

POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering



Master's Degree Thesis

**Multi-model approach for simulation of
Li-ion batteries**

Supervisors

Prof. Alessandro RIZZO

Ing. Giovanni GUIDA

Candidate

Luca BUSSI

December 2022

Abstract

The global effort to reduce fossil fuel consumption implies massive energy storage exploitation. The only available technology able to satisfy the required performance is the Li-ion cell. To ensure a sustainable cycle life, it is necessary to exploit the batteries during their overall possible lifetime.

BAT-MAN is a project by Brain Technologies to provide a product that offers an online, real-time, and non-intrusive estimation of a lead-acid battery's charge status and state of health.

The thesis aims to provide an extension of BAT-MAN to Li-ion batteries. The thesis aims to investigate the behavior of different Li-ion cells to build a general model for simulation and identification purposes. A rigid methodology definition is needed to standardize the experiments and collect consistent data for the successive steps. Some different approaches are investigated to find the best trade-off model between accuracy and computational complexity. Candidate models are tested in closed-loop validation using an Extended Kalman Filter as an observer. The resulting parameters models are examined to extrapolate a simple relation between different SoC and SoH of the batteries. The last phase is the application of the ERMES algorithm, patented by Brain Technologies, to provide a fast and computational inexpensive estimation of the State of Health of the investigated cell.

Table of Contents

List of Tables	VI
List of Figures	VII
Acronyms	XII
1 Introduction	1
1.1 Thesis Outline	2
2 Lithium Batteries	3
2.1 A brief history	3
2.2 How do cells work?	3
2.3 Materials	4
2.3.1 Cathodes	5
2.3.2 Anodes	5
2.4 Temperature effects	6
2.4.1 Low temperature effects	7
2.4.2 High temperatures effects	7
2.5 Aging	8
3 State of Art	11
3.1 Mathematical model	11
3.2 SoC estimation	12
3.3 SoH estimation and aging effects simulation	14
3.3.1 Fatigue theory and equivalent cycle counting	14
3.3.2 Voltage relaxation	15
3.3.3 Other methods	16
3.4 Critiques	16
4 Energetic Framework	17
4.1 Nominal or Rated Capacity	17

4.2	Real Capacity	17
4.3	Lost Capacity	18
4.4	Coulomb Counting	18
4.5	Interpretation of the Capacity	19
4.6	State of Charge SoC	19
4.7	State of Health SoH	20
4.8	Depth of Discharge DoD	21
5	Simulation and data generation	22
5.1	Model	22
5.2	Charge simulation	25
5.2.1	Charging algorithm	25
5.2.2	Simulation	26
5.2.3	Results	30
5.3	Discharge simulations	31
5.3.1	Test definition	31
5.3.2	Simulation	31
5.3.3	Results	34
5.4	OCV-SoC characteristic	34
6	Identification	38
6.1	Single RC, static model	38
6.2	Single RC, dynamical model using SoC_R	39
6.2.1	Identification from static parameters	40
6.2.2	Direct dynamical model identification	42
6.2.3	Results	45
6.3	Single RC, dynamical model using SoC_N	45
7	Validation	50
7.1	Open loop validation	50
7.1.1	New battery validation	51
7.1.2	Old battery validation	52
7.1.3	Results	53
7.2	Minimum number of points	53
7.3	Closed loop validation	56
7.3.1	Observability analysis	56
7.3.2	Extended Kalman Filter	57
7.3.3	Constrained Extended Kalman Filter	59
7.3.4	New battery validation	60
7.3.5	Old battery validation	60
7.4	Comparison between OL and CL approach	61

8	Multi-model approach	64
8.1	Analysis of SoC and Current dependencies considering SoC_R	64
8.1.1	V_{OCV} approximation	64
8.1.2	R_0 approximation	65
8.1.3	Current independent τ approximation and R identification .	66
8.1.4	Current dependent τ approximation and R identification . .	67
8.2	Analysis of SoH dependency considering SoC_R	69
8.2.1	V_{OCV} approximation	69
8.2.2	R_0 approximation	69
8.2.3	τ and R approximation	70
8.2.4	C_R definition	75
8.3	Analysis of SoC and Current dependencies considering SoC_N	75
9	ERMES	78
9.1	Introduction of a new battery	78
9.2	Multi-model definition	78
9.3	ERMES definition	80
9.4	SoH estimation with correct values set of EKF	82
9.5	SoH estimation with linearly separated values set of EKF	91
10	Conclusions and Future Works	97
10.1	Conclusions	97
10.2	Future works	98
A	Battery model for data generation	100
B	Matlab codes: Identification	104
B.1	Static identification	104
B.2	Identification from static parameters	106
B.3	Direct dynamical model identification	110
C	Matlab codes: Validation	115
C.1	Open-loop validation	115
C.2	Closed-loop validation	117
D	Matlab codes: Parameters approximation	121
D.1	τ current independent	121
D.2	τ current dependent	125
E	ERMES scheme	130
	Bibliography	131

List of Tables

2.1	Characteristics of the most common intercalation cathode materials. [2]	6
2.2	Impact of different stress factor on cell aging [16]	10
5.1	Cell parameters provided by the manufacturer	23
5.2	CC/CV algorithm parameters	26
5.3	Capacity test performed on new battery	28
5.4	Charging duration test performed on new battery	29
5.5	Capacity test performed on old battery	30
5.6	Charging duration test performed on old battery	30
5.7	Current values used in discharging simulations	31
5.8	Simulation parameters	32
7.1	RMSE values for open loop validation for new battery model	53
7.2	RMSE values for open loop validation for old battery model	53
7.3	RMSE values for some N-steps identification tests	55
7.4	RMSE values for closed loop validation for new battery model	61
7.5	RMSE values for closed loop validation for old battery model	61
8.1	Coefficients of the straight lines approximating V_{OCV}	65
8.2	Coefficients of the straight lines approximating R_0	66
8.3	Coefficients of the straight lines approximating current independent τ	67
8.4	Coefficients p_1 of the straight lines approximating current dependent τ	69
8.5	Coefficients p_2 of the straight lines approximating current dependent τ	69
9.1	Parameters of the multi-model	80

List of Figures

2.1	Heat generation processes [7]	8
3.1	Schematic of the circuit model with n RC groups	11
3.2	Schematic of the circuit model with a different time constant for charge and discharge process [20]	12
3.3	Schematic of the thermal equivalent circuit of a Li-ion cell. [20] . .	13
4.1	Capacity evolution during cell lifetime [16]	18
5.1	Schematic of the circuit model used for simulations	22
5.2	Parameter LUTs of the considered 3-RC groups model	24
5.3	CC-CV algorithm, current and voltage profiles	25
5.4	CC-CV algorithm [33]	27
5.5	CC-CV charging simulation for new battery	28
5.6	CC-CV charging simulation for old battery	29
5.7	Example of voltage relaxation in the new battery	32
5.8	New battery, discharge simulation exploiting <i>Approach 2</i>	33
5.9	Example of voltage relaxation in the old battery	34
5.10	Old battery, discharge simulation exploiting <i>Approach 2</i>	35
5.11	Comparison on the releasable charge between new and old battery, exploiting <i>Approach 1</i> and <i>Approach 2</i>	36
5.12	Comparison between SoC_N and SoC_R choice in the OCV-SoC char- acteristic	37
6.1	Circuit model for static identification	39
6.2	Static model parameters for old and new battery	39
6.3	Circuit model for dynamical identification	40
6.4	Dynamical model parameters of the new battery calculated using the mixed static-dynamic approach	42
6.5	Dynamical model parameters of the old battery calculated using the mixed static-dynamic approach	43

6.6	Dynamical model parameters of the new battery calculated using direct <i>Approach 1</i>	44
6.7	Dynamical model parameters of the new battery calculated using direct <i>Approach 2</i>	45
6.8	Dynamical model parameters of the old battery calculated using direct <i>Approach 1</i>	46
6.9	Dynamical model parameters of the old battery calculated using direct <i>Approach 2</i>	47
6.10	Dynamical model parameters of the new battery calculated using direct <i>Approach 2</i> and SoC_N	48
6.11	Dynamical model parameters of the old battery calculated using direct <i>Approach 2</i> and SoC_N	49
7.1	Validation dataset currents for new battery	51
7.2	Validation dataset currents for old battery	52
7.3	Open loop validation of new battery model	54
7.4	Open loop validation of old battery model	55
7.5	Inference diagram of the original state-space model	56
7.6	Inference diagram of the augmented state-space model	57
7.7	Closed-loop configuration	58
7.8	Closed loop validation of new battery model	62
7.9	Closed loop validation of old battery model	63
8.1	Comparison between new and old parameters identified using SoC_R	65
8.2	Approximation of V_{OCV} curves for new and old batteries	66
8.3	Approximation of R_0 curves for new and old batteries	67
8.4	Approximation of current independent τ curves for new and old batteries	68
8.5	Approximation of R with current independent τ for new and old batteries	68
8.6	Approximation of current dependent τ curves for new and old batteries	70
8.7	Approximation of R with current dependent τ for new and old batteries	70
8.8	Approximation of V_{OCV}	71
8.9	Approximation of R_0	71
8.10	Approximation of R	72
8.11	R quadratic approximation	72
8.12	Current dependent τ approximation	73
8.13	m^τ current dependency	73
8.14	Approximation of τ	74
8.15	Comparison between new and old parameters identified using SoC_N	75
8.16	Approximation of V_{OCV} and R_0 using SoC_N	76

8.17	Approximation of current independent τ and R using SoC_N	76
8.18	Approximation of current dependent τ and R using SoC_N	77
9.1	Parameter LUTs of <i>medium battery</i>	79
9.2	ERMES scheme	81
9.3	New battery, ERMES with correct SoH, $I = 0.5C$	83
9.4	New battery, ERMES with correct SoH, $I = 0.75C$	83
9.5	New battery, ERMES with correct SoH, $I = 1C$	83
9.6	New battery, ERMES with correct SoH, $I = 1.5C$	84
9.7	New battery, ERMES with correct SoH, $I = 2C$	84
9.8	Medium battery, ERMES with correct SoH, $I = 0.5C$	84
9.9	Medium battery, ERMES with correct SoH, $I = 0.75C$	85
9.10	Medium battery, ERMES with correct SoH, $I = 1C$	85
9.11	Medium battery, ERMES with correct SoH, $I = 1.5C$	85
9.12	Medium battery, ERMES with correct SoH, $I = 2C$	86
9.13	Old battery, ERMES with correct SoH, $I = 0.5C$	86
9.14	Old battery, ERMES with correct SoH, $I = 0.75C$	86
9.15	Old battery, ERMES with correct SoH, $I = 1C$	87
9.16	Old battery, ERMES with correct SoH, $I = 1.5C$	87
9.17	Old battery, ERMES with correct SoH, $I = 2C$	87
9.18	New battery D1, SoH estimation boxplot	88
9.19	Medium battery D1, SoH estimation boxplot	88
9.20	Old battery D1, SoH estimation boxplot	89
9.21	New battery D2, SoH estimation boxplot	89
9.22	Medium battery D2, SoH estimation boxplot	90
9.23	Old battery D2, SoH estimation boxplot	90
9.24	Comparison of test <i>D1</i> and <i>D2</i>	91
9.25	New Battery D1, linear SoH values, error boxplot	91
9.26	New Battery D2, linear SoH values, error boxplot	92
9.27	Medium Battery D1, linear SoH values, error boxplot	92
9.28	Medium Battery D2, linear SoH values, error boxplot	92
9.29	Old Battery D1, linear SoH values, error boxplot	93
9.30	Old Battery D2, linear SoH values, error boxplot	93
9.31	New Battery D1, linear SoH values, SoH estimation boxplot	94
9.32	New Battery D2, linear SoH values, SoH estimation boxplot	94
9.33	Medium Battery D1, linear SoH values, SoH estimation boxplot	94
9.34	Medium Battery D2, linear SoH values, SoH estimation boxplot	95
9.35	Old Battery D1, linear SoH values, SoH estimation boxplot	95
9.36	Old Battery D2, linear SoH values, SoH estimation boxplot	95
9.37	Summary of the test, SoH estimation boxplot	96

A.1	Simulink model for data generation	100
A.2	Detail of the cell model block	101
E.1	Simulink ERMES block scheme	130

Acronyms

CEKF

Constrained Extended Kalman Filter

CL

Closed Loop

DoD

Depth of Discharge

EKF

Extended Kalman Filter

ERMES

Extendable Range MultiModal Estimator Sensing

OCV

Open Circuit Voltage

OL

Open Loop

SoC

State of Charge

SoH

State of Health

Chapter 1

Introduction

Energy storage is one of the key topics for the transition to a sustainable economy. Renewable energy is the core of future power production, but for its intrinsic characteristics, they are not an on-demand source. The solution is the storage of the energy over-production to exploit it in the future.

According to the *Sustainable Development Goals* and the *Paris Agreement*, the transition to a sustainable energy production system is not sufficient to guarantee the achievement of the goals. Transport is one of the most climate-impactful systems globally. To reduce the dependency on fossil fuels, the automotive sector is slowly substituting conventional heat engines with more efficient electric motors. The transition requires huge investments in energy storage solutions, now focused on Li-ion batteries.

Lithium batteries for automotive use have to achieve strict performance. Firstly, the power delivery must be high, to guarantee comfort, security, and maneuverability to the vehicle. The total capacity is one of the key aspects, to ensure enough range for the vehicle. On the other hand, the weight and volume of the battery packs limit the above performance index. All the previous requirements are achieved by high-cost battery packs, that guarantee high capacity and performance. With the actual technologies, the life of an automotive battery pack is lower than expected: to ensure the correct achievement, it is dismissed at approximately 80% of its total life-cycle.

On the opposite side, batteries for energy storage for daily use do have not the same strict requirements. The currents needed are lower and the size of the pack is not a problem. One of the key parameters for the spread of the technology is the cost. Automotive battery packs affect over half of the total cost of the vehicle, but storage batteries are economically unsustainable at the same price.

In the future, the entire ecological transition may depend on the use of Li-ion batteries and their exploitation during the complete theoretical life cycle. An automotive *old* battery pack could be used for years in the energy storage field.

At the moment, there is not a widespread method to certify a used battery for a re-qualification. At the same time, the first use of the battery must be stopped at the optimal point to guarantee a long time use in the second life. To ensure the achievement of this goal, a simple and inexpensive device to monitor and measure the *State of Health* of a battery is needed. BAT-MAN project delivers a solution to the problem, with a fast and online estimation of the required parameters.

1.1 Thesis Outline

This work aims to extend the concepts at the basis of the BAT-MAN project to the Li-ion batteries field. The main steps followed in the thesis are:

- In-depth study of the literature to define a starting framework. The most used techniques for the characterization and the simulation of a cell are evaluated to find the most promising for the actual goal.
- Data generation from an experimentally-defined model. The data will be used in the next steps. During this phase, charging and discharging algorithms are studied to define a strict *Energetic Framework*.
- Identification of a simpler model that approximates the data. Some identification methods are applied and evaluated through validation tests. All the models are tested both in an open and closed-loop configuration.
- Multi-model approach and ERMES algorithm application to estimate the *State of Health*.

Chapter 2

Lithium Batteries

2.1 A brief history

The first experiments on the possible exploitation of battery systems of lithium were done by G.N. Lewis in 1912. The results were not consistent and for fifty years the research did not consider anymore the material for this purpose. In the late 1960s, the request for high-density energy sources for military, medical, and consumer electronic devices sharply grew. Exploiting lithium as cathode material, the first lithium-iodine batteries were deployed.

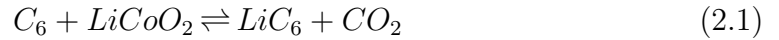
In the 1970s lithium batteries were common in consumer electronics. Lithium ions were expected to plate back in metal on the anode during the charge, and the attention was on the cathode to guarantee a long life for the component. Until 1977, the efforts to create a rechargeable lithium-based battery fail. First batteries were extremely unstable due to the presence of metallic lithium in the anode. In 1980, a battery with a lithium-ion cathode is presented by Goodenough. In 1991, Sony presented the first lithium-ion battery that exploited graphite and lithium-cobalt oxide as lithium source and sink.

The research is focused on different materials to guarantee both higher capacity and power capability., the electrode combination will suggest a possible path for a new level of power density. Commercial lithium-ion batteries have a specific power density of $150 - 250Wh/kg$, and future lithium-sulphur and lithium-air cell promise respectively $2600Wh/kg$ and $11400Wh/kg$. [1]

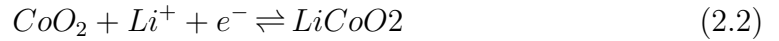
2.2 How do cells work?

The operating principle of a Lithium-ion battery is the electrochemical potential. Lithium has the highest potential ($3.04V$) among all the elements. Consequently, Lithium atoms have a high tendency to lose an electron and become ions, so it

is a highly unstable materials. On the other hand, Lithium is almost stable if it is part of a metal oxide. In the cathode, Lithium is bounded to the metal oxide. During the charging period, the positive side of the power source removes metal oxide's electrons. The electrons are stored in the anode, made of graphite. The electrolyte prevents the electrons to move freely between the anode and cathode. The Lithium-ions are attracted by the negative charge of the anode and cross the electrolyte. The structure of graphene avoids the lithium-ions to bond the electrons. During discharging phase, lithium ions cross the electrolyte to return to a more stable status. The electrons flow through the circuit and reach the cathode, to be caught by the metal oxide. If the internal temperature of the cell rises enough, the electrolyte will dry up and the electrons will flow freely. This condition leads to a short circuit and the consequent explosion of the cell. To avoid this effect, an insulating layer, called a separator, is added to the electrolyte. The separator is permeable to the Lithium-ions because of its porosity, but its impenetrable to the electrons. The electrolyte will be degraded if the electrons come into contact with it. During the first charge, the electrolyte reacts with Lithium-ions and graphite, forming a solid layer that prevents any direct contact between electrons and the electrolyte. This layer is called the Solid Electrolyte Interface (SEI). The formation of the layer consumes approximately 5% of the Lithium-ions and the process is optimized during the design phase to achieve the best performance possible. A common choice for the metal oxide is Lithium-Cobalt oxide ($LiCoO_2$). The overall reaction is:



The cathode half reaction is:



The anode half reaction is:



The conductivity of the anode is usually augmented by adding a copper plate, while the same thing is done on the cathode with an aluminum one. These plates are called collectors. Different anode and cathode materials allow the enhancement of some intrinsic characteristics of the cell. For example, an energy cell is optimized to achieve the maximum specific energy at a medium-low load, while a power one offers the best peak performances. Most of the Lithium-ion cells on the market have the structure illustrated above, with some differences in the choice of materials.

