ln(Q) = \gamma \ln(I_d) + \ln(t_e).
\] (2.46)

Subtracting the term containing $\gamma$ from both sides of Equation 2.46 yields
\[
K_Q - \gamma \ln(I_d) = \ln(t_e),
\] (2.47)

where $K_Q$ is a constant representing the $\ln(Q)$. Using batch least squares (which is discussed in Section 2.6 of this paper) and the currents and times from Figure 2.6, the value of $K_Q$ and $\gamma$ were found to be 8.9947 and 1.1813 respectively. The nominal capacity of the cell, $Q$, is then calculated as
\[
Q = e^{K_Q} = 2.2389 \text{ Ah},
\] (2.48)

which is very close to the rated capacity of the cell. The next section of this paper addresses the diffusion properties of an electrochemical cell.

![Figure 2.6: Discharge of 2.4 Ah lithium-ion cell at three different constant currents.](image-url)
2.2.4 Diffusion

There are three processes that control mass transport, i.e., ion flow through the electrolyte: convection, the electric field potential, and diffusion (Linden, 1995). Convection effects are minimal in most electrochemical cells due to the consistency of the electrolyte used. The effect from the electric field is also minimal due to the electrolyte’s high conductivity (Bergveld et al., 2002). Therefore, diffusion is the primary process for ion flow through the electrolyte.

Almost everyone has experienced the effect of diffusion at one time or another. Consider a flashlight. After the flashlight has been used for an extended time, the light becomes very dim. When the flashlight is turned off and the batteries are allowed to rest, the charge distribution in the electrolyte will equalize throughout the cell. When the flashlight is turned back on, it will initially provide a brighter beam because the electrolyte has returned to its equalized state.

The delayed effect of diffusion on the terminal voltage of a cell can be easily seen in Figure 2.7, which shows step changes in the current applied to a 9.5 Ah NiMH cell. When the cell has been at rest for a significant amount of time, the species within the electrolyte are at a constant concentration throughout the cell. As current flows through the cell, the concentration of ions in the electrolyte will increase or decrease at the electrodes as illustrated in Figure 2.8.

The diffusion within the cell can be described using Fick’s first and second laws (Bergveld et al., 2002). Fick’s first law, in one dimension, is given as

\[
 k_d \frac{\partial q_f(x,t)}{\partial x} = -j_d(x,t),
\]

(2.49)

where \( q_f(x,t) \) is the effective concentration of diffusing charge, \( t \) is time, \( x \) is the position
Figure 2.7: Step changes in cell voltage showing diffusion effects.

Figure 2.8: Illustration of ion diffusion within a cell (Hartley & Lorenzo, 2002).
in the electrolyte, $k_d$ is the diffusion coefficient, and $j_d(x,t)$ is the charge flux at location $x$. Since only one dimension and one charged ionic species are being considered in Equation 2.49, the flux $j_d(x,t)$ is equal to the current $i(t)$. Fick’s second law describes the dynamics of the diffusion process and is given by

$$\frac{\partial q_f(x,t)}{\partial t} = k_d \frac{\partial^2 q_f(x,t)}{\partial x^2}.$$  \hspace{1cm}(2.50)

The boundary conditions for Equation 2.50 are

$$k_d \frac{\partial q_f(0,t)}{\partial x} = -j_{d0}(t)$$ \hspace{1cm}(2.51)

and

$$k_d \frac{\partial q_f(L,t)}{\partial x} = -j_{dl}(t)$$ \hspace{1cm}(2.52)

with an initial condition of

$$q_f(x,0) = q_{bulk},$$ \hspace{1cm}(2.53)

where $L$ represents the right boundary, $q_{bulk}$ is the equilibrium value of $q_f$, also known as the bulk charge density, and $j_{d0}$ and $j_{dl}$ are the flux at zero and $L$ respectively. For one diffusing species, Hartley and Lorenzo (2002) give the solution to Equation 2.50 in the Laplace domain as

$$q_f(x,s) = \frac{\cosh(\sqrt{\beta s} x)}{\sqrt{\beta s} \sinh(\sqrt{\beta s} L)} J_L(s) - \frac{\cosh(\sqrt{\beta s} (L-x))}{\sqrt{\beta s} \sinh(\sqrt{\beta s} L)} J_0(s) + \frac{q_{bulk}}{s},$$ \hspace{1cm}(2.54)

where $J_L(s)$ and $J_0(s)$ are the Laplace transforms of the terms $j_{dl}(t)$ and $j_{d0}(t)$ respectively, and $\beta$ is $1/k_d$. As $L$ approaches infinity and letting $x = 0$, Equation 2.54 then becomes

$$q_f(0,s) = -\frac{1}{\sqrt{\beta s}} J_0(s) + \frac{q_{bulk}}{s}.$$ \hspace{1cm}(2.55)
Electrochemists refer to the term $1/\sqrt{\beta s}$ in Equation 2.55 as the Warburg impedance, which will be discussed further in Section 2.3. Figure 2.9 shows a plot of the impedance of an electrochemical cell. This plot is referred to as a Cole-Cole plot (IUPAC, 1994), which is a Nyquist plot with the vertical axis flipped, and is often used by electrochemists. This impedance plot is comprised of the ohmic resistance of the cell, $R_{\Omega}$, in series with the diffusive capacitor term $C_d$, which is in parallel with the diffusion resistance $R_d$ and the Warburg impedance. The values of the parameters for the plot were chosen to show the typical shape of the impedance of an electrochemical cell as reported in literature (Bergveld et al., 2002) (Linden, 1995) but do not correspond to values from an actual physical cell.

As the plot in Figure 2.9 illustrates, at very low frequencies, the $1/\sqrt{\beta s}$ term from Equation 2.55 dominates, and the impedance of the cell is large and has a phase angle near $-45^\circ$. The Warburg impedance term is sometimes replaced with a constant phase element, CPE, where the lower frequency phase shift may be a value other than $-45^\circ$.

One method of approximating Fick’s second law is to lump space using a central difference as

$$k_d \frac{q_{f1}(t) - q_{f0}(t)}{H} = -j_{d0}(t) \quad (2.56)$$

and

$$k_d \frac{q_{fL}(t) - q_{f(L-1)}(t)}{H} = -j_{dL}(t). \quad (2.57)$$

For $n$ lumps, the diffusion vector, $X$, is defined as
\begin{equation}
X = \begin{bmatrix}
q_{f_1}(t) & q_{f_2}(t) & q_{f_3}(t) & \cdots & q_{f(n-2)}(t) & q_{f(n-1)}(t) & q_{jn}(t)
\end{bmatrix}^T,
\end{equation}

and the input vector, $U$, is defined as

\begin{equation}
U = \begin{bmatrix}
-j_{d_0} & -j_{d_t}
\end{bmatrix}^T.
\end{equation}

Then the approximation to Fick’s second law given in Equation 2.55 becomes

\begin{equation}
\dot{X} = \Psi X + \Lambda U,
\end{equation}

where
Given that the matrix $\Psi$ is real and symmetric, it has real eigenvalues with one eigenvalue at the origin. Due to Kirchoff’s current law, the charge fluxes $j_{d0}$ and $j_{dL}$ must be equal, therefore Equation 2.62 may be replaced by

$$\lambda = \frac{1}{H} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & -1 \end{bmatrix},$$  \hspace{1cm} (2.63)
resulting in a zero at the origin which cancels out the single eigenvalue at the origin from the matrix $\Psi$. Assuming the smallest of the remaining eigenvalues is dominant, the diffusion equation can be approximated as

$$\dot{q}_d(t) = k_{d1}a(q_s)i(t) - k_{d2}q_d(t), \quad (2.65)$$

where $q_d(t) = h(q_f(0,t) - q_{bulk})$, $k_{d1}$ is a gain constant, $-k_{d2}$ is the remaining dominant eigenvalue from the matrix $\Psi$, $q_{bulk}$ is from the initial condition given in Equation 2.53, and $a(q_s)$ is the charge acceptance term from Equation 2.40 (Hartley & Lorenzo, 2002). If necessary, more terms could be kept in the approximation and similar equations could be applied for each diffusion species.

2.2.5 Cell Current Summary

This section has discussed the current through the electrochemical cell, and how this current affects the amounts of chemical species within the cell. The following section combines the material from Sections 2.1 and 2.2 to generate mathematical models of the electrochemical cell.

2.3 Electrochemical Cell: Dynamic Models

Due to transient effects, measurements from sensors are not sufficient to determine the $SOC$ within a cell. Therefore, many types of dynamic models have been developed to estimate the $SOC$ of a cell. Feedback from sensors is normally used to minimize errors from the model. As discussed earlier, one method of determining $SOC$ is to create a pre-defined look-up table that maps $E_{oc}$ (the open circuit voltage) to $SOC$ for various currents. Another method is to calculate $SOC$ by integrating the current passing through the cell. More advanced methods create state observers using a derived model.
This section discusses some common modeling methods based on the physical properties of the electrochemical cell presented in the previous sections.

2.3.1 Electric Circuit Models

An accurate model is typically required to create a controller. Often, a constant DC voltage source is used to model a battery. This section will build upon this simple model by incorporating the dynamics of the electrochemical cell discussed in the previous sections. The simplest addition to the constant voltage source model is a resistor in series with the constant voltage source as shown in Figure 2.10. This simple model accounts for the instantaneous voltage drop across the cell when a current is applied to the cell, but it does not address the other dynamics of the cell.

One of the key dynamics of a cell that should be addressed is diffusion. In Section 2.2.4, the diffusion in a cell was discussed, and an approximate solution to the change in the diffusing charge was given in Equation 2.65. This differential equation has the same form as the voltage across the parallel RC circuit shown in Figure 2.11.

Rewriting Equation 2.65 in terms of the circuit elements \( R_d \) and \( C_d \) yields

\[
\frac{dV_d(t)}{dt} = \frac{1}{C_d}i(t) - \frac{1}{R_d C_d}V_d(t),
\]

where the voltage, \( V_d(t) \), dropped across the capacitor \( C_d \) is proportional to the diffused charge, \( V_d(t) = k_d q_d(t) \). As illustrated in Figure 2.12, the values for the circuit elements \( R_d \) and \( C_d \) can be determined experimentally by applying a step change in current to the circuit in Figure 2.11.

Eveready Battery Company (2001) provides an equivalent linear electric circuit model that combines Figure 2.10 and Figure 2.11 together as shown in Figure 2.13. This
Figure 2.10: Simple resistive cell model.

Figure 2.11: RC circuit representing diffusion within a cell.

Figure 2.12: Voltage across parallel RC circuit due to a step change in current (Hartley & Lorenzo, 2002).
model is for a NiMH cell. In the circuit, $E_0$ is defined as the effective open circuit cell voltage which is in series with the resistor $R_\Omega$, the effective instantaneous resistance, and the resistor $R_d$, which is in parallel with the capacitor $C_d$. Eveready defines $R_d$ as the delayed resistance and $C_d$ as the effective parallel capacitance.

Another key cell dynamic, which is not included in the Eveready model, is the effect of the state of charge on the terminal voltage of the cell. Figure 2.1, which is repeated in Figure 2.14 for convenience, shows how the terminal voltage of an electrochemical cell decreases as the cell is discharged. Figure 2.14 also indicates the areas of the discharge cycle that are approximately linear with respect to state of charge and those that are nonlinear.

Often the terminal voltage of the electrochemical cell is assumed to be constant within the middle of the operating range. However, as seen in Figure 2.14, the terminal voltage of the cell near the middle of the graph is approximately linear, but gradually decreases as the cell is discharged. This approximately linear decrease in voltage can be modeled by inserting a capacitor in series with the voltage source as shown in Figure 2.15. The voltage across the capacitor $C_s$ is proportional to the stored charge in the cell; that is, $v_s(t) = k_s q_s(t)$. As the state of charge in the cell increases or decreases, the voltage $v_s(t)$ across this capacitor will increase or decrease accordingly.

None of the models introduced in this section so far has addressed the loss of charge while the circuit is at rest. To account for this loss of charge, a resistor can be placed in parallel with the storage capacitor. Figure 2.16 shows a full circuit model that includes a resistive term, $R_{sd}$, representing the self-discharge of the cell.

In Section 2.2.4, a model of the diffusion process was presented. This model of the diffusion process resulted in Equation 2.54, which contains the Warburg impedance.
Figure 2.13: Linear circuit model for NiMH cell as described by Eveready Company (2001).

Figure 2.14: Voltage discharge curve for a NiMH cell indicating linear and nonlinear regions.
In the electrical circuit models presented above, the entire diffusion process is approximated by the parallel circuit branch containing the elements $C_d$ and $R_d$. A more complete model is shown in Figure 2.17 with the Warburg impedance added in series with the resistor $R_d$. In Figure 2.18, the Warburg impedance is replaced with a constant phase element. The next section presents a model that incorporates the nonlinear aspects of the electrochemical cell voltage.

![Figure 2.15: Circuit model including a capacitor representing capacity of the cell.](image)
Figure 2.16: Circuit model including self-discharge resistor.

Figure 2.17: Circuit model including Warburg impedance.
2.3.2 Plett’s Model

Plett (2004) developed a model that incorporated the nonlinear voltage characteristics of the electrochemical cell by combining the Shepherd, Unnewehr, and Nernst models given in Equations 2.36, 2.37, and 2.38 respectively. Combining these models, Plett defines the cell voltage as

\[
E(t) = E_o + \frac{K_1}{q_s(t)} + R_2 i(t) + K_2 q_s(t) + K_3 \ln(q_s(t)/Q) + K_4 \ln\left(\frac{Q - q_s(t)}{Q}\right)
\]  

(2.67)

and the stored charge as

\[
q_s(t) = q_s(0) - \int_{0}^{t} \eta i(\tau) d\tau,
\]  

(2.68)
where \( \eta \) is the charge efficiency, which is the average value of the charge acceptance.

Using Equation 2.68 to determine the stored charge and batch least squares to identify the unknown coefficients in Equation 2.67, Plett was able to closely model the cell voltage even in the nonlinear regions near full charge and near full discharge.

### 2.3.3 First-Principles Model

Although the Plett model described in the previous section provides good results, it lacks the dynamics of the cell caused by the diffusion of the charge within the cell. Hartley and Jannette (2005) provide a more complete model which is an approximation based on first-principles. Their model includes the stored charge equation, repeated here as

\[
\dot{q}_s = a(q_s)\dot{i}(t) - k_{sd}q_s(t), \tag{2.69}
\]

which is based on Faraday’s law. Their model also includes the diffusion equation (Equation 2.66),

\[
\dot{q}_d = k_{d1}a(q_s)\dot{i}(t) - k_{d2}q_d(t), \tag{2.70}
\]

which is based on Faraday’s and Fick’s laws. Hartley and Jannette’s model also includes the nonlinear voltage equation, Equation 2.39, repeated here as

\[
E(t) = E_0 + R_{\text{li}}i(t) + k_{\text{li}}\ln(1 + |i(t)|)\text{sign}(i(t)) + k_{\text{qd}}\ln(1 + |q_d(t)|)\text{sign}(i(t)) + k_{\text{qs}}q_s(t) - k_{\text{qsNF}}\ln\left(\frac{q_s(t)}{Q}\right) + k_{\text{qsNF}}\ln\left(\frac{Q - q_s(t)}{Q}\right), \tag{2.71}
\]

which is based on the Butler-Volmer equation. The differential Equations 2.69 and 2.70 each represent one of the parallel RC branches of the electrical circuit model shown in Figure 2.16. By combining these two equations along with the nonlinear voltage Equation 2.71, Hartley and Jannette (2005) were able to create a model that accurately
tracks the state of charge even within the nonlinear regions near full charge and full
discharge of the cell as shown in Figure 2.14. By applying a step change in current to the
cell, the parameters for $R_\tau$, $R_d$, and $C_d$ can be determined. Likewise, by fully charging
the cell, allowing it to sit unused for an extended period of time and then recharging it,
the value for the product $C_d R_d$ can be obtained. Once these parameters have been
determined, the constants in Equation 2.71 can be determined from measured data using
batch least squares. Since this model includes a dynamic state variable representing the
stored charge, $q_s(t)$, this model can be used to track the state of charge in the cell.

2.3.4 Modeling Temperature Effects

As discussed in Section 2.1.2, the rate of the chemical reactions of the cell is
temperature dependent based on the Arrhenius equation given as Equation 2.30. To
address the temperature dependence of the cell voltage, Annavajjula et al. (2008) used a
modified version of Equation 2.71 including second-order temperature effects as

$$
E(t) = (k_{AT_0} + k_{AT_1} T + k_{AT_2} T^2)(E_0 + R_\tau \dot{i}(t) + k_{qd} q_s(t) + k_{qT} q_{sT}(t) + k_{qN} q_{sN}(t)
+ k_{qd} q_{dT}(t) + k_{qT} q_{sT}(t) + k_{qN} q_{sN}(t))
$$

In Equation 2.72 above, the $\ln(\dot{i}(t))$ term from Equation 2.71 was removed, the logarithm
terms associated with the stored charge near empty and near full were replaced with
exponentials, and the logarithm associated with diffused charge was replaced with the
linear diffused charge term. Although this model makes adjustments to account for the
temperature changes in the cell, identification of the unknown coefficients becomes more
difficult.
2.3.5 Chosen Dynamic Models

The previous subsections of Section 2.3 presented a number of dynamic cell models. This study uses three dynamic models. These models are based on the first-principles model presented in Section 2.3.3 because it is the most complete. The first model contains no temperature correction. The first equation for this model addresses the change in stored charge as

\[ \dot{q}_s(t) = i(t), \]  

(2.73)

where \( a(q_s) = 1 \) for all states of charge, and the self discharge term \( k_{sd} \) is assumed to be zero. The second equation for this model is the diffusion equation

\[ \dot{q}_d(t) = \frac{1}{k_{Cd}} i(t) - \frac{1}{k_{Cd} k_{Cd}} q_d(t), \]  

(2.74)

where the unknown coefficients \( k_{d1} \) and \( k_{d2} \) from Equation 2.70 have been replaced with \( \frac{1}{k_{Cd}} \) and \( \frac{1}{k_{Cd} k_{Cd}} \) to match the electrical circuit analogy given in Equation 2.66. The third equation for this model is the voltage equation

\[
E(t) = k_{E0T0} + k_{RT0} \dot{i}(t) + k_{i(T0)} \ln(1 + |i(t)|) \text{sign}(i(t)) + k_{d0} \dot{q_d}(t) + k_{qT0} q_s(t) + k_{qNET0} \ln\left(\frac{q_s(t)}{Q}\right) + k_{qNET0} \ln\left(\frac{Q - q_s(t)}{Q}\right),
\]

(2.75)

and the diffusion term is assumed to be linear as in Equation 2.72. Table 2.1 lists each of the seven coefficients shown in Equation 2.75 and describes their relationship with the physical cell.

The other two models also use Equations 2.73 and 2.74, but they include temperature correction terms added to Equation 2.75. A first-order temperature-corrected model is created by setting the coefficient \( k_{AT2} \) from Equation 2.72 to zero and
Table 2.1: Description of the non-temperature corrected coefficients of the voltage Equations 2.75-2.77.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{E0T0}$ (V)</td>
<td>Base voltage of a cell when the cell current, diffused charge, and stored charge are all zero.</td>
</tr>
<tr>
<td>$k_{R0T0}$ (Ω)</td>
<td>Primary resistance of the cell. Caused primarily by the SEI layer and the diffusion through the electrolyte.</td>
</tr>
<tr>
<td>$k_{iT0}$ (V)</td>
<td>Polarization adjustment caused by the current flowing in the cell.</td>
</tr>
<tr>
<td>$k_{qdT0}$ (V/seconds)</td>
<td>Potential adjustment caused by the diffused charge within the cell.</td>
</tr>
<tr>
<td>$k_{qsT0}$ (V/seconds)</td>
<td>Linear change in potential due to the stored charge within the electrode.</td>
</tr>
<tr>
<td>$k_{qsNET0}$ (V)</td>
<td>Potential of the cell when it is near empty (very little charged material remaining), resulting in a majority of uncharged material available within the electrode.</td>
</tr>
<tr>
<td>$k_{qsNFT0}$ (V)</td>
<td>Potential of the cell when it is near full (very little uncharged material remaining), resulting in a majority of charged material available within the electrode.</td>
</tr>
</tbody>
</table>

then conceptually multiplying the temperature correction terms with Equation 2.75 resulting in a total of 14 coefficients to be determined. The first-order temperature-corrected voltage equation is

$$E(t) = k_{E0T0} + \left(k_{E0T1} + k_{R0T0} + k_{R0T1}T\right)i(t) + \left(k_{iT0} + k_{iT1}T\right)\ln\left(1 + \left|i(t)\right|\text{sign}(i(t))\right) + \left(k_{qdT0} + k_{qdT1}T\right)q_d(t) + \left(k_{qsT0} + k_{qsT1}T\right)q_s(t) + \left(k_{qsNET0} + k_{qsNET1}T\right)\ln\left(\frac{q_s(t)}{Q}\right) + \left(k_{qsNFT0} + k_{qsNFT1}T\right)\ln\left(\frac{Q - q_s(t)}{Q}\right). \quad (2.76)$$

A second-order temperature-corrected model assumes $k_{AT2}$ is not zero. Conceptually, multiplying the temperature correction terms with Equation 2.75 results in a total of 21 coefficients to be determined. The second-order temperature-corrected voltage equation is
\[ E(t) = k_{E0T0} + k_{E0T1}T + k_{E0T2}T^2 + (k_{R3T0} + k_{R3T1}T + k_{R3T2}T^2)\rho(t) + \\
(k_{gT0} + k_{gT1}T + k_{gT2}T^2)\ln(1 + |i(t)|)\text{sign}(i(t)) + \\
(k_{qT0} + k_{qT1}T + k_{qT2}T^2)q_d(t) + (k_{qsT0} + k_{qsT1}T + k_{qsT2}T^2)q_s(t) + . \] (2.77)

\[ (k_{qsNET0} + k_{qsNET1}T + k_{qsNET2}T^2)\ln\left(\frac{q_s(t)}{Q}\right) + \\
(k_{qsNFT0} + k_{qsNFT1}T + k_{qsNFT2}T^2)\ln\left(\frac{Q - q_s(t)}{Q}\right) \]

The coefficients with the subscripts of \( T1 \) and \( T2 \) in Equations 2.76 and 2.77 show the first- and second-order temperature effects of their corresponding \( T0 \) terms listed in Table 2.1. Each of these three dynamic cell models is used for identification which will be discussed in Chapter IV. The next section of this chapter discusses how the electrochemical cell changes as the cell ages.

2.4 Electrochemical Cell: Aging

Cell aging is a complex mechanism. There are two primary observable changes in a cell as it ages. As an electrochemical cell ages, it loses capacity. This loss of capacity is usually referred to as capacity fade (Spotnitz, 2003). In addition to the loss of capacity, the internal resistance of the cell increases with age. This increased resistance is sometimes referred to as power fade (Plett, 2004). Many studies on the aging of lithium-ion cells have been documented in the literature, which include capacity fade, power fade, and other aging mechanisms. Table 2.2 provides a summary of these studies. This section reviews the results of these studies.
2.4.1 Cycle Life and Life Fraction

Cycle life of an electrochemical cell refers to the number of times a cell can be cycled before the capacity of the cell fades below a minimum value (Bergveld et al., 2002). This minimum acceptable value of cell capacity is defined by the United States Advanced Battery Consortium (USABC) to be 80% of the rated capacity of the cell (Dubarry et al., 2007).

Since any two cells will reach their end of life during different cycles, it is sometimes convenient to normalize the cycle life for comparing the performance of the cells. This paper defines the term life fraction \( LF \) as

\[
LF = \frac{\text{cyclenumber}}{\text{lastcycle}},
\]

where \( \text{cyclenumber} \) is the present cycle of the cell, and \( \text{lastcycle} \) is the cycle where the capacity of the cell falls below the minimum acceptable value.

2.4.2 Maximum Charging Voltage

The capacity of a lithium-ion cell is dependent on the maximum charging voltage (MCV) of the cell (Bergveld et al., 2002). As the MCV is increased, the stored capacity increases, however, increasing the MCV of the cell dramatically reduces the cycle life of the cell (Spotnitz, 2003) (Bergveld et al., 2002). Figure 2.19 shows a plot of data presented by Bergveld et al. (2002). This plot displays the inverse relationship of cycle life versus the MCV. This figure also displays the capacity of a cell versus the MCV. As this plot shows, changing the MCV of the cell from 4.1 V to 4.7 V results in an almost 50% increase in the capacity of the cell, but reduces the cycle life of the cell from approximately 600 cycles down to 10 cycles, which is a 98% decrease in cycle life.
Table 2.2: Summary of aging studies presented in the literature.

<table>
<thead>
<tr>
<th>Author</th>
<th>Size (Ah)</th>
<th>Cell</th>
<th>Cycling Discharge</th>
<th>Temp. (°C)</th>
<th>Testing</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abraham et al. 2004</td>
<td>1</td>
<td>Modified 18650</td>
<td>Added reference electrode</td>
<td>Calendar</td>
<td>55</td>
<td>Reference electrode, EIS, CF, Z,</td>
</tr>
<tr>
<td>Abraham et al. 2007</td>
<td>1</td>
<td>18650</td>
<td>C and C/25, 3.72-4.1 V</td>
<td>25, 45, 55</td>
<td>HPPC, EIS Individual electrodes analyzed CF, Z, SEI</td>
<td>During early weeks capacity fade was faster at C/25, but then faster at C/1 rate. Fading faster for cycling than calendar time. Positive electrode main cause of impedance rise</td>
</tr>
<tr>
<td>Dubarry et al. 2007</td>
<td>0.9</td>
<td>18650</td>
<td>Pulsed 9C, C/3, C/10</td>
<td>Room</td>
<td>CF, PF, Z</td>
<td>Early capacity loss is due to NiO thin-film on the positive electrode. Later capacity loss due to SEI of negative electrode. Capacity loss coincides with increase in polarization or the activation energy.</td>
</tr>
<tr>
<td>Eom et al. 2007</td>
<td>1</td>
<td>Canister</td>
<td>C/2</td>
<td>23</td>
<td>CF, TTF</td>
<td>Mean time to failure 1024 cycles.</td>
</tr>
<tr>
<td>Kobayashi et al. 2007</td>
<td>0.4</td>
<td>Cylinder</td>
<td>Pulse C, C/2</td>
<td>0, 20, 40, 60, 80</td>
<td>CF, Z, XANES</td>
<td>Change in shape of the positive electrode to inactive (Li deficient) cubic shape. A film of Li₂Co₃ covered the positive electrode.</td>
</tr>
<tr>
<td>Ning et al. 2003</td>
<td>1.4</td>
<td>18650</td>
<td>C, 2C, 3C</td>
<td>Room</td>
<td>CF, Z, individual electrode impedance</td>
<td>Tested at 300 cycles. Two electrode Equation. Increase in impedance of positive electrode is the dominant cause of aging for at 1C rate, but impedance of negative electrode dominates at higher discharge rates.</td>
</tr>
<tr>
<td>Ning et al. 2006</td>
<td>1.7</td>
<td>18650</td>
<td>C/2</td>
<td>Room</td>
<td>Concentration of Li, size of activation energy</td>
<td>Due to parasitic reactions, not all lithium intercalates back into the negative electrode, therefore the lithium concentration in the anode is smaller at the beginning of the next discharge cycle. As the potential of the electrodes varies, the activation energy varies. The majority of cycleable lithium is lost during CV charging phase due to parasitic reactions based on the negative electrode activation energy.</td>
</tr>
</tbody>
</table>
Table 2.2: Summary of aging studies presented in the literature (continued).

<table>
<thead>
<tr>
<th>Author</th>
<th>Size (Ah)</th>
<th>Cell Type</th>
<th>Cycling Discharge</th>
<th>Temp. (°C)</th>
<th>Testing Method(s)</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ploehn et al. 2004</td>
<td>40 5</td>
<td>Cylinder &amp; Prismatic</td>
<td>Calendar</td>
<td>15, 30, 40, 60</td>
<td>CF, SEI thickness</td>
<td>Capacity loss increases at $t^{0.5}$. SEI growth at $t^{0.5}$.</td>
</tr>
<tr>
<td>Santhana-gopalan et al. 2008</td>
<td>1.7</td>
<td>Pouch C/2</td>
<td>5, 15, 25, 35, 45</td>
<td>CF, OCP, Reference electrode</td>
<td>Capacity fade from cathode is not significant. Loss of active material at anode is dominate. Presented model that tracks weight and Li inside the electrode.</td>
<td></td>
</tr>
<tr>
<td>Shim et al. 2002</td>
<td>1.4m</td>
<td>Custom Built C/5, C/2, C, 2C</td>
<td>25, 60</td>
<td>PF, CF, Z, Individual electrodes analyzed</td>
<td>Positive electrode impedance increased. Capacity of positive electrode decreased significantly with temperature.</td>
<td></td>
</tr>
<tr>
<td>Shim et al. 2003</td>
<td>1.4m</td>
<td>Custom Built C/2 to 70% and 100% DOD</td>
<td>Room</td>
<td>OCEOV CF, Z, EIS</td>
<td>Open-circuit end of voltage decreased with cycle number Discharging to 100% DOD show significantly faster capacity fade and impedance rise. Losses come from interfacial impedance at the positive electrode.</td>
<td></td>
</tr>
<tr>
<td>Stamps et al. 2005</td>
<td>1.4</td>
<td>18650 Calendar C</td>
<td>25, 35, 45, 55</td>
<td>PF</td>
<td>Power fade model $Y(t; T,SOC) = A(T,SOC) - B(T,SOC) t^{3.2}$. A is rapid fade based on temperature, B is slower. Model accurate up to 40% DOD</td>
<td></td>
</tr>
<tr>
<td>Wright et al. 2003</td>
<td>0.8 1</td>
<td>18650 25Wh Power assist</td>
<td>25, 45</td>
<td>PF</td>
<td>Linear power fade for 0.8 cell at 25°C. Power fade rate at rate of $t^{0.5}$ at 45°C. Power fade rate increases with temperature.</td>
<td></td>
</tr>
</tbody>
</table>

HPPC - Hybrid pulse power characterization, EIS - Electrochemical impedance spectroscopy, XANES - X-ray absorption near-edge analysis, PF - power fade, CF - capacity fade, Z - impedance, SEI - solid electrolyte interface, TTF - Time to failure, OCP - open circuit potential of electrodes, OCEOV - open-circuit end of voltage.