2.3 Materials

The choice of lithium is not random: it is the third lightest element, and it has the lowest reduction potential of any element, consequentially the cells have the highest

possible specific power. Furthermore, the global amount of lithium is sufficient to guarantee the possibility of the electrical transition for overall global mobility. Nevertheless, the rising price of the element is one of the major inhibitions to the expansion of the lithium-battery market and production. Cathode materials and the cost of processing are far more expensive than lithium itself. For this reason, most of the research on Li-ion batteries is invested in the development of new electrode materials, to enhance the performance of the cells. [2, 3]

2.3.1 Cathodes

An intercalation cathode material is a solid network that can store ions. The process must be reversible. The intercalation compound can be divided into different structures: layered, spinel, olivine, and tavorite. $LiCoO_2$ (LCO) is the first commercial intercalation material launched on the market. It is a convenient cathode material due to its high specific capacity (274 mAh/g), low self-discharge, high discharge voltage, and discrete cycling performance. The major limitation is the high cost, low thermal stability, and fast capacity fade at a high current rate. Other materials (Mn, Al, Fe, Cr, Al_2O_3 , B_2O_3) are studied as partial substitutes of cobalt, but the results are cells with limited performances. [2, 4] $LiNiO_2$ (LNO) has a similar specific capacity concerning LCO. The cells are less expensive due to the substitution of cobalt with nickel. Against, Ni-ions tend to substitute Li-ions and the result is a decrease in diffusion. LNO cells are also more thermally unstable than LCO. [2, 5] $LiMnO_2$ (LMO) seems to be promising because of the lower cost of the manganese for the cobalt. However, the cycling performance of LMO batteries is not consistent. [2] $Li(Ni_{0.5}Mn_{0.5})O_2$ (NMO) is an interesting material because of its similar properties to LCO. Cobalt is substituted with less expensive metals. The specific capacity is near 180mAh/g . [2]

2.3.2 Anodes

Anode materials are necessary to reduce the instability of the batteries caused by a metallic lithium anode. Lithium anode could form dendrites that, under some circumstances, cause a short circuit and the consequent thermal runaway of the battery. [2] Graphitic anodes are used since the first Li-ion commercial battery by Sony in 1992 and they are already the most common choices. Carbon is both low-cost and abundantly available. The intercalation of Li-ions between graphene planes offers good stability and high conductivity. The research is currently focused on the limitations, such as the low volumetric capacity ($330 - 430\text{mAh/cm}^{-3}$). [2] Lithium titanium oxide (LTO) has been introduced due to its superior thermal stability concerning graphite. On the other hand, the use of this material introduces a reduction in cell capacity and lower cell voltage. Also, LTO active material could

Compound	Specific capacity [mAhg ⁻¹]	Volumetric capacity mAhcm ⁻³	Voltage	Status
<i>LiTiS₂</i>	225	697	1.9	Commercial
<i>LiCoO₂</i>	274	1363	3.8	Commercial
<i>LiNiO₂</i>	275	1280	3.8	Research
<i>LiMnO₂</i>	285	1148	3.3	Research
<i>LiNi_{0.33}Mn_{0.33}Co_{0.33}O₂</i>	280	1555	3.7	Research
<i>LiNi_{0.8}Co_{0.15}Al_{0.05}O₂</i>	279	1284	3.7	Commercial
<i>Li₂MnO₃</i>	458	1708	3.8	Research
<i>LiMn₂O₄</i>	148	596	4.1	Commercial
<i>LiCo₂O₄</i>	142	704	4.0	Research
<i>LiFePO₄</i>	170	589	3.4	Commercial
<i>LiMnPO₄</i>	171	567	3.8	Research
<i>LiCoPO₄</i>	167	510	4.2	Research
<i>LiFeSO₄</i>	15	487	3.7	Research
<i>LiVPO₄F</i>	156	484	4.2	Research

Table 2.1: Characteristics of the most common intercalation cathode materials. [2]

react with organic electrolytes. The problem can be resolved using carbon coating, but the introduction of carbon material can accelerate electrolyte decomposition at high temperatures. [2, 6]

2.4 Temperature effects

The effects of temperature on Li-ion batteries are due both to the migration of the electrons and the properties of the material used in the design. Chemical reactions are regulated by Arrhenius law, which defines the relationship between temperature and reaction rate. While an acceptable operating range could be identified from 20°C to 60°C, the optimal range to reach consistent performances is limited to 15°C to 35°C. Outside this region, the cell can quickly degrade its performance and increase the risk of a dangerous malfunction. [7, 8] The impact on Li-ion batteries can be divided into effects attributable to high and low temperatures. Low temperatures lead to a decrease in ionic conductivity and diffusivity, resulting in a reduction of power capability. Considering low temperatures, the environment plays a critical role, while in high temperatures, heat is usually produced inside the battery proportional to the stress level of the operation. High internal temperatures are common both in fast charging and discharging. Some of the processes are

irreversible and the lack of accurate control could result in thermal runaway. [7]

2.4.1 Low temperature effects

Li-ion batteries have a strong decrease in performance below 0°C . The influence of temperature is traceable to different sources. The electrolyte is influenced by low temperatures: the viscosity increases and the ionic conductivity is consequently reduced. This effect leads to an increase in the internal impedance of the cell. The inclusion of electrolyte additives, as LiPo_2F_2 , is a partial solution to the problem. Low temperatures also slow the diffusion of Li-ions through the electrodes. [7] Another important factor is the increase of charge-transfer resistance: at -20°C is three times higher than at 25°C . The kinetics of the cell is deeply affected by this phenomenon. Also, during the charging the resistance is higher than during a discharging cycle, so the charging of the battery becomes a critical operation at low temperatures. [7] Cold conditions accelerate lithium plating. The anode is polarized, and the intercalation of Li-ions is slowed during the charging. Aggregated Li-ions are deposited on the electrodes, lowering the total capacity of the cell. Also, lithium plates can grow up becoming dendrites, which are dangerous for possible damage to the SEI and the consequently short circuit. [7, 9]

2.4.2 High temperatures effects

The influence of high temperature on cell behavior is more complex than the ones imputable to low temperature. Heat generation is associated with charge transfer and chemical reactions during the normal cycling of the battery. Heat is produced both in reversible and irreversible processes. Entropic heating is generated by reversible entropy fluctuations during electrochemical reactions. One of the main sources of irreversible heat generation is the so-called polarization process. It is a result of an excessive voltage between the operating point of the circuit and the open circuit voltage of the batteries. Overcoming this resistance, Li-ions generate heat. Another component of the irreversible process is ohmic heating, present both in the charging and discharging processes. Finally, phase changes in the cathode, due primarily to the diffusion of Li-ions, generate heat. [7] Aging is affecting the thermal behavior of batteries. Both calendar and cycling aging occur in combination, generating complex effects on the normal performance of the cell. The most intuitive contribution to heat generation is the growth of internal resistance of the cell, which produces heat through the ohmic effect. Not only does aging influences thermal properties, but also high temperature speed up aging both on calendar and cycling processes. [7, 10] Thermal runaway is common under non-suitable conditions use of Li-ion batteries. In high-temperature states, it may occur due to the trigger of exothermic reactions. The studies on the

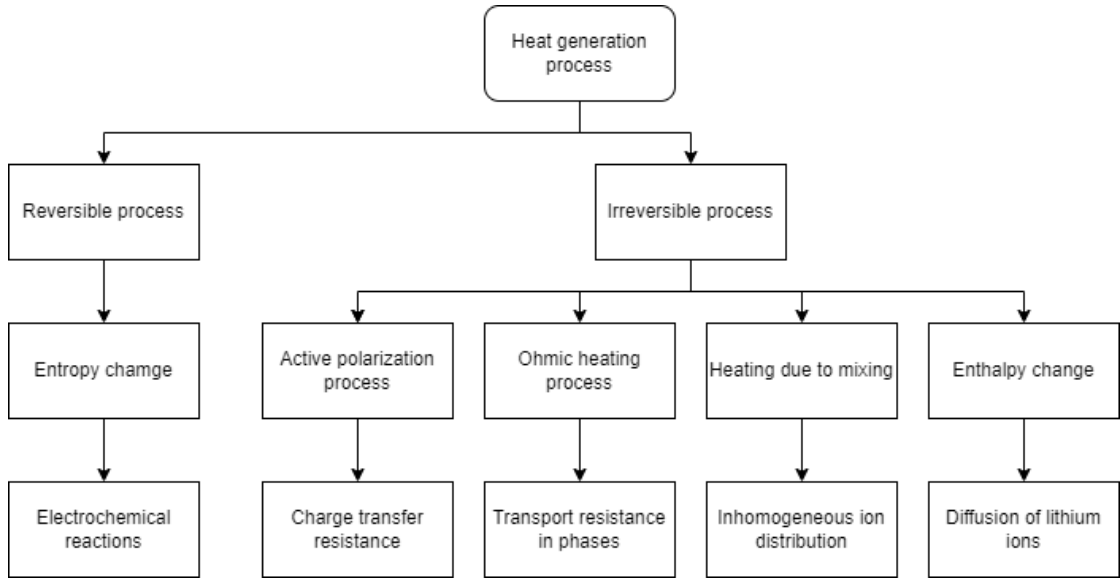


Figure 2.1: Heat generation processes [7]

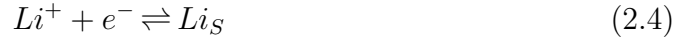
single effects occurring on the cell are done using a synchrotron X-ray tomography and the simulation of suitable conditions. [11] The shell temperature of 200°C corresponds to the trigger of the exothermic reaction. Under these conditions, the internal temperature of a single cell can reach up to 1000°C , damaging also the structural components of the battery. [7] To prevent any damage to the battery, a sophisticated thermal model is required. The easiest method is to measure the shell temperature of the cell and estimate the internal temperature using an empirical model. Internal temperature sensors are under study, being able to offer a more precise measurement at the cost of a more complex construction of the cell. [7]

2.5 Aging

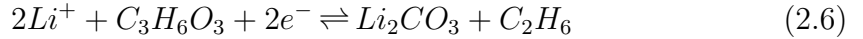
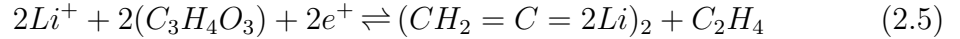
Aging is the most complex phenomenon to model regarding Li-ion batteries. It involves a wide range of side effects of the principal reactions; it is strongly non-linear and condition dependent. At the moment, there is no agreement on the influence of this mechanism on cell performance. The aging process can be divided into two sub-categories: calendar and cycle aging. Calendar aging is the most investigated due to some practical reasons. After production, batteries must be stocked in the warehouse for long periods. Also considering EVs, vehicles idle for more than 90% of their life cycle. Research agrees that a stock charge of 40% is optimal to reduce calendar aging, but there is contrasting proof regarding the best temperature. Research about cycle aging is mostly focused on the reduction of

usury due to charging and discharging processes. Cycling aging is strongly related to the stress applied to the battery. High temperature, current, and depth of discharge accelerate the aging process. [12, 13, 14]

The two more common consequences of aging are capacity loss and internal resistance increase. The phenomena involved are several and difficult to investigate. Research focuses on post-mortem methods such as X-ray photoelectron spectroscopy, X-ray diffraction, and scanning electron microscopy. While there is no evidence of modification of the cathode, aging is commonly attributed to lithium plating on the anode and changes in the solid electrolyte interface. Side reactions, such as gas production, are not considered the primary aging mechanism. [12, 15] The first process analyzed is lithium plating. Metallic lithium is produced based on the reaction



And it deposits on the surface of the graphite layer. This process both decreases the total active quantity of Li-ions in the cell and limits the transfer of the ions, increasing the amount of lithium subjected to the plating. [12] SEI growth is the secondary process that defines the aging of the cell. The thickness of the layer reduces the mass transport of the ions and influences the kinetics of the electrons on the anode. The phenomenon is described by the following reactions: [12]



Another process double linked to aging is thermal stability. Not only aging can be accelerated by temperature, but also the thermal property of the battery depends on the aging of the cell. Different operating temperatures during the life cycle may produce different levels of aging, triggering side reactions more than in other cases. [15] Aging is the result of a large variety of processes difficult to predict. The overall State of Health of the battery is influenced by its cycling history. The large number of materials used to build Li-ion cells complicates the identification of possible side reactions and a generalization is very unlikely. Cells with different technologies respond in different ways to the same stress. In table 2.2 it is shown the influence of the most common stress factors on different battery technologies. [16]

Stress factor	Cell chemistry					
	LCO	NMC	NCA	LMO	LFP	LTO
High SoC cycling	+++	+++	++	++	++	+
Low SoC cycling	++	+	+	+	+	++
High temperature	+++	++	++	++	+	+
Low temperature	+++	+++	+++	+++	+++	n.a.
High current	+++	++	+	n.a.	++	+

Table 2.2: Impact of different stress factor on cell aging [16]

Chapter 3

State of Art

3.1 Mathematical model

While in the first years of Li-ions battery studies electro-chemical model was the most exploited, now almost all the research is focused on equivalent electric models. Electrical models are accurate enough for simulation purposes, computationally efficient, and simple to manage. The circuit can be modified to satisfy different requirements, such as a higher precision or a lighter complexity for limited computational power applications. The most general circuit is composed of an independent voltage generator, a series resistor, and one or more parallel RC branches, as shown in figure 3.1. R_0 models the static response of the cell. V_{OCV} is the so-called open

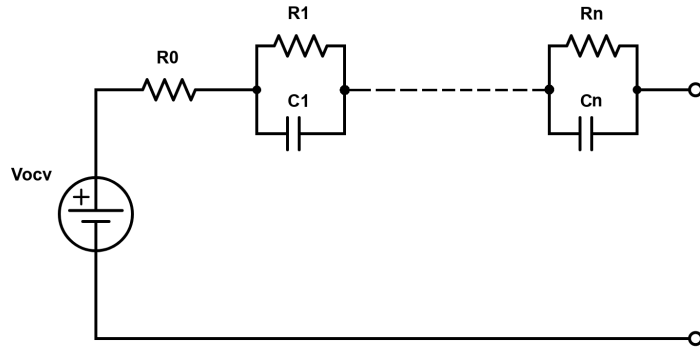


Figure 3.1: Schematic of the circuit model with n RC groups

voltage circuit, the terminal voltage of the cell under no load condition. It is the voltage measured after a long rest period of the battery. The RC branches model the dynamic behavior of the cell. The input of the model is the current, taken positive during the charging process, while the output is the terminal voltage. All

the parameters are dependent on SoC, temperature, aging, and load current. To reach a higher approximation of the non-linear behavior, a non-linear block can be applied to the scheme in the so-called Wiener structure. [17] A commonly used model presents two RC branches. The first RC has usually a small time constant, that models double-layer capacitance and charge transfer phenomena. The second one is significantly larger and takes count of the diffusion processes. [18, 19, 20, 21] A variation of the 2-RC model is to split the second time constant between the charging and discharging processes. The difference between the time constants is quite pronounced and the simulation results are improved. To achieve the separation, two ideal diodes are inserted into the circuit. [20] A major part of the

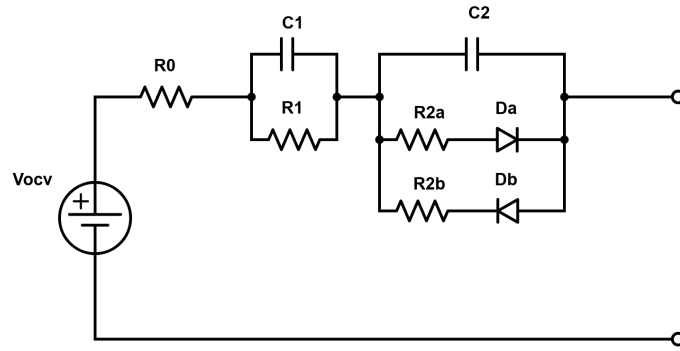


Figure 3.2: Schematic of the circuit model with a different time constant for charge and discharge process [20]

research does not consider temperature's influence on the parameters. A thermal model is decisive only in a critical situation, not very common in the general use of batteries. The dependence of the parameters on the temperature is determined experimentally and thermal models are required to consider the self-heating of the cell. A widespread equivalent thermal-electrical model is reported in figure 3.3.

3.2 SoC estimation

The common factor of all the techniques is the Coulomb-counting algorithm. To evaluate SoC, the current is integrated over time and compared with the total capacity of the cell. The resulting equation is quite standard:

$$\frac{dSoC(t)}{dt} = \frac{1}{C} I(t) \quad (3.1)$$

The process needs a periodical recalibration to avoid iterative errors due to uncertainties in the current measurements. While the evolution of the SoC curve is

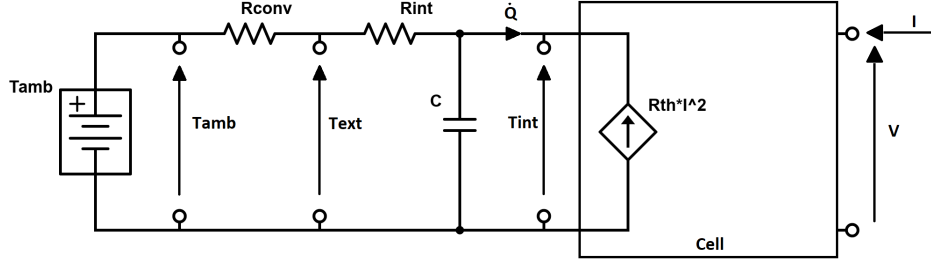


Figure 3.3: Schematic of the thermal equivalent circuit of a Li-ion cell. [20]

easy to replicate with the above method, a precise value for the SoC to calibrate the curve is difficult to find. To identify the SoC level of the cell under the test, two major approaches are exploited. The easiest one is to consider the verified relationship between open circuit voltage and SoC. The method is effective, and the results are good, but for the intrinsic characteristics of the measurement, it is impossible to apply it in a real-time application. Moreover, some cells present an almost flat characteristic at a medium value of SoC, reason why the uncertainty is not guaranteed to be acceptable. Another important drawback is the need for a precise characterization of the cell under test, due to the differences in the V_{OCV} -SoC curves considering different technical solutions in the design of the cell and the production process. The second and most used method is the inclusion of an observer in the simulation. The SoC appears as a state in the state space representation of the system and the convergence is usually quite fast. As an observer, Extended Kalman Filter or its derived algorithms are used. The results are consistent at the cost of high computational complexity. [22, 17]

An innovative approach exploits the correlation between SoC, and a new variable called unit time voltage drop and defined as

$$V' = \frac{\Delta V}{\Delta t} \quad (3.2)$$

The average correlation between SoC and $1/V'$ is 0.984 while the same between SoC and V is 0.988. This approach provides a simple correlation between SoC and SoH. The variable V' is correlated with the overpotential of the cell from V_{OCV} during a transient. SoC equation can be written as:

$$SoC = aV + b(1/V') + c \quad (3.3)$$

Where the coefficients are estimated from the data through linear regression. SoH

can be expressed as a function of SoC and V' as:

$$SoH = \alpha(SoC) \cdot [A(1/V') + B] \quad (3.4)$$

With A and B coefficients. SoC is calculated as a linear function of V and $1/V'$. This model structure guarantees high precision where the two variables have linear behavior, but it has poor performance at the borders of the SoC span due to their non-linearities. [23]

3.3 SoH estimation and aging effects simulation

Aging effects are difficult to estimate because of the number of different phenomena involved. The most used parameter to quantify the effect is the State of Health. SoH quantifies the real capacity of the cell with the initial or nominal one.

$$SoH(t) = \frac{real\ capacity(t)}{nominal\ capacity} \quad (3.5)$$

Other parameters used are State of Function SoF, End of Life EoL, and Remaining Useful Life RUL. All these quantities are related only to the capacity of the batteries, other critical characteristics of the cell are not considered in the health evaluation, but only as consequences of aging. [24] Calendar aging models are based on the electrochemical reactions that take place inside the cell. The equations are structured under Arrhenius kinetic and depend on temperature, time, and SoC. However, the parameters are determined experimentally. [24, 25] To simulate cycle aging, there is no agreement on the strategy to analyze the phenomenon. The results are not generalizable and the efficiency of the computation is usually very low.

3.3.1 Fatigue theory and equivalent cycle counting

Based on the equivalent cycle number, and stress parameters related to the depth of discharge DoD, C-rate, and temperature, an aging index is calculated for each charge-discharge cycle. The effect of cycling on the battery capacity and internal resistance is then depicted by this aging index. The model is applicable to any type of lithium-ion battery because it is based on physical equations and comparable cycle counts. The number of cycles a battery can withstand before reaching EoL when repeatedly discharged and charged at a specific DoD, C-rate, and temperature is referred to as the maximum number of cycles. When a battery has lost 20% of its initial capacity, it is said to have reached its end of life. The calculated stress factor is used to evaluate the maximum number of cycles. [26] The equivalent

number of cycles at a given DoD is defined as the number necessary to simulate the current situation with a standard cycle beginning and ending at full SoC:

$$N_{eq}(n) = 0.5 \left(2 - \frac{DoD(n-2) + DoD(n)}{DoD(n-1)} \right) \quad (3.6)$$

The aging index is defined as:

$$\epsilon(n) = \epsilon(n-1) + \frac{N_{eq}(n)}{N_c(n-1)} \quad (3.7)$$

where N_c is the maximum number of cycles to EoL.

3.3.2 Voltage relaxation

Voltage relaxation techniques are based on the collection of different parameters related to the standard charging and discharging cycle. The most significant test and parameters are reported:

- Voltage decrease rate during constant current discharging. The voltage decrease is faster as the cell ages. It is due to the decrease in the battery's total capacity.
- Voltage increase rate during constant current charging. The terminal voltage of the cell reaches faster than the nominal voltage when the cell is old.
- Constant current charging time. The time required to reach nominal voltage decreases when the battery ages.
- Total charging time using a CC/CV algorithm. The overall time to charge the battery grows for older cells.
- Current decrease rate during constant voltage charge. The current decrease rate increases for older batteries.
- Temperature decrease rate during constant voltage charge. The temperature decreases faster in old cells than in new ones.

The above parameters are combined with a machine learning algorithm to provide an accurate estimation of SoH. [27, 28, 29]

3.3.3 Other methods

Another investigated method is the statistical approach. Collecting a large amount of data, the researchers try to establish statistical connections and correlations. Prior information or models are not necessary, and they don't use physical or chemical equations. [24]

The application of the Degradation-Entropy Generation Theorem provides a new approach in the simulation field. The degradation of the cell is directly correlated with the rate of entropy generation, using irreversible thermodynamics. [30]

3.4 Critiques

The simulation of a battery exploiting an electric equivalent circuit is quite standardized and it is consistent with most possible scenarios. It requires a previous identification of the parameters through collected data and a calibration of the observer used to correct the small deviations due to model error. The main drawback is the impossibility of including an identification of the state of health of the cells to simplify the overall model. Using the current framework, the same battery considered in two different instants of its life is not considered as a single entity, because of the requirement of two sets of parameters to be simulated. A major part of the research is focused on the application of theoretical results to real-world data. The result is a simulation of the oncoming phenomena, but the goal to correctly classify a battery with its state of health with a simple test is always far. The investigated approaches are not suitable for real-time applications, of the high complexity or the large history needed to guarantee a consistent result. At the methodological level, accelerated aging tests are the most used. Due to the complexity of the aging phenomenon, it is impossible to generalize results achieved with specific and repetitive cycle tests. Cycling at high temperatures, DoD and C-rate return the enhancement of irreversible process with a different proportion of real-world application. To conclude, all the studies consider the end of life of a cell when it reaches the 80% of the initial capacity. This assumption includes the idea that a battery must be used only in the originally designed field of application. There is an absolute lack of research that looks for a re-characterization of the cell to designate them to a lighter use.

Chapter 4

Energetic Framework

The need to define a rigid energetic framework rises with an accurate analysis of the literature. Most of the considered articles do not report the assumptions on the basis on which the work is done. Without a clear definition of such quantities as SoC and Capacity, it is impossible to generalize the obtained results. Also, little changes in the assumption can generate huge discrepancies during the elaboration. In this work, when the approach to follow is not a priori clear, all the hypotheses are investigated until one result is the best for the research. In this section, the most relevant definitions and assumptions are reported. Some of these affirmations will be discussed and developed in the following chapters.

4.1 Nominal or Rated Capacity

Nominal Capacity is defined as the ideal capacity of the cell. It is always provided by the manufacturer, and, before any analysis, it is the only information available on the battery. It is expressed in Ah and it is a constant quantity.

$$C_N \tag{4.1}$$

4.2 Real Capacity

Real capacity corresponds to the actual capacity of the cell, considering a cycle from fully charged to completely discharged. It is expressed in Ah and it is dependent on SoH and T. During a single charge or discharge cycle, the real capacity is considered constant.

$$C_R(SoH, T) \tag{4.2}$$

Despite what common sense could suggest, the relation $C_R \leq C_N$ is not always true. The rated value is a mean value provided by the manufacturer, so the real

quantity can be also slightly higher than that. Additionally, initial cycles can release an additional quantity of lithium that enhances the capacity performance of the cell. After this short period, the capacity will decrease first linearly and then in a non-linear way.

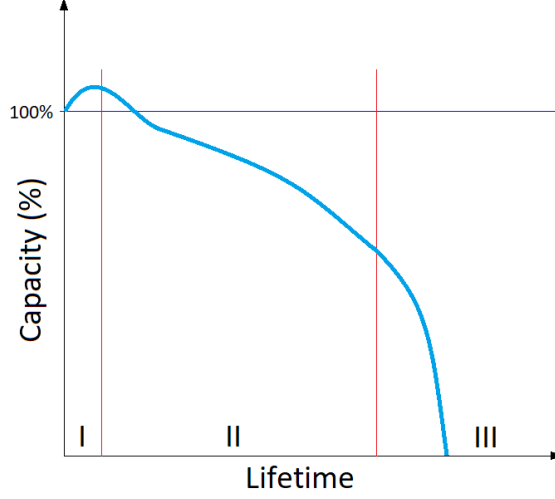


Figure 4.1: Capacity evolution during cell lifetime [16]

4.3 Lost Capacity

It is the difference between nominal capacity and real one. It is measured in Ah, and, excluding the first cycles, defined as positive.

$$C_{Lost}(SoH, T) = C_N - C_R(SoH, T) \quad (4.3)$$

4.4 Coulomb Counting

It is the algorithm used to quantify the charge transfer from and to the battery. Due to the measurement uncertainties on current and time, it needs periodical recalibrations to be effective.

$$Q(t) = \int_{t_0}^t I(\tau) d\tau \quad (4.4)$$

The current is defined as positive when it is provided to the battery. Generally, the absolute value of this quantity is considered.

4.5 Interpretation of the Capacity

Real capacity can be divided into released and releasable capacity. [31]

$$C_R(SoH, T) = C_{Released}(t) + C_{Releasable}(t) \quad (4.5)$$

The released part is defined as the energy already extracted from the battery, beginning the experiment with the battery fully charged. During the charging phase, the released capacity results negative.

$$C_{Released}(t) = \int_{t_0}^t I(\tau) d\tau \quad (4.6)$$

It is clear that:

$$C_{Released}(0) = 0 \quad (4.7)$$

$$C_{Released}(t_f) = C_R \quad (4.8)$$

with t_f time when the cell is completely discharged. Releasable capacity refers to the energy already present in the cell. In an experiment design to begin with the battery completely charged, it can be defined as the charge that will be extracted until the cell is defined as completely discharged.

$$C_{Releasable}(t) = \int_t^{t_f} I(\tau) d\tau \quad (4.9)$$

It can be seen that:

$$C_{Releasable}(0) = C_R \quad (4.10)$$

$$C_{Releasable}(t_f) = 0 \quad (4.11)$$

Also the nominal capacity can be decomposed as:

$$\begin{aligned} C_N &= C_{Lost}(SoH, T) + C_R(SoH, T) = \\ &= C_{Lost}(SoH, T) + C_{Released}(t) + C_{Releasable}(t) \end{aligned} \quad (4.12)$$

4.6 State of Charge SoC

The definition of SoC is strictly dependent on the assumptions of fully charged and fully discharged cells. The easiest approach to make the assumption is defining one of the two points and correlating the other. For simplicity, it is fixed at the fully charged point. A battery is considered completely charged or full when the battery charger stops its action. Despite the simplicity of the problem, there is no

agreement on what consider a fixed point on the characteristic of the battery. It is not a universal assumption, but it is quite general if the algorithm used for the charging process is the same. The importance of this assumption is to fix the point of $SoC = 1$, corresponding to the full cell. At this point, the definition of SoC is not unique. A general description is a ratio between the charge present in cell Q and its total capacity C .

$$SoC(t) = \frac{Q(t)}{C} \quad (4.13)$$

C is considered constant across the discharging cycle. There are two workable solutions for the choice of C . Both C_R and C_N can be used and no one seems to be a better choice. The two definitions are considered together in this section. SoC nominal is defined using C_N :

$$SoC_N(t) = \frac{Q(t)}{C_N} \quad (4.14)$$

SoC real is defined using C_R :

$$SoC_R(t) = \frac{Q(t)}{C_R} \quad (4.15)$$

Considering the experiments performed in this work, the starting point of the discharging process is always $SoC = 1$. To allow easier data collection and elaboration, the SoC definition is written considering the relation:

$$Q(t) = Q_{Releasable}(0) - \int_0^t I(\tau) d\tau = C - \int_0^t I(\tau) d\tau \quad (4.16)$$

The two above equations become:

$$SoC_N(t) = \frac{Q(t)}{C_N} = 1 - \frac{1}{C_N} \int_0^t I(\tau) d\tau \quad (4.17)$$

$$SoC_R(t) = \frac{Q(t)}{C_R} = 1 - \frac{1}{C_R} \int_0^t I(\tau) d\tau \quad (4.18)$$

As seen before, the real capacity of the cell can be slightly higher than the expected one. The result is the possibility of SoC lower than 0. The assumption has no physical meaning, it is a consequence of fixing the point of full charge at $SoC = 1$.

4.7 State of Health SoH

There is no unique parameter to define the aging condition of a cell. Different uses enhance different side reactions, resulting in different behaviors of the battery.

SoH is defined according to the literature as the capacity fade of the cell. The other parameters will be related to SoH to identify a clear framework for aging phenomena.

$$SoH(t) = \frac{C_R(t)}{C_N} \quad (4.19)$$

Where C_R is the real capacity of the cell and C_N is the nominal one. Due to the initial increasing performance of most of the batteries, SoH can be greater than 1 for a short period at the beginning of life.

4.8 Depth of Discharge DoD

Depth of discharge is a parameter used to quantify the stress due to a deep discharge of the battery. For a given discharging cycle, it is possible to define DoD as the difference between the initial and the final SoC.

$$DoD = SoC(t_0) - SoC(t_f) \quad (4.20)$$

Chapter 5

Simulation and data generation

5.1 Model

Due to the difficulties of performing many tests on real batteries, the research is done using mathematical models found in the literature. The main obstacle to the use of real data is the a priori definition of the experiments. This work aims to define a strict framework also in experimental data collection.

The model used to simulate the behavior of the battery is the three RC-groups circuit shown in figure 5.1. [32] The values of the parameters are experimentally

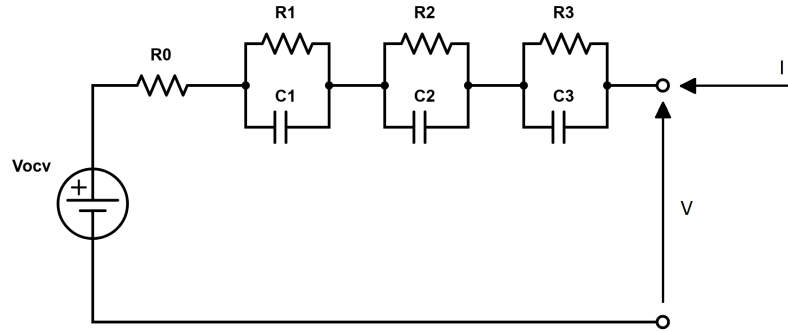


Figure 5.1: Schematic of the circuit model used for simulations

determined and collected in LUTs, depicted in figure 5.2. Two different aging values are considered. The rated parameters are reported in the table: From here, the batteries are called new and old referring to the aging values. As could be seen, the values of the parameters are strongly influenced by SoH and SoC. The

Cell parameters	
Capacity	5.4 Ah
Minimum Voltage	2.8 V
Rated current	2 C

Table 5.1: Cell parameters provided by the manufacturer

characteristic of the circuit can be written as:

$$V_T = V_{OCV} + R_0 I + V_1 + V_2 + V_3 \quad (5.1)$$

The three dynamical voltages are resolved using Laplace transform. Considering one of the three RC-groups:

$$V(s) = \frac{R}{sCR + 1} I(s) \quad (5.2)$$

$$V(s) \cdot sCR + V(s) = RI(s) \quad (5.3)$$

Defining the time constant τ :

$$\tau = \frac{1}{RC} \quad (5.4)$$

the equation can be rewritten as:

$$V(s) \cdot \frac{s}{\tau} + V(s) = RI(s) \quad (5.5)$$

Transforming again in time-domain:

$$V(t) = -\frac{1}{\tau} \cdot \frac{dV(t)}{dt} + Ri(t) \quad (5.6)$$

To resolve the equation, the previous expression is integrated:

$$V(t) = \int_{t_0}^{t_1} (-\tau V(t) + \tau Ri(t)) dt + V(t_0) \quad (5.7)$$

In real word applications, time is discretized. Considering a time sample T_s :

$$V(k+1) = V(k) + [-\tau V(k) + \tau Ri(k)] T_s \quad (5.8)$$

Now it is possible to write the state space equation of the three RC-groups models:

$$\begin{cases} V_T(k) = V_{OCV} + R_0 I(k) + V_1(k) + V_2(k) + V_3(k) \\ V_1(k+1) = V_1(k) + [-\tau_1 V_1(k) + \tau_1 R_1 i(k)] T_s \\ V_2(k+1) = V_2(k) + [-\tau_2 V_2(k) + \tau_2 R_2 i(k)] T_s \\ V_3(k+1) = V_3(k) + [-\tau_3 V_3(k) + \tau_3 R_3 i(k)] T_s \end{cases} \quad (5.9)$$

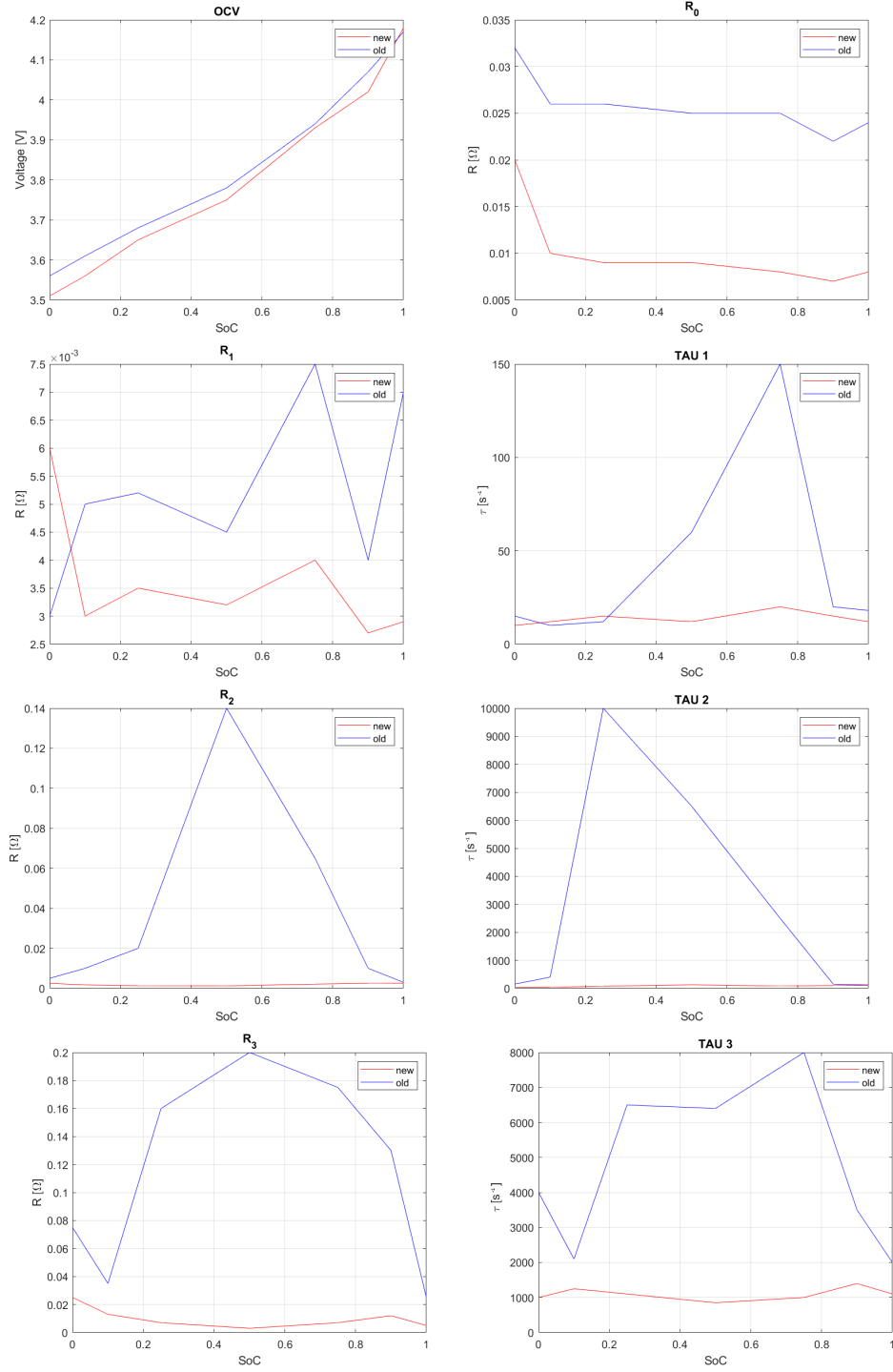


Figure 5.2: Parameter LUTs of the considered 3-RC groups model

The model is enough complex to guarantee good performance also in open-loop simulation. In the identification process, a simple model is considered: the difference in complexity between the two models is crucial to guarantee the generalization of the results.

5.2 Charge simulation

5.2.1 Charging algorithm

The charging process is a crucial part of the framework definition. The process must have consistent results across different experiments to guarantee repeatability. As depicted in the Energetic Framework, the charger is the only way to objectify the status of *completely charged battery* or *full battery*, corresponding to $SoC = 1$. Another use of charging simulation is the definition of the real capacity of the considered cell.

To charge a Li-ion battery many different algorithms can be used. The main differences among them are the time duration of the process and the stress applied to the cell. In this work all of these parameters are not important, so the easiest and most used algorithm is exploited. The *constant current / constant voltage* CC/CV algorithm consists of a two-step charging profile. The cell is charged at constant current (CC) until a suitable voltage $V_{pre-set}$ is reached. Then the voltage is kept constant (CV) until the current overpass a minimum current threshold. In real word applications, for safety reasons, little changes to the algorithm are

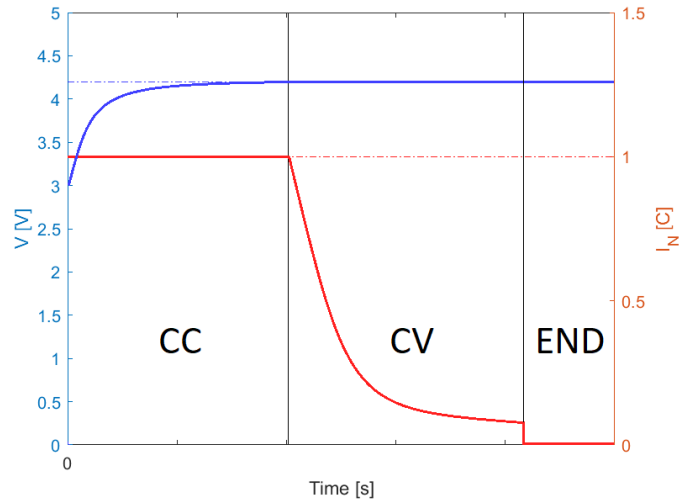


Figure 5.3: CC-CV algorithm, current and voltage profiles

needed. The resulting algorithm is reported in figure 5.4 [33]

- Lithium cells could be damaged by a charging process outside the optimal temperature range. If the temperature is not suitable, the charge cannot start.
- A high current provided at low SoC could be dangerous. When the voltage is below V_{cutoff} a lower current is needed. This process is called *Trickle Charge* TC and it exploits a typical current of $0.1C$.
- Particular health state of the battery could lead to constant current absorption greater than the current threshold. This phenomenon leads to the overcharging of the cell and the consequent irreversible damage. A limited time for the overall process is set to avoid the problem.

The CV portion of the algorithm is performed by a PID controller to maintain a stable voltage to the desired value.

5.2.2 Simulation

A charging process simulation is performed on the two batteries. The parameters of the CC/CV algorithm are reported in the table.

CC/CV parameters	
CC current	1C
Voltage pre-set threshold	4.2V
Voltage cutoff threshold	3.6V
Minimum Current	0.01C
Sample time	0.1s

Table 5.2: CC/CV algorithm parameters

New Battery

In figure 5.5 is reported the result for *new battery*. Until 5000s the TC mechanism is prevalent. The terminal voltage is slowly increasing due to a low input current. From 5000s to 8000s, the charger operates in CC mode. The current is maintained at 1C and the voltage is increasing. Reaching the voltage of 4.2V, the charger switches to CV mode, where the PID controller acts to stabilize the voltage. Here the current is decreasing until it reaches the minimum current threshold. To investigate the repeatability of the results, some tests are performed. Starting for a defining SoC, the cell is completely charged. The test is repeated five times, each time measuring the absorbed charge. The total capacity is computed by adding the theoretical charge already present in the cell at the beginning of the charging process. The results are consistent with the requirements. The charger defines a

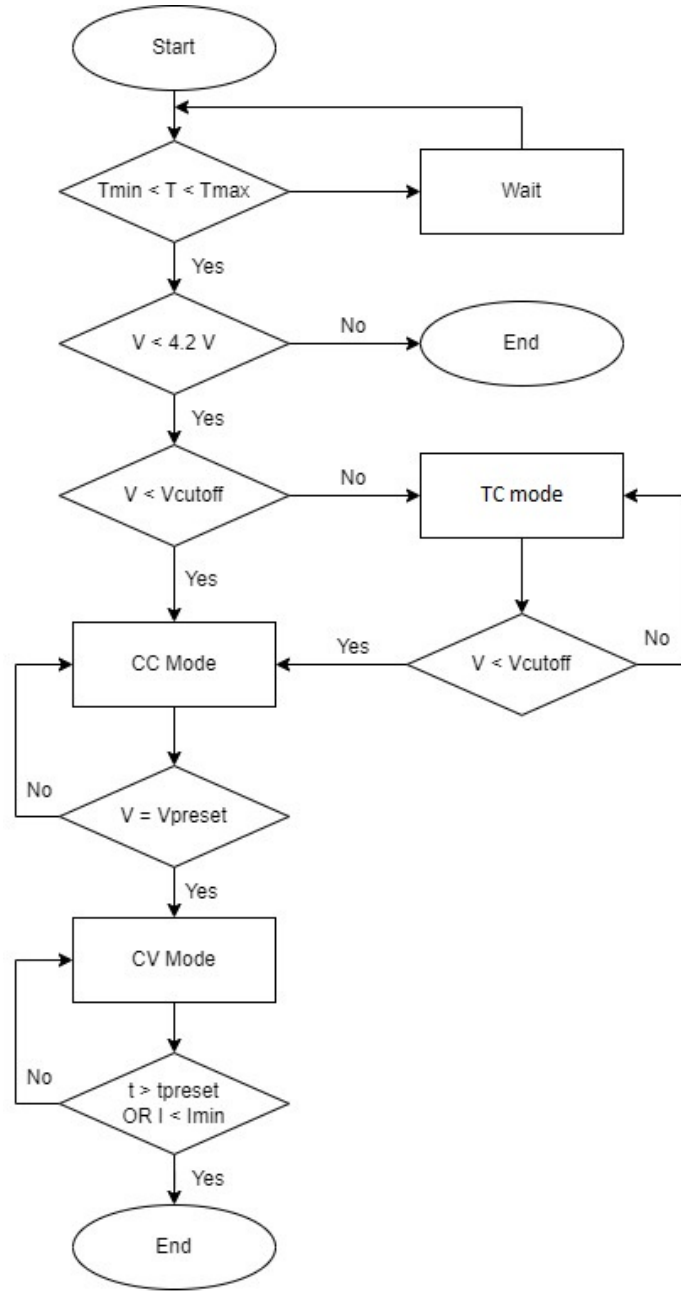


Figure 5.4: CC-CV algorithm [33]

unique point for *full battery*, so the association with $SoC = 1$ is univocal. The same test is performed to investigate the overall charging time of the process. Again, the results are consistent with the requirements. The charging process takes the same amount of time for every iteration.