2.4.3 Calendar Aging

Calendar aging refers to a cell’s loss of capacity while it is in storage. Spotnitz (2003) reports that the capacity fade during calendar aging decays at an Arrhenius, *i.e.* exponential, rate as shown in Figure 2.20. Abraham et al. (2004) inserted a reference
electrode into 18650 class lithium-ion cells to study the effects of calendar aging. Using accelerated aging by storing the cells at 55 °C, Abraham et al. (2004) determined that the positive electrode was the main cause of impedance increase during the first month of electrode, including the growth of the solid electrolyte interface (SEI) layer. The study also showed a definite shift in the voltage of the negative electrode from a negative value to a positive value.

Ploehn et al. (2004) and Thomas et al. (2003) also present calendar aging results using elevated temperatures. Ploehn’s results showed the capacity fading at a rate of \( t^{1/2} \). Ploehn also reported that the SEI layer grew at a rate of \( t^{1/2} \). Thomas et al. presents a power fade model of

\[
PF(t, T, SOC) = A(T, SOC) - B(T, SOC)t^{3/2}, \tag{2.79}
\]

where \( A(T, SOC) \) is a rapid fade rate and \( B(T, SOC) \) is a much slower rate. This power fade model was used to predict the mean cell lifetime where end of life was defined as the time to reach 23% power fade. Calendar aging consists primarily of two fade rates, a fast rate near the beginning of the cell’s life and a slower rate that continues through the cell’s life. The next section discusses cell aging due to cycling.

2.4.4 Cycle Aging

Spotnitz (2003) reports that capacity loss occurs faster when the cell is being cycled than when it is being stored. Capacity fade due to cycling changes at four different rates are shown in Figure 2.20. The first fade rate, designated A in the figure, is a fast rate that does not last very long. Next, are two slower but different rates, given as B and C in Figure 2.20. Finally, there is the fade rate D, which is extremely fast near the end of the cell’s life. The different fade rates are due to changes within different areas of
Figure 2.19: Cell capacity and cycle life versus maximum charging voltage (Bergveld et al., 2002).

Figure 2.20: General examples of cell aging (Spotniz, 2003).
the cell. These changes include the growth of the SEI layer at each electrode, as well as the loss of active materials due to side reactions.

The first fade rate in Figure 2.20 is due to the changes within the positive electrode. Dubarry et al. (2007) determined that early capacity fade was primarily due to a NiO insulating layer that formed on the positive electrode which resulted in impedance rise.

Abraham et al. (2007) also showed that the capacity loss coincides with the increase in polarization of the activation energy. The faster fade in later cell life is due to the conductive network breakdown caused by restructuring at the negative electrode, which is referred to as carbon retreat.

Kobayashi et al. (2007) studied the valence state of nickel and cobalt in the positive electrode and discovered that it did not significantly change, but that as the cell aged, the surface of the positive electrode showed a cubic phase due to a deficiency of lithium cations. A growth of Li$_2$Co$_3$ also occurred on the surface of the positive electrode, resulting in increased resistance.

Capacity fade is also dependent on how the cell is used. Ning et al. (2003) used discharging rates of 1C, 2C, and 3C and found that capacity fade was proportional to the discharge rate. Abraham et al. (2007) used two charging rates, C/1 and C/25. Cells cycled at the C/25 rate were found to have larger capacity fade than the cells cycled at 1C in the early weeks of cycling, which indicates that lithium intercalation is easier at the slower rate. However as cycling continued, the cells cycled at the higher rate aged faster. Analysis of the cell showed that the impedance rise at the positive electrode was the main cause of aging. Aging of the positive electrode was consistent with aging of the entire
cell. Analysis of the negative electrode showed some capacity fade, but this electrode still had sufficient capacity to accept lithium-ions from the positive electrode. Charging at the higher rate caused side reactions which resulted in restructuring of the graphite particles, insulating portions of the SEI and consuming lithium at the negative electrode. This restructuring increases cell impedance and reduces the ability of the electrode to deliver and accept lithium.

Shim et al. (2002) show that capacity fade is slightly faster as the charge rate is increased. Shim et al. (2003) cycled to 70% depth of discharge (DOD) and 100% DOD and showed that deeper cycling causes significant capacity loss.

Ning et al. (2006) determined that most of the cycleable lithium is lost during the CV phase of the cycle. This loss of cycleable lithium occurs because of the entrapment of the lithium within the SEI layer. Santhanagopalan et al. (2008) showed that the loss of available lithium at the negative electrode results in the very fast capacity fades at the beginning and end of cell life. The loss of cycleable lithium increases as the temperature of the cell increases, resulting in shorter cell life at higher temperatures.

2.4.5 Aging Models

Several of the aging studies discussed in the literature include aging models. These aging models focus on the growth of the SEI layer or the amount of lithium available at the electrodes. Both Ning et al. (2006) and Rong and Pedram (2006) gave a linear equation for the SEI film resistance of the form

\[ R_j|_N = \frac{\delta|_N}{\kappa}, \] (2.80)
where \( \delta \) is the thickness of the film in meters, \( \kappa \) is the conductivity of the film in Siemens/m, and \( N \) is the cycle number.

Ploehn et al. (2004) gave a similar model. This model defined the loss of cycleable lithium proportional to the thickness of the SEI layer. Ploehn et al. went on to define the capacity loss as

\[
Q_{\text{loss}} = k_{\text{loss}} \sqrt{k_{\text{Arr}} t},
\]

(2.81)

where \( k_{\text{loss}} \) is a constant based on the concentration of lithium at the SEI layer and the physics of the cell, and \( k_{\text{Arr}} \) is the Arrhenius constant discussed in Section 2.1.2.

Other models presented in the literature (Spotnitz, 2003) (Santhanagopalan et al. 2008) also focus on the internal physics of the cell. They either address the thickness of the SEI layer or the concentration of lithium at the surface of the electrodes. While these internal models are useful tools for cell designers, they cannot be easily implemented in a controller to be used in a BMS for determining remaining cell capacity.

2.4.6 Cell Aging Summary

This section discussed studies on cell aging and capacity fade that have been presented in the literature. Cell aging is a complex mechanism. Charging the cell to higher maximum charge voltages increases the cell capacity but has a drastic effect on the cell life. Likewise, discharging the cell at higher currents and to deeper depths of discharge also results in much shorter cell life. Aging models presented in the literature describe the model reactions at the surface of the electrodes. While these models are useful for cell designers, they are not practical for use in battery management systems.
The previous sections of this chapter discussed the physical properties of the electrochemical cell and how it ages. The last two sections of this chapter discuss filtering and identification methods that can be used to identify the cell models discussed.

2.5 Moving Average Filter

Data collected from real systems usually requires filtering before it can be analyzed. The moving average filter is good for retaining a step response while reducing random noise (Smith, 2003). The symmetric moving average filter is

\[
\sum_{m=-M}^{M} m n x \frac{1}{2M + 1} \sum_{m=-M}^{M} x[n+m] , \quad (2.82)
\]

where \(M\) is the number of points to be used on each side of the filter, \(x[n]\) is the input data point to be averaged around, and \(y[n]\) is the filtered output value. Since Equation 2.82 is symmetric about the point, \(n\), the first \(M\) points are filtered using

\[
y[n] = \frac{1}{M + n} \sum_{m=0}^{M} x[n+m] , \quad (2.83)
\]

and the last \(N - M\) points are filtered using

\[
y[n] = \frac{1}{M + n} \sum_{m=0}^{M} x[N - m - n] . \quad (2.84)
\]

To validate the moving average filter, a test waveform was created. Noise was added using Matlab’s rand() function, then the moving average filter was applied to the noisy data. The original waveform together with the noisy waveform are shown in the top plot of Figure 2.21, and the filtered waveform together with the original waveform are shown in the bottom plot of Figure 2.21. The filtered waveform contains very little noise and has only a slight widening of the step increase. The value of \(M\) used for filtering in this
paper was chosen to be 20 because it empirically gave smoother waveforms without losing too much of the dynamics.

![Sample waveform to illustrate the moving average filter.]

**Figure 2.21**: Sample waveform to illustrate the moving average filter.

2.6 Batch Least Squares Identification

Each of the cell models presented in the previous sections had parameters that were dependent on the type of cell used. As mentioned previously, these parameters can be identified using batch least squares. System identification consists of four steps (Landau & Zito, 2006): data acquisition, model selection, parameter estimation, and model validation.

The first step, data acquisition, is to excite the system with a sequence of known inputs $u_k$ and to record the outputs $y_k$. The excitation input must be sufficient to excite all the dynamics of the system that are to be modeled and must be persistently exciting.
After collecting the data, a model for the system must be chosen. For illustration, assume a system model of the form

\[ y_{k+1} = af(y_k) + bg(u_k), \]  

(2.85)

where \( u_k \) is the input to the system, \( y_k \) is the output at the time \( k \), \( a \) and \( b \) are the parameters to be identified, and \( f(y_k) \) and \( g(u_k) \) are functions that describe the response of the system.

The general idea of system identification is to select the parameters of the model that give the best fit. "Best fit" is usually interpreted as minimizing the error between the real system response and the response of the model. One of the most common methods used to determine unknown model coefficients while minimizing the error between the model and the measured data is batch least squares (Landau & Zito, 2006). If \( y_k \) is the output of the true system and \( \hat{y}_k \) is the output of the model to the same input at time \( k \), then the error \( e_k \) is defined as

\[ e_k = y_k - \hat{y}_k. \]  

(2.86)

Letting

\[ \Theta = [a \ b]^T \]  

(2.87)

and

\[ \Phi = \begin{bmatrix} f(y_1) & g(u_1) \\ f(y_2) & g(u_2) \\ \vdots & \vdots \\ f(y_m) & g(u_m) \end{bmatrix}, \]  

(2.88)

where \( m \) is the number of data samples collected, and

\[ Y = \begin{bmatrix} y_2 & y_3 & \cdots & y_{m+1} \end{bmatrix}^T, \]  

(2.89)
the system response to the input \( u_k \) can be written as

\[
Y = \Phi \Theta + \varepsilon. \tag{2.90}
\]

Given

\[
J = \frac{1}{2} \varepsilon^T \varepsilon = \frac{1}{2} \sum_{i=1}^{k} \varepsilon_i^2. \tag{2.91}
\]

as the performance measure representing the sum of the squared errors, the vector \( \Theta \) that results in the smallest averaged squared error is found by letting

\[
\varepsilon = Y - \Phi \Theta \tag{2.92}
\]

and solving

\[
\frac{dJ}{d\Theta} = 0. \tag{2.93}
\]

Substituting Equations 2.91 and 2.92 into Equation 2.93 gives

\[
\frac{dJ}{d\Theta} = \frac{d}{d\Theta} \left( \frac{1}{2} (Y - \Phi \Theta)^T (Y - \Phi \Theta) \right) = 0. \tag{2.94}
\]

Evaluating Equation 2.94 gives

\[
\frac{dJ}{d\Theta} = \frac{d}{d\Theta} \left( \frac{1}{2} (Y^T Y - \Phi^T \Phi \Theta - \Theta^T \Phi^T Y - \Theta^T \Phi \Theta \Theta) \right) = 0. \tag{2.95}
\]

Evaluating the derivative above yields

\[
\frac{1}{2} \left( 0 - \Phi^T Y - \Phi^T Y - 2 \Phi^T \Phi \Theta \right) = 0. \tag{2.96}
\]

Simplifying Equation 2.96 gives

\[
(\Phi^T \Phi) \Theta = \Phi^T Y. \tag{2.97}
\]

Finally, solving Equation 2.97 for \( \Theta \) gives

\[
\Theta = (\Phi^T \Phi)^{-1} \Phi^T Y. \tag{2.98}
\]
For the solution in Equation 2.98 to be a unique solution, the rank of the \( m \times n \) matrix \( \Phi \) must be \( n \) (Recktenwald, 2000). The identified model can be validated by comparing the model response to the response of the actual system to the same input using Equation 2.86.

The discussion above described the method of batch least squares for system identification. There are many other methods of identification, including total least squares, recursive least squares, and recursive maximum likelihood (Landau & Zito, 2006) (Recktenwald, 2000). Batch least squares was chosen for its robustness and ease of use for offline identification.

2.7 Chapter Summary

This chapter began by introducing the physical principles of an electrochemical cell. Next, several simple models for a cell were presented. These models increased in complexity as various physical aspects of the electrochemical cell were added to the electrical model. Next, a basic dynamic model of the electrochemical cell as described by Hartley and Jannette was presented, followed by a similar dynamic model developed by Plett. Then a temperature dependent model was presented.

After reviewing the models documented in the literature, three dynamic models were presented for use in this paper. These models are based on the first-principles model of Hartley and Jannette, but use a linear diffusion term in place of the logarithmic diffusion term. The other two models add both first-order and second-order temperature correction terms to the base model.

The next section presented aging results from the literature. These aging studies concentrate on the internal changes of the cell. The results of the studies indicate that
aging is primarily caused by the growth of the SEI layer and loss of active lithium in the electrodes.

The last two sections of this chapter introduced the filtering and identification methods selected for use in this paper. The next chapter will present the cycling method used and show general results from cycled cells.
CHAPTER III
THE CYCLING PROCESS

To perform the aging study, nine lithium-ion cells were cycled. These cells were commercial 18650 cells with a rated capacity of 2.4 Ah. The nine cells were labeled with the letters A through I for identification purposes. The cells were separated into three groups. Each group had a different maximum charging voltage (MCV). Table 3.1 lists each of the nine cells and their respective MCV.

This chapter discusses the method used for cycling the cells, collecting the data, and processing the collected data for use in MATLAB. This chapter also presents some sample cycles and discusses some general observations concerning how these cells changed with age.

3.1 Cycling Circuit

To cycle the cells and collect the data, a simple control circuit was built for each cell. The circuit diagram is shown in Figure 3.1. Table 3.2 lists the values of the components in the circuit. At the center of the circuit is the PIC 16F876A microcontroller (Microchip Technology Inc., 2003). This microcontroller contains five analog-to-digital input circuits for data collection, two pulse width modulator (PWM) outputs for analog control, as well as 16 other digital I/O ports. The PIC was driven by a 4 MHz crystal. Accurate voltage measurements were obtained using the 2.5 V zener
reference diode, $Z_1$ (Texas Instruments Inc., 2003). Since the voltage of the cell and the voltages from the current sensing resistors were larger than the 2.5 V reference, voltage divider circuits were used to drop the voltage down below the reference voltage. After the voltages were measured, the digital value was multiplied by the voltage divider value to obtain the real voltage value.

Charging of the cell was done through the transistor $Q_1$. During the charging phases of the cycle, the PIC sent out a PWM pulse to the RC circuit, consisting of $R_{10}$ and $C_3$. The output of the PWM would in turn provide a DC voltage reference to the base of $Q_2$ through the voltage divider, consisting of resistors $R_3$ and $R_4$. The voltage reference

<table>
<thead>
<tr>
<th>Cell</th>
<th>Charge Voltage (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.2</td>
</tr>
<tr>
<td>B</td>
<td>4.2</td>
</tr>
<tr>
<td>C</td>
<td>4.2</td>
</tr>
<tr>
<td>D</td>
<td>4.3</td>
</tr>
<tr>
<td>E</td>
<td>4.3</td>
</tr>
<tr>
<td>F</td>
<td>4.3</td>
</tr>
<tr>
<td>G</td>
<td>4.4</td>
</tr>
<tr>
<td>H</td>
<td>4.4</td>
</tr>
<tr>
<td>I</td>
<td>4.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuit Element</th>
<th>Circuit Value</th>
<th>Circuit Element</th>
<th>Circuit Value</th>
<th>Circuit Element</th>
<th>Circuit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>22 pF</td>
<td>$R_7$</td>
<td>1 kΩ</td>
<td>$R_{17}$</td>
<td>220 Ω</td>
</tr>
<tr>
<td>$C_2$</td>
<td>22 pF</td>
<td>$R_8$</td>
<td>1 kΩ</td>
<td>$R_{18}$</td>
<td>2 kΩ</td>
</tr>
<tr>
<td>$C_3$</td>
<td>10 μF</td>
<td>$R_9$</td>
<td>1 kΩ</td>
<td>$R_{19}$</td>
<td>1 kΩ</td>
</tr>
<tr>
<td>$C_4$</td>
<td>10 μF</td>
<td>$R_{10}$</td>
<td>1 kΩ</td>
<td>$R_{20}$</td>
<td>1 kΩ</td>
</tr>
<tr>
<td>$R_1$</td>
<td>0.5 Ω</td>
<td>$R_{11}$</td>
<td>3 kΩ</td>
<td>$R_{21}$</td>
<td>0.2 kΩ</td>
</tr>
<tr>
<td>$R_2$</td>
<td>1 Ω</td>
<td>$R_{12}$</td>
<td>1 kΩ</td>
<td>$Q_1$</td>
<td>TIP147</td>
</tr>
<tr>
<td>$R_3$</td>
<td>2 kΩ</td>
<td>$R_{13}$</td>
<td>1 kΩ</td>
<td>$Q_2$</td>
<td>TIP31</td>
</tr>
<tr>
<td>$R_4$</td>
<td>1 kΩ</td>
<td>$R_{14}$</td>
<td>220 Ω</td>
<td>$Q_3$</td>
<td>TIP142</td>
</tr>
<tr>
<td>$R_5$</td>
<td>3 kΩ</td>
<td>$R_{15}$</td>
<td>220 Ω</td>
<td>$U_1$</td>
<td>LM61</td>
</tr>
<tr>
<td>$R_6$</td>
<td>2 kΩ</td>
<td>$R_{16}$</td>
<td>220 Ω</td>
<td>$Z_1$</td>
<td>LM336</td>
</tr>
</tbody>
</table>
Figure 3.1: Control circuit used for cycling cells.
controlled the current through the collector of $Q_2$, thereby controlling the current within $Q_1$. The charging current was adjusted by changing the duty cycle of the PWM. The charging current was monitored using the voltage through the one ohm resistor $R_2$.

The discharge current was controlled through the transistor $Q_3$. Current through $Q_3$ was controlled by sending a PWM signal to the RC combination $R_{20}$ and $C_4$. The discharge current was monitored using the voltage across the resistor $R_{21}$.

The cell temperature was measured using an LM61 (National Semiconductor Corporation, 2005) temperature sensor, shown in the circuit as $U_1$. The voltage, $V_T$, of the temperature sensor is given as

$$V_T = 600mV + \frac{10mV/°C}{T} ,$$

where $T$ is the temperature given in °C. The maximum allowable temperature for the cell during charging is 45 °C (Sanyo, n.d.), which corresponds to 1.05 V from the sensor. The maximum allowable temperature during discharge is 60 °C or 1.2 V from the sensor.

Four LEDs were included to give a visible indication of the present state of the circuit. There were two red LEDs, of which one had a square shape. This square LED was toggled on then off each loop through the program to indicate that the circuit was operating. The other LEDs were all round-shaped and these LEDs were either on, off, or toggled on and off each loop depending on the state of the circuit as described in Table 3.3.

Programming of the microcontroller was done using PICBASIC Pro™ (MicroEngineering Labs, 2005). The flowchart for the PICBASIC program is shown in Figure 3.2. A charge/discharge cycle consists of five phases: charge at a constant current (CC) until the maximum charging voltage is reached, charge at a constant voltage (CV),
Table 3.3: State of LEDs on circuit board.

<table>
<thead>
<tr>
<th>State</th>
<th>Red</th>
<th>Green</th>
<th>Yellow</th>
<th>Red Square</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial State</td>
<td>Toggle</td>
<td>Toggle</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
<tr>
<td>Cell Temperature Too Hot</td>
<td>Toggle</td>
<td>Toggle</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
<tr>
<td>Opposite Of Green</td>
<td>OFF</td>
<td>Toggle</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
<tr>
<td>Opposite Of Red</td>
<td>Toggle</td>
<td>OFF</td>
<td>Toggle</td>
<td></td>
</tr>
<tr>
<td>Communications Failure</td>
<td>-</td>
<td>-</td>
<td>ON</td>
<td>Toggle</td>
</tr>
<tr>
<td>Cell Voltage Too Low</td>
<td>-</td>
<td>-</td>
<td>Toggle</td>
<td>Toggle</td>
</tr>
<tr>
<td>Constant Current</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
<tr>
<td>Constant Voltage</td>
<td>OFF</td>
<td>Toggle</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
<tr>
<td>Rest after Charge</td>
<td>Toggle</td>
<td>ON</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
<tr>
<td>Discharge</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
<tr>
<td>Rest after Discharge</td>
<td>ON</td>
<td>Toggle</td>
<td>OFF</td>
<td>Toggle</td>
</tr>
</tbody>
</table>

Toggle - indicates that the LED was turned on then off once during a program loop.

rest after charging (CR), discharge (D), and rest after discharging (DR). At the beginning of each cycle, the cell is charged at a constant current of 1C or 2.4 A. Constant current charging continues until the cell voltage reaches the maximum charging voltage designated for that cell. Once the cell voltage reaches the MCV, the PIC controller slowly reduces the current, maintaining the cell at the MCV until the current drops below 120 mA (5% of the 1C-rate) according to standard charging procedure.

After the current drops below 120 mA, the cell enters the CR phase, where no current passes through the cell for 10 minutes. The cell is then discharged at a rate of 2C or 4.8 A. Discharging continues until the cell voltage drops to 2.75 V, at which time the cell enters the DR phase and rests for another 10 minutes. The rest phases allow the cell to reach equilibrium. Data collected during the rest phases will be used to determine the coefficients for the diffusion equation given in Equation 2.74.

The main portion of the PICBASIC program consisted of an infinite loop where the PIC read the analog inputs, checked the cell temperature, analyzed the state of the cell, calculated the output signals, sent the data to the PC, awaited an acknowledgement.
Figure 3.2: Flowchart of the Cycle.bas PICBASIC program.
Figure 3.2: Flowchart of the Cycle.bas PICBASIC program (continued).
back from the PC, then paused until the next iteration. The program loop executed once per second. Timing of the loop was measured using an oscilloscope and adjusted using a delay in the loop in the PIC. The actual timing of a normal loop varied between 999 and 1002 ms. This variation was attributed to the delays in the PC sending back the acknowledgement.
Since the PIC microcontroller can do only integer math, the program was designed to minimize math calculations. Voltages read from the A/D were left in integer form. Before connecting the cell, each control circuit was calibrated, and the integer equivalent to the set voltages, respective currents, and temperature were stored inside the PIC program for later use during cycling. Table 3.4 contains the list of the set voltages, currents, and temperature, along with their respective scaled integer value for each of the nine circuits.

The data string sent to the PC was also left in integer form. The individual values were separated by commas and were periodically monitored in real-time on the PC as the data was being logged. Figure 3.3 shows a single dataline from a log file. Data logging is discussed in the next section.

Table 3.4: List of voltage or current with corresponding integer values.

<table>
<thead>
<tr>
<th>Cell</th>
<th>Voltage Volts</th>
<th>Charge Current Integer</th>
<th>Discharge Current Integer</th>
<th>Temperature °C</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.2</td>
<td>1719</td>
<td>2.4</td>
<td>974</td>
<td>4.8</td>
</tr>
<tr>
<td>B</td>
<td>4.2</td>
<td>1719</td>
<td>2.4</td>
<td>1005</td>
<td>4.8</td>
</tr>
<tr>
<td>C</td>
<td>4.2</td>
<td>1719</td>
<td>2.4</td>
<td>998</td>
<td>4.8</td>
</tr>
<tr>
<td>D</td>
<td>4.3</td>
<td>1760</td>
<td>2.4</td>
<td>998</td>
<td>4.8</td>
</tr>
<tr>
<td>E</td>
<td>4.3</td>
<td>1765</td>
<td>2.4</td>
<td>1005</td>
<td>4.8</td>
</tr>
<tr>
<td>F</td>
<td>4.3</td>
<td>1770</td>
<td>2.4</td>
<td>1045</td>
<td>4.8</td>
</tr>
<tr>
<td>G</td>
<td>4.4</td>
<td>1815</td>
<td>2.4</td>
<td>1045</td>
<td>4.8</td>
</tr>
<tr>
<td>H</td>
<td>4.4</td>
<td>1824</td>
<td>2.4</td>
<td>1020</td>
<td>4.8</td>
</tr>
<tr>
<td>I</td>
<td>4.4</td>
<td>1814</td>
<td>2.4</td>
<td>1040</td>
<td>4.8</td>
</tr>
</tbody>
</table>
Dataline:
2007-08-01-15:00:00-
Ar, 1719, 298, 43373, 366, 1556, 11, 14, 492, A, 569, 0, 0, 121, 504, 7

The breakdown of the dataline is as follows:
the first line is the timestamp added by the PC,
Ar indicates cell A in rest after charging,
1719 is the maximum charging voltage reference as scaled integer,
298 is the cycle number,
43373 is a second counter,
366 is the cell temperature as scaled integer,
1556 is the cell voltage as scaled integer,
11 is the cell charging current as scaled integer,
14 is the cell discharging current,
492 is the 2.5 reference voltage as scaled integer,
A is the acknowledgement value sent back to the PIC and then resent to the PC,
569 is the number of seconds the cell has been in the current phase (rest phase),
0 is the current value of the duty cycle of the charge circuit PWM,
0 is the current value of the duty cycle of the discharge circuit PWM,
121 is the value in duty cycle adjustment which is four times the actual value sent to
the PWM,
504 is the actual integer value read in from the charging resistor current sensor, and
7 is the actual integer value read in from the discharge resistor current sensor.

Figure 3.3: Sample dataline from raw data file.

3.2 Data Storage

Once per second, data was sent to a PC for storage. Five PC’s were used for
logging the data. Each PC ran the Linux operating system. Actual logging was done
using the PERL script, Logger.pl. The flow chart for the logging script is shown in
Figure 3.4. A separate logger program was used for each cell.

The logger script had four main functions: receive the dataline sent from the PIC,
send an acknowledgement back to the PIC that the data was received, apply a time stamp
to the data collected, and store the data in a file. The acknowledgement sent back to the
PIC was toggled between an upper case ‘A’ and a lower case ‘a’. The acknowledgement
was only used to verify to the PIC that communication with the PC was working.
Figure 3.4: Flowchart of the Logger.pl PERL script.

Verification was done within the PIC program through the SERIN command. If no data was received within 300 milliseconds, the PIC went to the COMMFAIL subroutine, which started a counter and turned on the yellow LED. If communications failed five times in a row, the PIC went into the INIT state and cycling stopped.

For ease of data management, the logger script created a new data file each hour. Data files have the naming convention of Cpcname-YYYY-MM-DD-HH-PORTNAME.dat where C is the letter of the cell being logged; pcname is the name of the PC collecting the data; YYYY-MM-DD-HH is the year, month, day, and hour the data was collected; and PORTNAME was the serial port on the PC, either ttyS0 for the first port or ttyS1 for the second port. As mentioned in the previous section, the data received by the logging program was periodically monitored using the tail command provided with the Linux operating system.
3.3 Data Processing

The data needed to be processed before it could be loaded into MATLAB for identification and modeling. Data processing was done in three steps: interpolation of missing datalines and creation of daily files; scaling integer values into floating point values and combining all data for a cell into one large file; and then separating it into files that contained one complete cycle for easy analysis in MATLAB. Because of its easy string manipulation features, the PERL scripting language was selected to perform the data processing. The complete PERL programs are included in Appendix A.

Due to communication disturbances, there were a few datalines that contained invalid characters in them. Also, there were some gaps between datalines because the PIC program cycle ranged from 999 to 1002 ms or because there was a communication error. To verify the data, a PERL program called Createdaily.pl was created. The flowchart for this program is shown in Figure 3.5 and the full program is listed in Appendix A. This program executed offline and would read the hourly data files created by the logging program. If a dataline contained invalid characters, it was discarded, a warning message was printed to the screen, and the dataline was replaced by performing linear interpolation between the next valid line and the previous valid lines. If there were two or more seconds between the timestamps of the two datalines, linear interpolation was performed creating the datalines for each missing row between those two datalines. All corrected data was stored in a daily file named for the day the data was collected. Daily files were then manually verified to insure that they contained 86,400 datalines.

As mentioned previously, raw data sent from the PIC to the PC was formatted so that it was easy to view the information and to verify that the program was working correctly. Another PERL script, called Convertdata.pl, was created to read the
interpolated data and convert the bit-valued numbers from the PIC into floating point values representing the voltage, current and temperature of the cell. The flow chart for this program is shown in Figure 3.6. This data was then saved into one large data file per cell so that all cycles were continuous within this one file.

The continuous data file was then separated into individual files for each cycle using a third PERL script called Createcycles.pl (see flow chart in Figure 3.7). Separate cycles could then be loaded into MATLAB for viewing and identification. The next section discusses some sample cycles from each of the cells.

Figure 3.5: Flowchart of the Createdaily.pl PERL script.
Figure 3.6: Flowchart of the Convertdata.pl PERL script.

Figure 3.7: Flowchart of the Createcycles.pl PERL script.
3.4 Sample Cycles

Due to the large number of cycles collected for each cell, it is impractical to include them all in this paper, therefore one complete cycle from the beginning, middle, and end of each cell’s life will be presented for each of the nine cells. These 27 cycles are shown in Figures 3.8-3.34 and will be used for comparisons of the dynamic and aging models discussed in Chapters IV and V respectively. MATLAB program files for plotting and simulating cycle data are included in Appendix B.

The vertical dotted lines shown in Figures 3.8-3.34 indicate the start time of each phase of the cycle as discussed in Section 3.1. The horizontal dotted lines for the voltage plot indicate the maximum charging voltage of the cell and the end of discharge voltage of 2.75 V. The horizontal lines on the current plots represent the maximum charge current and maximum discharge currents, 2.4 A and -4.8 A respectively. The two horizontal dotted lines on the temperature plot are simply reference lines for ease of reading. The green dotted lines in all the plots show the actual measured values, while the solid blue lines are the filtered data using the moving average filter discussed in Section 2.5. For consistency, the vertical axes of all 27 plots are the same; however, due to the large time difference between cycle lengths, the horizontal axes of the cells change.