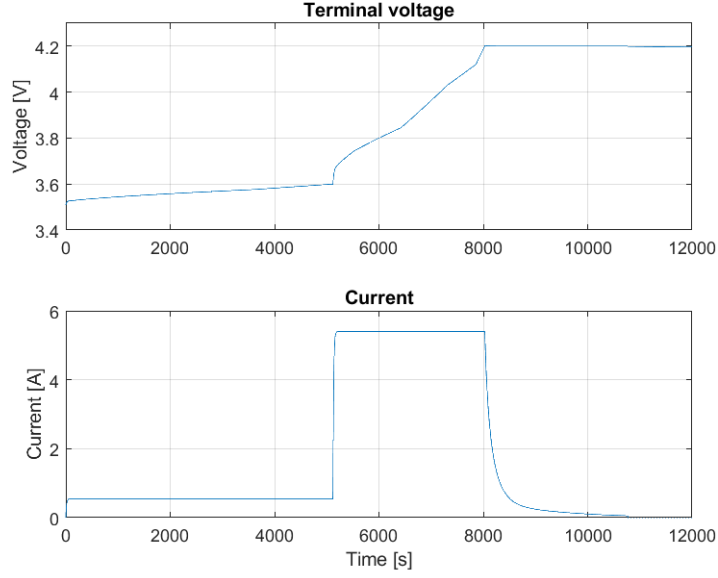


Figure 5.5: CC-CV charging simulation for new battery

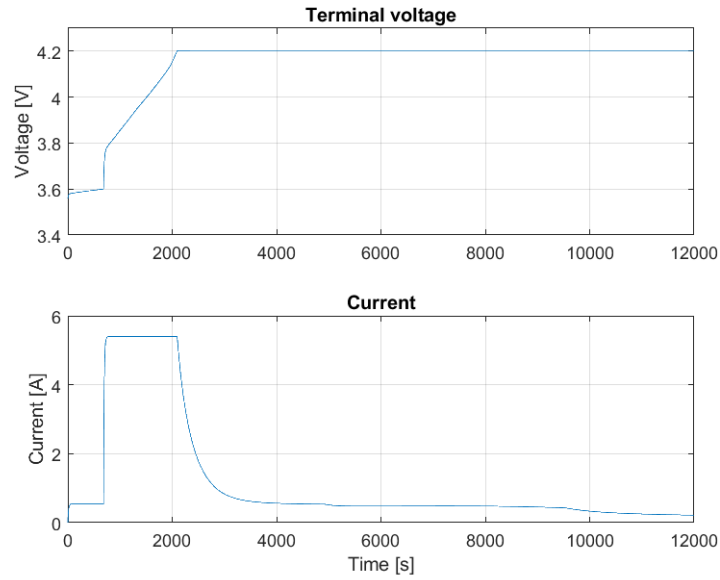
New battery, Capacity test						
Initial SoC	Charge absorbed [Ah]					Mean Capacity [Ah]
0	5.4515	5.4515	5.4515	5.4515	5.4515	5.4515
0.1	4.9115	4.9115	4.9115	4.9115	4.9115	5.4515
0.3	3.8315	3.8315	3.8315	3.8315	3.8315	5.4515
0.5	2.7515	2.7515	2.7515	2.7515	2.7515	5.4515
0.7	1.6715	1.6715	1.6715	1.6715	1.6715	5.4515
0.9	0.5923	0.5923	0.5923	0.5923	0.5923	5.4515

Table 5.3: Capacity test performed on new battery

Old battery

In figure 5.6 is reported the result for *old battery*. Until 1000s the TC mechanism is prevalent. The terminal voltage is slowly increasing due to a low input current. From 1000s to 2000s, the charger operates in CC mode. The current is maintained at $1C$ and the voltage is increasing. Reaching the voltage of $4.2V$, the charger switches to CV mode, where the PID controller acts to stabilize the voltage. Here the current is slowly decreasing until it reaches an almost stationary value. To investigate the repeatability of the results, some tests are performed. Starting for a defining SoC, the cell is completely charged. The test is repeated five times, each time measuring the absorbed charge. The total capacity is computed by adding

New Battery, Charging time test					
Initial SoC	Charging time [s]				
0	10770	10770	10770	10770	10770
0.1	7295	7295	7295	7295	7295
0.3	5010	5010	5010	5010	5010
0.5	4204	4204	4204	4204	4204
0.7	3270	3270	3270	3270	3270
0.9	1572	1572	1572	1572	1572

Table 5.4: Charging duration test performed on new battery**Figure 5.6:** CC-CV charging simulation for old battery

the theoretical charge already present in the cell at the beginning of the charging process. The results are consistent with the requirements. The charger defines a unique point, with a tolerance of 0.08% for *full battery*, so the association with $SoC = 1$ is univocal. The same test is performed to investigate the overall charging time of the process. Again, the results are consistent with the requirements. The charging process takes the same amount of time for every iteration.

Old battery, Capacity test						
Initial SoC	Charge absorbed [Ah]					Mean Capacity [Ah]
0	4.1221	4.1221	4.1221	4.1221	4.1221	4.1221
0.1	3.7171	3.7171	3.7171	3.7171	3.7171	4.1221
0.3	2.9071	2.9071	2.9071	2.9071	2.9071	4.1221
0.5	2.0971	2.0971	2.0971	2.0971	2.0971	4.1221
0.7	1.2871	1.2871	1.2871	1.2871	1.2871	4.1221
0.9	0.4805	0.4805	0.4805	0.4805	0.4805	4.1255

Table 5.5: Capacity test performed on old battery

Old Battery, Charging time test					
Initial SoC	Charging time [s]				
0	17962	17962	17962	17962	17962
0.1	16393	16393	16393	16393	16393
0.3	14177	14177	14177	14177	14177
0.5	11158	11158	11158	11158	11158
0.7	7643	7643	7643	7643	7643
0.9	2882	2882	2882	2882	2882

Table 5.6: Charging duration test performed on old battery

5.2.3 Results

As expected, the old battery has a capacity lower than the new one. It can be defined the SoH of the two batteries:

$$SoH_{NEW} = 1 \quad (5.10)$$

$$SoH_{OLD} = \frac{C_R}{C_N} = \frac{4.12Ah}{5.4Ah} = 0.75 \quad (5.11)$$

Considering charging time, the older battery takes more time to be completely charged. This is due to the faster increase of the terminal voltage during the CC phase. The time distribution between CC and CV is modified with an increase in the duration of the CV phase, which is slower than the CC one. Despite the lower capacity, the duration of the charging process for an old battery is almost doubled concerning a new one.

5.3 Discharge simulations

5.3.1 Test definition

The batteries are first charged with the charging algorithm defined in the previous section. Starting from $SoC = 1$, the cells are completely discharged using a sequence of current pulses. To investigate the response of the cells, more than one current value is used. The standard values of current are reported in table 5.7. The test

Current values	
Standard value	Normalized value
2.7A	0.5C
4.05A	0.75C
5.4A	1C
8.1A	1.5C
10.8A	2C

Table 5.7: Current values used in discharging simulations

aims to provide a suitable set of data for the identification phase and to define the characteristic $V_{ocv}(SoC)$ of the two batteries. To accelerate the extrapolation of the characteristic, a suitable idle time after each current pulse is needed to guarantee the OCV measurement. The parameters used in the simulations are reported in table 5.8. The end of the test can be determined following two approaches:

- **Approach 1:** The cell is considered empty when the terminal voltage reaches a suitable threshold. It is the method used in real word applications, because of the impossibility of the electronics working under the voltage $V = 3.4V$.
- **Approach 2:** The cell is considered empty when a charge equivalent to C_R is extracted.

In both cases, the test is immediately stopped if the safety voltage threshold (2.8V) provided by the manufacturer is reached. To obtain a sufficient resolution, 20 pulses are used to discharge *new battery* and the same settings are used for the *old* one. The parameters are reported in table 5.8

5.3.2 Simulation

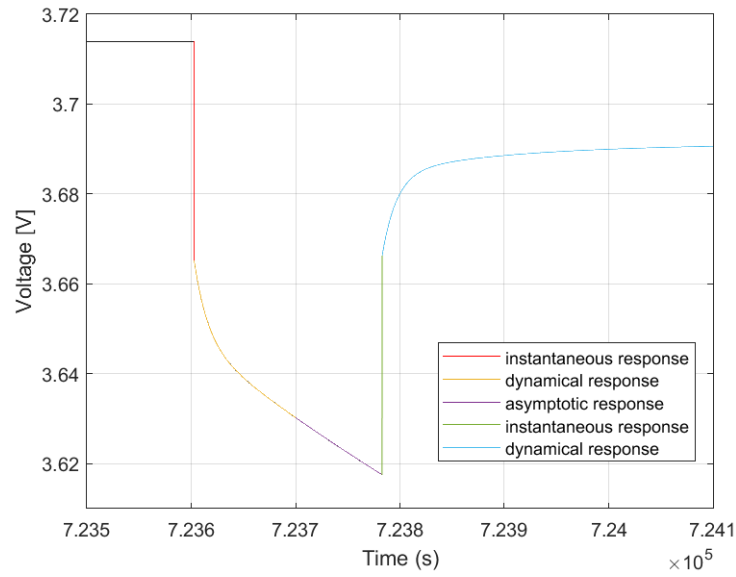
New Battery

It could be seen as the typical characteristic of a battery under current pulses. When the current is applied, the instantaneous response is due to the static resistance

Simulation parameters	
Time after charge	28800s
Current period	57600s
Current duty cycle	$\frac{I_N}{C_N} \cdot 0.3\%$

Table 5.8: Simulation parameters

of the cell. Figure 5.7 it is depicted the instantaneous response, the dynamical one, and the asymptotic behavior of the circuit after the end of the transient. Considering the entire discharging process, it is evident a dependency of the sharply

**Figure 5.7:** Example of voltage relaxation in the new battery

decreasing of the terminal voltage to the applied current: higher the current and deeper the voltage relaxation. The phenomenon is accentuated at low SoC when the terminal voltage is near the limit threshold. When the current is eliminated, the battery has a fast response and the OCV is reached in a short time. In figure 5.8 are reported only the results referring to *Approach 2*.

Old Battery

When the current is applied, the instantaneous response is due to the static resistance of the cell. In figure 5.9 it is depicted the instantaneous response and the dynamical one. Considering the entire discharging process, it is evident

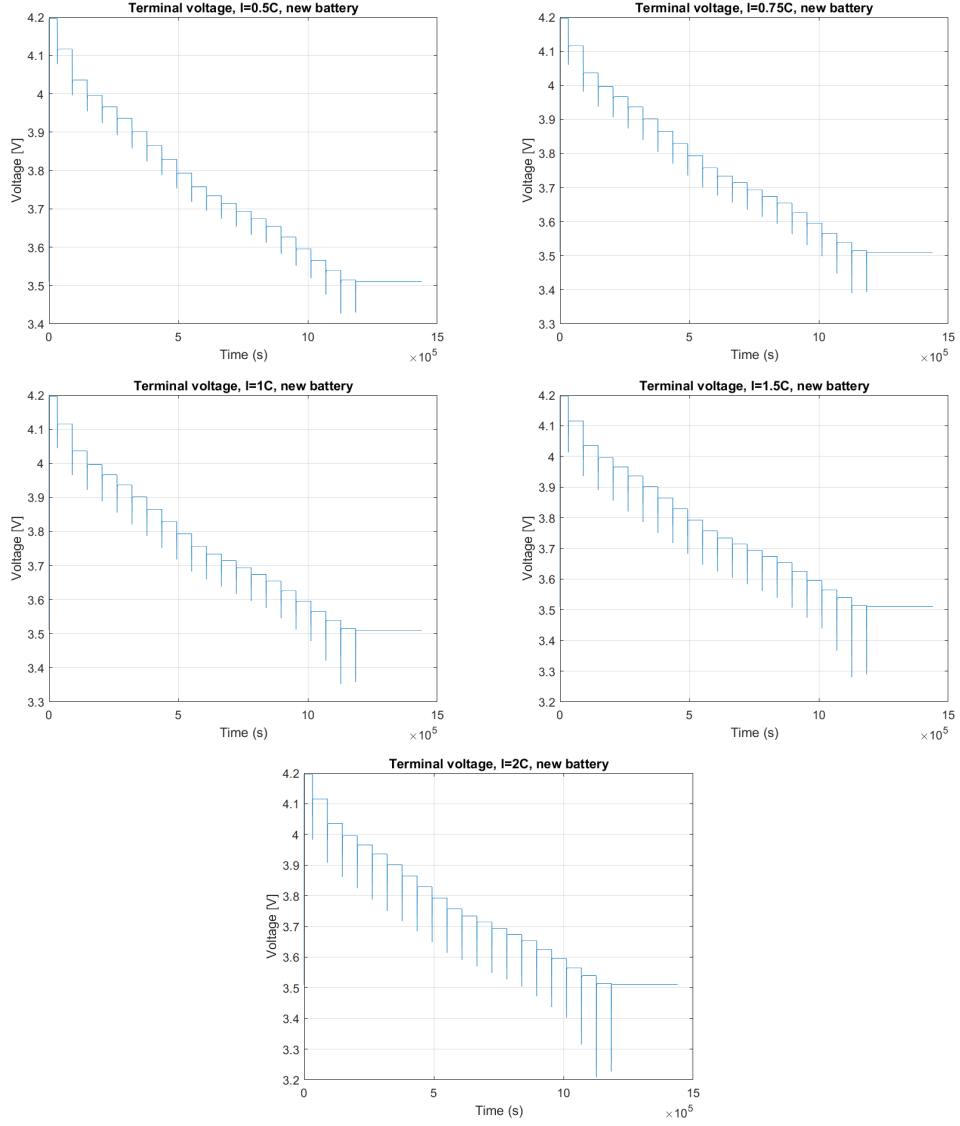


Figure 5.8: New battery, discharge simulation exploiting *Approach 2*

a dependency of the sharply decreasing of the terminal voltage to the applied current: higher the current and deeper the voltage relaxation. When the current is eliminated, the battery has a slow response and the OCV is reached in a non-negligible time. In figure 5.10 are reported only the results referring to *Approach 2*.

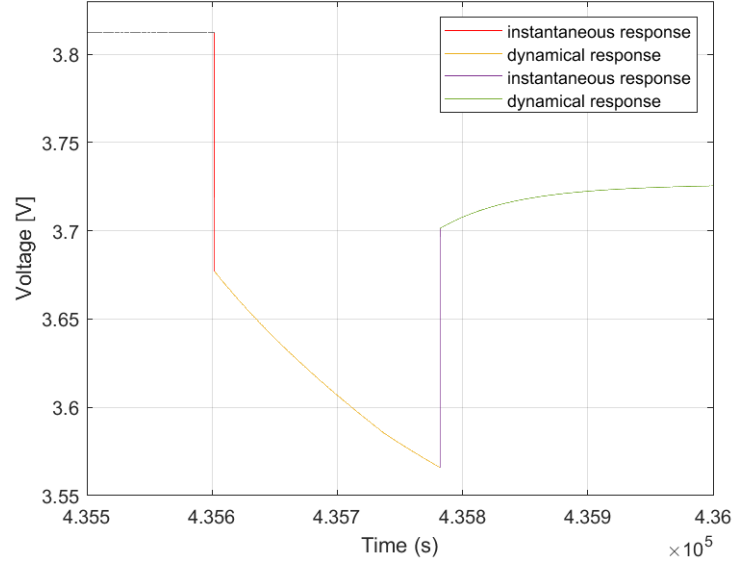


Figure 5.9: Example of voltage relaxation in the old battery

5.3.3 Results

The difference between *Approach 1* and *Approach 2* is evident in figure 5.11. The extractable charge from the batteries in *Approach 1* is current dependent. This is due to the deeper relaxation appearing in the older battery. In real word applications, this is a physical limitation due to the design choice of the circuit powered by the cell, but for research, the purpose is a clear limitation. To avoid possible ambiguity on the definition of *empty battery*, *Approach 2* only is now considered. The capacity is consistent with the SoH value defined in the previous paragraph. The dynamical transient of the new battery result shorter than the old one. This phenomenon leads to a slower response to the current pulses and a slower reaching of the steady-state terminal voltage. The voltage relaxation is more pronounced in the old battery and less dependent on the SoC value compared to the new one. The deeps are almost constant across all the SoC intervals, while in the new battery a low SoC, the relaxation is more pronounced. This result suggests a similar behavior of a new battery at low SoC and an older battery at high SoC.