Figures 3.8-3.10 show cycles near the beginning, middle and end of life for cell A. As shown in the figures, more noise occurred during cycle 1 (Figure 3.8) than was present in the other cycles shown for cell A. In reality, cycle 230 (Figure 3.10) lasted longer than the cycle 1 shown in Figure 3.9. The lengths of the individual cycles were affected by noise. If the voltage of the cell was too high due to a noisy input reference, the cell would stop charging early and go into rest mode, which resulted in differences in cycle lengths.
The cycling circuit proved to be susceptible to noise and outside disturbances, such as the cooling fans that were used to cool the power transistors. Cell D, cycle 1, shown in Figure 3.17 has the most noise of all the samples. The second cycle of cell F was selected as a sample cycle due to a large disturbance that occurred in the first cycle of this cell. The position of the cooling fans was adjusted to reduce noise. Shielding was also added to the circuits to help reduce noise. The effect of the noise reduction is visible in the later cycles shown in the sample figures.

As indicated in Table 3.1, cells A through C all had a maximum charging voltage of 4.2 V, cells D through F had a MCV of 4.3 V, and cells G through I had a MCV of 4.4 V. As shown in the figures, the currents and voltages of the cells are consistent with one
Figure 3.9: Voltage, current, and temperature of cell A, cycle 230.

Figure 3.10: Voltage, current, and temperature of cell A, cycle 460.
Figure 3.11: Voltage, current, and temperature of cell B, cycle 1.

Figure 3.12: Voltage, current, and temperature of cell B, cycle 120.
Figure 3.13: Voltage, current, and temperature of cell B, cycle 240.

Figure 3.14: Voltage, current, and temperature of cell C, cycle 1.
Figure 3.15: Voltage, current, and temperature of cell C, cycle 225.

Figure 3.16: Voltage, current, and temperature of cell C, cycle 450.
Figure 3.17: Voltage, current, and temperature of cell D, cycle 1.

Figure 3.18: Voltage, current, and temperature of cell D, cycle 120.
Figure 3.19: Voltage, current, and temperature of cell D, cycle 250.

Figure 3.20: Voltage, current, and temperature of cell E, cycle 1.
Figure 3.21: Voltage, current, and temperature of cell E, cycle 125.

Figure 3.22: Voltage, current, and temperature of cell E, cycle 250.
Figure 3.23: Voltage, current, and temperature of cell F, cycle 2.

Figure 3.24: Voltage, current, and temperature of cell F, cycle 145.
Figure 3.25: Voltage, current, and temperature of cell F, cycle 290.

Figure 3.26: Voltage, current, and temperature of cell G, cycle 1.
Figure 3.27: Voltage, current, and temperature of cell G, cycle 100.

Figure 3.28: Voltage, current, and temperature of cell G, cycle 200.
Figure 3.29: Voltage, current, and temperature of cell H, cycle 1.

Figure 3.30: Voltage, current, and temperature of cell H, cycle 100.
Figure 3.31: Voltage, current, and temperature of cell H, cycle 200.

Figure 3.32: Voltage, current, and temperature of cell I, cycle 1.
Figure 3.33: Voltage, current, and temperature of cell I, cycle 95.

Figure 3.34: Voltage, current, and temperature of cell I, cycle 180.
another. The temperature plots are also consistent, showing a slight temperature rise during the constant current phase. This temperature rise is due to the heating caused by the internal resistance, $R_\Omega$. As the current is decreased, the temperature also decreases. During the rest period after charging, the temperature is roughly constant as expected. During the discharge, there is a sharp decrease in temperature due to the endothermic nature of the chemical reactions. This decrease in temperature causes the cell voltage to drop due to the Arrhenius effect discussed in Section 2.1.2. As the cell continues discharging, its temperature steadily rises due to the internal resistance. This rise in temperature causes the cell voltage to also rise slightly. After the slight rise, the cell voltage drops, due to the loss of charge in the cell, until the voltage reaches the 2.75 V limit set by the control circuit. As expected, the temperature effects increase when the maximum charging voltage is increased.

This section presented a selection of cycles over the life of each of the nine cells that were cycled. The next section discusses the behavior of the cells over their entire life span.

3.5 Cell Behavior

In the previous section, cycles from the beginning, middle and end of life for each cell were presented. This section analyzes the performance of the cells over their entire life.

3.5.1 Capacity

The parameter of interest to most cell users is the charge capacity of the cell over its life span. Figures 3.35-3.37 show the charge stored in ampere-hours by each of the
nine cells. Since the initial capacity of the cell is not actually known, this study assumed that 10% of the rated cell capacity was left in the cell after every discharge; therefore, the capacity shown in these plots is the charge stored in the cell during the charging phases of the cycle plus the 10% left from the previous cycle. The charge removed from the cell during discharge was also calculated and found to be roughly the same as the charge stored in the cell during charging, in every case considered.

As mentioned earlier, the rated capacity of the cells used in this paper are 2.4 Ah. The plots shown in Figures 3.35-3.37 include four red horizontal lines labeled 20%, 50%, 80%, and 100%. These four lines correspond to the percentage of the rated capacity of the cell.

As these plots indicate, initially the capacity remains fairly constant for the first 100 cycles, it then begins to decrease at a linear rate. After the capacity decreases to approximately 80% of the rated value, there appears to be a slight increase in the slope of the line. As the cell capacity nears the line indicating 50%, there is a sharp drop to approximately 20% of the rated capacity. These plots are consistent with the general cycling plot presented by Spotnitz (2003) as shown in Figure 2.21. Due to the sharp drop at 50% capacity, this paper defines the end of life of the cell to be when the capacity of the cell reaches 50% of the rated cell value. Table 3.5 shows the list of cells and the cycle number where they reach both 80% and 50% of their rated capacity.

For the cells with the MCV of 4.2 V, after the cells dropped to 20% capacity, some cycles showed that the capacity actually jumped back to around 50% then immediately dropped back down to 20%. This pulsating action also occurred to the cells charged to the higher MCVs, but it lasted for a much shorter length of time.
Cell B died much faster than the other two cells with an MCV of 4.2 V. As the plot shows, however, after pulsating between capacities of 20% to 50% for over a hundred cycles, cell B produced many cycles above the 50% capacity line with some cycles above 80% capacity. This fast capacity fade is probably due to the early formation of an electronic conductive path presented by Dubarry et al. (2007) and discussed in Section 2.6. As the cell was continuously cycled, this path later broke down, resulting in the increase in capacity in later life. Chapter V presents a more detailed analysis of the capacity fade of the experimental cells.

Figure 3.35: Capacity plots of cells with maximum charging voltage of 4.2 V.
Figure 3.36: Capacity plots of cells with maximum charging voltage of 4.3 V.

Figure 3.37: Capacity plots of cells with maximum charging voltage of 4.4 V.
Table 3.5: Cycle at which the cell reached minimum capacity.

<table>
<thead>
<tr>
<th>Cell</th>
<th>Cycle Number 80% Capacity</th>
<th>Cycle Number 50% Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>331</td>
<td>536</td>
</tr>
<tr>
<td>B</td>
<td>99</td>
<td>343</td>
</tr>
<tr>
<td>C</td>
<td>245</td>
<td>456</td>
</tr>
<tr>
<td>D</td>
<td>179</td>
<td>258</td>
</tr>
<tr>
<td>E</td>
<td>150</td>
<td>256</td>
</tr>
<tr>
<td>F</td>
<td>170</td>
<td>314</td>
</tr>
<tr>
<td>G</td>
<td>156</td>
<td>234</td>
</tr>
<tr>
<td>H</td>
<td>137</td>
<td>234</td>
</tr>
<tr>
<td>I</td>
<td>126</td>
<td>216</td>
</tr>
</tbody>
</table>

3.5.2 Open-Circuit Voltage: Charged Cell

Figure 3.38 shows the filtered open-circuit voltage of the cells after they had been completely charged and allowed to rest. As the plots show, the open-circuit voltage of the charged cells slightly decreases as the cell ages. This is consistent with results presented in the literature discussed in Section 2.4. As expected, when the MCV of the cell is increased, its open-circuit voltage after charge also increases. As the cell ages, the open-circuit voltage after charge decreases at a faster rate.

The decrease in the open-circuit voltage of the cells with the same MCV is consistent between cells. There is a sharp decrease in open-circuit voltage of cell B after 500 cycles. This sharp decrease follows the large spike in capacity of cell B at the same point in the cell’s age as discussed in the previous section.

As the maximum charge voltage of the cells is increased, the open-circuit voltage of the charged cells decreases at a faster rate. This faster decrease in open-circuit voltage is consistent with the faster rate of capacity fade of the cell.
3.5.3 Open-Circuit Voltage: Discharged Cell

As mentioned above and in Chapter V, when the cell voltage is considered in context with cell aging, it is reported as decreasing with cell age. However, when the open-circuit voltage of a fully discharged cell is analyzed, this voltage actually increases. Figures 3.39-3.41 show the unfiltered open-circuit voltage for each of the nine cells at the end of the discharge rest cycle. As these plots show, the open-circuit voltage after discharge remains constant during the early stages of the cell’s life at approximately 3.8 V. During the middle period of the cell’s life, the open-circuit voltage of the discharged cell increases. At the end of the cell’s life, there is a sharp increase in this open-circuit voltage. After the open-circuit voltage of the discharged cell reached a certain value, the voltage fluctuated between the two values before remaining constant at the larger value.
Comparing the fully discharged open-circuit voltage plots of Figures 3.39-3.41 with the capacity (charge) plots shown in Figures 3.35-3.37, they appear to be correlated. To determine if there is any direct correlation between the capacity and discharged open-circuit voltage plots, the plot of the capacity must be scaled and shifted to fit on the same axis as the voltage. Scaling and shifting of the capacity is done using

\[ V_{\text{capacity}} = V_{\text{ref}} - \frac{Q}{k_{\text{scale}}} \]  

(3.2)

where \( V_{\text{capacity}} \) is a voltage representation of the cell capacity, \( V_{\text{ref}} \) is a reference voltage chosen empirically to be 4.0 V, \( Q \) is the capacity of the cell, and \( k_{\text{scale}} \) is the scaling factor chosen empirically to be 8 V/Ah. The value \( V_{\text{capacity}} \) from Equation 3.2 is plotted on the same graph as the fully discharged open-circuit voltage for each of the nine cells in Figures 3.42-3.44. To fully show the close relationship between \( V_{\text{capacity}} \) and the discharged open-circuit voltage, these plots use the unfiltered data. As these plots clearly illustrate, the open-circuit voltage of a fully discharged cell closely follows the capacity of the cell.

3.5.4 Cell Temperature

No direct temperature control was applied to the cells during cycling. The cells were cycled in an air conditioned room where the temperature ranged between 20 and 27 °C. Figures 3.45-3.47 show the minimum, average and maximum temperature of each of the nine cells over their life span. As these figures clearly indicate, the maximum temperatures of the cells ranged between 35 and 45 °C, well below the maximum discharge temperature of 60 °C discussed in Section 3.1.
Figure 3.39: Open-circuit voltage of fully discharged cells with MCV of 4.2 V.

Figure 3.40: Open-circuit voltage of fully discharged cells with MCV of 4.3 V.
Figure 3.41: Open-circuit voltage of fully discharged cells with MCV of 4.4 V.

Figure 3.42: Open-circuit voltage of fully discharged cells along with scaled capacity for cells with MCV of 4.2 V.
Figure 3.43: Open-circuit voltage of fully discharged cells along with scaled capacity for cells with MCV of 4.3 V.

Figure 3.44: Open-circuit voltage of fully discharged cells along with scaled capacity for cells with MCV of 4.4 V.
Figure 3.45: Filtered maximum, minimum, and average temperatures for cells with MCV of 4.2 V.

Figure 3.46: Filtered maximum, minimum, and average temperatures for cells with MCV of 4.3 V.
3.6 Chapter Summary

This chapter presented the method used for cycling the cells and also presented some sample cycles and general results. Overall, the cycling performed as expected and the cell capacity also faded as expected. For the cells charged to 4.2 MCV, cells A and C performed similarly whereas cell B’s capacity faded abnormally quickly. The cells with MCVs of 4.3 V and 4.4 V performed very consistently and they had very similar capacity fades.

As discussed in Section 3.4, the cycling circuit was susceptible to outside noise and disturbances. One major cause of noise was the amplification due to the use of voltage divider resistors that were required to decrease the voltage below the reference voltage and subsequent multiplication of the voltage value inside the PIC. This external
noise could have been reduced by using a larger voltage reference such as connecting two of the LM336 reference diodes in series which would remove the need for the voltage dividers. Another way the outside noise could have been reduced is to add active filters between the sensors and the A/D inputs. The other major cause of noise and disturbances was the close proximity of the cycling circuit to the cooling fans and air conditioner required to keep the room within the operating temperature. This noise was reduced by adjusting the location of the cycling circuits and increased shielding around wires. These enhancements resulted in the reduced noise during later cycles as discussed in Section 3.4. The next chapter presents the identification of the dynamic models that were chosen in Section 2.3.4.
Chapter II discussed how the parameters of a dynamic model could be identified using batch least squares. This chapter presents the process of parameter identification for the dynamic models chosen in Section 2.3.4. First, the effect of the PWM control circuit is analyzed and identified. Next, the common impedance characteristics of the cell are identified. Finally, the remaining coefficients in the nonlinear voltage equations are determined. Identification for the dynamic models is performed on each cycle for all nine cells. After identification is complete, sample plots are presented and general results of the identified parameters are discussed.

4.1 Current Response

In Section 2.3.1, the determination of the primary parameters $R_\Omega$, $R_d$, and $C_d$ was discussed. The value of $R_\Omega$ can be determined through the change in the instantaneous voltage due to a step response, while the diffusion parameters can be identified from the RC response of the voltage of the cell due to a step change in the cell current. The most consistent change in current occurs at the end of the discharge cycle; therefore, the parameters $R_\Omega$, $R_d$, and $C_d$ were determined using the measured data from this time. Due to the nature of the PWM driving circuit, the change in current is not an instantaneous step change, but an exponentially decaying waveform as shown in Figure 4.1. The
equation for the current waveform in the time domain is

\[ i(t) = I_0 e^{-t/\tau_{PWM}}, \]  \hspace{1cm} (4.1)

where \( \tau_{PWM} = R_{PWM} C_{PWM} \), is the RC time constant of the PWM circuit, and \( I_0 \) is the initial current. Taking the logarithm of both sides of Equation 4.1 gives the linear equation

\[ \ln|i(t)| = \ln|I_0| - \frac{t}{\tau_{PWM}}. \]  \hspace{1cm} (4.2)

The time constant, \( \tau_{PWM} \), is roughly the same for all cycling circuits and can be determined using batch least squares (Recktenwald, 2000). Using this method, the time constant for the current decay was calculated for the cycles of all cells and then averaged. The average value of the time constant for the current decay waveform was determined to be 7.6 s. The dashed red line in Figure 4.1 shows the simulated current decay using the averaged value of 7.6 s for the time constant.

![Figure 4.1: Current decay after the discontinuation of discharge of a lithium-ion cell.](image)
4.2 Resistance and Diffusion

Once the time constant of the current decay was determined, the values of the parameters could be determined. Assuming the cell impedance matches the impedance circuit shown in Figure 2.13, the equation for the impedance is

\[ Z(s) = R_\Omega + \frac{1/C_d}{s + 1/R_d C_d}. \]  

(4.3)

From Ohm’s law, the voltage in the Laplace domain is written as

\[ E(s) = I(s)Z(s). \]  

(4.4)

The Laplace transform of the current in Equation 4.1 is

\[ I(s) = \frac{I_{d0}}{s + 1/\tau_{PWM}}, \]  

(4.5)

which gives the cell voltage to be

\[ E(s) = \left( \frac{I_{d0}}{s + 1/\tau_{PWM}} \right) \left( R_\Omega + \frac{1/C_d}{s + 1/R_d C_d} \right). \]  

(4.6)

Taking the inverse Laplace transform of Equation 4.6 yields

\[ E(t) = R_\Omega I_{d0} e^{-t/\tau_{PWM}} + R_d \frac{I_{d0}}{R_d C_d \tau_{PWM}} \left( e^{-t/(R_d C_d)} - e^{-t/\tau_{PWM}} \right). \]  

(4.7)

To determine the coefficients \( R_\Omega \) and \( R_d \), batch least squares was used, while searching over the \( R_d C_d \) products. A for loop was used to iterate through a range of times for the \( R_d C_d \) time constant to find the value of \( R_d C_d \) that gave the minimum \( V_{rms} \) error of the batch least squares. This procedure was used to determine the diffusion time constant \( R_d C_d \) and the coefficients \( R_\Omega \) and \( R_d \) for every cycle of each cell. The diffusion time constant remained relatively constant for all cells at an average value of 30.97 s.
Figure 4.2 shows the voltage recovery after discontinuation of the discharge phase of a typical cycle. The solid blue line shows the actual voltage. The dashed red line shows the simulated value after identification using Equation 4.7 given above.

![Figure 4.2: Voltage recovery after the discontinuation of discharge of a lithium-ion cell.](image)

Plots of the parameters \( R_\Omega, R_d, \) and \( C_d \) over all cycles are shown in Figures 4.3, 4.4, and 4.5 respectively. Dotted lines are the actual parameters, while the solid lines are the moving averages. The top plot in each figure consists of the three cells that had a maximum voltage charge of 4.2 V. The middle plot shows the three cells charged to 4.3 V. The bottom plot shows the three cells charged to 4.4 V.

As expected, the plots for the resistance value \( R_\Omega \) show a linear increase in resistance as the cells age (refer to Figure 4.3). This linear increase is in agreement with the aging results that were discussed in Section 2.4. The slope of this linear increase in \( R_\Omega \) resistance increases as the MCV is increased. Identification was performed on the data from the plots of Figure 4.3 using the linear equation

\[
R_\Omega = k_{R\Omega 1} + k_{R\Omega 2} \text{cycle},
\]

where \( k_{R\Omega 1} \) and \( k_{R\Omega 2} \) are the constants to be identified, and \( \text{cycle} \) is the cycle number of the cell. The actual values of \( k_{R\Omega 1} \) and \( k_{R\Omega 2} \) obtained by batch least squares identification, as
Figure 4.3: Identified values for cell resistance, $R_\Omega$.

Figure 4.4: Identified values for diffusion resistance, $R_d$. 
well as the error, are given in Table 4.1. Figure 4.6 shows plots of the linear $R_\Omega$
identified using Equation 4.8. The dashed lines in Figure 4.6 are the filtered values from
Figure 4.3 for comparison.

Figure 4.4 shows the identified values for the diffusion resistance $R_d$. The actual
values for $R_d$, as shown by the dotted lines, vary due to noise and the identification
process. However, the average value for the cells, as shown by the solid lines, reveals
that $R_d$ remains relatively constant over the life of the cell with only a slight decrease in
the resistance at the large voltages. This slight decrease is probably due to parasitic
reactions within the cell. For identification of the dynamic model, the value of $R_d$ is
considered to be constant at $0.03234 \ \Omega$. 

Figure 4.5: Identified values for diffusion capacitance, $C_d$. 

![Graph showing identified values for diffusion capacitance](image)
The capacitance term $C_d$ also remains relatively constant over the life of the cell.

The identified values for the parameter $C_d$ are shown in Figure 4.5. $C_d$ is also considered to be constant, having a value of 957.6 F.

### Table 4.1: List of identified parameters for linear approximation to $R_Ω$.  

<table>
<thead>
<tr>
<th>Cell</th>
<th>$k_{R_{11}}$ ($\Omega$)</th>
<th>$k_{R_{12}}$ ($\Omega$/cycle)</th>
<th>RMS Error (m$\Omega$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.161e-01</td>
<td>1.211e-04</td>
<td>15.95</td>
</tr>
<tr>
<td>B</td>
<td>1.272e-01</td>
<td>1.315e-04</td>
<td>5.58</td>
</tr>
<tr>
<td>C</td>
<td>1.202e-01</td>
<td>5.213e-05</td>
<td>6.53</td>
</tr>
<tr>
<td>D</td>
<td>1.096e-01</td>
<td>1.687e-04</td>
<td>9.30</td>
</tr>
<tr>
<td>E</td>
<td>1.432e-01</td>
<td>1.908e-04</td>
<td>8.30</td>
</tr>
<tr>
<td>F</td>
<td>1.240e-01</td>
<td>1.891e-04</td>
<td>8.83</td>
</tr>
<tr>
<td>G</td>
<td>1.253e-01</td>
<td>3.262e-04</td>
<td>14.00</td>
</tr>
<tr>
<td>H</td>
<td>1.221e-01</td>
<td>2.968e-04</td>
<td>11.27</td>
</tr>
<tr>
<td>I</td>
<td>1.256e-01</td>
<td>3.045e-04</td>
<td>8.16</td>
</tr>
</tbody>
</table>

**Figure 4.6:** Linear approximation of identified values of $R_Ω$, with filtered values of $R_Ω$ from Figure 4.3 shown as dashed lines.
4.3 Charge Vectors

The next step in the identification process was to create the charge vectors $q_s(t)$ and $q_d(t)$ for the cycle. At the end of a cycle, the cell is considered fully discharged. However, there is still some charge remaining in the cell, although the exact amount is not known. Therefore, for identification of the unknown cell parameters, it has been assumed that there is 10% of the rated capacity or 0.24 Ah left in the cell at the end of a cycle. Using an initial value of 0.24 Ah or 864 Aseconds, the vector $q_s(t)$ can be determined from Equation 2.73. Figure 4.7 shows the stored charge for cell D, cycle 120.

As the plot shows, the stored charge starts at the assumed value of 864 Aseconds. During the charging phase, the stored charge increases until it reaches a maximum value of approximately 7,390 Aseconds. Since $R_{sd}$ is assumed to be infinite, the stored charge remains constant during the rest phase. During the discharge phase, the stored charge decreases at a constant rate with the stored charge ending at a value near its starting value.

Since the cell was at rest at the beginning of each cycle, the initial value for the diffused charge $q_d(t)$ is zero. Using Equation 2.74 and the constant values of $R_d$ and $C_d$
discussed in the previous section, the diffused charge was calculated for each cycle. A
plot of the diffused charge is shown in Figure 4.8. As expected, the diffused charge
followed the current of the cell as shown in Figure 3.18. During the charging phase, the
charge within the cell is positive, decreasing as the current decreases. During the rest
periods, the diffused charge returns to zero, and during the discharge phase, the charge is
diffused in the opposite direction.

4.4 Dynamic Model Identification

Once the vectors for $q_d(t)$ and $q_s(t)$ have been created, identification of the
unknown coefficients of the voltage equations for the dynamic models was performed.
Two voltage equations were presented for the dynamic model in Section 2.3.5. These
models will be designated as DMT$^0$ for Equation 2.75 which contains no temperature
correction, DMT$^1$ for Equation 2.76 which contains the first order temperature correction
terms, and DMT$^2$ for Equation 2.77 which contains the second order temperature
correction terms. The equation for the cell voltage of model DMT$^0$, was given in
Equation 2.75 and is repeated here as
\[ E(t) = k_{E0T0} + k_{R_{Q1}T0}i(t) + k_{R1T0}\ln\left(1 + |i(t)|\right) \text{sign}(i(t)) + k_{qdT0}q_d(t) + k_{qST0}q_s(t) + k_{qsNET0}\ln\left(\frac{q_s(t)}{Q}\right) + k_{qsNFT0}\ln\left(\frac{Q - q_s(t)}{Q}\right) \]  \hspace{1cm} (4.9)

Since the value of \( R_G = k_{R_{Q1}T0} \) and has already been determined, the value of \( R_{Q1}i(t) \) will be subtracted from \( E(t) \) before identification. Therefore, the equation for building the rows of the \( \Phi \) matrix of Equation 2.88 is

\[ \Phi_i = \begin{bmatrix} 1 & \ln(1 + |i(t)|) \text{sign}(i(t)) & q_d(t) \\ q_s(t) & \ln\left(\frac{q_s(t)}{Q}\right) & \ln\left(\frac{Q - q_s(t)}{Q}\right) \end{bmatrix}. \hspace{1cm} (4.10) \]

The equation for the parameter matrix \( \Theta \) from Equation 2.98 is

\[ \Theta = \left(\Phi^T \Phi\right)^{-1} \Phi^T \left(E(t) - k_{R_{Q1}T0}i(t)\right), \hspace{1cm} (4.11) \]

where \( k_{R_{Q1}T0} \) is the cell impedance \( R_G \) identified in Section 4.2. Plots of the identified system compared with some actual data are shown in Figures 4.9-4.17. These figures show the identified voltage in addition to the actual cell voltages that were also shown in Figures 3.8-3.34 from Chapter III. As the plots indicate, the voltage from the identified system matches the actual voltage very closely. The largest RMS error was 37.29 mV\(_{\text{rms}}\) for cell G, cycle 1 shown in the bottom plot of Figure 4.15.

The largest instantaneous error for the identified models occurred during the discharge phase of the cycles. This large deviation of the identified model from the actual voltage directly corresponds to the temperature of the cell. To help visualize this deviation, an expanded view of the actual and identified cell voltages of the discharge phase of cell E, cycle 1 along with the temperature is shown in Figure 4.18. When the discharge phase began, there was a sudden drop in cell temperature due to the endothermic nature of the chemical reactions during charging. However, as the cell
discharge continues, heating due to the internal resistance of the cell builds up, increasing the cell temperature. Since the voltage of the cell is dependent on the temperature, initially there is a sharp decrease in cell voltage. However, as the temperature of the cell increases, the voltage of the cell also increases. As expected, this deviation in cell voltage is even more pronounced as the MCV of the cells is increased.

To improve the cycle model near the start and end of the discharge phase, the model coefficients $k_i$ were adjusted to be temperature dependent. The first- and second-order temperature correction equations for model DMT$^1$ and DMT$^2$ were given in Equations 2.76 and 2.77 and are repeated here as

![Sample cycles for cell A showing identified dynamic model DMT$^0$.](image)
Figure 4.10: Sample cycles for cell B showing identified dynamic model DMT$^0$.

Figure 4.11: Sample cycles for cell C showing identified dynamic model DMT$^0$. 

114
Figure 4.12: Sample cycles for cell D showing identified dynamic model DMT0.

Figure 4.13: Sample cycles for cell E showing identified dynamic model DMT0.
Figure 4.14: Sample cycles for cell F showing identified dynamic model DMT<sup>0</sup>.

Figure 4.15: Sample cycles for cell G showing identified dynamic model DMT<sup>0</sup>.
Figure 4.16: Sample cycles for cell H showing identified dynamic model DMT$.^0$

Figure 4.17: Sample cycles for cell I showing identified dynamic model DMT$.^0$.
\[ E(t) = k_{E0T0} + k_{E0T1}T + (k_{R2T0} + k_{R2T1})i(t) + (k_{R2T0} + k_{R2T1})\ln(1 + |i(t)|)\text{sign}(i(t)) \]
\[ + (k_{qT0} + k_{qT1})q_{d}(t) + (k_{qT0} + k_{qT1})q_{s}(t) + (k_{qsNET0} + k_{qsNET1}T)\ln\left(\frac{q_{s}(t)}{Q}\right) + (k_{qsNET0} + k_{qsNET1}T)\ln\left(\frac{Q - q_{s}(t)}{Q}\right) \] (4.12)

and

\[ E(t) = k_{E0T0} + k_{E0T1}T + k_{E0T2}T^2 + (k_{R2T0} + k_{R2T1}T + k_{R2T2}T^2)i(t) + \]
\[ (k_{R2T0} + k_{R2T1}T + k_{R2T2}T^2)\ln(1 + |i(t)|)\text{sign}(i(t)) + \]
\[ (k_{qT0} + k_{qT1}T + k_{qT2}T^2)q_{d}(t) + (k_{qT0} + k_{qT1}T + k_{qT2}T^2)q_{s}(t) + (k_{qsNET0} + k_{qsNET1}T + k_{qsNET2}T^2)\ln\left(\frac{q_{s}(t)}{Q}\right) + \]
\[ (k_{qsNET0} + k_{qsNET1}T + k_{qsNET2}T^2)\ln\left(\frac{Q - q_{s}(t)}{Q}\right) \] (4.13)

Using Equation 4.12, the rows in the first-order temperature-corrected matrix \( \Phi \) become

\[
\Phi_i = [1, T, i(t)T, \ln(1 + |i(t)|)\text{sign}(i(t)), \ln(1 + |i(t)|)\text{sign}(i(t))T, q_{d}(t), q_{d}(t)T, q_{s}(t), q_{s}(t)T, \ln\left(\frac{q_{s}(t)}{Q}\right), \ln\left(\frac{q_{s}(t)}{Q}\right)T, \ln\left(\frac{Q - q_{s}(t)}{Q}\right), \ln\left(\frac{Q - q_{s}(t)}{Q}\right)T].
\] (4.14)

Using this modified \( \Phi \) matrix, the system was re-identified. Similarly, using Equation 4.13, the rows of \( \Phi \) matrix were modified using

\[
\Phi_j = [1, T, T^2, i(t)T, i(t)T^2, \ln(1 + |i(t)|)\text{sign}(i(t)), \ln(1 + |i(t)|)\text{sign}(i(t))T^2, q_{d}(t), q_{d}(t)T, q_{d}(t)T^2, q_{s}(t), q_{s}(t)T, q_{s}(t)T^2, \ln\left(\frac{q_{s}(t)}{Q}\right), \ln\left(\frac{q_{s}(t)}{Q}\right)T, \ln\left(\frac{q_{s}(t)}{Q}\right)T^2, \ln\left(\frac{Q - q_{s}(t)}{Q}\right), \ln\left(\frac{Q - q_{s}(t)}{Q}\right)T, \ln\left(\frac{Q - q_{s}(t)}{Q}\right)T^2].
\] (4.15)

and the system was re-identified. An expanded view of the discharge curve for cell E, cycle 1 is displayed in Figure 4.18. This figure depicts the identified system using each
of the three models: DMT\textsuperscript{0}, DMT\textsuperscript{1} and DMT\textsuperscript{2}. The instantaneous error for each of the identified models in Figure 4.18 is 28.28 mV\textsubscript{rms}, 11.81 mV\textsubscript{rms}, and 8.43 mV\textsubscript{rms} for the models DMT\textsuperscript{0}, DMT\textsuperscript{1} and DMT\textsuperscript{2} respectively.

As Figure 4.18 illustrates, adding the temperature correction improved the accuracy around the discharge curves, but there is very little visible difference between the models DMT\textsuperscript{1} and DMT\textsuperscript{2}. The condition number of the identification matrices for each of the three models was 7.44x10\textsuperscript{5}, 1.28x10\textsuperscript{8}, and 2.91x10\textsuperscript{10} respectively. The growing condition number as the models increase in size indicates that the added temperature terms are correlated (Recktenwald, 2000). Because model DMT\textsuperscript{2} adds much more complexity but provides only slightly less error, it was not used for developing the aging model discussed in Chapter V. The next section will compare the performance of the identified parameters from the two dynamic models DMT\textsuperscript{0} and DMT\textsuperscript{1}. Simulations of model DMT\textsuperscript{1} cell voltages for the 27 sample cycles are shown in Figures 4.19-4.27.

4.5 Identified Parameters for Dynamic Models DMT\textsuperscript{0} and DMT\textsuperscript{1}

This section analyzes the performance of the identification process and how this process affected the value of the identified parameters. An analysis of how the parameters change as the cell ages and how these results compare to the literature presented in Section 2.4 is given in Section 5.1 of the next chapter.

The life times of the identified parameters were normalized using the life fraction concept defined in Equation 2.78. Normalization of the parameters is necessary for accurate comparison of the parameters because the number of cycles produced by each cell over its life span is different. The life fraction ranges from zero, when the cell is new, to one when the cell has reached the end of its life. The cell data was placed within
Figure 4.18: Close up of discharge cycle showing identified dynamic models.

Figure 4.19: Sample cycles for cell A showing identified dynamic model DMT¹.
Figure 4.20: Sample cycles for cell B showing identified dynamic model $\text{DMT}^1$.

Figure 4.21: Sample cycles for cell C showing identified dynamic model $\text{DMT}^1$.

121
Figure 4.22: Sample cycles for cell D showing identified dynamic model DMT$^1$.

Figure 4.23: Sample cycles for cell E showing identified dynamic model DMT$^1$. 

122
Figure 4.24: Sample cycles for cell F showing identified dynamic model DMT$^1$.

Figure 4.25: Sample cycles for cell G showing identified dynamic model DMT$^1$.
Figure 4.26: Sample cycles for cell H showing identified dynamic model DMT$^1$. 

Figure 4.27: Sample cycles for cell I showing identified dynamic model DMT$^1$. 

124
an array containing 1000 rows. Each cycle row from the data was placed in a row of the array using

\[ \text{row} = \text{floor} \left( \frac{\text{cyclenumber} \times 1000}{\text{lastcycle}} \right), \]

where the function \text{floor} selects the lowest integer. For example, cell A’s last cycle was defined as cycle 536 in Table 3.5; the life fraction of cycle 400 is 400/536 or 0.7463.

Using Equation 4.15, the identified parameters for cycle 400 were then placed in row 746 of the 1000 row array. Similarly, cycle 401 has a life fraction of 0.7481. Parameters for cycle 401 were then placed in row 748 of the array. Missing rows such as row 747 in the example above, were generated using linear interpolation of parameters between the adjacent cycles. Using the normalized arrays, the identified parameters from different cells could be averaged and plotted using the MATLAB program.

4.5.1 Capacity Versus Life Fraction

Figure 4.28 shows the identified capacity for each of the nine cells plotted versus life fraction. The figure contains three plots: the top plot of each figure shows the specified parameter for all cells with an MCV of 4.2 V, the middle plot contains parameters for cells with an MCV of 4.3 V, and the bottom plot contains parameters for the cells charged to 4.4 V. The dotted lines show the actual identified parameter values while the solid lines show the filtered data using the moving average filter described in Section 2.6.
4.5.2 Identified Parameters for model DMT$^0$ Versus Life Fraction

This subsection looks at the behavior of the identified coefficients of the dynamic model DMT$^0$ which contains no temperature correction terms. Each one of the coefficients in the voltage Equation 4.9 for the dynamic model DMT$^0$ represent a specific physical aspect of the electrochemical cell as described in Table 2.1. Plots of each identified parameter for the dynamic model DMT$^0$ are shown in Figures 4.29-4.35 with the RMS error shown in Figure 4.36.

As the plots in Figures 4.29-4.35 show, each of the identified values for each cell with the same MCV are consistent with the other cells. For example, Figure 4.29 shows the coefficient $k_{E070}$ for model DMT$^0$. This voltage coefficient represents the standard voltage of the cell. Although the identified value of this voltage coefficient varies from cycle to cycle, it remains relatively constant throughout the life of the cell with a slight
overall increase in value. The values of $k_{E0T0}$ for each of the cells with an MCV of 4.2 V all start around 3.7 V and increase to around 3.8 V at the end of their life. For cells with an MCV of 4.3 V, $k_{E0T0}$ is around 3.6 V and increases to approximately 3.9 V. Likewise, the value of $k_{E0T0}$ for cells with an MCV of 4.4 V starts at an even lower value, 3.5 V, then increases to around 3.7 V at the end of cell life. Analysis of the other identified values for the coefficients of model DMT$^0$ shows that the identified parameters are consistent for all cells.

Figure 4.36 shows the $V_{rms}$ error between the identified model DMT$^0$ and the filtered data. For the majority of identified cycles, this error is well below 40 mV which validates this modeling approach.

![Figure 4.29: Identified values for the parameter $k_{E0T0}$ for model DMT$^0$.](image)
Figure 4.30: Identified values for the parameter $k_{R0T0}$ for model DMT$^{\text{III}}$.

Figure 4.31: Identified values for the parameter $k_{liT0}$ for model DMT$^{\text{III}}$.
Figure 4.32: Identified values for the parameter $k_{qdT0}$ for model DMT$^{43}$.

Figure 4.33: Identified values for the parameter $k_{qsT0}$ for model DMT$^{43}$.
Figure 4.34: Identified values for the parameter $k_{qsNET_0}$ for model DMT$^d$.

Figure 4.35: Identified values for the parameter $k_{qsNET_0}$ for model DMT$^d$. 
4.5.3 Identified Parameters for model DMT\textsuperscript{1} Versus Life Fraction

Coefficients for the voltage equation for dynamic model DMT\textsuperscript{1} given in Equation 4.12 contain the same coefficients as the DMT\textsuperscript{0} model, but this model also contains a temperature based coefficient to account for temperature changes for each term in the equation. Each of the identified parameters for this model is plotted in Figures 4.37-4.49 with the RMS error shown in Figure 4.50.

Figure 4.37 shows the coefficient $k_{E0T0}$ for the model DMT\textsuperscript{1}. Comparing this coefficient with the same coefficient from model DMT\textsuperscript{0} shown in Figure 4.29, the values of the coefficient $k_{E0T0}$ for the cells with an MCV of 4.2 V for the model DMT\textsuperscript{1} are all much higher and have a much large deviation than the same cells for the model DMT\textsuperscript{0}. Comparing the cells that had an MCV of 4.3 V, in model DMT\textsuperscript{1} cells D and E both have
voltage around 5.5 volts compared with the voltage of 3.6 V for all three cells of model DMT\(^0\). The third cell, cell F, actually starts out at a very low voltage, near 1 V, but then rises to meet the values of the other two cells as the cell loses capacity reducing the effect of temperature. Both cells G and I from the bottom plot of Figure 4.37 start out at a low voltage, approximately 2 V, while the other cell, cell H, starts at a high voltage.

The large deviation in the values for the coefficient \(k_{E0T0}\) of model DMT\(^1\) is explained by looking at the coefficient \(k_{E0T1}\) shown in Figure 4.38. All the values for the cells charged to 4.2 V have a coefficient \(k_{E0T1}\) term that is very small or negative. When these coefficients are combined as

\[
k_{E0C} = k_{E0T0} + k_{E0T1} T, \tag{4.17}
\]

using the average temperature of each cycle for the temperature \(T\), the result is very close to the values given for the \(k_{E0T0}\) only in model DMT\(^0\). Plots of Equation 4.17 are shown in Figures 4.51-4.53. The combined value \(k_{E0C}\) from Equation 4.17 shown in the bottom plot of Figure 4.51 compares reasonably well with the top plot of Figure 4.29, which shows the values of \(k_{E0T0}\) for model DMT\(^0\) for cells with an MCV of 4.2 V. Similarly, the bottom plots of Figures 4.52 and 4.53 are consistent with the middle and bottom plots of Figure 4.29. Figures 4.51-4.53 illustrate the correlation between the linked terms \(k_{E0T0}\) and \(k_{E0T1}\) for model DMT\(^1\). An analysis of the behavior of the other identified coefficients for model DMT\(^1\) shown in Figures 4.39-4.49 reveals similar correlation between each coefficient and its respective temperature corrective term. The large deviation in the coefficient \(k_{E0T0}\), and most of the other identified coefficients, for model DMT\(^1\) makes this model impractical for application of the aging model. However, as
Figure 4.37: Identified values for the parameter $k_{E070}$ for model DMT$^1$.

Figure 4.38: Identified values for the parameter $k_{E071}$ for model DMT$^1$. 

133
Figure 4.39: Identified values for the parameter $k_{\Omega T1}$ for model DMT$^1$.

Figure 4.40: Identified values for the parameter $k_{nT0}$ for model DMT$^1$. 
Figure 4.41: Identified values for the parameter $k_{LT1}$ for model DMT$^1$.

Figure 4.42: Identified values for the parameter $k_{qtD0}$ for model DMT$^1$.  

135
Figure 4.43: Identified values for the parameter $k_{qdt1}$ for model DMT$^\dagger$.

Figure 4.44: Identified values for the parameter $k_{qsT0}$ for model DMT$^\dagger$. 
Figure 4.45: Identified values for the parameter $k_{qsT1}$ for model DMT$^1$.

Figure 4.46: Identified values for the parameter $k_{qsNET0}$ for model DMT$^1$. 
Figure 4.47: Identified values for the parameter $k_{qsNET1}$ for model DMT1.

Figure 4.48: Identified values for the parameter $k_{qsNFT0}$ for model DMT1.
Figure 4.49: Identified values for the parameter $k_{q_{NFT1}}$ for model DMT$^1$.

Figure 4.50: Magnitude of model error for model DMT$^1$.
Figure 4.51: Combination of terms $k_{E0T0}$ and $k_{E1T1}$ for DMT using the average cell temperature for an MCV of 4.2 V.

Figure 4.52: Combination of terms $k_{E0T0}$ and $k_{E1T1}$ for DMT using the average cell temperature for an MCV of 4.3 V.
Figure 4.53: Combination of terms $k_{E0T0}$ and $k_{E0T1}$ for DMT$^1$ using the average cell temperature for an MCV of 4.4 V.

Figure 4.18 shows, temperature correction applied to the voltage equation does improve the response as shown in Figure 4.50.

4.5.4 Individual Temperature Correction Terms Versus Life Fraction

As Section 4.5.3 described, the strong temperature correlation between the like terms of the Equation 4.12 allow too much flexibility in the identification process and result in the large deviation of parameters between cells. To restrict the identification process and identify which of the coefficients from Equation 4.12 are most affected by the temperature of the cell, identification was redone for model DMT$^1$ while allowing only one of the first-order parameters in Equation 4.12 to be non-zero at a time.
Table 4.2 shows the average error from the identification of the modified DMT\textsuperscript{1} model along with the average error of the full DMT\textsuperscript{0} and DMT\textsuperscript{1} models. The data in Table 4.2 is sorted showing the model with the largest error at the top. The overall $V_{rms}$ error of the DMT\textsuperscript{0} model for all cells and all identified cycles was 29.80 mV\textsubscript{rms}. For the temperature correction terms for each of the coefficients in Equation 4.12, the term that has the largest error, or least impact on error reduction, for the DMT\textsuperscript{1} model is the $k_{qs\text{NET1}}$ term. This is logical since when the cell is near empty, there is little energy to affect the voltage regardless of the cell temperature. The next largest error term is $k_{E\text{O71}}$ indicating the temperature effect of the standard voltage is less important than that of the current or the charge within the cell.

### Table 4.2: Identification error values for the different temperature models.

<table>
<thead>
<tr>
<th>Model Label</th>
<th>Model Description</th>
<th>MCV 4.2 V (mV\textsubscript{rms})</th>
<th>MCV 4.3 V (mV\textsubscript{rms})</th>
<th>MCV 4.4 V (mV\textsubscript{rms})</th>
<th>Overall (mV\textsubscript{rms})</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMT\textsuperscript{0}</td>
<td>All Terms</td>
<td>27.34</td>
<td>30.02</td>
<td>32.05</td>
<td>29.80</td>
</tr>
<tr>
<td>DMT\textsuperscript{m1}</td>
<td>$k_{qs\text{NET1}} \neq 0$</td>
<td>26.98</td>
<td>29.22</td>
<td>31.15</td>
<td>29.11</td>
</tr>
<tr>
<td>DMT\textsuperscript{m2}</td>
<td>$k_{E\text{O71}} \neq 0$</td>
<td>24.85</td>
<td>28.21</td>
<td>31.04</td>
<td>28.03</td>
</tr>
<tr>
<td>DMT\textsuperscript{m3}</td>
<td>$k_{qs\text{NET1}} \neq 0$</td>
<td>20.09</td>
<td>24.52</td>
<td>29.95</td>
<td>24.85</td>
</tr>
<tr>
<td>DMT\textsuperscript{m4}</td>
<td>$k_{qs\text{NET1}} \neq 0$</td>
<td>20.00</td>
<td>24.28</td>
<td>29.71</td>
<td>24.66</td>
</tr>
<tr>
<td>DMT\textsuperscript{m5}</td>
<td>$k_{R\text{O71}} \neq 0$</td>
<td>22.75</td>
<td>23.55</td>
<td>27.16</td>
<td>24.49</td>
</tr>
<tr>
<td>DMT\textsuperscript{m6}</td>
<td>$k_{q\text{dT1}} \neq 0$</td>
<td>19.44</td>
<td>21.17</td>
<td>23.58</td>
<td>21.40</td>
</tr>
<tr>
<td>DMT\textsuperscript{m7}</td>
<td>$k_{l\text{T1}} \neq 0$</td>
<td>18.29</td>
<td>20.34</td>
<td>23.05</td>
<td>20.56</td>
</tr>
<tr>
<td>DMT\textsuperscript{m8}</td>
<td>$k_{l\text{T1}}$ and $k_{q\text{dT1}} \neq 0$</td>
<td>18.08</td>
<td>19.91</td>
<td>22.04</td>
<td>20.01</td>
</tr>
<tr>
<td>DMT\textsuperscript{m9}</td>
<td>$k_{R\text{O71}}$, $k_{l\text{T1}}$ and $k_{q\text{dT1}} \neq 0$</td>
<td>17.21</td>
<td>18.87</td>
<td>21.72</td>
<td>19.27</td>
</tr>
<tr>
<td>DMT\textsuperscript{1}</td>
<td>All Terms</td>
<td>13.18</td>
<td>14.25</td>
<td>18.82</td>
<td>15.42</td>
</tr>
</tbody>
</table>

The individual term having the greatest effect on temperature (i.e. producing the smallest error for a single temperature parameter) is the polarization term $k_{l\text{T1}}$, closely followed by the diffusion term $k_{q\text{dT1}}$ and the current term $k_{R\text{O71}}$. The high temperature
sensitivity of each of these terms indicates the current and polarization have a greater
effect on the cell voltage than the stored charge or standard voltage of the cell.

Since the terms $k_{l/\Omega T}$ and $k_{qTd}$ had the smallest individual errors, a new model was
identified using only these two first-order terms together with all the term from model
DMT$^0$. Similarly identification was also done using all three of the smallest error terms,
$k_{R\Omega T}$, $k_{l/\Omega T}$ and $k_{qTd}$. Each of these modified DMT$^1$ models were used to develop an
aging model as will be described in Chapter V.

4.6 Chapter Summary

This chapter presented the method of model parameter identification. Initially,
three dynamic models were chosen: DMT$^0$ containing no temperature correction terms,
DMT$^1$ containing first order temperature correction, and DMT$^2$ containing first and
second order temperature correction terms. Model comparisons showed that all three
models fit the cycle data well, with the temperature correction models having the smallest
errors as expected. The small difference in model error between the first and second
order temperature correction models showed that using the more complex second order
model was unnecessary.

Analysis showed that the coefficients showed consistency between all nine cells
for the model DMT$^0$. However, the correlation between the temperature correction terms
and their respective non-temperature correction terms for the model DMT$^1$ showed large
deviations in the identified parameters between cells. Alternatively, when identification
was performed using only one temperature correction term at a time, the polarization
term was revealed to have the largest reduction in the identification error, followed
closely by the diffusion term. Combinations of the first-order parameters that were most
sensitive to temperature produced smaller errors and consistent values of the identified parameters. Changes in the identified parameters as the cell ages are presented in the next chapter.
CHAPTER V
AGING RESULTS

Identification of model parameters for all the cycles for all the cells was performed using batch least squares as described in the previous chapter. This chapter analyzes the performance of the cells over their cycle life and compares that with the literature discussed in Section 2.4. An aging model is created and simulations of the predicted cell behavior are compared with the actual data. Finally, a method of using the aging model is presented.

5.1 Cell Performance

This section analyzes the performance of the cells over their life span and compares their performance with reports presented in Section 2.4. Identified values for capacity, resistance, and the identified coefficients of the cell voltage equation are studied as the cells age.

For comparison, each of the values of the identified parameters is placed within an array using the life fraction as described in Section 4.5. Each parameter for each of the three cells with the same MCV were then averaged together. Averaged vectors of each of the three MCVs were then plotted on the same axis for a visual reference of how the parameters changed over time.
5.1.1 Capacity and Cycle Life

Figure 2.19 given in Section 2.4 shows how cell capacity increases as the maximum charging voltage (MCV) is increased. Figure 5.1 shows the capacity plot of Figure 2.19 along with the averaged values of capacity from the experimental cells. This figure also plots the averaged values of the experimentally determined cycle life for comparison with the data presented by Bergveld et al. (2002). Figure 5.1 shows the cycle life of the experimental data using the end of life values of both 80% rated capacity, plotted as triangles, and 50% rated capacity, plotted as diamonds. The dashed lines show the estimated cycle life for the full range of voltages supplied by Bergveld et al. (2002). The estimated cycle life values shown in Figure 5.1 were determined by applying an exponential curve fit to the experimental data. The cycle life equation used was

\[ \text{cyclelife} = k_{cl1} e^{(-k_{cl2} \text{MCV})}. \] (5.1)

By taking the logarithm of both sides of Equation 5.1 and performing batch least squares with the three data points, the coefficients \( k_{cl1} \) and \( k_{cl2} \) were determined to be 2010 and 3.344 respectively for end of life at 50% capacity.

As Figure 5.1 shows, both the capacity and cycle life from the experimental cells are consistent with the data presented by Bergveld et al. (2002). The experimental cells had lower maximum capacities, primarily due to the large charging current which caused the cell to reach the maximum charging voltage quickly, thus reducing the total amount of charge added to the cells. However, as the MCV was increased, the cell capacity increased following the same trend as the Bergveld data. As the MCV was increased, the cycle life of the experimental cells decreased, although not as sharply as the cells from the Bergveld data.
5.1.2 Capacity Fade

Spotnitz (2003) reported that capacity fade due to cycling goes through four rates as shown in Figure 2.20. Figure 5.2 shows the capacity plot of cell F, along with the general fade rates presented by Spotnitz. Capacity plots for the experimental cells were given in Figures 3.35-3.37 of Chapter III. All of these plots contain the fade rates labeled B, C, and D in Figure 2.20. Figure 5.3 shows a close-up of the early cycles for each of the nine experimental cells. Cells C, F, and I also appear to have an early fade rate labeled A in Figure 2.21. The other cells do not show this fade rate A. The lack of a clear fade rate A is consistent with the large charging current used, as was presented by Abraham et al. (2007).
Figure 5.2: Capacity of cell F shown with general fade points presented by Spotnitz (2003).

Figure 5.3: Capacity of all nine cells during the first 45 cycles.

Section 3.5.1 presented individual capacity plots of the nine experimental cells. In order to clearly see how these cells aged with respect to the maximum charging voltages, the average capacity of each of the cells with the same MCV have been plotted.
versus life fraction in Figure 5.4. During the first 40% of cell life fraction, the average capacity for each of the three MCVs remains nearly constant with only a slight decrease. Beyond 40 percent, the capacity of the cells decreases at a linear rate.

Capacity of the cells with a MCV of 4.4 V decreased faster than the cells with lower MCVs. Since the increase in cell resistance (shown in Figure 4.6) is linear throughout the life of the cells, the capacity fade must also be affected by the side reactions discussed in Section 2.4.4. In the next section, the aging effects of the identified parameters from the dynamic models will be reviewed, and correlation with the cell resistance and capacity fade will be discussed.

![Figure 5.4: Average capacity of cells versus life fraction for different MCVs.](image)

5.2 Identified Parameters

In this section, the identified parameters of the dynamic models from Chapter IV are reviewed. The identified parameters for the cells with the same MCV were averaged together. These three averaged values were then filtered using a moving average filter and plotted on the same axis for comparisons. This section presents these plots and discusses them.
5.2.1 Aging Comparison: No Temperature Correction

In this section, the identified parameters of the voltage equation with no temperature correction from Equation 2.75 are analyzed. Figures 5.5-5.11 show the plots of the cell parameters versus life fraction for the dynamic model DMT\textsuperscript{0}. In each plot, the blue line indicates the average of the three cells charged to an MCV of 4.2 V. The magenta line is the average of the three cells charged to 4.3 V, and the black line is the average of the three cells charged to 4.4 V. A summary of how the coefficients change as the cells age is given in Table 5.1.

Figure 5.5 shows the averages of the parameter $k_{E0T0}$ for the model DMT\textsuperscript{0}. This figure shows the averaged values from the three plots in Figure 4.29. As the plot shows, the base voltage, $k_{E0T0}$, of the cell slightly increases as the cell ages. This small increase in voltage probably occurs due to remaining lithium in the negative electrode causing the shift in voltage of the negative electrode from a negative to positive value as discussed in Section 2.4.3. This base voltage is slightly lower for the cells with higher MCVs, indicating the other parameters within the model have more impact on the overall cell voltage when the ending charge voltage of the cell is larger.

Figure 5.6 shows the averages of the parameter $k_{R\Omega T0}$. This figure shows the identified value of $R\Omega$ given in Equation 4.8 and the coefficients are listed in Table 4.1. As discussed in Section 2.4.3, the cell resistance rises linearly as the cells age. The cell resistance increases at a faster rate as the maximum charge voltage is increased.

The parameter $k_{iIT0}$ is shown in Figure 5.7. This parameter models the effect of the current on the polarization of the electrodes. This parameter rises with age and MCV. This trend is consistent with the results presented by Ning et al. (2006) in Section 2.4.4.
Table 5.1: Description of how identified parameters change with age for model DMT0.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Aging</th>
<th>MCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{E0T0}$ (V)</td>
<td>Base voltage of a cell when the cell current, diffused charge, and stored charge are all zero.</td>
<td>Slightly Increases</td>
<td>Value smaller for larger MCV. Value increases slower for larger MCV.</td>
</tr>
<tr>
<td>$k_{R0T0}$ (Ω)</td>
<td>Primary resistance of the cell. Caused primarily by the SEI film and diffusion.</td>
<td>Increases</td>
<td>Value larger for larger MCV. Increases faster if charged to a higher MCV.</td>
</tr>
<tr>
<td>$k_{Q0T0}$ (V)</td>
<td>Polarization adjustment caused by the current flowing in the cell.</td>
<td>Increases</td>
<td>Value larger for larger MCV. Increases faster if charged to a higher MCV.</td>
</tr>
<tr>
<td>$k_{qdT0}$ (V/Aseconds)</td>
<td>Potential adjustment caused by the diffused charge within the cell.</td>
<td>Decreases</td>
<td>Value smaller for larger MCV. Value decreases faster for larger MCV.</td>
</tr>
<tr>
<td>$k_{qsT0}$ (V/Aseconds)</td>
<td>Potential scaling-factor due to the stored charge within the electrode.</td>
<td>Increases</td>
<td>Value larger for larger MCV. Increases faster if charged to a higher MCV.</td>
</tr>
<tr>
<td>$k_{qsNET0}$ (V)</td>
<td>Potential scaling-factor of the cell near empty, more abundant uncharged material.</td>
<td>Increases</td>
<td>Value smaller for larger MCV. Value increases slower for larger MCV.</td>
</tr>
<tr>
<td>$k_{qsNFT0}$ (V)</td>
<td>Potential scaling-factor of the cell near full charge, more abundant charged material.</td>
<td>Increases</td>
<td>Value is negative during first 55% of life, then goes positive. Cells with larger MCV start out more negative, but end more positive than cells with smaller MCVs.</td>
</tr>
</tbody>
</table>

Parameter $k_{qdT0}$ represents the effect of the diffused charge on the cell voltage. Plots showing how $k_{qdT0}$ changes over the life of the cells are shown in Figure 5.8. As the figure shows, $k_{qdT0}$ slowly decreases as the cells age, indicating that the effect of the diffused charge on the cell voltage is reduced in relation to the other terms of the voltage equation.

Parameter $k_{qsT0}$ represents the linear effect of the stored charge on the cell voltage. Plots for the averaged values of $k_{qsT0}$ are given in Figure 5.9. For the MCV of 4.2 V, this parameter is almost constant, having a slight linear increase. The increase is more pronounced for the MCV of 4.3 V. For the MCV of 4.4 V, the slope of the parameter
increase is small until the cell reaches 40% of its life, at which point the slope increases significantly until about 70%, when it levels out again.

The plots for parameter $k_{qs\text{NET}0}$ are shown in Figure 5.10. Parameter $k_{qs\text{NET}0}$ represents the sharp decrease in cell voltage as it nears the end of charge. This parameter remains fairly constant at a value slightly below zero for the first 40% of the cell’s life, then the value moves slightly positive. The plots for this parameter follow the plots for the parameter $k_{E0\text{T}0}$ shown in Figure 5.5 only shifted down near zero.

The final parameter $k_{qs\text{NFT0}}$ is shown in Figure 5.11. This parameter starts negative for all nine cells and moves positive as the cells age. As the MCV increases, the parameter $k_{qs\text{NFT0}}$ starts at lower values and then increases at a faster rate than cells with lower MCVs. During the first 40% of cell life, the parameter increases slowly. Between 40% and 70%, it increases at a much higher rate. After 70%, the parameter $k_{qs\text{NFT0}}$ levels off to a constant value.

Of the seven parameters listed in Table 5.1, six of them increase as the cells age. This increase is primarily due to the increase in cell resistance caused by the growth of the SEI layer. The one parameter that decreased, $k_{qd\text{T}0}$, decreased towards zero which indicates that the increase in cell resistance simply reduces the effect of the diffusion term on the cell voltage as the cells age.

5.2.2 Aging Comparison: With Temperature Correction

The previous subsection discussed the performance of the parameters in the voltage Equation 2.75. In Section 4.5.4, the temperature correction terms from Equation 2.76 were identified using one first-order temperature correction term at a time. This
Figure 5.5: Averaged parameter $k_{E070}$ for model DMT$^0$.

Figure 5.6: Averaged parameter $k_{R270}$ for model DMT$^0$.

Figure 5.7: Averaged parameter $k_{U70}$ for model DMT$^0$. 
Figure 5.8: Averaged parameter $k_{qTd0}$ for model $\text{DMT}^0$.

Figure 5.9: Averaged parameter $k_{qsT0}$ for model $\text{DMT}^0$.

Figure 5.10: Averaged parameter $k_{qsNET0}$ for model $\text{DMT}^0$. 
Figure 5.11: Averaged parameter $k_{qNFT0}$ for model DMT$^0$.

section analyzes the results of these individual first-order temperature correction terms, along with their corresponding zeroth-order terms and compares them with the results from the previous section. A summary of the findings is listed in Table 5.2.

Figures 5.12-5.18 show the averaged identified values for each of the terms in Equation 2.76. Each line in the figure represents the average of the three cells with the same MCV. Figure 5.12 contains the averaged values of the parameter $k_{E0T0}$ for model DMT$^1$ in the top plot. The middle plot in Figure 5.12 shows the averaged values for the parameter $k_{E0T1}$. The averaged values $k_{E0T0}$ remain relatively constant over the life of the cell. The temperature correction term $k_{E0T1}$ shows a rise as the cells age. The bottom plot in Figure 5.12 shows the combined plots of the averaged values of $k_{E0T0}$ plus the averaged values of $k_{E0T1} \times T$, where $T$ is the average temperature for each cycle. These combined plots of the bottom plot on Figure 5.12 agree closely with Figure 5.5 showing the $k_{E0T0}$ for the DMT$^0$ model.

Figure 5.13 shows the averaged values of the cell resistance term for $k_{R\Omega T0}$ along with its associated temperature correction term $k_{R\Omega T1}$. The resistance term $k_{R\Omega T0}$ is
Table 5.2: Description of how identified parameters change with age for modified models DMT\textsuperscript{mi-m7} (Each $T_1$ term was identified individually).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Aging</th>
<th>MCV</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{E_0T_0}$ (V)</td>
<td>Slight decrease</td>
<td>Close together during early portion of cell life, diverges as the capacity fade increases.</td>
<td>Combined values agree with parameter from the DMT\textsuperscript{0} model.</td>
</tr>
<tr>
<td>$k_{E_0T_1}$ (V$/^\circ$C)</td>
<td>Increases</td>
<td>Smaller value for large MCVs. Values converge as the cells age.</td>
<td></td>
</tr>
<tr>
<td>$k_{R_{Q_0T_0}}$ (Ω)</td>
<td>Increases</td>
<td>Increases faster for larger MCV.</td>
<td>Combined values decrease in disagreement with the DMT\textsuperscript{0} model.</td>
</tr>
<tr>
<td>$k_{R_{Q_0T_1}}$ (Ω$/^\circ$C)</td>
<td>Decreases</td>
<td>Relatively constant among different MCVs.</td>
<td></td>
</tr>
<tr>
<td>$k_{Q_0T_0}$ (V)</td>
<td>Increases</td>
<td>Large MCV starts out smaller but increases faster than other MCVs.</td>
<td>Combined values agree with parameter from the DMT\textsuperscript{0} model.</td>
</tr>
<tr>
<td>$k_{Q_0T_1}$ (V$/^\circ$C)</td>
<td>Decreases</td>
<td>Large MCV starts out large but decreases faster than other MCVs.</td>
<td></td>
</tr>
<tr>
<td>$k_{q_0T_0}$ (V/Aseconds)</td>
<td>Increases</td>
<td>Large MCV starts out smaller but increases faster than other MCVs.</td>
<td>Combined values disagree with the DMT\textsuperscript{0} model.</td>
</tr>
<tr>
<td>$k_{q_0T_1}$ (V/Aseconds$/^\circ$C)</td>
<td>Decreases</td>
<td>Large MCV starts out large but decreases faster than other MCVs.</td>
<td></td>
</tr>
<tr>
<td>$k_{q_0T_0}$ (V/ Aseconds)</td>
<td>Relatively constant</td>
<td>Larger MCV increases near end of life.</td>
<td>Value larger for larger MCV. Increases faster if charged to a higher MCV.</td>
</tr>
<tr>
<td>$k_{q_0T_1}$ (V/Aseconds$/^\circ$C)</td>
<td>Increases</td>
<td>Smaller MCV increases faster.</td>
<td></td>
</tr>
<tr>
<td>$k_{q_{NET_0}}$ (V)</td>
<td>Relatively constant</td>
<td>Close together during early portion of cell life, diverges as the capacity fade increases.</td>
<td>Combined values agree with parameter from the DMT\textsuperscript{0} model.</td>
</tr>
<tr>
<td>$k_{q_{NET_1}}$ (V$/^\circ$C)</td>
<td>Relatively constant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{q_{NFT_0}}$ (V)</td>
<td>Increases</td>
<td>Smaller MCV increases faster.</td>
<td>Combined values agree with parameter from the DMT\textsuperscript{0} model.</td>
</tr>
<tr>
<td>$k_{q_{NFT_1}}$ (V$/^\circ$C)</td>
<td>Slight decrease</td>
<td>Smaller MCV decreases faster.</td>
<td></td>
</tr>
</tbody>
</table>

identical as the resistance for model DMT\textsuperscript{0} since this value was identified from Equation 4.7 and subtracted out of Equation 2.76 before identification of the unknown terms. The middle plot of Figure 5.13 shows the temperature correction term $k_{R_{Q_0T_1}}$. The values for
the parameter $k_{RΩT_1}$ decrease as the cells age, thus reducing the effect of the cell resistance on the voltage. When the parameter $k_{RΩT_0}$ and $k_{RΩT_1}$ are combined using the average cell temperature, the overall cell resistance term is reduced. The reduction in the overall resistance of the cell does not agree with the measured results, indicating that this temperature correction term is not a good choice for the model.