5.4 OCV-SoC characteristic

The following consideration is done based on the results obtained by *Approach 2*. One of the advantages of the previous test is the possibility to extrapolate the OCV-SoC relation directly from the terminal voltage characteristic. The Open Circuit

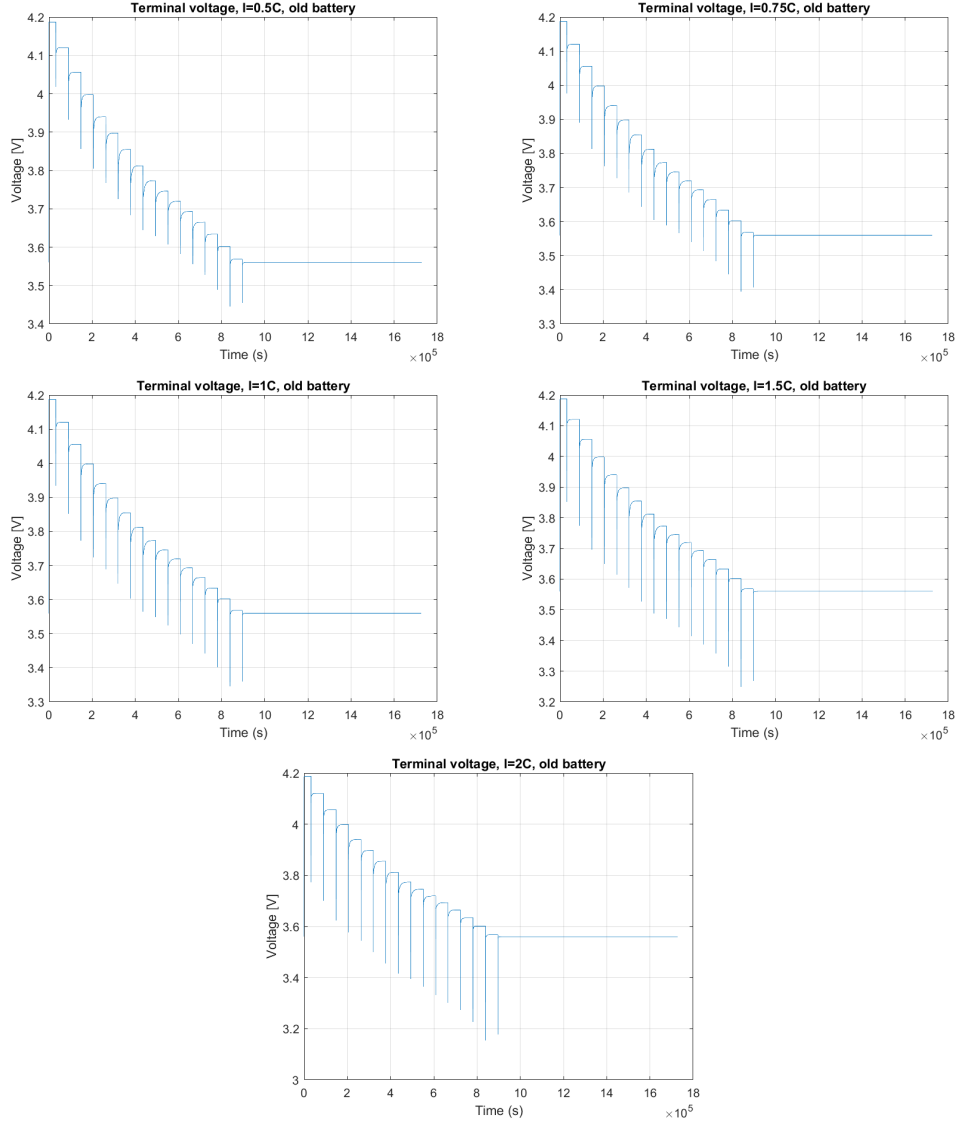


Figure 5.10: Old battery, discharge simulation exploiting *Approach 2*

Voltage is equal to the terminal voltage if the cell was idle for a sufficient period. The time interval needed for a correct measure is determined experimentally. In the previous test, the shorter period is equal to 57254s and the results are consistent with the required precision. After each current pulse and the consequent idle time, the voltage measurement is collected. Now there are two possible choices for the definition of SoC, as described in chapter 4:

- SoC_N

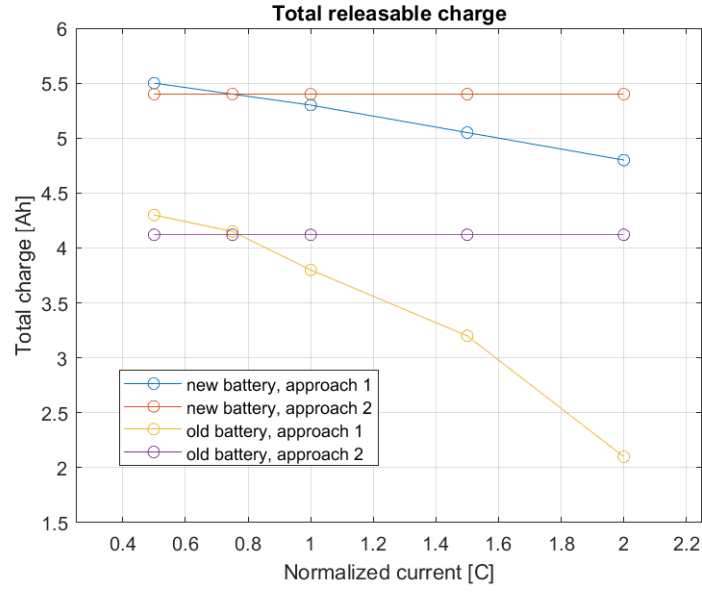


Figure 5.11: Comparison on the releasable charge between new and old battery, exploiting *Approach 1* and *Approach 2*

- SoC_R

The solutions appear to be equivalent. The choice of SoC_N redistributes the influence of the SoH on the capacity of the cell model, while SoC_R does not need a correction of the parameters. Both approaches are followed in the next steps. In figure 5.12 are reported the consequent characteristics. The first approach suggests the use of a unique OCV-SoC characteristic for all the batteries. The second one suggests a rotation of the characteristics around the point $SoC = 1$

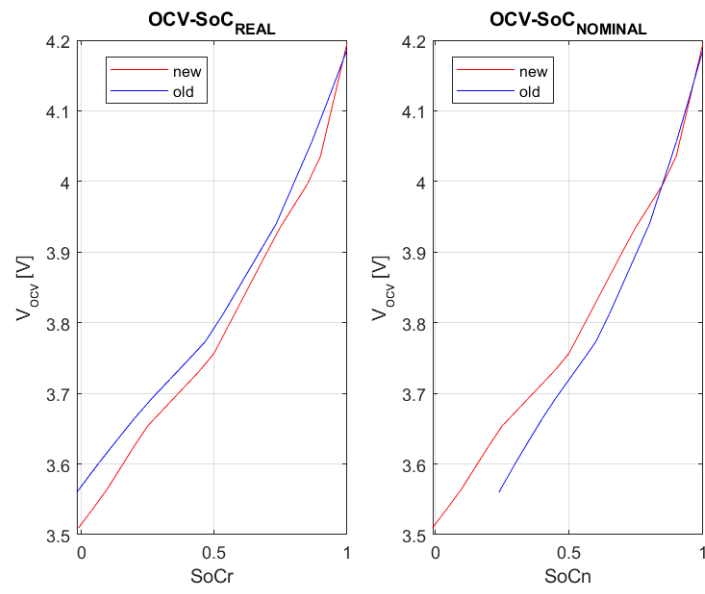


Figure 5.12: Comparison between SoC_N and SoC_R choice in the OCV-SoC characteristic

Chapter 6

Identification

The aim of this section is the detection of the simplest model consistent with the simulation performance required by a multi-model approach. The approach consists of the choice of a mathematical model and a fine-tuning of the process to obtain the best open-loop performance possible. Then, a closed-loop approach is exploited to improve the results. Another important factor is the resulting shape of the parameter LUTs: to investigate a multi-model approach, it is necessary the evidence of the correlation between parameters' changes. The identification must be guided to satisfy all the requirements. For simplicity, all the following tests are performed considering SoC_R were not specified. A different identification test is performed for each value of current and

6.1 Single RC, static model

The first step is to perform the static identification of the circuit. The simplifications done in this procedure cannot lead to a usable model, but the information regarding the static resistance can be useful for dynamical identification. The circuit used for the identification is reported in figure 6.1. The state-space equation representation of the system does not include dynamical terms:

$$V_T(k) = V_{OCV} + R_S I(k) \quad (6.1)$$

where V_{OCV} is extrapolated by the LUTs defined in the previous chapters. The algorithm used for the identification is the Least Square estimator. In this application, LS provides a punctual solution. Every data point leads to a different R_S value. The mathematical definition of the problem is:

$$R_S(SoC) = \frac{V_T(k) - V_{OCV}}{I(k)} \quad (6.2)$$

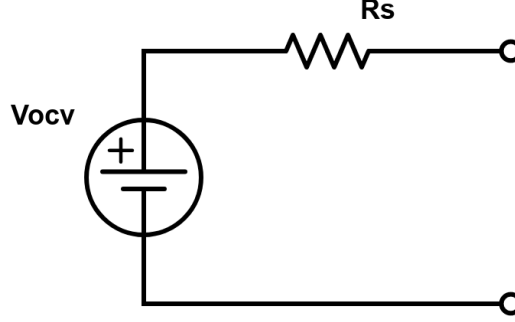


Figure 6.1: Circuit model for static identification

To not overfit the characteristic, for each period of the current (considered by the active and the idle part), the mean of the values is considered as an indicator. The SoC value associated with the point is the median data across the current period. The resulting LUTs are reported in figure 6.2.

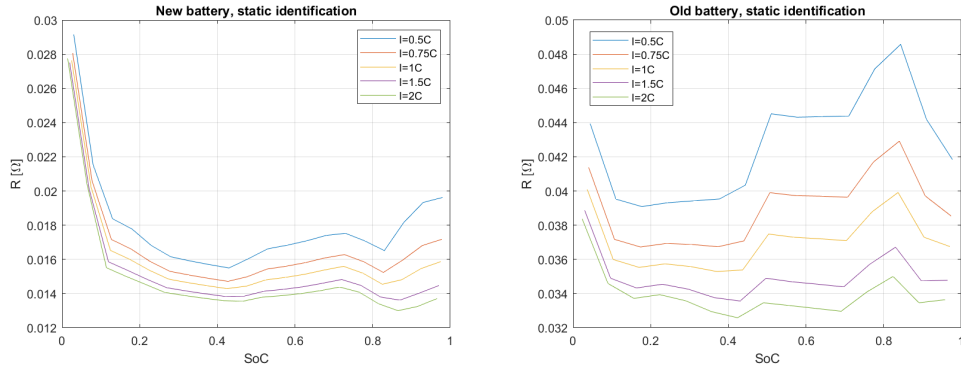


Figure 6.2: Static model parameters for old and new battery

6.2 Single RC, dynamical model using SoC_R

The circuit used as model is reported in figure 6.3. The dynamical part is defined by the single RC group. The state space representation of the system is:

$$\begin{cases} V_T(k) = V_{OCV} + R_0 i(k) + V(k) \\ V(k+1) = V(k) + [-\tau V(k) + \tau R i(k)] T_s \end{cases} \quad (6.3)$$

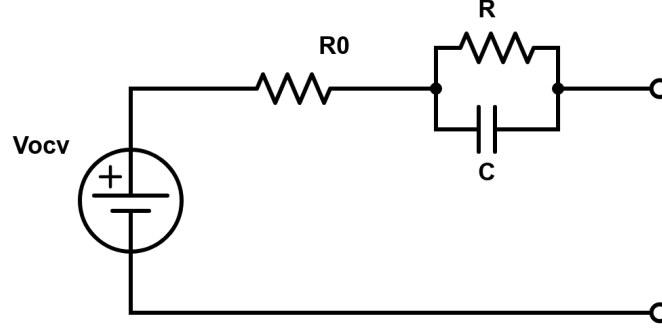


Figure 6.3: Circuit model for dynamical identification

The problem must be written according to LS theory. Firstly, $V(k)$ and $V(k-1)$ must be expressed as functions of know data exploiting the first equation:

$$V(k) = V_T(k) - V_{OCV}(k) - R_0 I(k) \quad (6.4)$$

$$V(k-1) = V_T(k-1) - V_{OCV}(k-1) - R_0 I(k-1) \quad (6.5)$$

Substituting in the second equation:

$$\begin{aligned} & V_T(k) - V_{OCV}(k) - V_T(k-1) + V_{OCV}(k-1) \\ &= (R_0 + R\tau T_s)I(k) - R_0 I(k-1) - \tau T_s [V_T(k-1) - V_{OCV}(k-1) - R_0 I(k-1)] \end{aligned} \quad (6.6)$$

From here, all the following methods write the problem in a slight different way.

6.2.1 Identification from static parameters

This approach exploits the knowledge of the static resistance R_S . It is possible to define a correlation between the static resistance and the two resistors present in the model:

$$R_S = R_0 + R \quad (6.7)$$

and consequently:

$$R = R_S - R_0 \quad (6.8)$$

Equation 6.6 can be rewritten as:

$$\begin{aligned} & V_T(k) - V_{OCV}(k) - V_T(k-1) + V_{OCV}(k-1) \\ &= (R_0 + R_S\tau T_s - R_0\tau T_s)I(k) - R_0 I(k-1) \\ &\quad - \tau T_s [V_T(k-1) - V_{OCV}(k-1) - R_0 I(k-1)] \end{aligned} \quad (6.9)$$

From the original three unknowns, the variable to be found become two: τ and R_0 . The successive step is to divide the parameters from the known data:

$$\begin{aligned} V_T(k) - V_{OCV}(k) - V_T(k-1) + V_{OCV}(k-1) \\ = I(k)[R_S\tau T_s + R_0 - R_0\tau T_s] + I(k-1)[-R_0 + \tau T_s R_0] \\ + [V_{OCV}(k-1) - V_T(k-1)]\tau T_s \end{aligned} \quad (6.10)$$

Defining three slack variables,

$$\begin{cases} \alpha = R_S\tau T_s + R_0 - R_0\tau T_s \\ \beta = \tau T_s R_0 - R_0 \\ \gamma = \tau T_s \end{cases} \quad (6.11)$$

the vector

$$Y = \begin{bmatrix} V_T(2) - V_{OCV}(2) - V_T(1) + V_{OCV}(1) \\ \dots \\ V_T(k) - V_{OCV}(k) - V_T(k-1) + V_{OCV}(k-1) \end{bmatrix} \quad (6.12)$$

and the vector

$$X = \begin{bmatrix} I(2) & I(1) & V_{OCV}(1) - V_T(1) \\ \dots & \dots & \dots \\ I(k) & I(k-1) & V_{OCV}(k-1) - V_T(k-1) \end{bmatrix} \quad (6.13)$$

The overall identification problem can be written as:

$$Y = X \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \quad (6.14)$$

and the original parameter is expressed as a function of the three slack variables. The values of R_s are defined by the static model identification. It is important to notice that the system is over-determined, so there are two expressions for R_0 :

$$\tau = \frac{\gamma}{T_s} \quad (6.15)$$

$$R_0 = \frac{\beta}{\tau T_s - 1} = \frac{\alpha - R_S\tau T_s}{1 - \tau T_s} \quad (6.16)$$

$$R = R_S - R_0 \quad (6.17)$$

The obtained estimator is applied to the data collected in chapter 5. A single iteration of the algorithm is applied to each current period. The resulting LUTs are reported in figure 6.4 and 6.5. As for the static case, the SoC is reported as the median value observed during the considered interval.

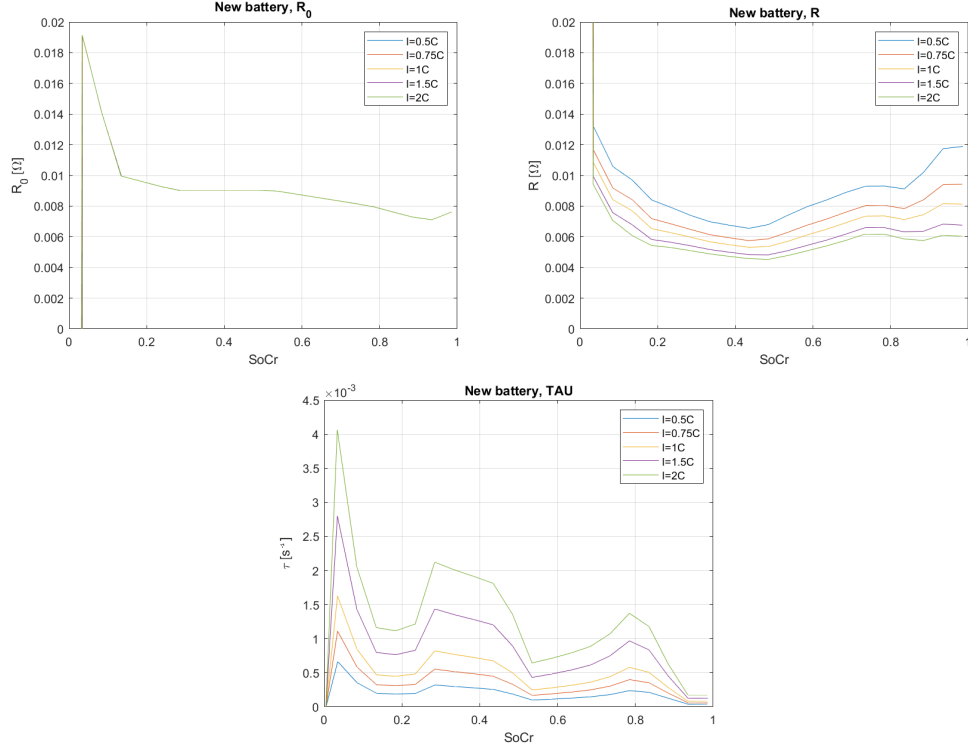


Figure 6.4: Dynamical model parameters of the new battery calculated using the mixed static-dynamic approach

6.2.2 Direct dynamical model identification

This approach tries to avoid the need for two separate steps to identify the parameters. Starting from equation 6.6, the LS theory is developed conserving all the three unknown parameters τ , R_0 , and R . The parameters are divided from the collected known data:

$$\begin{aligned} V_T(k) - V_{OCV}(k) - V_T(k-1) + V_{OCV}(k-1) \\ = (R_0 + R\tau T_s)I(k) + (R_0\tau T_s - R_0)I(k-1) - (V_{OCV}(k-1) - V_T(k-1))\tau T_s \end{aligned} \quad (6.18)$$

Defining the three slack variables:

$$\begin{cases} \alpha = R_0 + R\tau T_s \\ \beta = R_0\tau T_s - R_0 \\ \gamma = \tau T_s \end{cases} \quad (6.19)$$

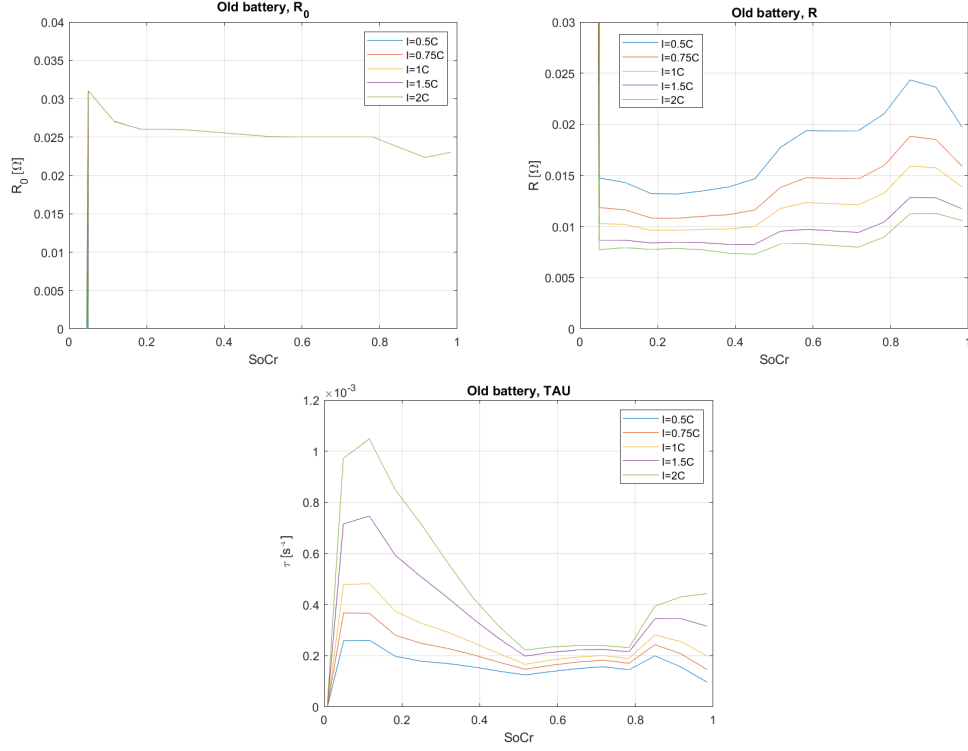


Figure 6.5: Dynamical model parameters of the old battery calculated using the mixed static-dynamic approach

the vector of known values

$$Y = \begin{bmatrix} V_T(2) - V_{OCV}(2) - V_T(1) + V_{OCV}(1) \\ \dots \\ V_T(k) - V_{OCV}(k) - V_T(k-1) + V_{OCV}(k-1) \end{bmatrix} \quad (6.20)$$

and the vector

$$X = \begin{bmatrix} I(2) & I(1) & V_{OCV}(1) - V_T(1) \\ \dots & \dots & \dots \\ I(k) & I(k-1) & V_{OCV}(k-1) - V_T(k-1) \end{bmatrix} \quad (6.21)$$

The overall identification problem can be written as:

$$Y = X \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \quad (6.22)$$

and the original parameter are expressed as function of the three slack variables

$$\tau = \frac{\gamma}{T_s} \quad (6.23)$$

$$R_0 = \frac{\beta}{\tau T_s - 1} \quad (6.24)$$

$$R = \frac{\alpha - R_0}{\tau T_s} \quad (6.25)$$

Now the identification framework is ready to be applied to the simulation data collected in chapter 5. Two possible subsets of the dataset are considerable:

- *Approach 1*: only the active part of the current period is considered.
- *Approach 2*: the overall current period is considered.

The resulting LUTs are reported in figure 6.6 and 6.8 for *Approach 1* and in figure 6.7 and 6.9 for *Approach 2*.