Averaged values for the terms $k_{liT_0}$ and $k_{liT_1}$ are shown in the top and middle plots of Figure 5.14. These two values show opposite behavior, with the $k_{liT_0}$ parameter increasing as the cells age and the $k_{liT_1}$ parameter decreasing. When these two terms are combined, however, they show the same general trend as the $k_{liT_0}$ parameter from the mode DMT$^0$.

Averaged values for the diffused charge parameters $k_{qdT_0}$ and $k_{qdT_1}$ are given in Figure 5.15. These parameters also oppose each other. When combined as shown in the bottom plot of Figure 5.15, they show an overall increase in the diffused charge coefficient, which is the opposite of the $k_{qdT_0}$ parameter for the model DMT$^0$.

Figure 5.16 shows the linear stored charge parameters $k_{qsT_0}$ and $k_{qsT_1}$. Parameter $k_{qsT_0}$ remains relatively constant for all cells as the cells age, with some divergence among the cells with different MCVs. Parameter $k_{qsT_1}$ shows a consistent increase as a cell ages. When combined, these parameters follow the same trend as the parameter $k_{qsT_0}$ from the DMT$^0$ model. Similarly, when combined, the parameters $k_{qsNFT_0}$ and $k_{qsNFT_1}$ as well as the parameters $k_{qsNET_0}$ and $k_{qsNET_1}$ shown in Figures 5.17 and 5.18 respectively show general agreement with their corresponding terms in the DMT$^0$ model.
Figure 5.12: Combination of averaged terms \( k_{E0T0} \) and \( k_{E0T1} \) of model DMTm2.

Figure 5.13: Combination of averaged terms \( k_{R\Omega T0} \) and \( k_{R\Omega T1} \) of model DMTm3.
Figure 5.14: Combination of averaged terms $k_{iT0}$ and $k_{iT1}$ of model DMT$_{m_f}$.

Figure 5.15: Combination of averaged terms $k_{qd0}$ and $k_{qd1}$ of model DMT$_{m_6}$. 

159
Figure 5.16: Combination of averaged terms $k_{qsT0}$ and $k_{qsT1}$ of model DMT$^{m^5}$.

Figure 5.17: Combination of averaged terms $k_{qsNET0}$ and $k_{qsNET1}$ of model DMT$^{m^1}$. 

160
5.2.3 Aging Comparison: Summary

This section discussed the behavior of the identified parameters of the model DMT⁰ as well as the modified first-order temperature models DMTᵐ¹ through DMTᵐ⁹. Identification of the parameters for the model DMT⁰ resulted in consistent parameter trends among cells as the cells aged. Identification of the parameters for the modified first-order temperature models DMTᵐ¹ through DMTᵐ⁹ resulted in conflicting parameter trends among cells as the cells aged in several cases. This was particularly true for the resistance and diffused charge parameters. The next section develops an aging model based on MCV and life fraction of the cells using the modified first-order temperature models DMT⁰ and DMTᵐ⁸ which gave the smallest errors. This aging model is applied to the dynamic models and results are discussed.

Figure 5.18: Combination of averaged terms $k_{qsNFT0}$ and $k_{qsNFT1}$ of model DMTᵐ⁴.
5.3 Aging Model

This section discusses the development of an aging model that defines how the identified parameters for the dynamic models DMT₀ and DMT¹ change over the life of the cell. First, a method is given for determining how much life fraction is used. Next, the aging model is developed, and finally, some results of the aging model applied to the cycled cells are discussed.

5.3.1 Life Fraction

Equation 5.1 gave the cyclelife of a cell based on the MCV. Using the identified values for the cells in this cycle, Equation 5.1 is given as

\[ \text{cyclelife} = 2010e^{(-3.344MCV)} . \]  

(5.2)

The amount of life used during a given cycle is the reciprocal of Equation 5.2. The accumulation of the life used, or the life fraction (LF) of the cell, can then be determined using

\[ LF_{n+1} = LF_n + \frac{1}{2010e^{(-3.344MCV_n)}} , \]  

(5.3)

where \( n \) is the cycle number. Using Equation 5.3, the total used life fraction can be estimated. This value can then be used to determine the coefficients of the voltage equation using the aging model presented next.

5.3.2 Aging Model Definition

Equation 4.8 gave the linear equation for the cell resistance as

\[ R_{z1} = k_{Rz1} + k_{Rz2} \text{cycle} . \]  

(5.4)
If the variable *cycle* from Equation 5.4 is replaced with the life fraction of the cell and general coefficients are used, Equation 5.4 can be applied to all the parameters in the dynamic model as

\[ k_{ijc} = k_{y1} + k_{y2}LF. \]  

(5.5)

Since the parameters of the dynamic model are dependent on the maximum charging voltage, Equation 5.5 is modified as

\[ k_{ix} = (k_{ix1} + k_{ix2}MCV) + (k_{ix3} + k_{ix4}MCV)LF, \]  

(5.6)

where the coefficients \( k_{ix} \) are identified using batch least squares applied to the averaged identified parameters as they aged. Equation 5.6 will be referred to as the linear aging model, LAM. Adding a quadratic life fraction term to Equation 5.6 gives

\[ k_{ix} = (k_{ix1} + k_{ix2}MCV) + (k_{ix3} + k_{ix4}MCV)LF + (k_{ix5} + k_{ix6}MCV)LF^2, \]  

(5.7)

where \( k_{ix} \) are coefficients to be determined also using batch least squares. Equation 5.7 will be referred to as the quadratic aging model, QAM.

5.3.3 Aging Model Results

The top plot in Figure 5.19 shows the averaged cell capacity for the three MCVs. The linear aging model for the cell capacity is also shown in the top plot as the color coded dotted lines with symbols. The bottom shows the averaged cell capacity along with the quadratic aging model. As Figure 5.19 indicates, the single linear line of the LAM model has an overlarge estimate of the cell capacity near the beginning of a cell’s life and a reduced capacity near the middle of a cell’s life, where its capacity begins to fade at a much faster rate. The quadratic aging model, shown in the bottom plot of Figure 5.19, provides a more accurate match to the measured capacity. Therefore, the
QAM was used to estimate the cell capacity, while the LAM was used to determine the remaining parameters in dynamic models. The parameters for the capacity are given for the aging model QAM in Table 5.3.

Table 5.3: Parameters for the capacity aging model QAM.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (Aseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{l1}$</td>
<td>$-3.3276e+04$</td>
</tr>
<tr>
<td>$k_{l2}$</td>
<td>$9.6673e+03$</td>
</tr>
<tr>
<td>$k_{l3}$</td>
<td>$9.2981e+03$</td>
</tr>
<tr>
<td>$k_{l4}$</td>
<td>$-2.3143e+03$</td>
</tr>
<tr>
<td>$k_{l5}$</td>
<td>$2.5031e+04$</td>
</tr>
<tr>
<td>$k_{l6}$</td>
<td>$-6.6453e+03$</td>
</tr>
</tbody>
</table>

The linear aging model was applied to each of the parameters in the voltage equations for the dynamic models DMT$^0$ and DMT$^1$, as well as each of the modified versions of DMT$^1$ listed in Table 4.2. The linear aging models were calculated for all cycles identified in Chapter IV. The average error over all cycles was then calculated for each aging model. Table 5.4 lists the average errors over all cycles for each of the dynamic models given in Table 4.2. The aging models are labeled to match the labels given to the dynamic models in Table 4.2. The data listed in Table 5.4 is sorted with the aging model having the largest error listed at the top and the aging model having the smallest error at the bottom. From these results, it can be seen that the largest error occurs from applying the aging model LAM$^1$, which uses all 14 of the identified parameters from the complete model DMT$^1$. This large error is a direct reflection of the strong correlation between the first-order temperature correction terms and their corresponding zeroth-order terms, as discussed in Section 4.5.3.

The linear aging model LAM$^0$ applied to model DMT$^0$ actually had smaller errors than the modified dynamic models DMT$^{m1}$, DMT$^{m2}$, DMT$^{m3}$ and DMT$^{m4}$, having non-zero first-order parameters of $k_{qsNET1}$, $k_{E0T1}$, $k_{qST1}$ and $k_{qsNET1}$ respectively. The best aging model was created using the two temperature correction terms of model DMT$^{m8}$ from
Table 5.4: Average RMS error values for aging models.

| Aging Model | Dynamic Model (Label - Description) | MCV 4.2 V (mVrms) | MCV 4.3 V (mVrms) | MCV 4.4 V (mVrms) | Overall (mVrms) |
|-------------|-------------------------------------|-------------------|-------------------|-------------------|----------------
| LAM⁰        | DMT¹ - All Terms                     | 140.81            | 137.91            | 136.84            | 138.52          |
| LAM¹⁰¹      | DMT¹⁰¹ - k_{q,t1} ≠ 0                | 139.85            | 118.60            | 105.43            | 121.29          |
| LAM¹⁰²      | DMT¹⁰² - k_{q,t1} ≠ 0                | 113.17            | 89.20             | 78.60             | 93.66           |
| LAM¹⁰³      | DMT¹⁰³ - k_{q,t1} ≠ 0                | 83.45             | 76.13             | 71.04             | 76.87           |
| LAM¹⁰⁴      | DMT¹⁰⁴ - k_{q,NFT1} ≠ 0              | 75.82             | 62.54             | 57.45             | 65.27           |
| LAM¹⁰⁵      | DMT¹⁰⁵ - All Terms                   | 71.80             | 62.64             | 58.76             | 64.40           |
| LAM¹⁰⁶      | DMT¹⁰⁶ - k_{q,t1} ≠ 0                | 61.04             | 59.84             | 61.52             | 60.80           |
| LAM¹⁰⁷      | DMT¹⁰⁷ - k_{q,t1} ≠ 0                | 48.49             | 48.62             | 49.85             | 48.99           |
| LAM¹⁰⁸      | DMT¹⁰⁸ - k_{q,t1} ≠ 0                | 46.55             | 46.12             | 47.17             | 46.62           |
| LAM¹⁰⁹      | DMT¹⁰⁹ - k_{q,t1}, k_{q,t1} ≠ 0     | 47.06             | 43.85             | 46.32             | 45.74           |
| LAM¹¹⁰      | DMT¹¹⁰ - k_{q,t1}, k_{q,t1} ≠ 0     | 45.75             | 44.57             | 46.79             | 45.70           |

Figure 5.19: Capacity plots showing comparison between linear and quadratic aging models.

Table 4.2. From analysis of Table 5.4, two aging models were selected. The first aging model is LAM⁰ due to its small error and small size, and the second model is LAM¹⁰⁸ because it has the smallest error.
Coefficients for the aging model LAM\(^0\) are given in Table 5.5. The coefficients from the aging model LAM\(^0\) are plotted (dotted lines with symbols) in Figures 5.20-5.26. The identified averaged parameters from model DMT\(^0\) were plotted using solid lines.

Figure 5.20 shows the averaged values from the identified parameter \(k_{E0T0}\) of the model DMT\(^0\) for each of the three maximum charge voltages as solid lines. As discussed in Section 5.2, these parameters fluctuate, but over the life of the cell they show a general increase in value. The linear aging model LAM\(^0\), shown as the dotted lines, follows the general trend of the parameter \(k_{E0T0}\) for each of the three MCVs.

Figure 5.21 shows the parameter \(k_{R\Omega T0}\) for the aging model LAM\(^0\) along with the averaged values for each of the three MCVs from the model DMT\(^0\). Since the values of \(k_{R\Omega T0}\) from the model DMT\(^0\) were already a linear approximation for each of the nine models, the aging model for this parameter follows this model DMT\(^0\) with only a slight deviation due to minor differences in the nine cells.

Figures 5.22-5.26 show the other five parameters for the aging model LAM\(^0\). As each of the figures show, the linear aging model follows the averaged values from the model DMT\(^0\) fairly closely. Due to the built-in correlation of the MCV from Equation 5.6, the aging model actually cancels out the variations of the parameters caused by the differences in the cells, the charging MCVs, and the identification process, making the coefficients from the aging model consistent for all MCVs throughout the life of the cells.

Sample cycle plots of the cell voltages using the aging model LAM\(^0\) are shown in Figures 5.27-5.35. The blue line in these figures is the actual cell voltage of the cycle. The dashed red line shows the simulation of the aging model. The dashed green line shows the aging model voltage shifted by the difference between the actual cell voltage and the predicted value of the aging model voltage at the start of the cycle. The RMS
error between the voltage from the simulated aging model voltage and the actual voltage is given in the lower left-hand corner of each plot. As the plots show, the aging model matches the actual data closely, with the largest difference occurring during the discharge phase of the cells due to the large temperature changes.

Figure 5.36 shows a plot of the RMS error for each cycle of each cell for the aging model LAM$. The solid lines in Figure 5.36 show the error of the aging model simulation. The dotted lines show the error after shifting the aging model so that both the model and the actual data have the same starting voltage. As the plot shows, using the known starting voltage to shift the simulated cell voltages up or down provides a slight reduction in the error between the simulated and actual voltages.

Table 5.5: Coefficients for the aging model LAM$.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>$k_{i1}$</th>
<th>$k_{i2}$</th>
<th>$k_{i3}$</th>
<th>$k_{i4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{E0T0}$</td>
<td>7.1850e+00</td>
<td>-8.4542e-01</td>
<td>3.2799e+00</td>
<td>-7.2582e-01</td>
</tr>
<tr>
<td>$K_{R0T0}$</td>
<td>6.2993e-02</td>
<td>1.4186e-02</td>
<td>-5.0649e-01</td>
<td>1.3055e-01</td>
</tr>
<tr>
<td>$k_{R0T0}$</td>
<td>-4.1199e-01</td>
<td>9.4180e-02</td>
<td>-1.5634e+00</td>
<td>3.9405e-01</td>
</tr>
<tr>
<td>$k_{W0T0}$</td>
<td>1.2251e+01</td>
<td>-2.6978e+00</td>
<td>5.7338e+00</td>
<td>-1.4789e+00</td>
</tr>
<tr>
<td>$k_{W0T0}$</td>
<td>3.0585e-04</td>
<td>-5.7396e-05</td>
<td>-2.8057e-03</td>
<td>6.6606e-04</td>
</tr>
<tr>
<td>$k_{WNET0}$</td>
<td>9.3034e-01</td>
<td>-2.2877e-01</td>
<td>1.6211e+00</td>
<td>-3.6390e-01</td>
</tr>
<tr>
<td>$k_{WNET0}$</td>
<td>1.5897e+00</td>
<td>-3.8242e-01</td>
<td>-2.6797e+00</td>
<td>6.4211e-01</td>
</tr>
</tbody>
</table>

Figure 5.20: Averaged parameter $k_{E0T0}$ for model DMT$ shown with aging model LAM$.
Figure 5.21: Averaged parameter $k_{R\Omega T_0}$ for model DMT$^0$ shown with aging model LAM$^0$.

Figure 5.22: Averaged parameter $k_{liT_0}$ for model DMT$^0$ shown with aging model LAM$^0$.

Figure 5.23: Averaged parameter $k_{qdT_0}$ for model DMT$^0$ shown with aging model LAM$^0$. 
Figure 5.24: Averaged parameter $k_{qT0}$ for model DMT$^0$ shown with aging model LAM$^0$.

Figure 5.25: Averaged parameter $k_{qNET0}$ for model DMT$^0$ shown with aging model LAM$^0$.

Figure 5.26: Averaged parameter $k_{qNFT0}$ for model DMT$^0$ shown with aging model LAM$^0$. 
Figure 5.27: Sample cycles for cell A showing aging model LAM$^0$.

Figure 5.28: Sample cycles for cell B showing aging model LAM$^0$. 
Figure 5.29: Sample cycles for cell C showing aging model LAM⁰.

Figure 5.30: Sample cycles for cell D showing aging model LAM⁰.
Figure 5.31: Sample cycles for cell E showing aging model LAM\(^0\).

Figure 5.32: Sample cycles for cell F showing aging model LAM\(^0\).
Figure 5.33: Sample cycles for cell G showing aging model LAM$^0$.

Figure 5.34: Sample cycles for cell H showing aging model LAM$^0$. 
Figure 5.35: Sample cycles for cell I showing aging model LAM$^0$.

Figure 5.36: Errors between aging model LAM$^0$ and actual voltages for all cycles.
The values for the parameters of the aging model LAM^{m8} are given in Table 5.6. Using the values from Table 5.6, the calculated parameters from the aging model LAM^{m8} are plotted (dotted lines with symbols) in Figures 5.37-5.45 along with the identified averaged parameters from the modified model DMT^{m8} from Table 4.2. Figure 5.37 shows the averaged values from the identified parameter $k_{E0T0}$ for the modified model DMT^{m8} discussed above. Since this parameter has no first-order temperature corrected parameter associated with it, the values for this parameter agree closely with the values from the base model DMT^{0} as shown in Figure 5.20. The parameter $k_{E0T0}$ from the aging model LAM^{m8} is plotted in Figure 5.37 as the dotted lines. The values for the aging model also agree closely with the aging model values from Figure 5.20.

Table 5.6: Coefficients for the aging model LAM^{m8}.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>$k_{i1}$</th>
<th>$k_{i2}$</th>
<th>$k_{i3}$</th>
<th>$k_{i4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{E0T0}$</td>
<td>7.0086e+00</td>
<td>-8.0072e-01</td>
<td>9.5309e+00</td>
<td>-2.2021e+00</td>
</tr>
<tr>
<td>$k_{R0T0}$</td>
<td>6.3288e-02</td>
<td>1.4118e-02</td>
<td>-5.0596e-01</td>
<td>1.3043e-01</td>
</tr>
<tr>
<td>$k_{0T0}$</td>
<td>-7.3489e+00</td>
<td>1.7919e+00</td>
<td>1.0462e+01</td>
<td>-2.4582e+00</td>
</tr>
<tr>
<td>$k_{a0T0}$</td>
<td>1.2102e+02</td>
<td>-2.8888e+01</td>
<td>-2.1997e+02</td>
<td>5.2889e+01</td>
</tr>
<tr>
<td>$k_{t0T0}$</td>
<td>5.8869e+04</td>
<td>-1.2137e+04</td>
<td>-4.6905e+03</td>
<td>1.1059e-03</td>
</tr>
<tr>
<td>$k_{qNET0}$</td>
<td>8.1297e-01</td>
<td>-2.0146e-01</td>
<td>4.3742e+00</td>
<td>-1.0111e+00</td>
</tr>
<tr>
<td>$k_{qNFT0}$</td>
<td>1.6764e+00</td>
<td>-4.0063e+01</td>
<td>3.4834e+00</td>
<td>8.3565e-01</td>
</tr>
<tr>
<td>$k_{T1}$</td>
<td>2.2713e-01</td>
<td>-5.6009e-02</td>
<td>-3.4465e-01</td>
<td>8.1279e-02</td>
</tr>
<tr>
<td>$k_{qT1}$</td>
<td>-3.5720e+00</td>
<td>8.6461e-01</td>
<td>6.4640e+00</td>
<td>-1.5568e+00</td>
</tr>
</tbody>
</table>

Figure 5.38 shows the plot for the parameter $k_{R0T0}$ from the aging model LAM^{m8}. Since the resistance was determined separately from the other parameters, this plot is identical to the plot for the parameter $k_{R0T0}$ from the aging model LAM^{0} shown in Figure 5.21.

Figures 5.39 and 5.40 show the plots of the parameters $k_{T0}$ and $k_{T1}$ for aging model LAM^{m8}. As discussed in Section 4.5, the correlation between these two parameters causes the identification process to generate larger magnitude values that tend
to oppose each other. Similarly, the parameters $k_{qdT0}$ and $k_{qdT1}$ shown in Figures 5.41 and 5.42 also tend to oppose each other, yet they combine to reduce the overall error.

Figures 5.43-5.45 show plots of the last three parameters for the aging model LAM$^{m8}$. Since these parameters do not have associated first-order temperature correction terms, they follow their respective parameters from the aging model LAM$^0$ shown in Figures 5.24-5.26.

Sample cycle plots of the cell voltages using the aging model LAM$^{m8}$ are shown in Figures 5.46-5.54. The blue line in these figures is the actual cell voltage of the cycle. The dashed red line shows the simulation of the aging model. The dashed green line shows the aging model shifted by the difference between the actual cell voltage and the estimated value of the aging model at the start of the cycle. Simulations of the aging model LAM$^{m8}$ show a small dip in voltage at the beginning of the discharge phase of the cycles caused by the two first-order temperature terms contained in this model. Error plots for the aging model LAM$^{m8}$ for all the cells throughout their lives are shown in Figure 5.55. The next section of this chapter provides some suggestions on how to apply the aging model.

Figure 5.37: Averaged parameter $k_{E0T0}$ for model DMT$^{m8}$ shown with aging model LAM$^{m8}$.  

176
Figure 5.38: Averaged parameter $k_{R\Omega T0}$ for model DMT$^{m8}$ shown with aging model LAM$^{m8}$.

Figure 5.39: Averaged parameter $k_{liT0}$ for model DMT$^{m8}$ shown with aging model LAM$^{m8}$.

Figure 5.40: Averaged parameter $k_{liT1}$ for model DMT$^{m8}$ shown with aging model LAM$^{m8}$.
Figure 5.41: Averaged parameter $k_{qdT0}$ for model DMT\textsuperscript{m8} shown with aging model LAM\textsuperscript{m8}.

Figure 5.42: Averaged parameter $k_{qdT1}$ for model DMT\textsuperscript{m8} shown with aging model LAM\textsuperscript{m8}.

Figure 5.43: Averaged parameter $k_{qsT0}$ for model DMT\textsuperscript{m8} shown with aging model LAM\textsuperscript{m8}.
Figure 5.44: Averaged parameter \( k_{q_{0}NE70} \) for model DMT\(^{m8} \) shown with aging model LAM\(^{m8} \).

Figure 5.45: Averaged parameter \( k_{q_{0}NFT0} \) for model DMT\(^{m8} \) shown with aging model LAM\(^{m8} \).
Figure 5.46: Sample cycles for cell A showing aging model LAM$^{mk}$.

Figure 5.47: Sample cycles for cell B showing aging model LAM$^{mk}$.
Figure 5.48: Sample cycles for cell C showing aging model LAM\textsuperscript{m8}.

Figure 5.49: Sample cycles for cell D showing aging model LAM\textsuperscript{m8}.
Figure 5.50: Sample cycles for cell E showing aging model LAM$^{m8}$.

Figure 5.51: Sample cycles for cell F showing aging model LAM$^{m8}$. 

182
Figure 5.52: Sample cycles for cell G showing aging model LAM\textsuperscript{MK}.

Figure 5.53: Sample cycles for cell H showing aging model LAM\textsuperscript{MK}.
Figure 5.54: Sample cycles for cell I showing aging model LAM\textsuperscript{m8}.

Figure 5.55: Errors between aging model LAM\textsuperscript{m8} and actual voltages for all cycles.
5.4 Application of the Aging Model

This section provides some suggestions on how to apply the aging model presented in the previous section. The aging model LAM$^0$ for the dynamic model DMT$^0$ produced good results. This model was also the simplest and most direct. If the cells are going to be used at room temperature and lower currents, the DMT$^0$ model should be used. If the cells are going to be used at temperatures above room temperature, or at high currents that cause the cells to rapidly change temperatures, the aging model LAM$^m8$ should be used.

Since the capacity of the cell is predicted, this predicted value should be scaled by 110% to prevent the log terms from going negative. A check should also be performed to verify that the value of $q_s(t)$ does not go negative due a low predicted value of capacity, which would cause the $\ln(q_s/Q)$ term to generate complex numbers. If $q_s(t)$ is approaching zero, the logarithm term should be forced to zero.

Predictions of a cell’s capacity can be generated by first determining the total accumulated life fraction using Equation 5.3. Then the capacity of the cell is predicted using the quadratic life fraction model given as Equation 5.7.

Given the cell current, the charge vectors $q_s(t)$ and $q_d(t)$ can be predicted using Equations 2.73 and 2.74, respectively. The voltage of the cell can then be predicted using the appropriate voltage equation from the chosen aging model.

5.5 Chapter Summary

This chapter compared the aging results from the nine cells with results from the literature. This chapter also developed an aging model using both the life fraction and MCV of the cells. The aging model was used to calculate the coefficients for the
dynamic model built on first principles from Chapter II. The linear aging model fit the
dynamic model coefficients with small error, with the exception of capacity which used
the quadratic aging model. It is recommended that the selected linear aging model be
used to calculate the coefficients of the voltage equation for the dynamic cell model. The
quadratic aging model should be used to generate the estimated capacity of the cell.
Two aging models were selected for use. The first aging model ignores temperature
effects of the cell. The second model adds first-order temperature correction terms to two
of the voltage equation parameters. Simulations of the selected dynamic models built
from their corresponding linear aging models tracked the original cycle data with small
error. Chapter VI presents closing remarks and suggests some areas for future work.
CHAPTER VI

CONCLUSION

Nine lithium-ion cells were cycled until they reached their end of life. Data collected during cycling was used to identify parameters in a first principles dynamic model of the cells. The cells used in the aging study performed well. Cells charged with the same MCV had consistent results. Cells charged to higher MCVs had larger capacities but shorter life spans as expected.

Generally, cell capacity remained relatively constant during the first 40% of the cell’s life, before decreasing at a linear rate until reaching approximately 50% of its rated capacity, when the cell was declared to have reached its end of life. A direct correlation was found between the cell capacity and the open-circuit voltage of a fully discharged cell. Cell resistance increased at a linear rate throughout the life of the cells.

The dynamic models chosen for identification fit the collected data closely. Models containing temperature correction terms produced smaller error than the model without any temperature correction. The correlation between temperature correction terms and their corresponding non-temperature dependent terms caused inconsistencies in the model parameters. When the temperature correction terms were applied individually, the polarization term was found to provide the most reduction in the overall error, closely followed by the diffusion term.
An aging model was presented that generated the coefficients for the dynamic model equations using life fraction and maximum charging voltage of the cells. The aging model applied to the dynamic model that included temperature corrections terms for the polarization and diffusion terms provides the coefficients that generated the smallest voltage. The aging model applied to the dynamic model with no temperature correction also produced good results with less complexity.

Unlike aging models that determine the internal properties of a cell directly, the aging model presented here can be readily used in a battery management system. Incorporation of the aging model into a battery management system will allow the BMS to better track capacity and remaining life. With this knowledge of the changing cell parameters and life fraction, the BMS can potentially extend the life of the battery pack.

This paper has presented a methodology to develop an aging model for lithium-ion cells. This same methodology can be applied to cells of other chemistries. This methodology consists of data collection during cell cycling. Collected data is then used to perform parameter estimation of the first principles dynamic model throughout the life of the cells. These parameter estimates are then used to create the cell aging model that can be used to accurately predict the behavior of a cell as it ages. The resulting aging model can be used to predict a cell’s behavior in specific applications.

There are several areas of this work that could be expanded. Future work could consist of performing an aging study on the same type of cells, but with lower maximum charging voltages and at different charging and discharging rates to compare with the aging models presented here. Other work could include experiments showing how the presented aging model performs on other types of cell chemistry. A third way of
expanding this work would be to implement the aging model into a battery management system and determine the accuracy of the results when used in a specific application.
REFERENCES


Hansen, Terry, and Chia-Jiu Wang. “Support Vector Based Battery State of Charge

Hartley, Tom T., and Carl F. Lorenzo. “Optimal Battery Charging for Damage

Batteries.” American Institute of Aeronautics and Astronautics’ 3rd International
CA.

International Union of Pure and Applied Chemistry, Physical Chemistry Division
Commission on Electrochemistry. “Impedances of Electrochemical Systems:
Terminology, Nomenclature and Presentation, Part I: Cells with Metal
Electrodes and Liquid Solutions.” Pure and Applied Chemistry. 66.9 (1994):
1831-1891.

Kobayashi, Hironori, Masahiro Shikano, Shinji Koike, Hikari Sakaibe, and Kuniaki
Tatsumi. “Investigation of Positive Electrodes After Cycling Testing of High-
Power Li-ion Battery Cells: I. An Approach to the Power Fading Mechanism

Landau, Ioan, and Gianluca Zito. Digital Control Systems: Design, Identification and
1-84628-055-9. Rev. Trans. of Commande des systèmes: conception,

07-037921-1.


MicroEngineering Labs, Inc. PICBASIC PRO™ Compiler. Colorado Springs, Colorado:
microEngineering Labs, Inc. 2005

National Semiconductor Corporation. LM 61 2.7V, SOT-23 or TO-92 Temperature

Ning, Gang, Bala Haran, and Branko N. Popov. “Capacity Fade Study of Lithiumion
Batteries Cycled at High Discharge Rates.” Journal of Power Sources. 117

Ning, Gang, Ralph E. White, and Branko N. Popov. “A Generalized Cycle Life Model of


APPENDICES
APPENDIX A

BASIC AND PERL PROGRAM LISTINGS

This appendix contains the listing of programs used to collect cycle data from the PIC microcontroller, and PERL programs used to preprocess the data for loading into MATLAB. M-files for processing data inside of MATLAB are included in Appendix B.

A.1 Cycle.bas

Cycle.bas is the PICBASIC Pro™ program used to control the charging circuits. Reference values in this program were determined by applying the reference voltages and currents to the circuit and recording the bit values produced by the PIC. The measured reference values for the individual cells are listed in Table 3.4.

```
' Charge/discharge program
' Circuit has 4 LED's
'     solid green - red off - charging - constant current
'     blink green - red off - charging - constant voltage
'     solid green - red blinking - rest after charge
'     solid red - green off - discharging - constant voltage
'     solid red - green blinking - rest after discharge
'     yellow blinking - communication failure
'     square red - blinks for 1 second clock
' Sat Jun 16 10:54:28 EDT 2007
'Thu Aug 30 19:43:23 EDT 2007 - reduced initial current to 100
'Set data in EEPROM
'start at location 0, cyclecount, cell name, status, Vmax, Vmin, Imin
'DATA word 0, byte "A",byte "l", word 420, word 275, word 12

' Define variables
comma       var byte    ' comma for serial output formatting
cyclecnt    var word    ' Used to count cycles
modecnt     var word    ' Used to display how long in this mode
secondcnt   var word    ' Used to count seconds
```
limitcnt var word ' Used to count how many times reached
limit
writecnt var word ' Used to count number of trys to write
to flash
inputbuffer var byte[4] ' buffer serial input
cellname var byte ' Cell names A-J, store in flash
curstatus var byte ' Status of cycle: c - charge CC
'                   C - charge CV
'                   r - rest after charge
'                   d - discharge CC
'                   R - rest after discharge
'                   i - in initialization state
'                   H - Cell Temp. too high
prevstatus var byte ' Previous status used in initialization
phase
restcnt var word ' Count seconds for 10 minute rest period
(600 seconds)
timeoutcnt var word ' Count how many times communication has
failed in a row
Vcell var word ' cell voltage, in binary
Vcharge var word ' Voltage of charging resistor (1 ohm)
Vdisch var word ' Voltage of discharging resistor (0.2
ohm)
Vmax var word ' max cell voltage, in binary
Vmin var word ' min cell voltage, in binary
Vref var word ' Reference value
Vdd var word ' Power reference value
Ich max var word ' max cell charge current (CC), in binary
Ich max var word ' Max cell charge current (CC), in binary
Ichref var word ' Constant current charging
Idisch max var word ' Constant current discharging
Idisch max var word ' Max cell discharge current, in binary
Temp var word ' cell temperature, in binary
Temp var word ' cell temperature, in binary
chargeDC var word ' duty cycle of first PWM, charging
dischDC var word ' duty cycle of second PWM, discharging
adjustDC var word ' Seems to be changing to fast, so divide
by 8
waitcnt var word ' wait
Vcell12 var word ' Second value, to take average
Icharg e2 var word ' Second value, to take average
Idisch2 var word ' Second value, to take average
Temp2 var word ' Second value, to take average
Vcell13 var word ' Third value, to take average
Icharge3 var word ' Third value, to take average
Idisch3 var word ' Third value, to take average
Temp3 var word ' Third value, to take average
Vcell14 var word ' Fourth value, to take average
Icharge4 var word ' Fourth value, to take average
Idisch4 var word ' Fourth value, to take average
Temp4 var word ' Fourth value, to take average
' input variables
ADtemp var word ' A/D value of temp sensor
ADcell var word ' A/D value of cell voltage
ADcharge var word ' A/D value of charge resistor
ADref var word ' A/D value of reference LM336
ADdisch var word ' A/D value of charge resistor
fromPC var PORTC.5 ' Serial in from PC, receive
PCstatus var word ' Store input from PC
' output variables
chargePWM var word ' Charging PWM output
dischPWM var word ' Discharging PWM output
toPC var PORTC.4 ' Serial out to PC, transmit

' other I/O variables
startbtn var PORTA.4 ' Switch to start program
sqredLED var PORTB.4 ' square red - 1 second
redLED var PORTB.5 ' round red - discharge
greenLED var PORTB.6 ' round green - charge
yellowLED var PORTB.7 ' round yellow - comm fail
FcycleB0 var word ' Cycle number byte 0
FcycleB1 var word ' Cycle number byte 1
Fcellname var word ' Cell name 1 byte, byte 0
Fstatus var word ' Status value 1 byte, byte 0

' Initialize variables

' DEFINE hardware PWM settings
DEFINE CCP1_REG PORTC
DEFINE CCP1_BIT 2
DEFINE CCP2_REG PORTC
DEFINE CCP2_BIT 1

' DEFINE A/D settings
DEFINE ADC_BITS 10
DEFINE ADC_CLOCK 3
DEFINE ADC_SAMPUS 50
TRISA = %00111111 ' PORTA all input

' ADCON1 =0 ' Set PORTA analog
ADCON1.0=1; ' First four bits of A/D reg1 = 0001 turns all A/D on, ch3 is Vref+
ADCON1.1=0; ' All zeros turns on all A/D with Vdd as reference
ADCON1.2=0;
ADCON1.3=0;
ADCON1.7=1; ' A/D result right justified PIC16F876A.pdf page 130

Vcell2=0
Icharge2=0
Idisch2=0
Temp2=0
Vcell3=0
Icharge3=0
Idisch3=0
Temp3=0
Vcell4=0
Icharge4=0
Idisch4=0
Temp4=0
comma =","
cyclecnt =0
secondcnt =0
restcnt =0
timeoutcnt =0
waitcnt =0
limitcnt =0
modecnt =0

197
Flash addresses

FcycleB0 = 0     ' Cycle number byte 0
FcycleB1 = 1     ' Cycle number byte 1
Fcellname = 2     ' Cell name 1 byte, byte 0
Fstatus = 3     ' Status value 1 byte, byte 0

PWM init
chargePWM = 1  ' Charging PWM channel
dischPWM = 2  ' Discharging PWM channel
chargeDC = 0  ' see CHARGECC
dischDC = 500  ' see DISCHARGE
adjustDC = 0  ' see DISCHARGE
HPWM chargePWM, 0, 1000  ' Turn off PWMs
HPWM dischPWM, 0, 1000  ' Turn off PWMs

Read data from flash
read FcycleB0, cyclecnt.BYTE0
read FcycleB1, cyclecnt.BYTE1
read Fcellname, cellname
read Fstatus, prevstatus

Reference variables
Vdd is 5.2 volts
Vref is 2.5 V, 1023 or 10 bits, to minimize calculations in the program, all references are scaled for direct comparison of 10 bits
Set references for max/min values
Vdd = 1023*52  ' 5.2 Volts from power supply
Voltages measured, and values recorded here
Cell A on DVO board
Tmax = 440   ' 45 oC, 0 oC = 600 mV, +10 mV, 1.05 V, 60 oC for discharge
Vmax = 1719  ' Maximum charge voltage 4.2 V
Vmin = 1142  ' Minimum discharge voltage 2.75 V
Ichref = 974  ' 1C, 2400 milliamps ' Max (constant) charge current (Cell A, 1.00 ohm resistor)
Idisref = 2040  ' Constant discharge current, 2 C, 0.974 V across 0.203 Ohm resistor
Imin = 117  ' Minimum charge current, 5% of Ichref, 120 mA, 0.120 V (1 ohm)
Ichmax = Ichref ' Set max value

At poweron, turn all LED's on
pause to allow crystals to setup
and temperature to equalize to prevent going to TOHOT
High yellowLED
pause 2000
High greenLED
pause 2000
High redLED
pause 2000
High sgredLED
pause 2000
Low yellowLED ' Turn off
Low greenLED ' Turn off

Initialization, stay here until input received
INIT:
HPWM chargePWM, 0, 1000  ' Turn off PWMs
HPWM dischPWM, 0, 1000  ' Turn off PWMs
curstatus = "I";

' read/send data
' Leave init, when communications is received from PC

'Toggle LED's to show in INIT
if (redLED) THEN' At poweron, turn all LED's on
  Low redLED
  Low greenLED
else
  High redLED
  High greenLED
endif
SEROUT2 toPC, 16572, [cellname, curstatus, comma, dec Vmax, comma, dec cyclecnt, comma, dec secondcnt, comma, dec Temp, comma, dec Vcell, comma, dec Icharge, comma, dec Idisch, comma, dec Vref, comma, inputbuffer, comma, dec modecnt, comma, dec chargeDC, comma, dec dischDC, comma, dec adjustDC, comma, dec ADcharge, comma, dec ADdisch, 10]
  secondcnt = secondcnt + 1
SERIN2 fromPC, 16572, 300, INIT, [inputbuffer]
curstatus = prevstatus ' Set the status

CYCLESELECT:

' Collect data
gosub collectdata

' Check Temperature, if HOT, set status = H
if curstatus == "d" then
  Tmax = 503 ' 60 oC for discharge
else
  Tmax = 440 ' 45 oC for charge
endif
if Temp > Tmax then
  curstatus = "H"
  restcnt = 0 ' Keep at zero until cooled off
endif

' Adjust PWM duty cycles
select case curstatus
  case "H"
    gosub TOHOT
  case "c"
    gosub CHARGECC
  case "C"
    gosub CHARGECV
  case "r"
    gosub CHARGEREST
  case "d"
    gosub DISCHARGE
  case "R"
    gosub DISCHARGEREST
  case else
    ' Then start charging
    gosub CHARGECC
  end select
  'ENDIF
end select

' Advance second counter
secondcnt = secondcnt + 1
modecnt = modecnt + 1 ' Advance the mode counter

'send data to PC for logging
SEROUT2 toPC, 16572, [cellname, curstatus, comma, dec Vmax, comma, dec cyclecnt, comma, dec modecnt, comma, dec Temp, comma, dec Vcell, comma, dec Icharge, comma, dec Idisch
comma, dec Vref, comma, inputbuffer, comma, dec modecnt, comma, dec chargeld, comma, dec discld, comma, dec
ADcharge, comma, dec ADisch, 10]
' Verify data was received
SERIN2 fromPC, 16572, 300, COMMMFAIL, [inputbuffer]
  Low yellowLED ' Turn low, comm worked

' Wait for one second to end
' Time was measured using an oscilloscope
for waitcnt = 1 to 790
  pause 1 ' wait 1 millisecond
next
Toggle spreadLED
goto CYCLESELECT

TOHOT:
  ' If cell temperature is above rated value,
  ' wait 5 minutes to allow cooling, and blink
  ' lights, reset
  HPWM chargePWM, 0, 1000 ' Turn off PWMs
  HPWM dischPWM, 0, 1000 ' Turn off PWMs
  ' 'Alternate LED's to show in hot
if (redLED) THEN
  Low redLED
  High greenLED
else
  High redLED
  Low greenLED
endif
restcnt = restcnt + 1
if restcnt > 300 then
  ' Go back to previous mode
  curstatus = prevstatus
endif
RETURN

' Enter charging
CHARGECC: ' Charge Constant Current
  ' If the first time charging, charge at 1/4 C, 0.6 A.
  ' Assume the cell is at Vmin, 2.75 V, then need
  ' 5.85V at base of Q1 to start charging, at 2.4 A.
  ' Gain of U1 is 4, so input voltage needs to be
  ' 1.4625 V.  PWM duty cycle range is 0 to 255 where
  ' 255 is Vdd or 5.2 V.  Given 255 = 5.2, X = 1.4625,
  ' gives X to be 71.72.  Let the current ramp up, so
  ' start PWM at 50 DC (duty cycle).
  ' Adjusting by 1/4 steps to prevent changing too quickly
if cyclecnt < 1 then
  ' First charge, max current 1.2 A, 1/2 C
  Ichmax = (Ichref >> 1) ' First charge is 1.2 A
else
  ' Beyond first charge, charge at 1 C.
  Ichmax = Ichref
endif
' Adjust current
if Icharge < Ichmax then
  chargeDC = chargeDC + 1
endif
if (Icharge > Ichmax) then
    chargeDC = chargeDC - 1
endif

dischDC = 0
adjustDC = chargeDC >> 2 ' divide by 4
HPWM chargePWM, adjustDC, 1000
HPWM dischPWM, 0, 1000 ' Set curstatus
curstatus = "c";

if (prevstatus != curstatus) THEN
    curstatus = "C";
    Toggle greenLED
endif

if (Vcell > (Vmax * 2)) then
    chargeDC = chargeDC - 20 ' Voltage way too high
endif
if (Vcell > Vmax) then
    chargeDC = chargeDC - 1
endif

if (Icharge > Ichmax) then
    chargeDC = chargeDC - 1
endif

dischDC = 0
adjustDC = chargeDC >> 2 ' divide by 4
HPWM chargePWM, adjustDC, 1000
HPWM dischPWM, 0, 1000 ' Set curstatus
curstatus = "c";

if (prevstatus != curstatus) THEN
    prevstatus = curstatus
    cyclecnt = cyclecnt + 1 ' Advance cycle
    write FcycleB0, cyclecnt.BYTE0
    write FcycleB1, cyclecnt.BYTE1
endif

if Vcell >= Vmax then
    limitcnt = limitcnt + 1
endif
if (limitcnt >= 5) then
    limitcnt = 0
    gosub CHARGECV
endif

' Do not charge longer than 2.4 hours at CV
if modecnt > 9000 then
    limitcnt = 0
    restcnt = 0     ' Reset rest counter (10 min, 600 sec)
    gosub CHARGEREST
endif

return

CHARGECV:    ' Charge Constant Voltage
if (Vcell < Vmax) then ' Keep at max voltage, but not above
    if (Icharge < Ichmax) then ' Don't let current above ref
        chargeDC = chargeDC + 1
    endif
endif
if (Vcell > (Vmax * 2)) then
    chargeDC = chargeDC - 20 ' Voltage way too high
endif
if (Vcell > Vmax) then
    chargeDC = chargeDC - 1
endif

' Check to ensure current not too large
if (Icharge > Ichmax) then
    chargeDC = chargeDC - 1
endif

return
Low redLED
  gosub writestatus
  prevstatus=curstatus
endif

' Go to CHARGEREST if current drops below Min
if Icharge < Imin then
  limitcnt=limitcnt+1
endif
if Icharge <= 5 then
  limitcnt = 0
  restcnt = 0  'Reset rest counter (10 min, 600 sec)
  gosub CHARGEREST
endif
'
Do not charge longer than 2.4 hours at CV
if modecnt > 9000 then
  limitcnt = 0
  restcnt = 0  'Reset rest counter (10 min, 600 sec)
  gosub CHARGEREST
endif
return

CHARGEREST:  ' Rest after charging
chargeDC =0
dischDC =0
HPWM chargePWM, 0, 1000
HPWM dischPWM, 0, 1000
  ' Set curstatus
  curstatus="r";
High greenLED
Toggle redLED
  if(prevstatus != curstatus) THEN
    modecnt=0
    gosub writestatus
    prevstatus=curstatus
    restcnt =0;
  endif
  restcnt=restcnt+1
  if(restcnt > 600) THEN
    gosub DISCHARGE  ' Go to DISCHARGE
  endif
return

DISCHARGE:  ' Discharge Constant Current
  ' Discharge is at 2 C, 4.8 A, measured
  ' from 0.2 ohm resistor which should
  ' have a 0.96 V drop. Base of Q2 needs
  ' to be 1.66 V, duty cycle of PWM
  ' should be 81.4, but start at 50
  ' and ramp up.
  ' Adjusting by 1/4 steps to prevent changing too quickly
  if Idisch < Idisref then
    dischDC = dischDC + 1  ' Increase the current
  endif
  if Idisch > Idisref then
    dischDC = dischDC - 1  ' Decrease the current
  endif
chargeDC =0
adjustDC = dischDC >>2  ' divide by 4
HPWM chargePWM, 0, 1000
HPWM dischPWM, adjustDC, 1000
  ' Set curstatus
  curstatus="d";
High redLED
if(prevstatus != curstatus) THEN
modecnt = 0
  dischDC = 500  ' Starting value, see DISCHARGE
  Low greenLED
gosub writestatus
  prevstatus = curstatus
endif
  ' Go to DISCHARGEREST if current drops below Min
if Vcell < Vmin then
  limitcnt = limitcnt + 1
endif
  if limitcnt >= 5 then
    limitcnt = 0
    restcnt = 0    'Reset rest counter (10 min, 600 sec)
gosub DISCHARGEREST
  endif
return

DISCHARGEREST:  ' Rest after charging
chargeDC = 0
dischDC = 0
  HPWM chargePWM, 0, 1000
  HPWM dischPWM, 0, 1000
  ' Set curstatus
curstatus = "R";
  High redLED
  Toggle greenLED
  if(prevstatus != curstatus) THEN
    modecnt = 0
  endif
  restcnt = restcnt + 1
  if(restcnt > 600) THEN
    gosub CHARGECC   ' Go to charge CC
  endif
return

  ' Timeout waiting for response back from PC
COMMFAIL:
  timeoutcnt = timeoutcnt + 2  ' communication failed, advance the counter
  High yellowLED
  if timeoutcnt > 5 then
    timeoutcnt = 0
    goto INIT   ' Five failures, go to init
  endif
  goto CYCLESELECT
End

writestatus:
  ' Write status, read to make sure it worked, if not, try again
WRITEDATA:
  ' try to write without masking interrupts first, works 99% of time
  write Fstatus, curstatus
  read Fstatus, prevstatus
if(prevstatus!=curstatus) THEN
  INTCON.7=0  ' Set mask and try to write again
  write Fstatus,curstatus
  INTCON.7=1
endif
if(writecnt >5) THEN
  writecnt=0;
  return
endif
' keep trying
read  Fstatus,prevstatus
if(prevstatus!=curstatus) THEN
  goto WRITEDATA
endif
writecnt=0;
return

collectdata:
  ' Save old values for reference
  Vcell4=Vcell3
  Icharge4=Icharge3
  Idisch4=Idisch3
  Temp4=Temp3
  Vcell3=Vcell2
  Icharge3=Icharge2
  Idisch3=Idisch2
  Temp3=Temp2
  Vcell2=Vcell1
  Icharge2=Icharge1
  Idisch2=Idisch1
  Temp2=Temp1
  ' Measure Temperature and discharge voltage
  ' using 2.5V ref for more accuracy, of small values
  ADCON1.0=1;  ' turns on ch3 as Vref+
adcin 0, ADtemp  ' pin 2
adcin 4, ADdisch  ' pin 7
adcin 1, ADcell  ' pin 3
adcin 2, ADcharge  ' pin 4
  ' Measure other voltages using 5.2V Vdd ref
  ' because voltages above the 2.5V ref
  ADCON1.0=0;  ' turns on ch3 as Vref+
adcin 3, ADref  ' pin 5 'Record the AD ref, versus 5.22V
  Temperature reading LM61  V=0.600 + Temp * 0.01 V/oC
  Temp=ADtemp
  ' Calculate voltages
  Vcell=ADcell * 2 ;  ' Was divided by 2
  Vref = ADref
  ' Calculate currents
  Vcharge=ADcharge * 3  ' Was divided by 3
  if Vcharge <= Vcell then
    Icharge = 0  ' No current flowing
  else
    Icharge=Vcharge-Vcell  ' 1 ohm resistor
  endif
  'Discharge
  Vdisch =ADdisch * 2  ' Use discharge voltage as reference
  Idisch =Vdisch * 5  ' 0.2 ohm resistor
Calculate average, >>2 divides by 4
Vcell = (Vcell + Vcell2 + Vcell3 + Vcell4) >> 2
Icharge = (Icharge + Icharge2 + Icharge3 + Icharge4) >> 2
Idisch = (Idisch + Idisch2 + Idisch3 + Idisch4) >> 2
Temp = (Temp + Temp2 + Temp3 + Temp4) >> 2

return
' End of Cycle.bas program

A.2 Logger.pl

Logger.pl is the PERL script used to add a timestamp and to record the data sent from the PIC into a file on the hard drive of the attached PC.

#!/usr/bin/perl
use Device::SerialPort;
$ARGC = @ARGV;  # get # of arguments
print ("arg count $ARGC \n");
if ($ARGC < 1) { die "USAGE: ./logger.pl ttyS#"; }

$PORTNAME = "$ARGV[0]";  # port without path
$PORT = "/dev/$ARGV[0]";   # full path of port

print "$PORT\n";
#
#
# Serial Port Settings
#
#
$comport = Device::SerialPort->new ($PORT) || die "Can't Open $PORT: $!";
$comport->baudrate(4800) || die "failed setting baudrate";
$comport->parity("none") || die "failed setting parity";
$comport->databits(8) || die "failed setting databits";
$comport->stopbits(1) || die "failed setting stopbits";
$comport->handshake("none") || die "failed setting handshake";
none, rts, cts
$comport->write_settings || die "no settings";
#
# Send a string to the port
#$pass=$comport->write("OK");
#sleep 1;
#
# open the logfile, and Port
#
#open(LOG, ">>$LOGDIR/$LOGFILE")
# | die "can't open smdr file $LOGDIR/$LOGFILE for append:\n$SUB $!\n";
#select(LOG), $| = 1;  # set nonbufferd mode
open(DEV, "+<$PORT")
| die "Cannot open $PORT: "$;
select(DEV), $| = 1; # set nonbufferd mode

# Loop forever, logging data to the log file
#$sec, $min, $hour, $mday, $mon, $year, $wday, $yday, $isdst) = localtime(time);
$mon++; # Month starts at zero, so add one to it
$year+=1900; # Year starts at 1900
filename = sprintf("%s-04d-%02d-%02d-%02d-
PORTNAME.dat", $ENV{HOSTNAME}, $year, $mon, $mday, $hour);
open(Flog, ">>A{filename}") or die "couldn't open
filename
";
$prevname="previous day's file";

while($data = <DEV>){ # print input device to file
  if($sec%2){ $pass=$comport->write("A\n"); } # Send back A
  else{ $pass=$comport->write("a\n"); } # Send back a for OK
  print Flog "$timestamp${data}";
  print STDOUT "$timestamp${data}";
  # Get current timestamp
  ($sec, $min, $hour, $mday, $mon, $year, $wday, $yday, $isdst) = localtime(time);
  $mon++; # Month starts at zero, so add one to it
  $year+=1900; # Year starts at 1900
  $timestamp = sprintf("%04d-%02d-%02d-%02d:%02d:%02d-
", $year, $mon, $mday, $hour, $min, $sec,);
  # Open new log file for every single day
  filename = sprintf("%s-04d-%02d-%02d-%02d-
PORTNAME.dat", $ENV{HOSTNAME}, $year, $mon, $mday, $hour);
  if($filename !~ /^$prevname$/) {
    close(Flog);
    print stderr "opening file $filename\n";
    open(Flog, ">>A{filename}") || die "can't open log file
filename for append: $SUB $!");
    $prevname = $filename;
  }
}

undef $comport;
# End of Logger.pl program;

A.3 Createdaily.pl

This PERL script reads the data from the program Logger.pl and checks for
missing lines. If a missing line is found, interpolation is used to estimate the missing
data. Valid lines are stored into files using the date the data was collected.
#!/usr/bin/perl
# CreateDaily.pl by rlh2
# Script to convert Hourly files to daily files, to fill in missing seconds, and to convert time into seconds
use Time::Local;

system("touch -t 200707010000 /tmp/julyfirst"); # Create reference
$JULYFIRST=`date --reference=/tmp/julyfirst +%s`;  # Get first of july
# How many days after july first, based on month value, needs shifted by 1
%DAYS AFTER=( '7' => 0, '8' => 31, '9' => 62, '10' => 92, '11' => 123, '12' => 153);

# Modes
%MODES=(I => 0, C => 1, R => 3, d => 4, r => 5, H => 6);
resetmodecount(); # Define mode count
%CYCLECNT=(
   'A' => 0, 'B' => 0, 'C' => 0, 'D' => 0, 'E' => 0, 'F' => 0, 'G' => 0, 'H' => 0, 'I' => 0);
); // Initialize Cycle cnt
%PTIMESECONDS=(
   'A' => 0, 'B' => 0, 'C' => 0, 'D' => 0, 'E' => 0, 'F' => 0, 'G' => 0, 'H' => 0, 'I' => 0);
); // Initialize Cycle cnt

$ODATADIR="/home/rhart/pic/data/origdata"; # Original data files
$DDATADIR="/home/rhart/pic/data/daily"; # Daily output data files
system("mkdir $DDATADIR");
if(!"$ARGV[0]") {die "USAGE: createdaily.pl CELLLETTER\n";}
@CELLS=("$ARGV[0]");
foreach $CELL (@CELLS)
{
   # Initialize variables
   $PSECONDCNT=0; # Previous second count to track missed points
   $DIR="/home/rhart/pic/data/origdata/$CELL/";
   $DDATADIR="/home/rhart/pic/data/daily/$CELL/";
   opendir (FDIR, $DIR) || next;

207
print(STDERR "Opened directory $DIR...
");
foreach $FILE (sort readdir($DIR))
{
    $OUTPUTFILE=substr($FILE,0,17) . ".dat"; # Create output file, for entire day
    if("$FILE" !~ /^$CELL\.*\.dat$/) { next; } # Not a data file, ignore
    print(STDERR "\tOpened file $FILE
");
    if("$FILE" /=^$CELL/ ) { print(STDERR "$DIR/$FILE does not belong to $CELL
");
        open(Fdat, "$DIR/$FILE") || print(STDERR "Could not open
$DIR/$FILE
");
        @LINES=<Fdat>; # read the data into array
        @LINESOUT=(); # empty the output array
        close(Fdat);
        @OUTPUTARRAY=(); # Initialize output array
    foreach $LINE (@LINES)
    {
        chomp $LINE; # remove EOL character
        ($STRING, $MV, $CYCLECNT, $SECONDCNT, $TEMP, $VCELL,
        $ICHARGE, $IDISCH, $CYCLECNT, $SECONDCNT, $TEMP, $VCELL,
        $ICHARGE, $IDISCH, $CYCLECNT, $SECONDCNT, $TEMP, $VCELL,
        $ICHARGE, $IDISCH, $ADJUSTDC,
        $ADCHARGE, $ADDISCH)=split(/,/,$LINE);
        if (!$PTEMP{$CELL}) { # Define the previous values for this cell
            storepreviousvalues();
        } # end if (!$PTEMP)
        ### DEBUGoff print(STDERR "-\$LINE\n");
        $TIMESTAMP=substr($STRING,0,19);
        $CELLVALUE=substr($STRING,20,1);
        $MODE=substr($STRING,21,1);
        $RAWTIME=$TIMESTAMP;
        $RAWTIME=~s/-//g;
        $RAWTIME=~s/://g;
        $TIMESECONDS=toseconds($TIMESTAMP);
        # Check for valid MODE I,c,C,R,d,r,H
        if($PREVMODE =~/r/ && $MODE =~/[cC]/)
        {
            $CYCLECNT{$CELL}++;
            if($MODECNT{r} < 600)
            {
                print(STDERR "Warning, change in cycle count ($CYCLECNT{$CELL}) at $TIMESECONDS\n");
                resetmodecount();
                $MODECNT{$MODE}++;
            } # Define mode count
            else {$MODECNT{$MODE}++;
        } # Check time with previous time to see if repeated/skipped, etc
        if ($TIMESECONDS == $PTIMESECONDS{$CELL} )
        {
            print(STDERR "Warning, time duplicated at $TIMESECONDS\n")
            next;
        } # end
        if ($PTIMESECONDS{$CELL} == 0) # Then first time
        {
            print(STDERR "1st\n");
            storepreviousvalues();
            print(data());
            next;
        }
# Check time shifts and interpolate
if ($TIMESECONDS > ($PTIMESECONDS{$CELL} + 1))
{
    $TIMEDIFF=$TIMESECONDS - $PTIMESECONDS{$CELL};
    if ($TIMEDIFF > 2) {
        print(STDERR "Error, time skip of $TIMEDIFF at $TIMESTAMP\n");
    # End if TIMEDIFF > 2 print error message
    if ($TIMESECONDS > ($PTIMESECONDS{$CELL} + 100))
    {
        printf(STDERR "Error, large time skip %d at $TIMESTAMP\n", $TIMEDIFF);
        $TIMEDIFF=100;
    }
} # End if ($TIMESECONDS > ...
    for ($i=1; $i < $TIMEDIFF; $i++)
    {
        # Interpolate missing data points
        interpolateall($i);
    } # End for ($i=0; $i < $TIMEDIFF; $i++)
} # End if ($TIMESECONDS > ...$

# Store into array for printing
printdata();

# Store PREV timeshift
storepreviousvalues();

} # End foreach LINE

# Store output in output data file
if (!open(Fout, ">$DDATADIR/$OUTPUTFILE")
{
    print(STDERR "Could not open OUTPUTFILE $DDATADIR/$OUTPUTFILE\n");
    close(Fout);
} # End for each FILE

} # End for each CELL

# FUNCTIONS
sub toseconds()
{
    # Split string 2007-01-02-HH:MM:SS into seconds
    ($YEAR, $MONTH, $DAY, $TIME) = split(/-/,$_[0]);
    ($HOUR, $MINUTE, $SECOND) = split(/:/,$TIME);
    $MONTH = sprintf("%d", $MONTH); # Convert to number, subtract the 0 from 09
    if ($CELL =~ /A/) { $MONTH++; } # Only cell A had month advanced

    #print("$YEAR, $MONTH, $DAY, $HOUR, $MINUTE, $SECOND -
    $DAYSAFTER("$MONTH")\n"); # Need to subtract one day, since this day is not over yet
    $SECONDS = $SECOND + 60 * $MINUTE + 3600 * $HOUR + 86400 * ($DAY + $DAYSAFTER($MONTH) - 1) + $JULYFIRST;
    return $SECONDS;
}

# end toseconds
sub interpolateall

# Input arguments $PIDISCH $IDISCH $STEPCNT $NSTEPS.
return sprintf("%d", $_[0] + ($_[1]-$_[0])*$_[2]/$_[3]);
} # end interpolate

sub interpolateall()
{
$ITEMP=interpolate1($PTEMP{$CELL},$TEMP,$_[0],$TIMEDIFF);
$IVCELL=interpolate1($PVCELL{$CELL},$VCELL,$_[0],$TIMEDIFF);
$IICHARGE=interpolate1($PICHARGE{$CELL},$ICHARGE,$_[0],$TIMEDIFF);
$IIDISCH=interpolate1($PIDISCH{$CELL},$IDISCH,$_[0],$TIMEDIFF);
$ITIMESTAMP=$PTIMESECONDS{$CELL}+$_[0];
# Print interpolated line
$IOUTPUTSTRING="$RAWTIME-$ITIMESTAMP-
$CELL$MODE,$MV,$CYCLECNT,
$SECONDCNT,$ITEMP,$IVCELL,$IICHARGE,$IIDISCH,$VREF,$IB,$MODECNT,
$CHARGEDC,$DIADDICHC,$ADDUSTDC,$ADCHARGE,$ADIDC"
;
push(@OUTPUTARRAY, "$IOUTPUTSTRING
");
}

sub fromseconds()
{
# Read data in seconds, convert back to 2007-07-15 15:08:46 format
$TimeZoneShift=3600 *4; # shift 4 hours, (one day = 86400)
($_[0],$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst)=gmtime($_[0]-$TimeZoneShift);
$year+=1900;
if($CELL =~/A/)
{
# Cell A was already shifted in logging program
$mon=$mon + 1; # Perl uses month starting at zero
}sprintf("%04d-%02d-%02d-\n%02d:%02d:%02d",$year,$mon,$mday,$hour,$min,$sec);
} # end fromseconds

sub resetmodecount()
{
# Start mode count, track how long in each mode
print(STDERR
"$CELL:MODECNT:'I' => 0, $MODECNT{'c'} => 0, $MODECNT{'C'} => 0, $MODECNT{'R'} => 0, $MODECNT{'d'} => 0, $MODECNT{'r'} => 0, $MODECNT{'H'} => 0
\n");
} # end sub resetmodecount()

sub storepreviousvalues()
{
$PTEMP{$CELL} =$TEMP;
$PVCELL{$CELL} =$VCELL;
$PICHARGE{$CELL} =$ICHARGE;
$PIDISCH{$CELL} =$IDISCH;
$PTIMESECONDS{$CELL} =$TIMESECONDS;

210
A.4 Convertdata.pl

This PERL script reads the daily data files and converts the scaled integer values of current, voltage, and temperature into floating point values. Data from all the daily files for each cell is stored in one large consecutive data file.