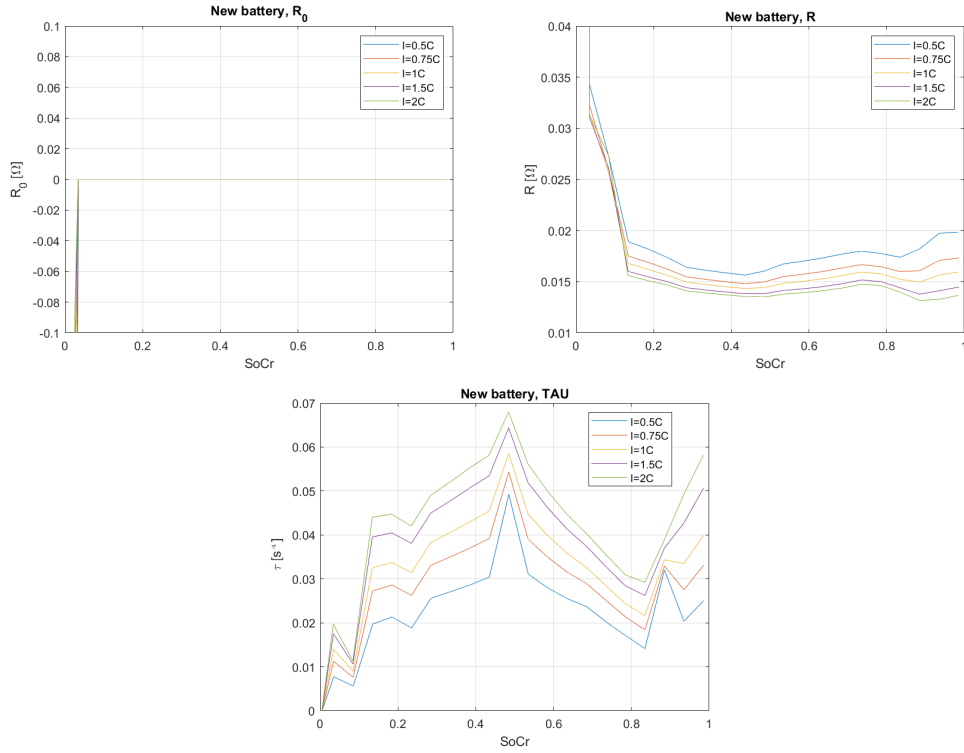


Figure 6.6: Dynamical model parameters of the new battery calculated using direct *Approach 1*

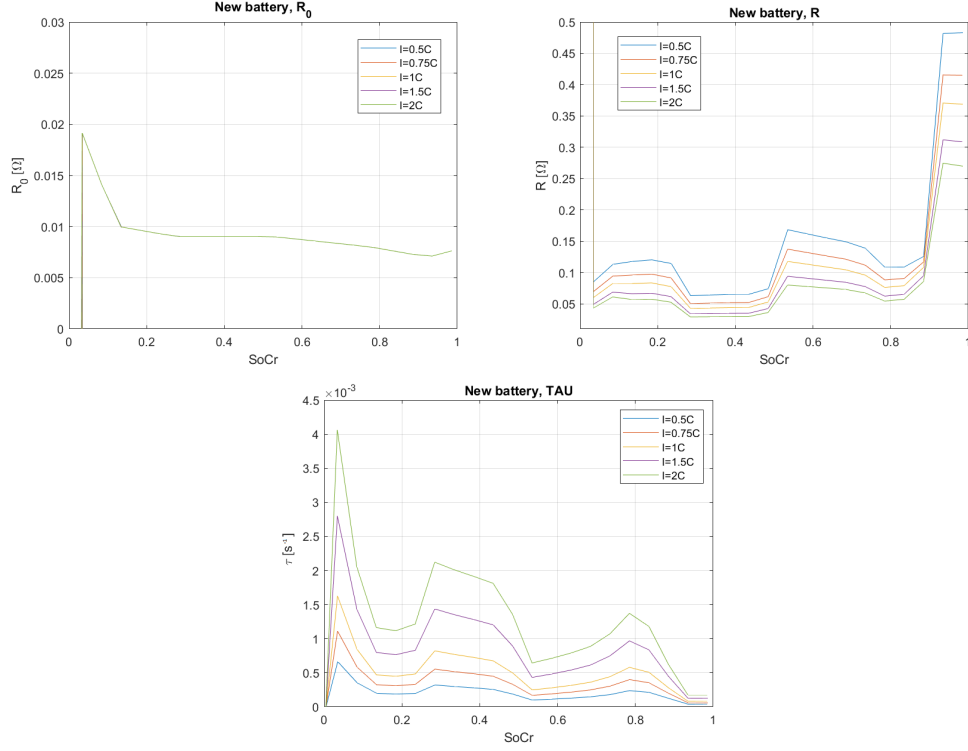


Figure 6.7: Dynamical model parameters of the new battery calculated using direct *Approach 2*

6.2.3 Results

The results achieved with the two-step identification are similar to the direct identification *Approach 2*. At this point, a clear preference cannot be expressed, but the complexity of the two-step identification is higher than other solutions. Comparing the direct identification, *Approach 1* has a problem in the estimation of R at low SoC. A negative value for the resistance is not consistent with the general electro-technical theory. The same approach leads to a non-observability of the series resistance R_0 , that for both batteries is fixed at zero. The comparison leads to the choice of *Approach 2 Direct Identification* as the best trade-off identification model for this section. In all the experiments, the resulting LUTs have to be filtered to delete outlier values.

6.3 Single RC, dynamical model using SoC_N

To guarantee a simpler presentation, only the results achieved with *Approach 2 Direct Identification* are proposed. The procedure is the same as presented in the

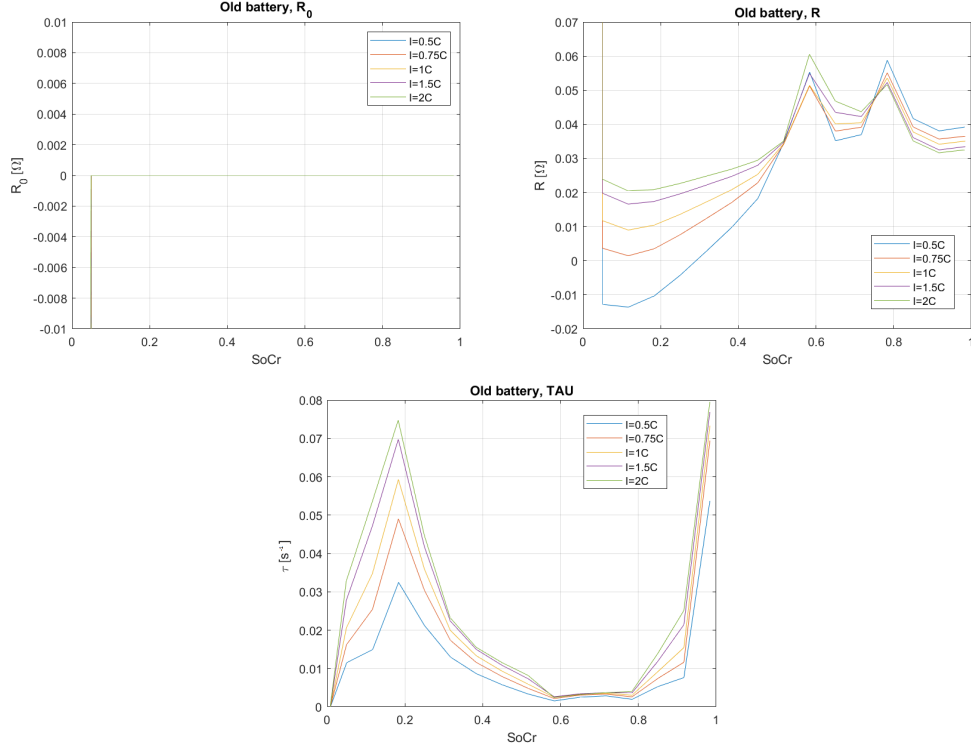


Figure 6.8: Dynamical model parameters of the old battery calculated using direct *Approach 1*

previous paragraph. The only difference is the use of SoC_N instead of SoC_R . The result is displayed in figure 6.10 and 6.11. For the new battery, it is valid the relation:

$$C_R = C_N \quad (6.26)$$

The tables relative to the new battery are equivalent to the previous one. The same cannot be seen for old batteries ones. The LUTs are translated into a set of sub-spaces of the previous graphs. There is not a clear preference for one approach, so both of them are developed in the next sections.

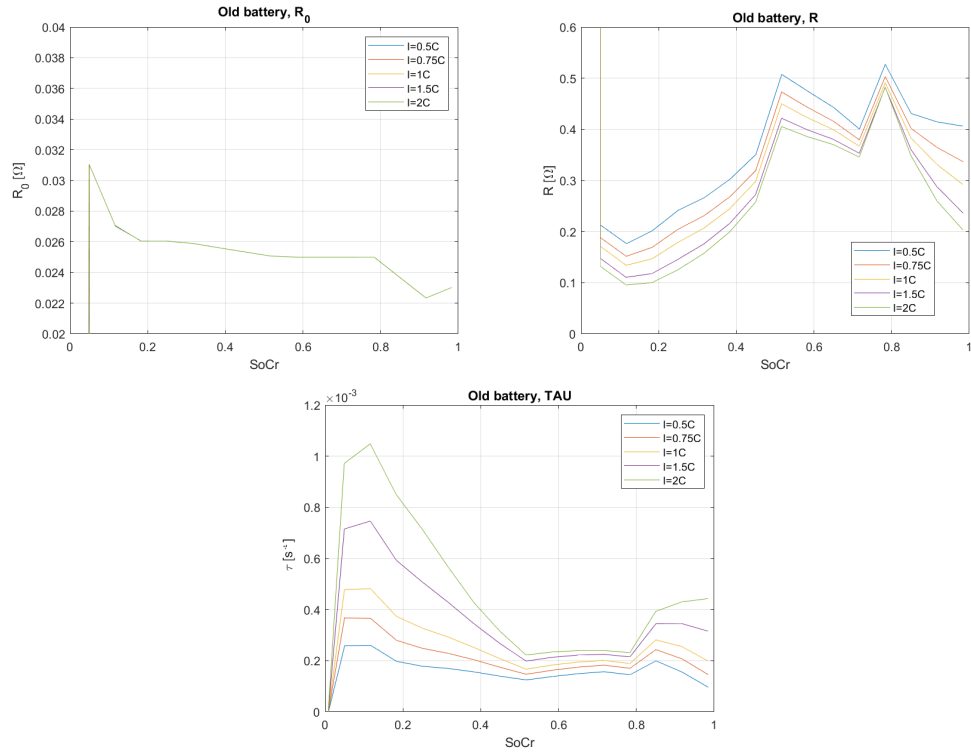


Figure 6.9: Dynamical model parameters of the old battery calculated using direct *Approach 2*

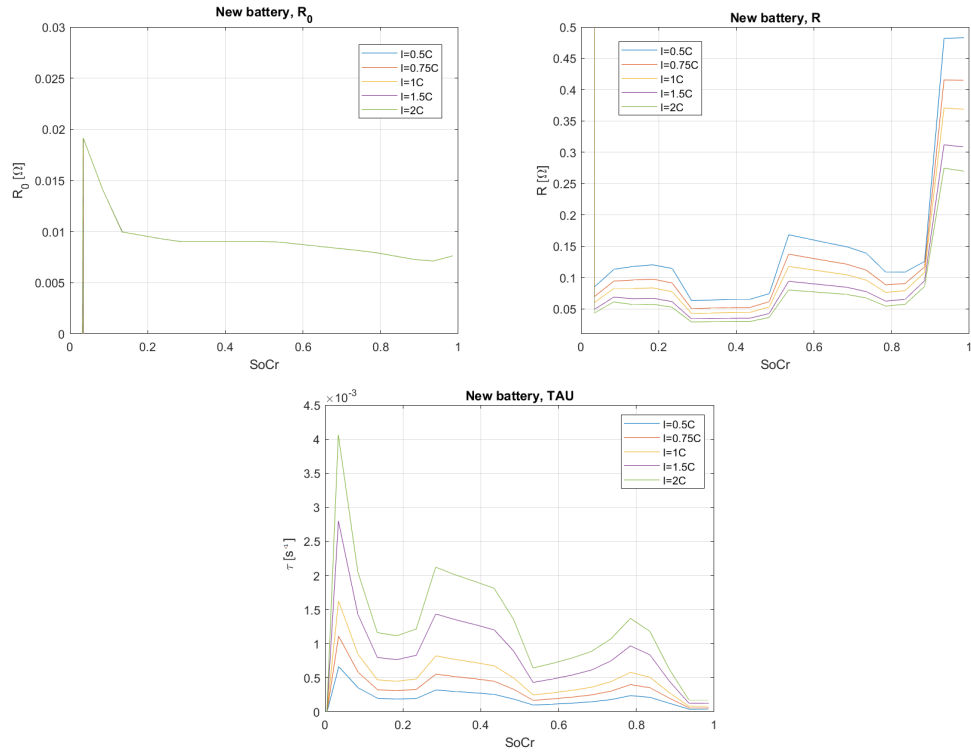


Figure 6.10: Dynamical model parameters of the new battery calculated using direct *Approach 2* and SoC_N

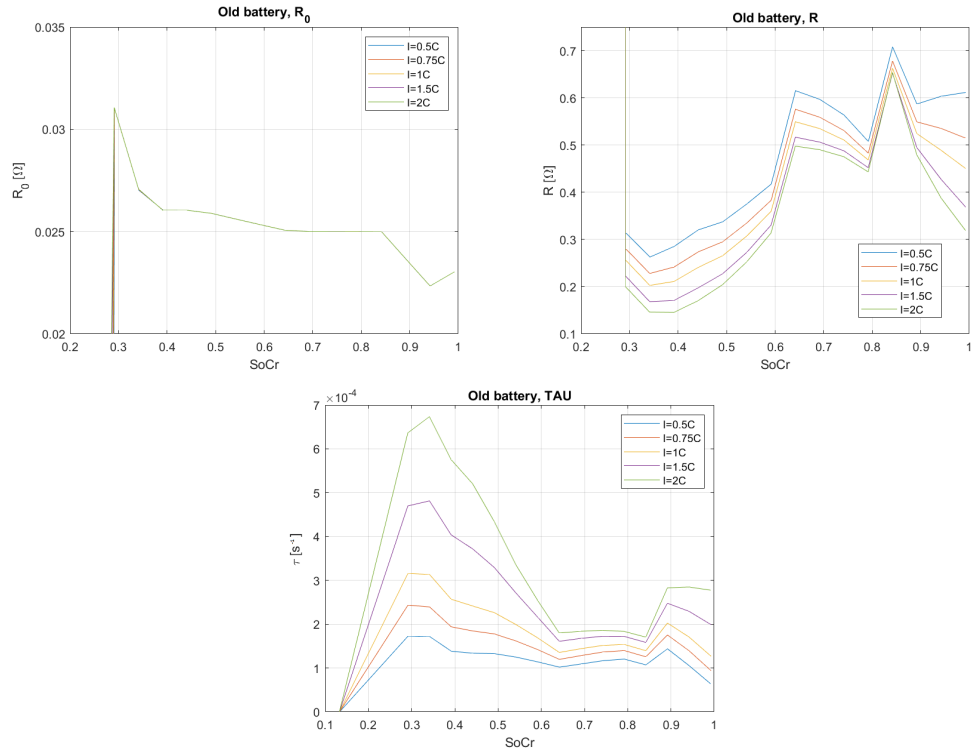


Figure 6.11: Dynamical model parameters of the old battery calculated using direct *Approach 2* and SoC_N

Chapter 7

Validation

The aim of this section is the validation of the models identified in the previous chapter with some validation datasets, different from the ones previously used. The analysis is performed through open-loop and closed-loop simulations. The closed-loop approach permits indirectly correcting part of the error generated by the identification process. The data are generated through the following procedure:

- The battery is completely charged at the beginning of the cycle.
- A square wave current is applied, with variable period and duty-cycle, but constant amplitude. The current values used in the experiment are the same used for the generation of the datasets used during the identification process.
- The test is interrupted when the battery is empty.

The validation current datasets are displayed in figure 7.1 and 7.2. The current is negative for the convention displayed in figure 5.1: an entering current generates a charging process.

7.1 Open loop validation

The parameter values used in the simulation are reported in figure 6.7 and 6.9. The OCV is displayed in figure 5.12. The test is performed using the current of the validation datasets as input. The initial SoC is fixed at one. Knowing with high precision the current and the real capacity of the cells, the uncertainty on the SoC is assumed as null. The parameters are calculated and updated online. The state-space model exploited is:

$$\begin{cases} V(k+1) = V(k) + [-\tau V(k) + \tau R i(k)]T_s \\ SoC(k+1) = SoC(k) + \frac{T_s}{3600 \cdot Capacity} i(k) \\ V_T(k) = V(k) + V_{OCV} + R_0 i(k) \end{cases} \quad (7.1)$$

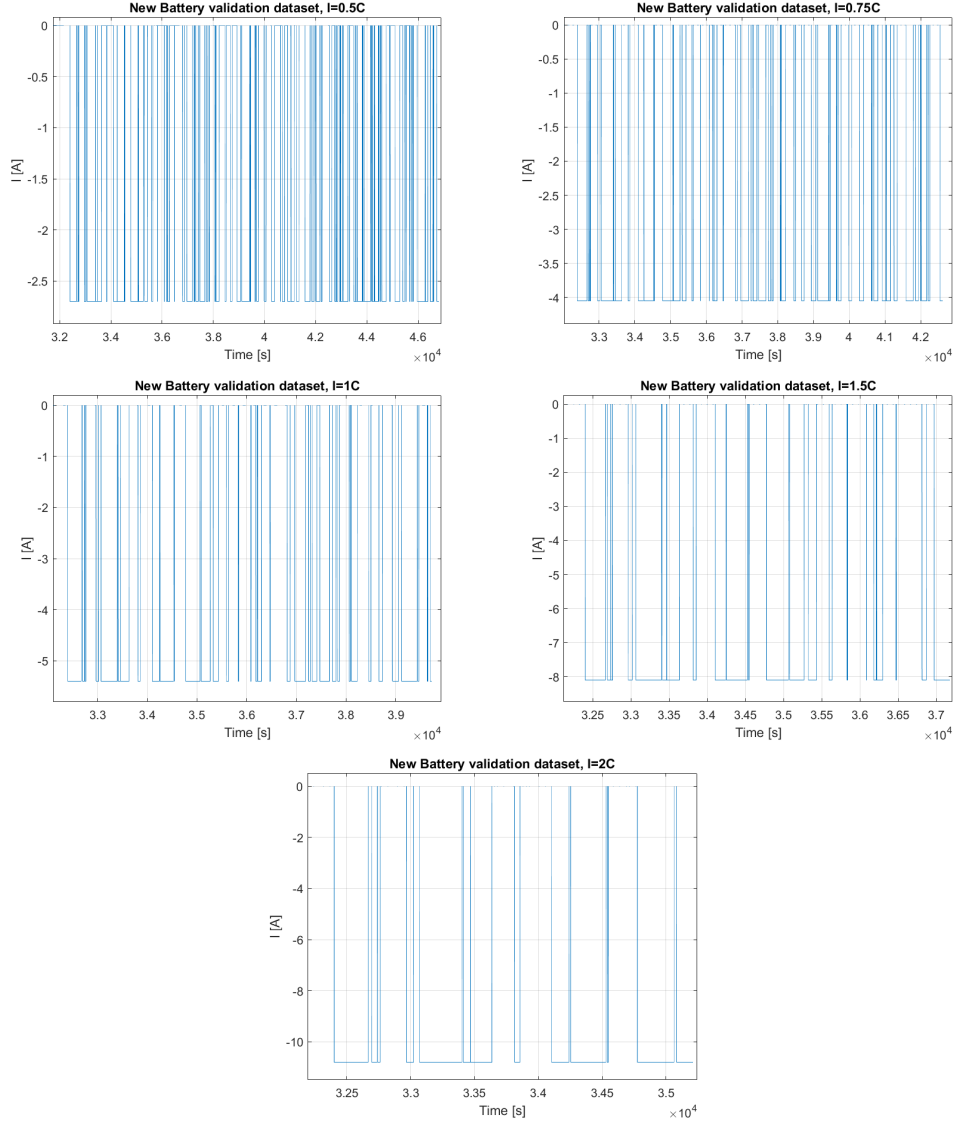


Figure 7.1: Validation dataset currents for new battery

7.1.1 New battery validation

The results of the simulation are reported in figure 7.3. The graph displays a direct comparison between the real data and the simulated ones. A single separate simulation is performed for each current value. The RMSE values of the validation are reported in table 7.1. The $RMSE\ SoC = [0.95, 0.05]$ is the value calculated in the region indicated. The necessity of a subset for the definition of the RMSE is due to the divergence of the model at low SoC values. This drawback can be

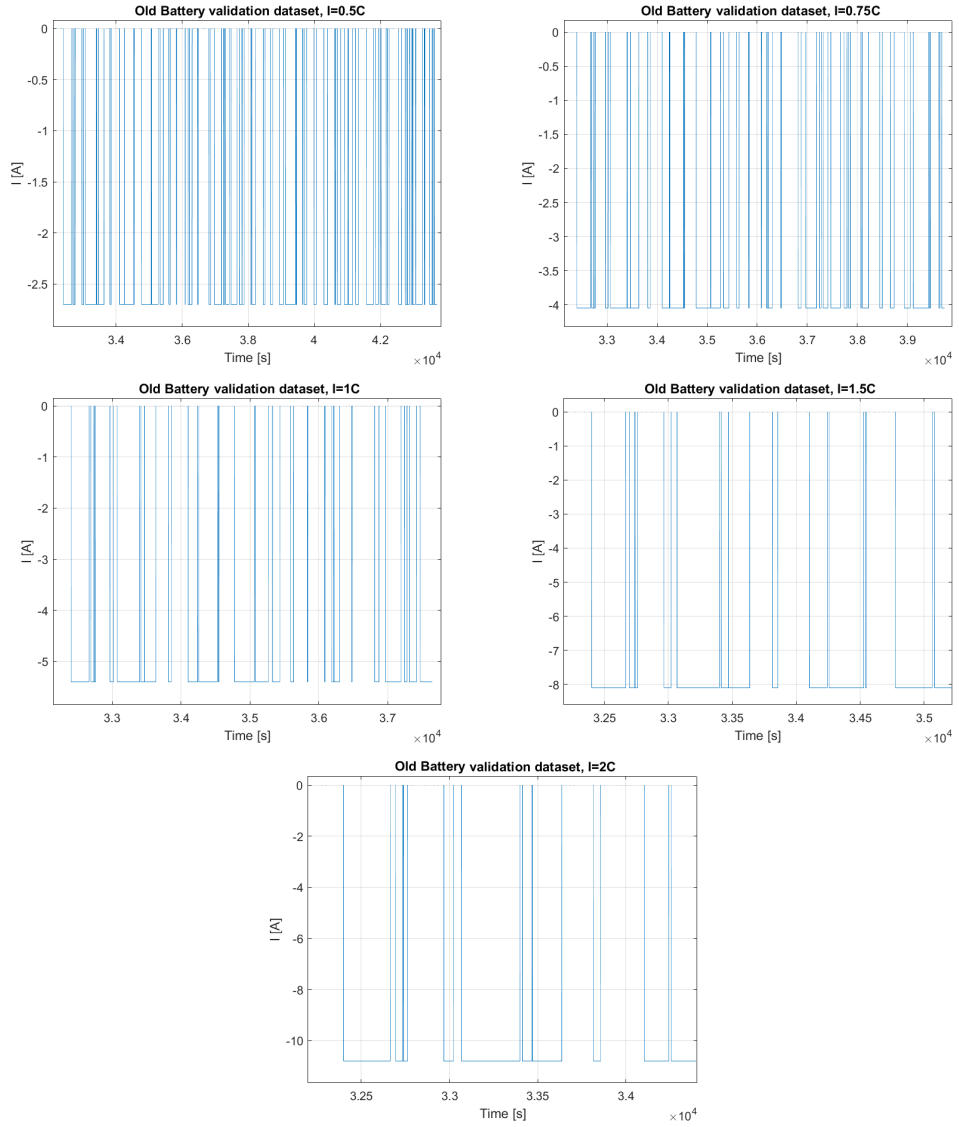


Figure 7.2: Validation dataset currents for old battery

removed by performing an accurate normalization of the LUTs deleting the outliers.