#!/usr/bin/perl

# Convertdata.pl
# Script to convert raw data into tab delimited numeric data for loading into Matlab.
# Bit values are scaled into voltage, current and temperature
# Tue Mar 11 22:40:05 EDT 2008
# rlh2

# Define default variables for checks
# Convert cell letters to numbers for upload into Matlab
%CELLS=( "A" => 1, "B" => 2, "C" => 3, "D" => 4, "E" => 5, "F" => 6, "G" => 7, "H" => 8, "I" => 9);
# Convert charging modes into numbers for Matlab
%MODES=( "I" => 0, "c" => 1, "C" => 2, "r" => 3, "d" => 4, "R" => 5, "H" => 6); # Valid modes

%CELLMODEprev=0;
$aftercharge=0; # Do not zero out until current is at rest
$afterdischarge=0; # Do not zero out until current is at rest
%CELLSPECS= (# Specs for the individual cells/boards
  # $Rcyclecnt, $RVmax, $RVmaxbit, $RIchV, $RIchbit, $RRch, $RIdisV, $RIdisbit,
  # $RRdis, $Rtimestamp) =
  # "A" => "0, 4.2, 1719, 2.4, 974, 1.0686, 0.974, 2040, 0.203, 0",
  # "B" => "0, 4.2, 1719, 2.4, 1005, 1.0667, 0.974, 2040, 0.205, 0",
  # "C" => "0, 4.2, 1719, 2.4, 998, 1.1012, 0.974, 2105, 0.206, 0",
  # "D" => "0, 4.3, 1760, 2.4, 998, 1.0702, 0.974, 2105, 0.203, 0",
  # "E" => "0, 4.3, 1765, 2.4, 1005, 1.0549, 0.984,
2140, 0.205, 0", "F" => "0, 4.3, 1770, 2.4, 1045, 1.0714, 0.974, 2220, 0.201, 0",
"G" => "0, 4.4, 1815, 2.4, 1045, 1.0806, 0.974, 2180, 0.206, 0",
"H" => "0, 4.4, 1824, 2.4, 1020, 1.0506, 0.974, 2080, 0.206, 0",
"I" => "0, 4.4, 1814, 2.4, 1040, 1.0626, 0.989, 2120, 0.206, 0"
); # END INDIVIDUAL SPECS

$argc=@ARGV;
$print("\nargc = #{@ARGV\n}");
if ($argc < 1) {die("\nUSAGE: $0 CELL_LETTER\n");}
$CELL_LETTER="@ARGV[0]";
$newtimestamp=0;
$cyclecounter=0;
# Get cell specs
($Rcyclecnt, $RVmax, $RVmaxbit, $RIchV, $RIchbit, $RRchV, $RIdis, $RIdisbit, $RRdis, $timestamp) = split(/,/,$CELLSPECs{$CELL_LETTER});
print(STDERR "CELL $CELL_LETTER: $Rcyclecnt, $RVmax, $RVmaxbit, $RIchV, $RIchbit, $RRchV, $RIdis, $RIdisbit, $RRdis, $timestamp
");

while (<STDIN>) {
$LINE=<>;
chomp $LINE; # Remove the EOL character
($HEADER, $Vref, $CYCLECNT, $SECONDNCY, $Tcell, $Vcell, $Ich, $Idis, $Vref, $LETTER, $MODECNT, $chargeDC, $dischDC, $adjustDC, $ADcharge, $ADDisch) = split(/,/, $LINE);
$newtimestamp++; # Advance timestamp
($rawtime, $internalTS, $CELL_A_MODE) = split(/-/,$HEADER); # Get PC timestamp,
$timejump=$internalTS-$PinternalTS;
if ($timejump > 1) { print(STDERR "Time jump of $timejump at $HEADER was $PinternalTS
"); } # Check timing
$PinternalTS=$internalTS;
if ($CELL_A_MODE !~ /^$CELL_LETTER/) { print(STDERR "Does not match $CELL_LETTER at $HEADER
"); next; } # Check cell letter
$CELLMODE=substr($CELL_A_MODE, 1, 1); # Get the cell mode c, c, r, d, R
if ($CELLMODEprev =~ /R/ && $CELLMODE =~ /c/) {
$cyclecounter++;

$VcellSC=bits2volts($Vcell, 2.5, 5.5, $RVmaxbit, $RVmax);
$TcellSC=(bits2temp($Tcell, 0.5, 1.2)-0.600)/0.01;
$IcellchSC=bits2amps($Ich, 0.0, 2.65, $RIchbit, $Ichnoise, $CELLMODE);
$IcelldisSC=-Idisbits2amps($Idis, 0.0, 5.5, $RIdisbit, $Idisnoise, $CELLMODE);

print(STDOUT "$CELLS{$CELL_LETTER}$CELLMODEprev\t$MODES{$CELLMODE})

212
sub bits2temp() { # Function to convert bit values to voltage
$bits =$_[0];
$minval =$_[1];  # max/min value, used to clip data
$maxval =$_[2];  # max/min value, used to clip data
$volts =$bits*2.5/1023;  # Convert bit value to voltages
if ($volts > $maxval) {$volts =$maxval;}  # Clip to maximum
if ($volts < $minval) {$volts =$minval;}  # Clip to minimum
return $volts;
}

sub bits2volts() { # Function to convert bit values to volts
$bits =$_[0];
$minval =$_[1];  # max/min value, used to clip data
$maxval =$_[2];  # max/min value, used to clip data
$refbits =$_[3];  # Charging voltage value in bits
$refvolt =$_[4];  # Charging voltage value in volts
$volts =$bits*$refvolt/$refbits;  # Convert bit value to voltage
if ($volts > $maxval) {$volts =$maxval;}  # Clip to maximum
value
if ($volts < $minval) {$volts =$minval;}  # Clip to minimum
value
return $volts;
}

sub Ichbits2amps() { # Function to convert bit values to volts
$bits =$_[0];
$minval =$_[1];  # max/min value, used to clip data
$maxval =$_[2];  # max/min value, used to clip data
$setval =$_[3];  # Value (in bits) board was set to
$mode =$_[5];  # set to zero if not charging
if ($mode =~ /\[cC\]/) {  # Then charging
$amps =$bits*2.4/$setval;  # Convert bit value to amps
if ($amps > $maxval) {$amps =$maxval;}  # Clip to maximum
value
if ($amps < $minval) {$amps =$minval;}  # Clip to minimum
value
$aftercharge=0;  # Reset aftercharge
} else {
if ($aftercharge++ > 36)
{ $amps = 0; # Current essentially zero 36 seconds after entering
}{ $amps=2.4/$setval; # Convert bit value to amps
  if($amps > $maxval) {$amps=$maxval;} # Clip to maximum
  if($amps < $minval) {$amps=$minval;} # Clip to minimum
}
# end else not in charge mode
return $amps;
} # end sub Ichbits2amps()

sub Idisbits2amps(){
  # Function to convert discharge bits to current
  $bits =$_[0];
  $minval =$_[1]; # max/min value, used to clip data
  $maxval =$_[2]; # max/min value, used to clip data
  $setval =$_[3]; # Value (in bits) board was set to
  $mode =$_[5]; # set to zero if not discharging
  if($mode =~ /d/){ # Then discharging
    $amps=$bits*4.8/$setval; # Convert bit value to amps
    if($amps > $maxval) {$amps=$maxval;} # Clip to maximum
    if($amps < $minval) {$amps=$minval;} # Clip to minimum
    $afterdischarge = 0; # Reset afterdischarge
  } else
  if($afterdischarge++ > 36) { # Current is off, force to zero to remove noise
    $amps = 0;
  } # Current essentially zero 36 seconds after entering rest mode
  else
    $amps=$bits*4.8/$setval; # Convert bit value to amps
    if($amps > $maxval) {$amps=$maxval;} # Clip to maximum
    if($amps < $minval) {$amps=$minval;} # Clip to minimum
  }
}
# end else not in charge mode
return $amps;
} # end sub Idisbits2amps()

# End of program Convertdata.pl

A.5 Createcycles.pl

This perl script reads the large file created by Convertdata.pl and saves each cycle
into a different file for upload into MATLAB.

#!/usr/bin/perl
# Create cycle.pl by rlh2
# Separate converted data into cycle files

@CELLS=("A", "B", "C", "D", "E", "F", "G", "H", "I");
$pcycle="0000";
$cell="@CELLS[0]";
open(Fout,">/home/rhart/pic/data/daily/cycles/$cell/$cell-0000.dat
);
@OUTPUTSTR=(); # Create output array
foreach $cell (@CELLS)
{
    print(STDERR "Processing $cell ...
");
    open(Fin,"/home/rhart/pic/data/daily/$cell.filtered.dat"
)and
        print(STDERR "opened $cell.filtered.dat"
);
    close(Fin);
    foreach $LINE (@LINES)
    {
        chomp $LINE;
        ($cellnum,$modeprev,$mode,$cycle,$voltage,$temperature,$currentch,$currentdis,$origtimestep,$counter,$rawtime)=split(/	/,$LINE);
        ###  Break raw time into separate columns, for Matlab
        $year=substr($rawtime,0,4);
        $mon=substr($rawtime,4,2);
        $day=substr($rawtime,6,2);
        $hr=substr($rawtime,8,2);
        $min=substr($rawtime,10,2);
        $sec=substr($rawtime,12,2);
        # Rest mode after discharge
        if($cycle != $pcycle)
        {
            print(STDERR "Changing from $pcycle to $cycle...
");
            print(Fout "@OUTPUTSTR"
);
            close(Fout);
            @OUTPUTSTR=(); # Empty output array
            $cyclestr=sprintf("%04d",$cycle); # Update pcycle
            $file="/home/rhart/pic/data/daily/cycles/$cell/$cell-$cyclestr.dat"
;
            open(Fout,">$file"
);
            print(STDERR "Opening file
"DAILY/cycles/$cell/$cell-$cyclestr.dat"
);
            if($cycle > 800) {last;} # Only process first 800 cycles
            if($cycle > 500 && $cell =~ /[FHIKL]/) {last;} # Only process first 500 cycles
        } # end foreach line
        push(@OUTPUTSTR,
            sprintf("$cellnum\t$modeprev\t$mode\t$cycle\t"$voltage\t$temperature\t$currentch\t$currentdis\t$origtimestep\t$counter\t$rawtime
        "$year\t$mon\t$day\t$hr\t$min\t$sec"
        ,
            $cycle,$voltage,$temperature,$currentch,$currentdis,$origtimestep,$counter,$rawtime
        ));
    } # end foreach line
    # Write entire array in one group to improve performance
)# end for each cell
close(Fout);
# End of program Createcycles.pl
APPENDIX B
MATLAB M-FILE PROGRAM LISTINGS

This appendix lists the MATLAB programs used to plot, identify, and simulate the cycle data and models. Programs are listed in the order they were executed.

B.1 Loopbls.m

Loopbls.m loads the file Globals.m and then steps through all cycles of every cell, plotting and identifying values. Other programs called by Loopbls.m are: Plot1cycle.m, Idbls.m, Vbls.m, and Logbls.m.

```matlab
% Loopbls.m
% Matlab file to loop through various cells,
% identifying current, and voltage responses
% relax after discharge

% Set timestamp for creating uniq file names
starttime=['-';datestr(now,'dd-mmm-yyyy HH:MM:SS')];
starttime=[starttime(1:18),'-'];starttime(13)='-';starttime(16)='-';

% Load global cell values
globals
for cellnum = 1:length(CELLNAMES) %9
Ro=0.16; % Starting point
Rd=0.03;
RdCd=50;
for cyclenum = 1:1:MAXCYCLES(cellnum) % H has 587 cycles??
try
% Build filename
filename=sprintf('s-%04d',CELLNAMES(cellnum),cyclenum);
matfile=filename;
plot1cycle % Plot voltage, current, temperature
idbls % Identify decaying discharge current timestamp
vbls % Identify Romega, Rd, and Cd
% Save identified values to data file
ferr=fopen(['parameters/','CELLNAME,date,'-RdCd.dat'],'A');
```
fprintf(ferr, '%s
', num2str([CYCLE(1), Ro, Rd, Cd, RdCd, Verrmin]));
fclose(ferr);

% After identifying Romega, Rd, and Cd, identify the rest of the voltage equation coefficients

end % end try/catch block
end % end for cyclenum
end % end for cellnum
% End of program Loopbs.m

B.2 Globals.m

The file Globals.m defines cell names, last cycle numbers, and other information needed by most of the m-files listed here.

%% Globals.m

% Global DESCRIPTION labelnames percentdeath
% Global APAR BPAR CPAR DPAR EPAR FPAR GPAR HPAR IPAR
% Global APAR2 BPAR2 CPAR2 DPAR2 EPAR2 FPAR2 GPAR2 HPAR2 IPAR2
% Global A_lastcycle Elastcycle Ilastcycle
% Global Blastcycle Flastcycle avelastcycle42
% Global C_lastcycle Glastcycle avelastcycle43
% Global D_lastcycle Hlastcycle avelastcycle44
% Global CELLNAMES CELLNUMS MAXCYCLES CELLVALUES CELLVNUM VCOLOR

% Individual cycle globals
% Global Vcell Icell Tcell Vcellf Icellf Tcellf MODE0 MODE CYCLE
% Global chargeAh dischargeAh aveTemp cell cyclenum minutes

CELLNAMES=['A';'B';'C';'D';'E';'F';'G';'H';'I'];
CELLNUMS=[1;2;3;4;5;6;7;8;9];

% Max cycle to identify
MAXCYCLES=[700;700;700;600;600;500;350;350;350];

%oldCELLVALUES=[4.2;4.2;4.2;4.3;4.3;4.3;4.4;4.4;4.4];
CELLVALUES=[4.2;4.2;4.2;4.3;4.3;4.3;4.4;4.4;4.4];
CELLVALUESTR=['4.2';'4.2';'4.2';'4.3';'4.3';'4.3';'4.4';'4.4';'4.4'];

CELLVNUM=[42;42;42;43;43;43;44;44;44]; % Put voltage as integer for selections
% Plot different voltages as different colors
VCOLOR=['b-';'r-.';'g--';'b-';'r-.';'g--';'b-';'r-.';'g--'];
PLOTLOC9=[1;4;7;2;5;8;3;6;9]; % Subplot for cells; 9 plots
PLOTLOC3=[1;1;1;2;2;2;3;3;3]; % Subplot 3 plots, 9 cells
PLOTLOC1=[1;2;3;1;2;3;1;2;3]; % Subplot 3 plots, 3 cells

LASTCYCLE80=[331,99,245,179,150,170,156,137,126]; % Last cycle before EOLife 80%
LASTCYCLE50=[536,343,456,258,256,314,234,234,216]; % Last cycle before EOLife 50%
MAXCAPACITY=[2,3283,2.2326,2.2372,2.5922,2.5395,2.5956,2.6294,2.6959,2.7355]; % Assuming, 10%, to start
% List of number of cycles to plot
B.3 Plot1cycle.m

Plot1cycle.m loads the one cycle data file, filters the data and plots voltage, current and temperature in one figure. The filtered data is then used by the identification programs. Filtering is done by calling the Movavevector.m program.

```matlab
% Plot1cycle.m
% Load global cell values
globals

% Use filename string variable, basename only
filename=input('Enter file name to load: ')';
CELLNAME=filename(1);
CYCLEstr=filename(3:length(filename));
CYCLENUM=str2num(CYCLEstr);
% Get the number of the cell for indexing CELLVALUES
for celli = 1:length(CELLNAMES)
    if(CELLNAME == CELLNAMES(celli))
        cell=celli;
    end
end

% Load data
s=sprintf(['./cycles/',filename(1),'/',filename,'.dat']);
s2=filename;s2(2)='_'; % Change - to _ like Matlab does
s3=sprintf(['Cell: %s Cycle: %s'],CELLNAME,CYCLEstr);
disp(['loading ' s])
celldata=load(s);
CELLNUM=celldata(:,1);
MODEO=celldata(:,2);
MODE=celldata(:,3);
CYCLE=celldata(:,4);
Vcell=celldata(:,5);
Tcell=celldata(:,6);
IcellC=celldata(:,7);
IcellD=celldata(:,8);
Icell=celldata(:,7)+celldata(:,8);
Timestamp=celldata(:,09);
Counter=celldata(:,10);
Year=celldata(:,11);
Month=celldata(:,12);
Day=celldata(:,13);
Hour=celldata(:,14);
Minute=celldata(:,15);
Second=celldata(:,16);
% Refilter with "step" steps, filter around center point;
filterstep=20;
Vcellf=movavevector(Vcell,filterstep);
```
Icellf = movavevector(Icell, filterstep);
IcellfC = movavevector(IcellC, filterstep);
IcellfD = movavevector(IcellD, filterstep);
Tcellf = movavevector(Tcell, filterstep);

StartTime = min(Counter) + filterstep;
EndTime = max(Counter) + filterstep;
minutes = Counter / 60; minutes = minutes - (StartTime / 60); % Reset to zero

disp(['Calculating amp hours'])
chargeAh = 0; dischargeAh = 0;
begin = 0; % Initialize previous mode
curmode = 0; % Initialize current mode
startcc = 0;
for j = 1:length(Icell)

% Calculate amp hours
if(Icell(j) > 0)
  chargeAh = chargeAh + Icell(j);
elseif(Icell(j) < 0)
  dischargeAh = dischargeAh + Icell(j);
end

% Store mode/phase changes for plotting indicator lines
if(MODEO(j) == 1 & pmode ~= 1 & startcc == 0)
  startcc = [j, minutes(j), celldata(j, :)];
end
if(MODEO(j) == 2 & pmode ~= 2)
  startcv = [j, minutes(j), celldata(j, :)];
end
if(MODEO(j) == 3 & pmode ~= 3)
  startcr = [j, minutes(j), celldata(j, :)];
end
if(MODEO(j) == 4 & pmode ~= 4)
  startd = [j, minutes(j), celldata(j, :)];
end
if(MODEO(j) == 5 & pmode ~= 5)
  startdr = [j, minutes(j), celldata(j, :)];
end
pmode = MODEO(j); % Keep last mode
end % end for j <= Icell

chargeAh = chargeAh / 3600;
dischargeAh = dischargeAh / 3600;
avetemp = mean(Tcell); % Get average temperature
minTemp = min(Tcell); % Get minimum temperature
maxTemp = max(Tcell); % Get maximum temperature

figure(1); clf;
subplot(3, 1, 1); hold off % Plotting VOLTAGE
plot(minutes, Vcellf); hold on
plot(minutes, Vcell, 'g:'); ylabel('Voltage (V)');
plot([1, max(minutes)], [CELLVALUES(cell), CELLVALUES(cell)], 'k:');

plot([1, max(minutes)], [2.75, 2.75], 'k:');
verticalrange = get(gca, 'YLim');
plot([startcc(2), startcc(2)], verticalrange, 'k:');
plot([startcv(2), startcv(2)], verticalrange, 'k:');
plot([startcr(2), startcr(2)], verticalrange, 'k:');
plot([startd(2), startd(2)], verticalrange, 'k:');
plot([startdr(2), startdr(2)], verticalrange, 'k:');
axis([0, max(minutes), 2.7, 4.5]);
text(startcc(2), 2.75, 'CC', 'FontSize', 8);
text(startcv(2), 2.75, 'CV', 'FontSize', 8);
text(startcr(2), 2.75, 'Rest', 'FontSize', 8);
text(startd(2), 2.75, 'Discharge', 'FontSize', 8);
text(startdr(2) + 0.5, 2.75, 'Rest', 'FontSize', 8); % Shift
lh = legend('Filtered', 'Unfiltered', 3);
lpos = get(lh, 'Position'); % Legend position
lpos(1) = lpos(1) + 0.1; % Shift right
set(lh, 'Position', lpos);

max(minutes), CELLVALUES(cell), [num2str(CELLVALUES(cell)), ' V'],
'FontSize', 8, 'Vertical Alignment', 'Top', 'Horizontal Alignment', 'Right');

subplot(3, 1, 2); hold off % Plotting CURRENT
plot(minutes, Icellf); hold on
plot(minutes, Icell, 'g:'); ylabel('Current (A)')
plot([1, max(minutes)], [2.4, 2.4], 'k:');
verticalrange = get(gca, 'YLim');
plot([startcc(2), startcc(2)], verticalrange, 'k:');
plot([startcv(2), startcv(2)], verticalrange, 'k:');
plot([startcr(2), startcr(2)], verticalrange, 'k:');
plot([startd(2), startd(2)], verticalrange, 'k:');
plot([startdr(2), startdr(2)], verticalrange, 'k:');
axis([0, max(minutes), -5.2, 2.8]);

ahs = sprintf('Ah: ch-%7.3f dis-%7.3f', chargeAh, abs(dischargeAh));
text(9, -1, ahs);

ahs = sprintf('diff: %6.4f', chargeAh - abs(dischargeAh));
text(9, -3, ahs);

subplot(3, 1, 3); hold off; % Plotting TEMPERATURE
plot(minutes, Tcellf); hold on
plot(minutes, Tcell, 'g:'); ylabel('Temperature (^oC)')
hold on; plot([1, max(minutes)], [aveTemp, aveTemp], 'k:');
verticalrange = get(gca, 'YLim');
plot([startcc(2), startcc(2)], verticalrange, 'k:');
plot([startcv(2), startcv(2)], verticalrange, 'k:');
plot([startcr(2), startcr(2)], verticalrange, 'k:');
plot([startd(2), startd(2)], verticalrange, 'k:');
plot([startdr(2), startdr(2)], verticalrange, 'k:');
axis([0, max(minutes), 12, 45]);

ats = sprintf('Ave temp: %7.4f', aveTemp);
text(mean(minutes), aveTemp + 5, ats, 'HorizontalAlignment', 'Center');

xlabel('Time (Minutes)');

% Set Ro, Rd, Cd, to fixed values (After these values were determined)
if(exist('Ak_Ro') == 0) load('k_Ro.mat'); end;
eval(['k_Ro=', CELLNAME, 'k_Ro;']);
Ro = k_Ro(1) + k_Ro(2) * CYCLENUM;
Rd = 0.03234;
Cd = 957.6;
RdCd = Rd * Cd;

% End of program Plot1cycle.m
B.4 Movavevector.m

Movavevector.m provides the filtering described in Equations 2.85-2.87.

```matlab
% Movavevector.m
function [VECTOROUT]=movavevector(VECTORIN,range);

% Function that takes an array and performs a moving average
% on the columns. range is the number of points to either side
% of current point in time to use in the moving average
% First column is assumed index(cycle) number, not averaged

%% From: The Scientist and Engineer's Guide to Digital
% Signal Processing By Steven W. Smith, Ph.D.

VECTORIN=crop(VECTORIN);
[maxrow,maxcol]=size(VECTORIN);
VECTOROUT=VECTORIN(1);
% At start and end of vector, small average range
VECTOROUT=zeros(size(VECTORIN)); % Make sure stays column vector
for n=1:range
    VECTOROUT(n)=mean(VECTORIN(1:n+range)); % Step up
    VECTOROUT(maxrow-n+1)=mean(VECTORIN(maxrow-n+range:maxrow)); % End range
end % end filter with range

% End of program Movavevector.m
```

B.5 Idblsm.m

The file Idblsm.m performs identification on the decaying current waveform after
discharge to determine the time constant of the PWM control circuit.

```matlab
% Idblsm.m by rlh2
% Used to identify the time constant of the current decay
curlen=60; % Length of current vector
tpercent=1-exp([-1, 2, 3, 3.5, 4]); % time constant values #
0.00155
s=startdr(1); % Start of discharge rest mode, (current shut off)
e=s+curlen; % Length(1cell);
Idrelax=1cell(s:e); % Current return after discharge
Vdrelax=Vcell(s:e); % Voltage return after discharge
Idrelaxlen=length(Idrelax);
Idfinal=mean(Idrelax(Idrelaxlen-curlen/10:Idrelaxlen)); % curlen/10=>10% of total points
Idnorm=1+Idrelax/max(abs(Idrelax)); % Normalize between 0 and 1
Idnormfinal=mean(Idnorm(Idrelaxlen-curlen/10:Idrelaxlen));

% for i=1:length(Idrelax)-1; % Create time (seconds) vector for
% current range
% i(td) = a(exp(-td/b)), BLS
```

222
\% A = \text{ones(size(td))} * -td; \\
\% Ainv = \text{pinv}(A); \\
\% a = \text{exp(Ainv(1,:)} * \text{log(abs(Idrelax))}); \% Take e^a to get actual value \\
\% b = Ainv(2,:)* \text{log(abs(Idrelax))}; \\
\% Get the time constant tRdCd \\
\% Irefmax = Idfinalpos * tpercent * 1.05; \% Shift above 0 for if statements \\
\% Irefmin = Idfinalpos * tpercent * 0.95; \% Shift above 0 for if statements \\
iRdCd = \text{zeros(1,4)}; \\
piref = 1; \% Reference into the tpercent array \\
for n = 2:Idrelaxlen \\
\% disp(n) \\
timeconst = tpercent(piref); \% Make local variable for easy reference \\
if (Idnorm(n-1) < timeconst & Idnorm(n) >= timeconst) \\
iRdCd(piref) = n/piref; \\
piref = piref + 1; \\
end \\
end \% end if matches timeconst \\
end \% end for i 1:length(Idref) \\
iRdCd; \\
RCdisTC = mean(iRdCd); \% Average value of iRdCd from tpercent vector \\
if (RCdisTC > 7.8 | RCdisTC < 7.6) \\
strRdCd = sprintf('RCdisTC (%7.4f) outside of expected range, setting to 7.6', RCdisTC); \\
RCdisTC = 7.6; \% Good match \\
else \\
strRdCd = '\'; \\
end \\
Aconstant = Idfinal; \% Create constants for simulation, and use for getting voltages \\
Bconstant = Idrelax(1) - Idfinal; \\
isim = 0*Idrelax; \% Initialize \\
\% Get the value \\
for n = 1:length(Idrelax) \\
isim(n) = Aconstant + Bconstant * \text{exp(-td(n)/RCdisTC)}; \\
end \% end for i 1:length(Vdrelax) \\
md = td/60; \% Minutes \\
figure(3); clf; subplot(2,1,1); \\
plot([md, Idrelax, 'b', md, isim, 'r-']); \\
title(['Current turning off after discharge for ', filename]); \\
xlabel('Time (Minutes)'); ylabel('Current (A)'); \\
Ierrref = Idrelax(1) + abs(Idrelax(1) - Idfinal) / 3; \% Gen reference line for error \\
Ierr = (Idrelax - isim); \% Plot error on same graph, using Ierrref hold on; plot \\
(md, Ierr + Ierrref, 'k', md, ones(size(Ierr)) * Ierrref, 'k'); \\
str = sprintf('Average Error_{rms} (%7.5f, sqrt(mean(Ierr.^2))); \\
text(floor(max(md)/3), floor(Ierrref) + 2, str); \\
text(floor(max(md)/2), floor(Ierrref) + 1, str); \\
hold off \\
\% End of program Idbls.m
B.6 Vdbls.m