7.1.2 Old battery validation

The results of the simulation are reported in figure 7.4. The graph displays a direct comparison between the real data and the simulated ones. A single separate simulation is performed for each current value. The RMSE values of the validation

Current	RMSE	RMSE [0.95,0.05]
0.5C	3765	0.0828
0.75C	6929	0.1054
1C	9779	0.1198
1.5C	10316	0.1649
2C	17728	0.1953

Table 7.1: RMSE values for open loop validation for new battery model

are reported in table 7.2. The $RMSE\ SoC = [0.95, 0.05]$ is the value calculated in the region indicated. The necessity of a subset for the definition of the RMSE is due to the divergence of the model at low SoC values. This drawback can be removed by performing an accurate normalization of the LUTs deleting the outliers.

Current	RMSE	RMSE [0.95,0.05]
0.5C	2234	0.0875
0.75C	4115	0.1038
1C	3176	0.1330
1.5C	4842	0.1981
2C	6726	0.2317

Table 7.2: RMSE values for open loop validation for old battery model

7.1.3 Results

The open-loop performance is not sufficient for an accurate simulation of the cell behavior. The restricted RMSE value seems to be low enough, but from the graphs, it is clear that the two datasets are too far from each other. This is due to the small relaxation of the curves compared to the overall voltage value at the terminal. A closed-loop approach is needed to reduce uncertainties.

7.2 Minimum number of points

Until this point, the choice of twenty sample points to characterize the LUTs of the parameters may appear not motivated. This section aims to justify the choice. The test follows the same procedure displayed in the previous chapters. For each set of points, the following steps are performed:

- Current pulse simulations with N steps, as done in chapter 5.

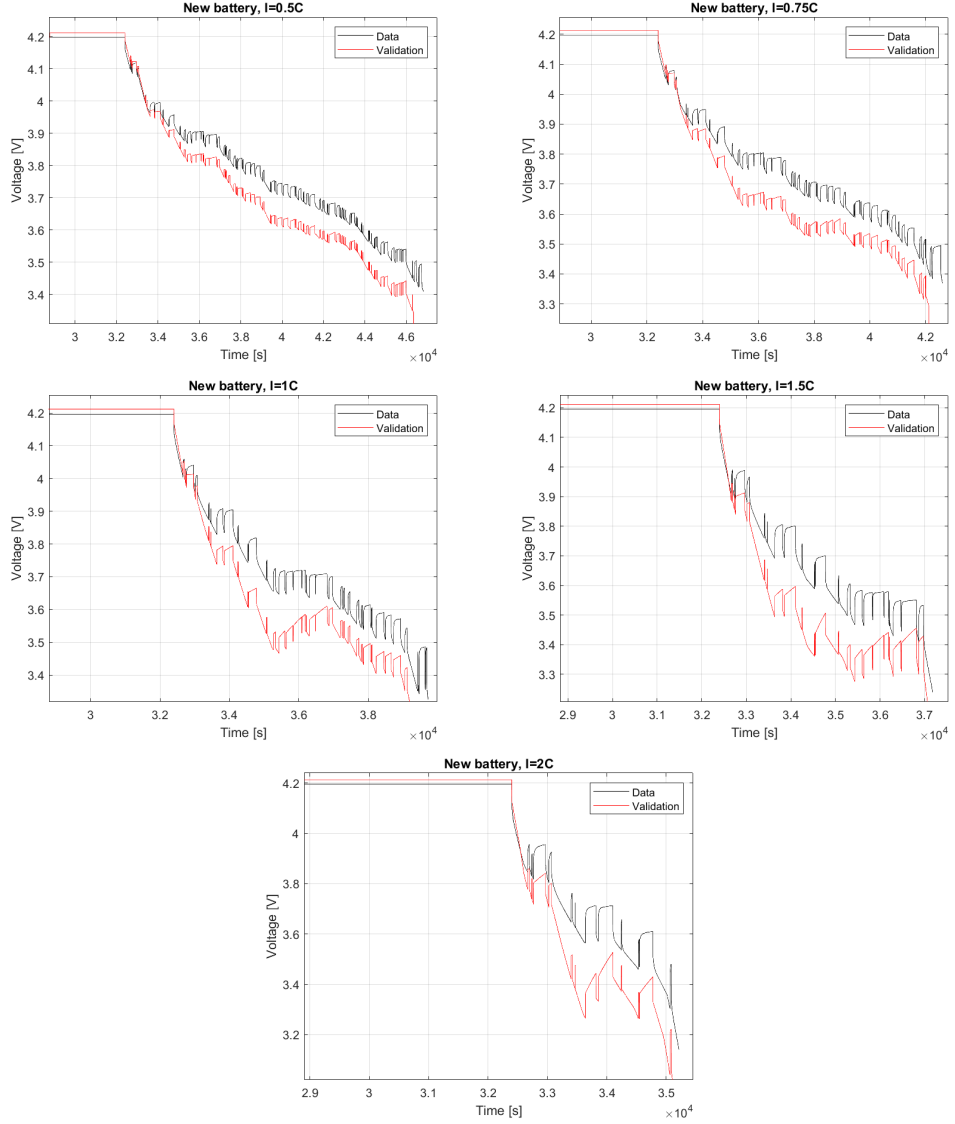


Figure 7.3: Open loop validation of new battery model

- Identification of the parameters of the single RC equivalent model following direct dynamic approach.
- Open loop validation and RMSE computation.

The results are displayed in table 7.3, where only the RMSE values for the restricted interval $SoC = [1, 0.1]$ are displayed. $N = 10$ has better performance for the new battery and a not-consistent one for the old. $N = 100$ suffers from overfitting concerning $N = 20$. The choice done in the results of the previous chapter is to be

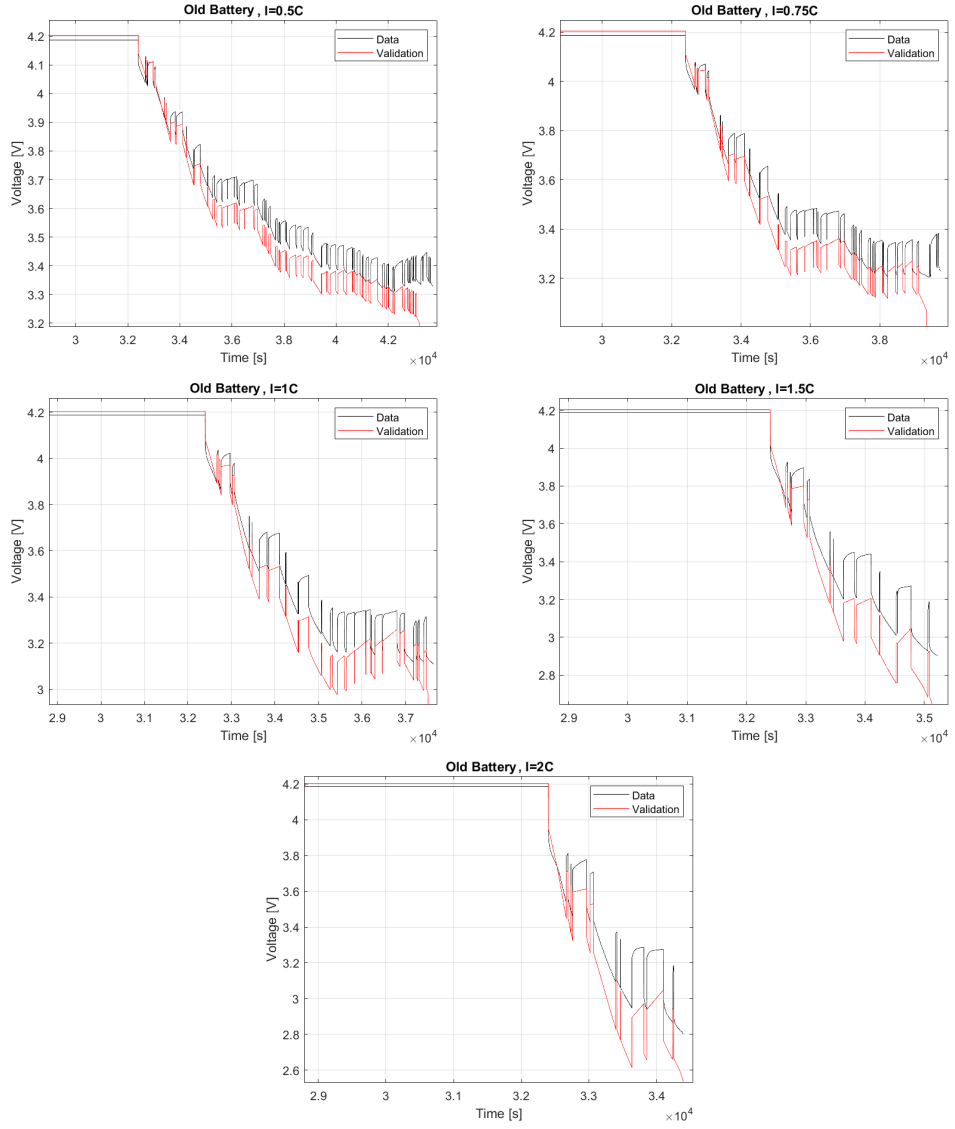


Figure 7.4: Open loop validation of old battery model

N	New Battery					Old Battery				
	0.5C	0.75C	1C	1.5C	2C	0.5C	0.75C	1C	1.5C	2C
10	0.032	0.041	0.043	0.053	0.053	247.2	221.1	275.7	192.1	193.9
20	0.069	0.088	0.095	0.122	0.121	0.068	0.082	0.099	0.0123	0.0135
100	0.348	0.457	0.512	0.696	0.636	0.370	0.425	0.444	0.432	0.429

Table 7.3: RMSE values for some N-steps identification tests

the best trade-off between complexity and performance.

7.3 Closed loop validation

The performance of the open loop simulations is not consistent with the precision required for the identification of SoH. To reduce the error, an observer is introduced. The analysis is performed using an Extended Kalman Filter.

7.3.1 Observability analysis

The observability of a non-linear system can be studied graphically through the inference diagram. [17] The diagram is composed of states and arcs. An arc from x_i to x_j defines a direct dependence on x_j in the expression of x_i . The states are called Strongly Connected Components SCC if connected with arcs to the overall graph, and Root Strongly Connected Components RSCC if arcs only depart from the state. The necessary and sufficient condition for observability is the possibility to measure at least one state for each RSCC subtree. Reporting the model exploited in the Open Loop analysis following the state-space notation: $x_1 = V$, $x_2 = SoC$, $y = V_T$

$$\begin{cases} x_1(k+1) = x_1(k) + [-\tau x_1(k) + \tau Ru(k)]T_s \\ x_2(k+1) = x_2(k) + \frac{T_s}{3600 \cdot Capacity} u(k) \\ y(k) = x_1(k) + V_{OCV} + R_0 u(k) \end{cases} \quad (7.2)$$

The inference diagram is reported in figure 7.5. From the diagram, it is not possible

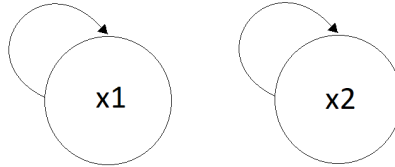


Figure 7.5: Inference diagram of the original state-space model

to determine the observability property. An augmented state model is proposed:

$$\begin{cases} x_1(k+1) = x_1(k) + [-\tau x_1(k) + \tau Ru(k)]T_s \\ x_2(k+1) = x_2(k) + \frac{T_s}{3600 \cdot Capacity} u(k) \\ x_3(k+1) = x_1(k+1) + V_{OCV} + R_0 u(k) \\ y(k) = x_3(k) \end{cases} \quad (7.3)$$

$x_1(k+1)$ can be substituted with the first relation in the model. The third equation becomes:

$$\begin{aligned} x_3(k+1) &= x_1(k) + [-\tau x_1(k) + \tau R u(k)]T_s + V_{OCV} + R_0 u(k) \\ &= [1 - \tau T_s]x_1(k) + [\tau R T_s + R_0]u(k) + V_{OCV} \end{aligned} \quad (7.4)$$

The inference diagram is reported in figure 7.6. It is clear that x_3 is an RSCC and

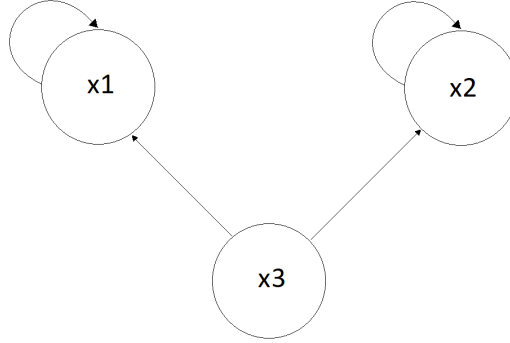


Figure 7.6: Inference diagram of the augmented state-space model

it can be measured. This proves the observability of the model. In the closed-loop validation, the augmented states are employed.

7.3.2 Extended Kalman Filter

The EKF is the extension of the Kalman Filter to non-linear systems. It is a covariance estimator, but, unlike Kalman Filter, it is not an optimal estimator. It linearizes the non-linear problem around the estimated covariance and mean. The closed loop configuration is displayed in figure 7.7. The EKF algorithm is provided in algorithm 1. Assuming the model is expressed through differentiable functions:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \quad (7.5)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (7.6)$$

where \mathbf{w}_k and \mathbf{v}_k are the process and observation noise assumed to be zero mean multivariate Gaussian noises with covariance matrices \mathbf{Q}_k and \mathbf{R}_k . The notation $\hat{\mathbf{x}}_{n|m}$ represents the estimate of \mathbf{x} at time n , given the observed data at time $m \leq n$. The transition matrices are defined as:

$$\mathbf{A}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k} \quad (7.7)$$

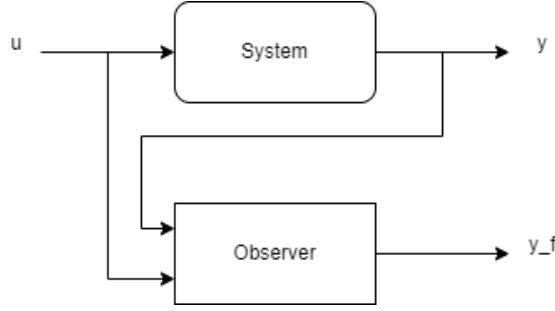


Figure 7.7: Closed-loop configuration

$$\mathbf{C}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}} \quad (7.8)$$

According to the previous model definition, the transfer functions are determined

Algorithm 1 EKF algorithm

Predict

Predicted state estimate

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k)$$

Predicted covariance estimate

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k$$

Update

Innovation

$$\mathbf{e}_k = \mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})$$

Innovation covariance

$$\mathbf{S}_k = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k$$

Near-optimal Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \mathbf{S}_k^{-1}$$

Updated state estimate

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k$$

Updated covariance estimate

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_{k|k-1}$$

by:

$$f(\mathbf{x}, u) = \begin{bmatrix} x_1[1 - \tau T_s] + \tau R T_s u \\ x_2 + \frac{T_s}{3600 \cdot Capacity} u \\ [1 - \tau T_s]x_1 + [\tau R T_s + R_0]u + V_{OCV} \end{bmatrix} \quad (7.9)$$

$$h(\mathbf{x}) = x_3 \quad (7.10)$$

The partial derivatives are calculated

$$\frac{\partial f_1}{\partial x_1} = 1 - \tau T_s, \quad \frac{\partial f_1}{\partial x_2} = 0, \quad \frac{\partial f_1}{\partial x_3} = 0 \quad (7.11)$$

$$\frac{\partial f_2}{\partial x_1} = 0, \quad \frac{\partial f_2}{\partial x_2} = 1, \quad \frac{\partial f_2}{\partial x_3} = 0 \quad (7.12)$$

$$\frac{\partial f_3}{\partial x_1} = 1 - \tau T_s, \quad \frac{\partial f_3}{\partial x_2} = 0, \quad \frac{\partial f_3}{\partial x_3} = 0 \quad (7.13)$$

$$\frac{\partial h}{\partial x_1} = 1, \quad \frac{\partial h}{\partial x_2} = 0, \quad \frac{\partial h}{\partial x_3} = 1 \quad (7.14)$$

and the Jacobians

$$A = \begin{bmatrix} 1 - \tau T_s & 0 & 0 \\ 0 & 1 & 0 \\ 1 - \tau T_s & 0 & 0 \end{bmatrix} \quad (7.15)$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad (7.16)$$

The matrices appear time-invariant. It is not the correct EKF filter, because the parameters are considered invariant during the design process. This simplification will be removed in the next chapters, where a clear dependence on SoC is defined. The model does not include physical boundaries to the states, so the filtered output is not correct. The main problem is the correction of the voltage on the capacitor with a positive factor also when the current is negative, during the discharging processes. Another problem is the limitation of the SoC to the interval $[0, 1]$ or similar values. To overcome the problem a constrained EKF is considered.

7.3.3 Constrained Extended Kalman Filter

There are two possible solution to the problem. [34, 35] The first is the restriction of the optimal Kalman gain to a feasible sub-space. The problem is written as:

$$\begin{aligned} \hat{K}_k^R &= \underset{\text{argmin}}{\text{trace}}[(I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T] \\ \text{s.t. } &A(\hat{x}_{k|k-1} + K_k v_k) = b \\ &C(\hat{x}_{k|k-1} + K_k v_k) \leq d \end{aligned} \quad (7.17)$$

The CEFK requires the solution of the optimization problem each cycle. The computational complexity is too high for real-time time application. The second approach is the projection of the unconstrained solution to a feasible set. It also involves an optimization problem:

$$\begin{aligned} \hat{x}_{k|k}^P = \operatorname{argmin} (x - \hat{x}_{k|k})^T W_k (x - \hat{x}_{k|k}) \\ \text{s.t. } Ax = b \\ Cx \leq d \end{aligned} \quad (7.18)$$

The solution seems to be computational expensive as the previous one, but observing the constraints specific for this problem:

$$x_1 \leq 0, 0 \leq x_2 \leq 1 \quad (7.19)$$

the feasible sets appear to be convex and mono-dimensional. The nearest point to a non-feasible point lies on the boundary. This characteristic of the problem is exploited to write the optimization problem with a simple *if-then-else* construct.

```

1  % Constraint on x1
2  if x1 > 0
3      x1 = 0;
4  end
5  % Constraint on x2
6  if x2 < 0
7      x2 = 0;
8  elseif x2 > 1
9      x2 = 1;
10 end

```

7.3.4 New battery validation

The results of the simulation are reported in figure 7.8. The graph displays a direct comparison between the real data and the simulated ones. A single separate simulation is performed for each current value. The RMSE values of the validation are reported in table 7.4. The $RMSE\ SoC = [0.95, 0.05]$ is the value calculated in the region indicated. The performance of the two RMSE indexes is almost equivalent.