The file Vdbls.m performs identification on the voltage waveform after discharge to determine the coefficients $R_\Omega$, $R_d$, and $C_d$.

```matlab
% Identify voltage diffusion after discharging
% b=1/Taupwm
% \( v(t_d) = \left[ I_d(0) \exp(-b*td) * R_0 + \{ R_d (1/(b(RdCd) -1)) \} * \exp(-td/RdCd) - \exp(-bt) \right] \)
% b=1/7.6
% $R_0$ expected to be .1-.2
% $R_d$ expected to be .03
% RdCd time constant expected to be 10-25

b=1/RCdisTC; % 1/timeconstant of current decay
id0=mean(Icell(s-2:s)); % Initial Current % Changed from 5 to 3 on 2008-07-17
if(exist('Ro')==0)
    RdCd=50; % Just chosen
    Ro=.16;
    Rd=0.02;
end
Vdfinal=Vdrelax(length(Vdrelax)-200):length(Vdrelax));

% 200 Seconds seems more than enough to catch everything
tmax=length(td);
% tbls=(0:1:tmax)'; % Create time vector (seconds)
tbls=td;%(1:150); % Create time vector (seconds)
%Ro=0.18;
Verrmin=50; % Initialize, find minimum error, then break loop
% Interate for RdCd
clear Verrvector; % Must not exist before starting the loop
RdCdcurrent=RdCd; % Keep previous value, adjust slowly
timeconstantrange=8:0.1:90;
for RdCd = timeconstantrange
    A=[id0*exp(-b*tbls),(id0/((b*RdCd)-1)) * (exp(-tbls/RdCd) - exp(-b*tbls)) ];
    k=pinv(A)*(Vdrelax(1:length(tbls))-Vdfinal);
    Rok=k(1); Rdk=k(2);
    A=k*RdCdcurrent+Vdfinal;
    Verr=sqrt(mean(Verr^2));
    Verrvector=[Verrvector,Verr];
    if( Verr < Verrmin )
        Verrmin=Verr;
        minRdCd=RdCdcurrent;
        minRok=Rok;
        minRdk=Rdk;
    end
end
if( Verrms < Verrmin )
    Verrmin=Verrms;
    minRdCd=RdCdcurrent;
    minRok=Rok;
    minRdk=Rdk;
    Vhat=A*k; % Get simulation value
end
```

224
% don't break RdCd=pRdCd; % Previous was optimal
% don't break break; % Getting larger, exit
end % end if( Verrrms < Verrmin)
%
end % end for RdCd
RdCd=minRdCd;
Rok=minRok;
Rdk=minRdk;
% Adjust slowly if out of range (added 2/10/2008)
if (abs(Rok)>0.3) Rok=0.3*sign(Rok); disp(Ro); Ro=Ro+(Rok-Ro)/100;
else Ro=Rok; end
if (abs(Rdk)>0.1) Rdk=0.3*sign(Rdk); Rd=Rd+(Rdk-Rd)/100;
else Rd=Rdk; end
if(Ro<0.1) Ro=Ro-(Rok-Ro)/100; end % Adjust negative Ro slowly
% Add lock on Ro
if (Ro <0.12) Ro =0.12; end
if (Ro >0.22) Ro =0.22; end

% Set values
Ro=Rok; Rd=Rdk;
% RdCd=RdCdcurrent-(RdCd-RdCdcurrent)/100;
Cd=RdCd/Rd;

% Fixed values
% Cd=3775.4; % Average of all nine cells
% Rd=0.0186; % Average of all nine cells
% RdCd=Rd*Cd;
% Cd=2471.3; % Average of all nine cells
% Rd=0.0220; % Average of all nine cells

figure(3); subplot(2,1,2);
try
Vdsimfixed=[id0*exp(-b*td)*Ro + (1/((b*RdCd)-1))*(exp(-td/RdCd) -exp(-b*td)))*Rd+Vdfinal];
Vdsimfixed=[id0*exp(-b*td)*Ro + (id0*(1/((b*RdCd)-1))*Rd+Vdfinal];
plot(md,Vdrelax,md,Vdsim,'r-','md,Vdsimfixed,'g');title('Simulated voltage relaxation after discharging');
end
xlabel('Time (Minutes)');ylabel('Cell Voltage (Volts)');
Verr=Vdrelax(1:length(Vdsim))-Vdsim; Verrstr=sprintf('Average Error [%ms]: %7.5f, mean(Verr));
pstr=sprintf('R_\Omega=%6.4f	R_d=%6.4f\nC_d=%7.4f\t\tau=%3.2f',Ro,Rd,RdCd/Rd,RdCd);
figure(floor(max(md)/3.3),3.4,pstr);
hold on; plot(md,Verr+3,'k:',[1,max(md)],[3,3],'k');hold off;
% End of program Vdbls.m

B.7 Logbls.m

Logbls.m performs identification on the voltage waveform to determine the voltage coefficients for Equations 2.78-2.80.
% Logbls.m
TS=1; % One second timestep
Cd0=0; % Initial Diffusion charge, chosen, assuming at rest
Range=[2.75:.01:4.2]/4.2; % Estimate capacity based on voltage at rest
Cs0=0.24*3600; % Initial Capacity, chosen ~10% Ah converted to seconds??
Cd=RdCd/Rd; % Capacitor value

%%% Set initial values
Cdt=0*Vcell;
Cst=0*Vcell;
Cdt(1)=Cd0; % set initial charge caught in diffusion
Cst(1)=Cs0; % A-hours, converted to seconds??
%%% Create the A matrix based inputs

% Create
for m = 1:(length(Vcell)-1)
    dCdt_dt=Icell(m)/Cd - Cdt(m)/(Rd*Cd);
    Cdt(m+1) = Cdt(m) + TS * dCdt_dt;
end
Capacity=1.05*max(abs(Cst)); % Make sure Q bigger than q_s

DESCRIPTION=['TCdt_1234567']; % Description, change for different identifications
modTcellf=Icellf; % Used for trying normalization of temp, had little effect
%%% Create A matrix Vcell=km + Icell*k2 +
%%% comment out Temperature correction terms as needed for particular model
Tsqcellf=Tcellf.*Tcellf; % Temperature value squared
CstNE=log(abs(Cst)/(Capacity)).*sign(Cst); % Dominant capacity term near Empty
CstNF=log((Capacity-Cst)/Capacity); % Dominant capacity term near Full
% Build up PHI (A) matrix in parts, comment out Tsquared, or T
terms if not using
Av0=[ones(size(Vcell)), modTcellf]; % , Tsqcellf]; % Initial voltage T0 value
k2,3
Alcell=[Icellf.*modTcellf]; %, Icellf.*Tsqcellf]; % Current temperature columns (T terms reduce)
% k4,5,6
Alcell=[Alcell, log(1+abs(Icellf)).*sign(Icellf),
    log(1+abs(Icellf)).*sign(Icellf).*modTcellf];
% ACDt = k7, 8, 9, 10, 11, 12
ACdt = [Cdt, Cdt.*modTcellf];
ACst = [Cst, Cst.*modTcellf];
ACst = [ACst, CstNE, CstNE.*modTcellf];
ACst = [ACst, CstNF, CstNF.*modTcellf];
% Combine into one large matrix
Alog=[Av0, Alcell, ACdt, ACst];

% Before BLS
ks=pinv(Alog)*(Vcellf-Icellf*Ro); % Take out Romega
VcellhatT=Alog*ks+Icellf*Ro;
% plot on same subplot as the voltage

226
B.8 Createmodel.m

The program Createmodel.m reads the data file created by Logbls.m, averages the three cells with the same MCV, and generates the aging model coefficients. This program calls the additional programs: Lastcycle.m, Linspacearray.m, Movingaverage.m, Chopspaces.m, Curvefit.m, PlotMCV.m, and PlotMCVquadcapacity.m.

```matlab
% Createmodel.m
% Create Model from Averaged values
% Calculate the average of each parameter and plot

close all; % Close all open graphs
% Load global cell values
globals
clear PAR;
for cellnum = 1:9 % For each cell
CELL=CELLNAMES(cellnum);
% End of life of cell is 50%
percentlife=50;
parpathname=[DESCRIPTION,'/',CELL,'-parameters.dat'];
PAR=load(parpathname);
[maxrow, maxcol] = size(PAR);
Vmeanerrcol=maxcol-29; % Count back from last column in array
Vrmserrcol=Vmeanerrcol+1;
chargeAhcol=Vmeanerrcol+16; % Column containing Ah
maxpowercol=chargeAhcol+3;
imagedir=[‘pngs/’, DESCRIPTION,’/’];
imagedir=[imagedir,num2str(percentlife),’/’];
plotn=1; % Plot number, put 3 on each figure
```
fign=1;  % Figure count

% Load cycles numbers to be skipped, due to noise errors
skipcycles;
{[skip='',CELL,'skip;']}
disp([Starting ',CELL])

[PAR2,lastcycle]=lastcycle(PAR,CELL,{\texttt{LASTCYCLE'}, num2str(per\texttt{centlife},',\{cel\l num\}')}, chargeAhcol,Vrmserrcol,0.08,skip);

PAR=movingaverage(PAR2,20);

{[CELL,'PAR=life\texttt{fraction}(PAR);']}
{[CELL,'PAR2=life\texttt{fraction}(PAR2);']}
{[CELL,'last cycle=',num2str(PAR(length(PAR),1)),';']};% Initialize to last cycle

end % end Create moving average

% Get averages of cells with same MCV
avelastcycle42=mean([A lastcycle,B lastcycle,C lastcycle]);
avelastcycle43=mean([D lastcycle,E lastcycle,F lastcycle]);
avelastcycle44=mean([G lastcycle,H lastcycle,I lastcycle]);
maxcycles42=min([max(APAR(:,1)),max(BPAR(:,1)),max(CPAR(:,1))]);% Select smallest of the three
maxcycles43=min([max(DPAR(:,1)),max(EPAR(:,1)),max(FPAR(:,1))]);% Select smallest of the three
maxcycles44=min([max(GPAR(:,1)),max(HPAR(:,1)),max(IPAR(:,1))]);% Select smallest of the three
maxcycles43=495;
maxall=min([maxcycles42,maxcycles43,maxcycles44]);% Create average vectors
maxcol=min(size(APAR));
mean42=zeros(size(APAR));
mean43=zeros(size(DPAR));
mean44=zeros(size(GPAR));
meanall=zeros(maxall,maxcol);
mean42=linspacearray(mean42,max(size(mean42)));% Moving average
mean43=linspacearray(mean43,max(size(mean43)));% Moving average
mean44=linspacearray(mean44,max(size(mean44)));% Moving average
meanall=averagearray(mean42,mean43,mean44);% Moving average
meanall=movingaverage(meanall,10);

% Set the cycle vector
cycles42=mean42(:,1);
cycles43=mean43(:,1);
cycles44=mean44(:,1);
cyclesall=meanall(:,1);

% Read in string array of labels for each column
eval(['labels',DESCRIPTION,';']); % Load labels
% Read in array of axis settings for each column
eval(['axisvalues',DESCRIPTION,';']); % Load axis values

% Open output file for model
starttime=['-', datestr(now, 'dd-mmm-yyyy HH:MM:SS')]; % set run
Timestamp
starttime(13)='_'; % Set space to underscore
fid=fopen(['CurveFit/CurveFit-Thalf-', DESCRIPTION, starttime, '.dat'], 'A');

fign=1;
plotn=1;
numplots=3;
for ncol = 1:maxcol
  % Get title and remove trailing spaces
  labelstr=chopspaces(labelnames(ncol,:));
  switch labelstr
    case {'Cycles', 'Vmeanerr', 'k_{RdT0} (\Omega)', 'k_{CdT0} (F)'}
      continue;
    case {'minTemp'}
      break;
    case {'chargeAh'}
      % Print the current plot
      print(gcf, '-dpng', [imagedir, DESCRIPTION, num2str(ncol-1, '%02d'), 'model-', removeSC(labelstr), num2str(percentlife), '.png']);
      plotn=1;
    case 'texec'
      break;
  end % end switch labelstr

  %%% Fit curve to the plot
  % Try linear interpolation
  range42=5:990; %12:400;   % Ranges for life fraction
  range43=5:990; %12:270;
  range44=5:990; %12:220;
  rangeall=5:990; %12:270;
  [prow,pcol]=size(mean42); % Skip first 50 rows to let it 'settle'
  [ylin42, klin42, errlin42]=curvefit(mean42(range42,1)/1000, mean42(range42,ncol), 'L');
  [yquad42, kquad42, errquad42]=curvefit(mean42(range42,1)/1000, mean42(range42,ncol), 'Q');
  fprintf(fid,'42lin>%s	err:%7.4f	%s
', labelstr, errlin42, num2str(klin42, '%7.4e	'));
  fprintf(fid,'42quad>%s	err:%7.4f	%s
', labelstr, errquad42, num2str(kquad42, '%7.4e	'));
  [prow,pcol]=size(mean43);
  [ylin43, klin43, errlin43]=curvefit(mean43(range43,1)/1000, mean43(range43,ncol), 'L');
  [yquad43, kquad43, errquad43]=curvefit(mean43(range43,1)/1000, mean43(range43,ncol), 'Q');
  fprintf(fid,'43lin>%s	err:%7.4f	%s
', labelstr, errlin43, num2str(klin43, '%7.4e	'));
  fprintf(fid,'43quad>%s	err:%7.4f	%s
', labelstr, errquad43, num2str(kquad43, '%7.4e	'));
  [prow,pcol]=size(mean44);
  [ylin44, klin44, errlin44]=curvefit(mean44(range44,1)/1000, mean44(range44,ncol), 'L');
  [yquad44, kquad44, errquad44]=curvefit(mean44(range44,1)/1000, mean44(range44,ncol), 'Q');
  fprintf(fid,'44lin>%s	err:%7.4f	%s
', labelstr, errlin44, num2str(klin44, '%7.4e	'));
  fprintf(fid,'44quad>%s	err:%7.4f	%s
', labelstr, errquad44, num2str(kquad44, '%7.4e	'));
fprintf(fid,'44lin>%s	err:%7.4f	%s
',labelstr,errlin44,num2str(klin44,'%7.4e	'));
fprintf(fid,'44quad>%s	err:%7.4f	%s
',labelstr,errquad44,num2str(kquad44,'%7.4e	'));

[prow,pcol]=size(meanall);
ylinall, klinall, errlinall] = curvefit(meanall(rangeall, 1)/1000, meanall(rangeall, ncol), 'L');
yquadall, kquadall, errquadall] = curvefit(meanall(rangeall, 1)/1000, meanall(rangeall, ncol), 'Q');

fprintf(fid,'alllin>%s	err:%7.4f	%s
',labelstr,errlinall,num2str(klinall,'%7.4e	'));
fprintf(fid,'allquad>%s	err:%7.4f	%s
',labelstr,errquadall,num2str(kquadall,'%7.4e	'));

% Save the identified values into a .mat file
for v=42:44
    voltstr=sprintf('%02d',v);
colstr=sprintf('%02d',ncol);
eval(['ylin', voltstr, colstr, '=ylin', voltstr, ';']);
eval(['yquad', voltstr, colstr, '=yquad', voltstr, ';']);
eval(['klin', voltstr, colstr, '=klin', voltstr, ';']);
eval(['kquad', voltstr, colstr, '=kquad', voltstr, ';']);
eval(['errlin', voltstr, colstr, '=errlin', voltstr, ';']);
eval(['errquad', voltstr, colstr, '=errquad', voltstr, ';']);
end

% Save the overall response
eval(['ylin', 'all', colstr, '=ylin', 'all', ';']);
eval(['yquad', 'all', colstr, '=yquad', 'all', ';']);
eval(['klin', 'all', colstr, '=klin', 'all', ';']);
eval(['kquad', 'all', colstr, '=kquad', 'all', ';']);
eval(['errlin', 'all', colstr, '=errlin', 'all', ';']);
eval(['errquad', 'all', colstr, '=errquad', 'all', ';']);

% Get full equation \[C1+C2*MCV]\]+[C3+C4*MCV]*LF, Added 9/5/2008

[ylinMCV0, klinMCV0, errlinMCV0] = curvefit([4.2, 4.3, 4.4]', [klin42(1), klin43(1), klin44(1)]', 'L'); % C1, C2
[ylinMCV1, klinMCV1, errlinMCV1] = curvefit([4.2, 4.3, 4.4]', [klin42(2), klin43(2), klin44(2)]', 'L'); % C3, C4
eval(['ylin', 'MCV0', colstr, '=ylinMCV0;']);
eval(['klin', 'MCV0', colstr, '=klinMCV0;']);
eval(['errlin', 'MCV0', colstr, '=errlinMCV0;']);
eval(['ylin', 'MCV1', colstr, '=ylinMCV1;']);
eval(['klin', 'MCV1', colstr, '=klinMCV1;']);
eval(['errlin', 'MCV1', colstr, '=errlinMCV1;']);

[yquadMCV0, kquadMCV0, errquadMCV0] = curvefit([4.2, 4.3, 4.4]', [kquad42(1), kquad43(1), kquad44(1)]', 'L'); % C1, C2
[yquadMCV1, kquadMCV1, errquadMCV1] = curvefit([4.2, 4.3, 4.4]', [kquad42(2), kquad43(2), kquad44(2)]', 'L'); % C3, C4
[yquadMCV2, kquadMCV2, errquadMCV2] = curvefit([4.2, 4.3, 4.4]', [kquad42(3), kquad43(3), kquad44(3)]', 'L'); % C3, C4
eval(['yquad', 'MCV0', colstr, '=yquadMCV0;']);
eval(['kquad', 'MCV0', colstr, '=kquadMCV0;']);
eval(['errquad', 'MCV0', colstr, '=errquadMCV0;']);
eval(['yquad', 'MCV1', colstr, '=yquadMCV1;']);
eval(['kquad', 'MCV1', colstr, '=kquadMCV1;']);
eval(['errquad', 'MCV1', colstr, '=errquadMCV1;']);

230
eval(['kquad', 'MCV1', colstr, '=kquadMCV1;']);
eval(['errquad', 'MCV1', colstr, '=errquadMCV1;']);
eval(['yquad', 'MCV2', colstr, '=yquadMCV2;']);
eval(['kquad', 'MCV2', colstr, '=kquadMCV2;']);
eval(['errquad', 'MCV2', colstr, '=errquadMCV2;']);

%%% end Fit curve
figure(fign);
subplot(2,1,1);
if(ncol==2) % Plot capacity in Ah, divide by 3600
    startname='cap-';
    plot(cycles42/1000,mean42(:,ncol)/3600, 'b');
    hold on
    plot(cycles43/1000,mean43(:,ncol)/3600, 'm');
    plot(cycles44/1000,mean44(:,ncol)/3600, 'k');
    % Plot identified MCV
    plotMCVquadcapacity;startname='caqmcv-';
    AXISVALUES(ncol,:)=[0,1,1,3] % Hard code axis
else % Else not capacity
    plot(cycles42/1000,mean42(:,ncol), 'b');hold on
    plot(cycles43/1000,mean43(:,ncol), 'm');hold off
    startname='s1-'; %plotMCV ;startname='mcv-';
end
% Label the plots
ylabel([labelstr]);
if(DESCRIPTION(6)=='0' & startname(1:3)=='mcv')
    ylabel(['^LAM^0',labelstr]);end;
if(DESCRIPTION(6)=='3') ylabel(['^LAM^{m8}',labelstr]);end
if(startname(1:3)=='cap') ylabel(['^LAM',labelstr]);end
if(startname(1:3)=='caq') ylabel(['^QAM',labelstr]);end
grid on;
axis(AXISVALUES(ncol,:));

lh=legend('4.2 V','4.3 V','4.4 V');
end
xlabel(['Life Fraction']);
print(gcf, '-dpng', [imagedir,startname,DESCRIPTION, num2str(ncol-1,'%02d-'), removeSC(labelstr), num2str(percentlife), '.png']);
fign=fign+1;
end % end for ncol = 1:Ic
% End of program Createmodel.m

B.9 Lastcycle.m

Lastcycle.m rebuilds the array, skipping rows that have parameters out of range or are listed in the array skip, which is a list of cycles manually created by viewing each cycle plot.

% Last cycle.m
function
[PAROUT, lastcyclenum] = lastcycle(PAR, CELL, lastcyclenum, chargeAhcol,

231
% Function to find the last cycle, skipping any out-of-range cycles

[maxrow, maxcol] = size(PAR);
lowAh = 0;
initial_capacity = mean(PAR(1:20, chargeAhcol)); % Get initial capacity
if(lastcyclenum < 95) % Last cycle for cell B @ 50%
    error('percent death is now lastcyclenum from globals')
    return;
end
PAROUT = PAR(1,:);
for n = 2:maxrow
    if (PAR(n,3) >= 1) disp([CELL,'- skipping big Romega=',
        num2str(PAR(n,3)),' ',num2str(n)]);
        continue; end;
    if (PAR(n,3) <= 0.000) disp([CELL,'- skipping small Ro=',
        num2str(PAR(n,3)),' ',num2str(n)]);
        continue; end;
    if (find(skip == n)) disp([CELL,'- skipping manual ',
        num2str(n)]);
        continue; end;
    if (PAR(n,Vrmserrcol) > maxrmserror) disp([CELL,'- big rms',
        num2str(PAR(n,Vrmserrcol))]);
        continue; end;
    if (PAR(n,1) > lastcyclenum)
        disp([CELL,'last cycle reached ' ,
            num2str(lastcyclenum)]);
        break;
    end

    % Create a moving average, to smooth values
    PAROUT = [PAROUT; PAR(n,:)];
end % End for n = length POLY
% End of program Lastcycle.m

B.10 Linspacearray.m

The program Linspacearray.m uses MATLAB's linspace function to perform linear interpolation between missing rows in the array.

% Linspacearray.m
function [ARROUT] = linspacearray(ARRIN, maxvalue);
% Function to create linear spacing between rows in an array
% Original array is assumed to have the first column of sequential numbers, with some of these numbers missing:
% 1, 3.3, 7.8
% 2, 5.3, 6.6
% 4, 5.2, 7.6
% This function will fill in the missing row between 2 and 4 with values of: 3, 5.25, 7.1
% Any rows beyond maxvalue will be chopped off
[rsize, csize] = size(ARRIN);
ARROUT = ARRIN(1,:); % Start off the first column
for n = 2:rsize
    prevn = ARRIN(n-1,1); % Get value in prev array
diff =ARRIN(n,1)-prevn;
if(diff >1)
    ROW=[prevn:prevn+diff]';
    for k = 2:cs
        ROW=[ROW,linspace(ARRIN(n-1,k),ARRIN(n,k),diff+1)']
    end
    ARROUT=[ARROUT;ROW(2:diff+1,:)]
else
    ARROUT=[ARROUT;ARRIN(n,:)]
end
ARROUT=ARROUT(1:maxvalue,:)

B.11 Movingaverage.m

The program Movingaverage.m takes the linearly spaced array and applies the movavevector.m program to columns two through N, where N is the total number of columns in the array. The first column is not averaged because it is an index column.

% Movingaverage.m
function [ARRAYOUT]=movingaverage(ARRAYIN,range);
% Function that takes an array and performs a moving average
% on the columns, range is the number of points to either side
% of current point in time to use in the moving average
% First column is assumed index(cycle) number, not averaged
% From: The Scientist and Engineer's Guide to Digital Signal Processing
% By Steven W. Smith, Ph.D.
[maxrow,maxcol]=size(ARRAYIN);
ARRAYOUT(:,1)=ARRAYIN(:,1);
for col=2:maxcol
    ARRAYOUT(:,col)=movavevector(ARRAYIN(:,col),range);
end % End for n = length ARRAYIN
% End of program Movingaverage.m

B.12 Chopspaces.m

The program Chopspaces.m removes trailing white space from a label string.

% Chopspaces.m
function [outstr]=chopspaces(instr)
% Function to cut off trailing spaces,
% Work back from end of string to find first character
B.13 Curvefit.m

The program Curvefit.m performs batch least squares to a vector using either a linear or quadratic equation.

```matlab
% Curvefit.m
function [yhat, k, err]=curvefit(LF, response, type)
% Function to do a curve fit for parameter response
% [yhat, k, err]=curvefit(LF, response, 'type')
% INPUTS:
% LF - life fraction vector
% response - response vector to be fit
% type - 1 character string  L - linear, Q - quadratic
% OUTPUTS:
% yhat - Response of model
% k - parameter matrix of model
% err - rms error between model and actually response
if(type == 'L')
    A=[ones(size(LF)), LF];
elseif(type == 'Q') % Quadratic
    A=[ones(size(LF)), LF, LF.*LF];
end
k=pinv(A)*response;
yhat=A*k;
err=sqrt(mean((response-yhat).^2));
% End of program Curvefit.m
```

B.14 PlotMCV.m

PlotMCV.m plots the linear aging model parameters versus the three MCVs onto the same plots as the averaged identified values.

```matlab
% PlotMCV.m
% Plot the full equation [C1+C2*MCV]+[C3+C4*MCV]*LF after identifying it.
% klinMCV0=klinMCV007;
% klinMCV1=klinMCV107;
```
B.15 PlotMCVquadcapacity.m

PlotMCVquadcapacity.m plots the quadratic aging equation for the capacity of the cell.

B.16 CompareModel.m

The program CompareModel.m estimates the parameters for the dynamic model from the aging model based on the MCV and current life fraction. These estimated values are then used to plot the simulation of the model with the actual data for comparison.
if (exist('starttime')==0) % Define Timestamp for this run
    starttime=['-',datenum(now,'dd-mmm-yyyy HH:MM:SS')];
    starttime=starttime(1:18);starttime(13)='-';starttime(16)='-';
end
[maxrow, maxcol]=size(PAR);
Vmeanerrcol=maxcol-29; % Count back from last column in array
Vmmserrcol=Vmeanerrcol+1;
chargeAhcol=Vmeanerrcol+16; % Column containing Ah
maxpowercol=chargeAhcol+3;

% Use filename string variable, basename only
filename=input('Enter file name to load: ');

%%% CELLNAME=filename(1);
%%% CYCLEstr=filename(3:length(filename));
%%% eval(['cyclenum=',CYCLEstr,';']); % Make integer

% Get average last value to calculate life fraction
VOLTG=num2str(CELLVNUM(cellnum));% Select appropriate voltage
eval(['avelastcycle=avelastcycle',VOLTG,';']);
lifefraction=cyclenum/avelastcycle;

%TREND3=['/sqrt(lifefraction)'];
TREND='TCdt_1234567';

eval(['Capacity=[kquadMCV002(1)+kquadMCV002(2)*',VOLTG,']/10+[kquadMCV102(1)+kquadMCV102(2)*',VOLTG,']/10)*lifefraction+(kquadMCV202(1)+kquadMCV202(2)*',VOLTG,']/10)*lifefraction*lifefraction];');
Capacity=Capacity*1.2;

Rd=0.03234;
Cd=957.6;

% Calculate coefficients from Aging equation
% Need to divide VOLTG by 10 to get 4.2, 4.3, 4.4
eval(['Ro=[klinMCV003(1)+klinMCV003(2)*',VOLTG,']/10+(klinMCV103(1)+klinMCV103(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[klinMCV006(1)+klinMCV006(2)*',VOLTG,']/10+(klinMCV106(1)+klinMCV106(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV007(1)+klinMCV007(2)*',VOLTG,']/10+(klinMCV107(1)+klinMCV107(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV008(1)+klinMCV008(2)*',VOLTG,']/10+(klinMCV108(1)+klinMCV108(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV009(1)+klinMCV009(2)*',VOLTG,']/10+(klinMCV109(1)+klinMCV109(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV010(1)+klinMCV010(2)*',VOLTG,']/10+(klinMCV110(1)+klinMCV110(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV011(1)+klinMCV011(2)*',VOLTG,']/10+(klinMCV111(1)+klinMCV111(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV012(1)+klinMCV012(2)*',VOLTG,']/10+(klinMCV112(1)+klinMCV112(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV013(1)+klinMCV013(2)*',VOLTG,']/10+(klinMCV113(1)+klinMCV113(2)*',VOLTG,']/10)*lifefraction];');
eval(['kapprox=[kapprox+klinMCV014(1)+klinMCV014(2)*',VOLTG,']/10)*lifefraction];');
Cd0=0; % Assume cell at rest to start cycle
Cs0col =chargeAhcol -6; % Get initial stored charge guess
Cs0=864;
Cdt(1)=Cd0;  %
Cst(1)=Cs0;  %

% Build up Cdt and Cst vectors

TS=1; % One second timestep
for m = 1:(length(Vcell)-1)
    dCdt_dt=Icellf(m)/Cd - Cdt(m)/(Rd*Cd);
    Cdt(m+1) = Cdt(m) + TS *dCdt_dt;
    dCst_dt=Icellf(m); % Need to add RselfDischarge and acceptance here
    Cst(m+1)=Cst(m) + TS * dCst_dt;
end

Cst_GT_Capacity=max(Cst) - Capacity; % Flag to determine if Cst >
Capacity value
% Keep log terms from going imaginary
if(Cst_GT_Capacity > 0) Capacity=Capacity+1.2*Cst_GT_Capacity;
end;

% Rebuild the A matrix

%% Create A matrix Vcell=km + Icell*k2 +...
modTcellf=Tcellf;
CstNE=log(0.01+abs(Cst)/(Capacity)).*sign(Cst); % Dominant capacity term near Empty
CstNF=log((Capacity-Cst)/Capacity); % Dominant capacity term near Full
% k2,3
Av0=[ones(size(Vcell)), modTcellf];
RoT1
Alcell=[Icellf.*modTcellf];
liT0, liT1 Alcell=[Icellf, log(1+abs(Icellf)).*sign(Icellf),
log(1+abs(Icellf)).*sign(Icellf).*modTcellf];
qdT0, qdT1 ACdt  =[Cdt, Cdt.*modTcellf]; , Cdt.*Tsqcellf];
qsT0, qsT1 ACst  =[Cst, Cst.*modTcellf];
qsNET0, qsNET1 ACst  =[ACst, Cst NE, Cst NE.*modTcellf];
qsNFT0, qsNFT1 ACst  =[ACst, Cst NF, Cst NF.*modTcellf];
% Complete phi (A) matrix
Aapprox = [Av0, Acell, ACdt, ACst];

Vapprox = Aapprox * kapprox + Ro * Icellf;

Vinit = mean(Vcell(1:10)); % Initial value of real voltage

Vapprox2 = Vapprox + Vinit - mean(Vapprox(1:10));

hold on; plot(minutes, Vapprox, 'r-.'); hold off

hold on; plot(minutes, Vapprox2, 'g--'); hold off

Verror = sqrt(mean((Vcellf - Vapprox).^2)); % rms error

Verror2 = sqrt(mean((Vcellf - Vapprox2).^2)); % rms error

Errstr = 'Model Error';

Errstr = [Errstr; sprintf('%6.2f V_{rms} %6.2f V_{rms}', Verror * 1000, Verror2 * 1000)];

xlim = get(gca, 'xlim');

th = text(xlim(2) * 0.02, 2.75, Errstr, 'VerticalAlignment', 'bottom');

if (Cst_GT_Capacity > 0)
    thandle = text(5, 2.8, num2str(Cst_GT_Capacity, 'q_s > CapacityValue by %7.4f'));
    set(thandle, 'Color', 'red', 'FontWeight', 'bold');
end

title([filename, '_{', DESCRIPTION, '} LFmodelMCV bf', TREND, '\rm ', VOLTG, ' voltages', num2str(percentdeath)]);

% End of program CompareModel.m