7.3.5 Old battery validation

The results of the simulation are reported in figure 7.9. The graph displays a direct comparison between the real data and the simulated ones. A single separate

Current	RMSE	RMSE [0.95,0.05]
0.5C	0.0136	0.0091
0.75C	0.0121	0.0085
1C	0.0114	0.0085
1.5C	0.0107	0.0090
2C	0.0104	0.0098

Table 7.4: RMSE values for closed loop validation for new battery model

simulation is performed for each current value. The RMSE values of the validation are reported in table 7.5. The $RMSE\ SoC = [0.95, 0.05]$ is the value calculated in the region indicated. The performance of the two RMSE index are almost equivalent.

Current	RMSE	RMSE [0.95,0.05]
0.5C	0.0160	0.0155
0.75C	0.0139	0.0124
1C	0.0125	0.0110
1.5C	0.0111	0.0101
2C	0.0084	0.0084

Table 7.5: RMSE values for closed loop validation for old battery model

7.4 Comparison between OL and CL approach

The closed-loop approach is highly more precise than the open-loop one. From the RMSE table, it could be seen values relative to OL are about seven orders of magnitude greater than CL. Also considering the filtered subset, the performance of CL are more effective than OL. This section highlights the necessity of an observer to guarantee low errors exploiting a low-order model. On the other hand, CL is more computationally-expensive than OL, but better performance can justify an increased complexity. With CL, the original data are completely tracked by the single RC model. The CL approach will be exploited in the next chapters to develop a multi-model approach to the estimation of SoH.

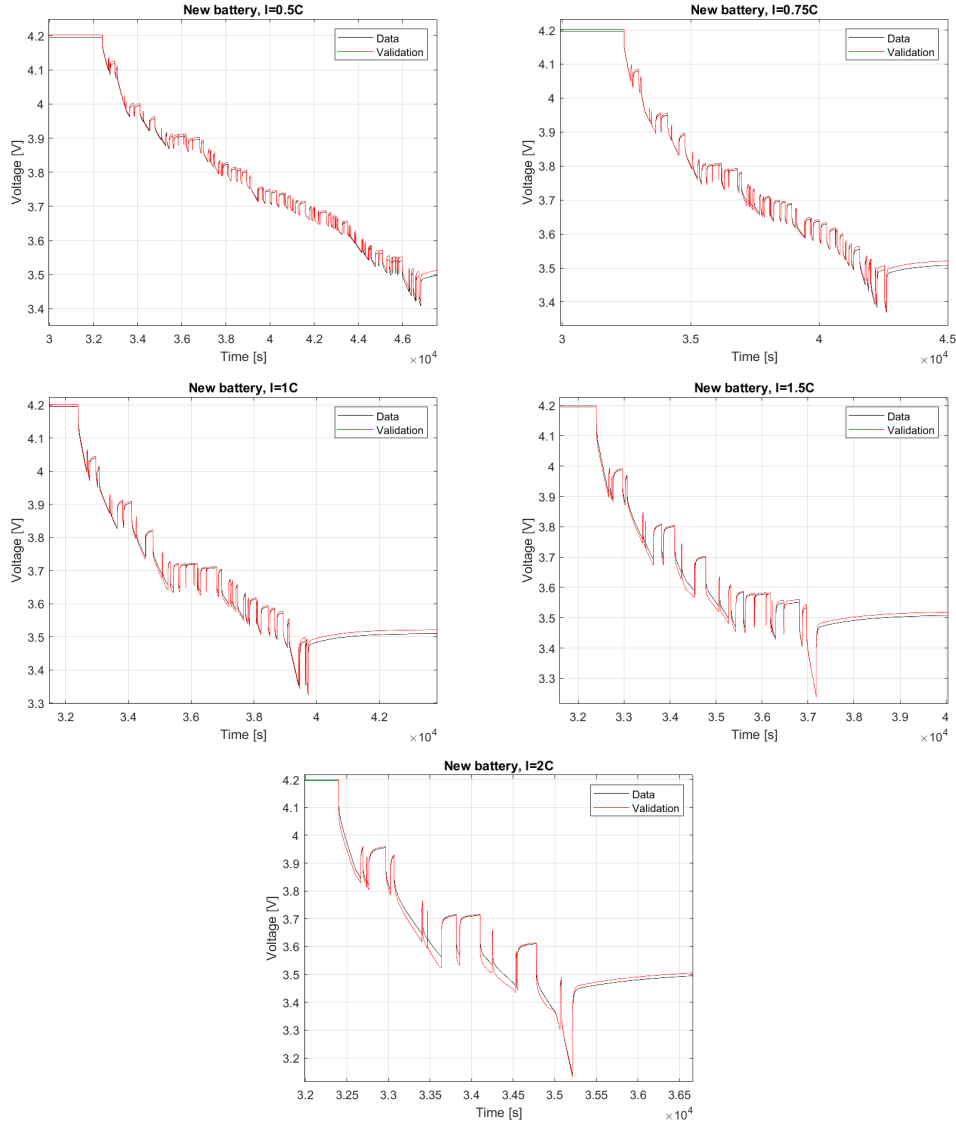


Figure 7.8: Closed loop validation of new battery model

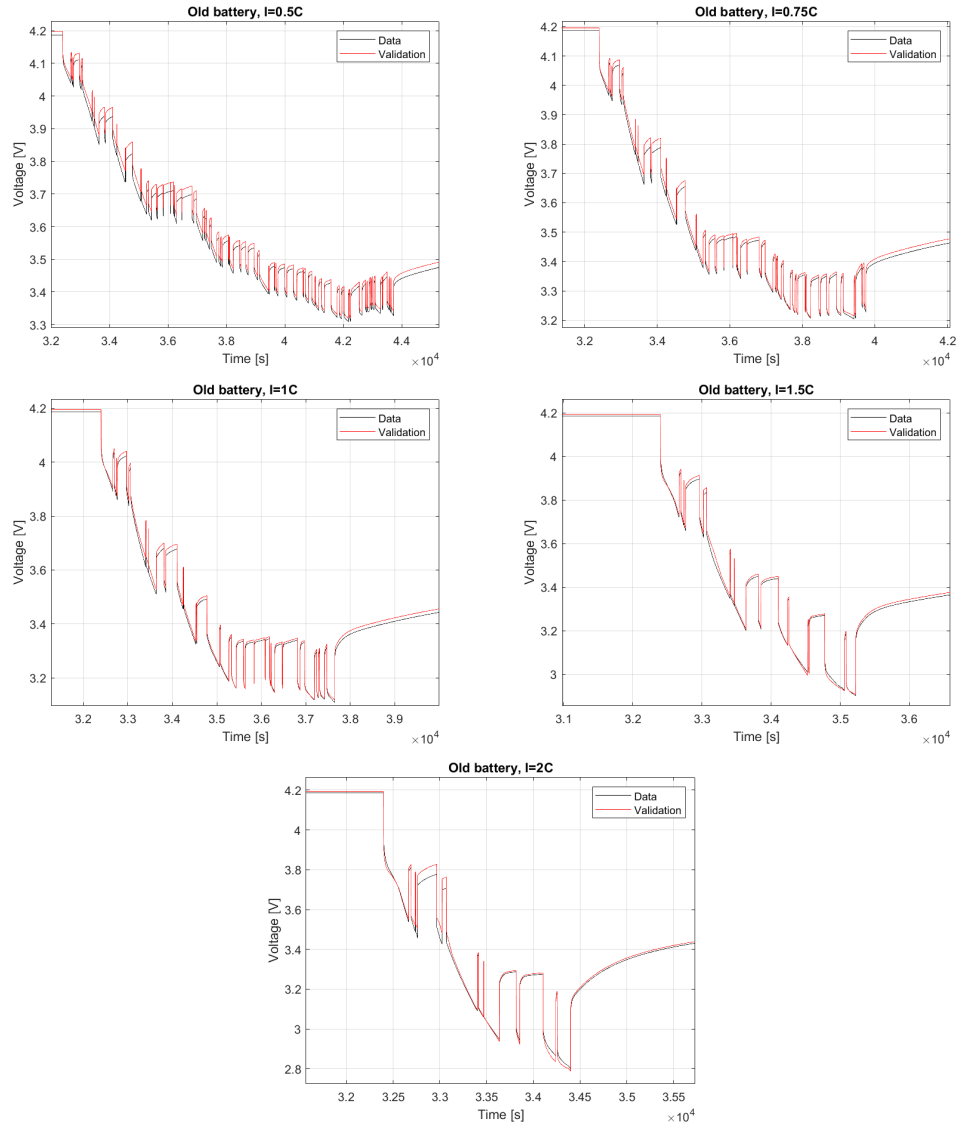


Figure 7.9: Closed loop validation of old battery model

Chapter 8

Multi-model approach

In the previous sections, the old and the new batteries are considered as two separate entities and not as a single cell in two different states. As explained in chapter 3, the most exploited approach is to consider different sets of LUTs parameters for different SoH values. It is the simplest way to simulate a previously characterized battery, but nothing could be told for a new cell under test. This section aims to provide a simple and computationally light solution that includes the dependency of the parameters on SoC, SoH, and current.

8.1 Analysis of SoC and Current dependencies considering SoC_R

Considering the LUTs generated during *Identification phase*, the idea is to simplify the graph with the simplest possible functions. The approximation cannot be perfect, but the CL approach guarantees good performances also with a simplified set of parameters. The LUTs of the new and old batteries are reported in figure 8.1.

8.1.1 V_{OCV} approximation

V_{OCV} curves do not present any dependence on the current applied to the cell. The line related to the new battery is very similar to the old one. The resulting curve can be linearly approximated by a straight line dependent on SoC.

$$V_{OCV}(SoC) = p_1^{V_{OCV}} SoC + p_2^{V_{OCV}} \quad (8.1)$$

The coefficients $p_1^{V_{OCV}}$ and $p_2^{V_{OCV}}$ are determined calculating the straight line passing through the points $SoC = 0$ and $SoC = 1$. Two independent lines are determined

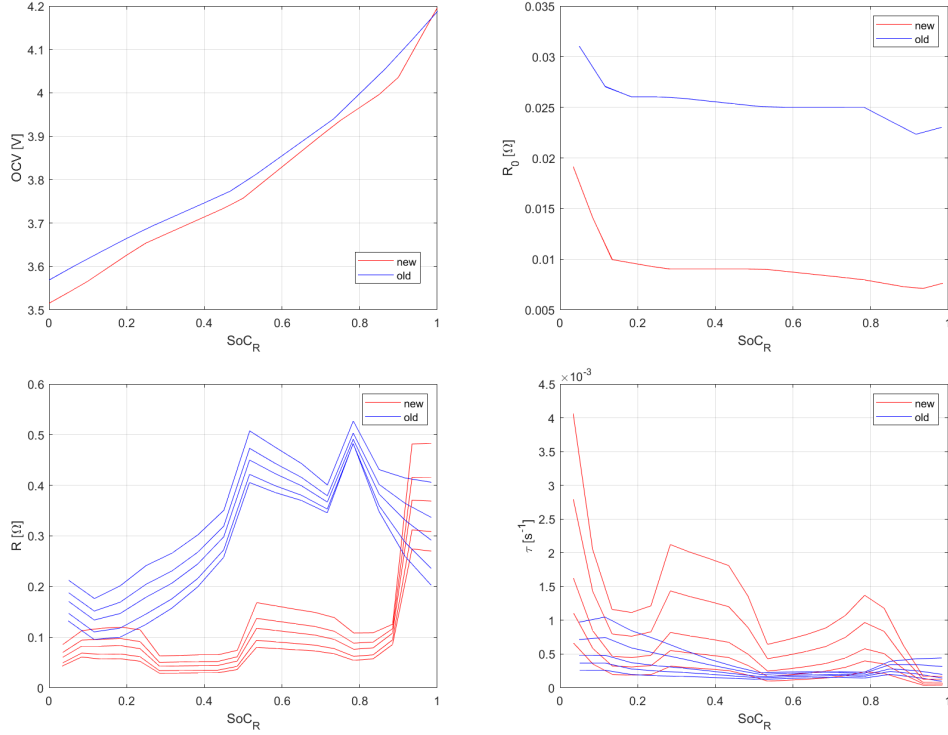


Figure 8.1: Comparison between new and old parameters identified using SoC_R

for the new and the old batteries. The coefficients are reported in table 8.1 and the corresponding graphs in figure 8.2.

Battery	p_1	p_2
New	0.6972	3.5168
Old	0.6165	3.5705

Table 8.1: Coefficients of the straight lines approximating V_{OCV}

8.1.2 R_0 approximation

R_0 curves do not present any dependence on the current applied to the cell. The resulting curve can be linearly approximated by a straight line dependent on SoC.

$$R_0(SoC) = p_1^{R_0} SoC + p_2^{R_0} \quad (8.2)$$

The coefficients $p_1^{R_0}$ and $p_2^{R_0}$ are determined by resolving the LS approximation related to all the points present in the corresponding LUTs. Two independent lines

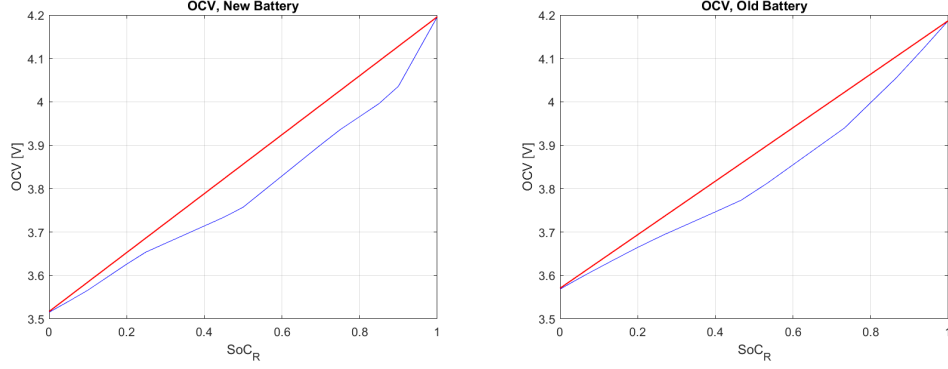


Figure 8.2: Approximation of V_{OCV} curves for new and old batteries

are determined for the new and the old batteries. The coefficients are reported in table 8.2 and the corresponding graphs in figure 8.3.

Battery	p_1	p_2
New	-0.0065	0.0127
Old	-0.0056	0.0283

Table 8.2: Coefficients of the straight lines approximating R_0

8.1.3 Current independent τ approximation and R identification

Observing the LUTs relatives to τ and R , it is clear that the dependence on current strongly influences both parameters. The aim is to reduce the dependence to only one parameter, fixing the other to only SoC dependence. The simplest to fix appears to be τ . The set of curves can be approximated as a straight line.

$$\tau(SoC) = p_1^\tau SoC + p_2^\tau \quad (8.3)$$

To simplify future SoH estimation, both the lines for new and old cells are fixed to pass through the point ($SoC = 1, \tau = 1 \cdot 10^{-4} s^{-1}$). The coefficients p_1^τ and p_2^τ are determined by resolving the LS approximation related to all the points present in the corresponding LUTs. Two independent lines are determined for the new and the old batteries. The coefficients are reported in table 8.3 and the corresponding graphs in figure 8.4. The last step is performing another identification batch using the new parameter approximations to find the R . In this way, all the uncertainties from the approximations are reflected on the new R LUT. Starting from the

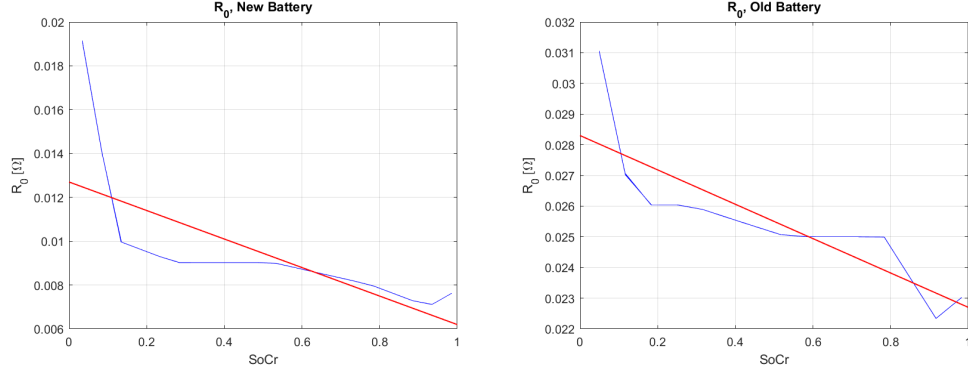


Figure 8.3: Approximation of R_0 curves for new and old batteries

Battery	p_1	p_2
New	-0.0011	0.0012
Old	$-4.02 \cdot 10^{-4}$	$5.02 \cdot 10^{-4}$

Table 8.3: Coefficients of the straight lines approximating current independent τ

state-space representation

$$\begin{cases} V_T(k) = V_{OCV} + R_0 i(k) + V(k) \\ V(k) = V(k-1) - \tau T_s V(k-1) + \tau R T_s i(k) \end{cases} \quad (8.4)$$

knowing all the parameters except for R ,

$$Y_k = V_T(k) - V_{OCV}(k) - V_T(k-1) + V_{OCV}(k-1) - R_0 i(k) + R_0 i(k-1) + \tau T_s [V_T(k-1) - V_{OCV}(k-1) - R_0 i(k-1)] \quad (8.5)$$

$$X_k = \tau T_s i(k) \quad (8.6)$$

the problem can be written as:

$$\begin{bmatrix} Y_2 \\ \dots \\ Y_k \end{bmatrix} = \begin{bmatrix} X_2 \\ \dots \\ X_k \end{bmatrix} \cdot R \quad (8.7)$$

The resulting R is reported in figure 8.5. The LUTs result is dependent both on current and SoC.

8.1.4 Current dependent τ approximation and R identification

τ is now approximated with a set of straight lines passing through the point ($SoC = 1, \tau = 1 \cdot 10^{-4} s^{-1}$). The current dependency is expressed in the angular

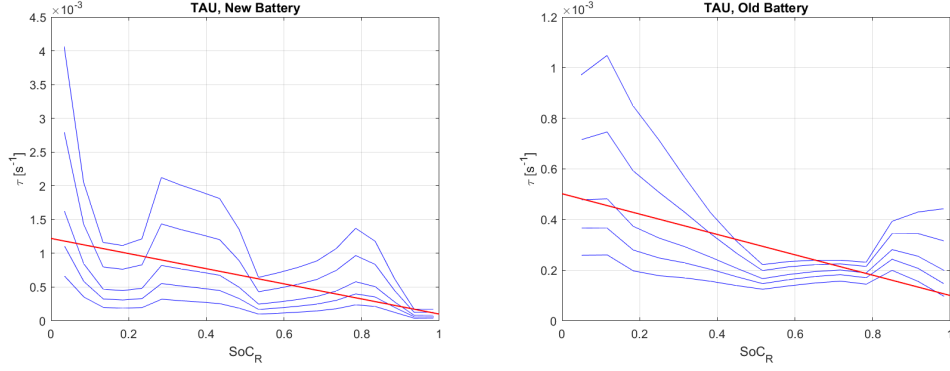


Figure 8.4: Approximation of current independent τ curves for new and old batteries

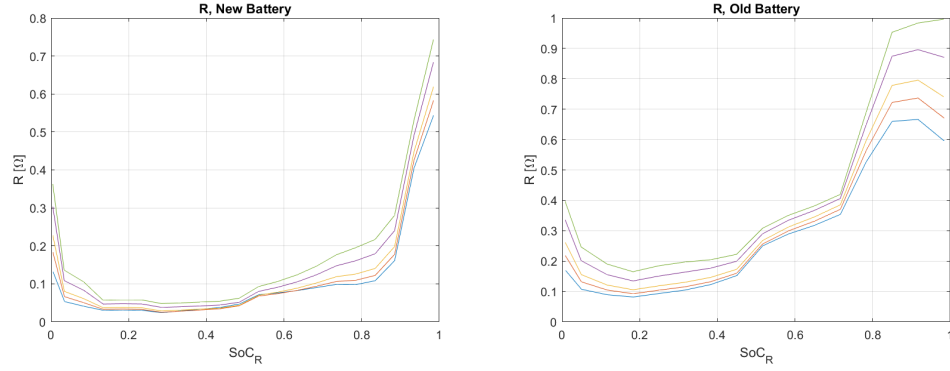


Figure 8.5: Approximation of R with current independent τ for new and old batteries

coefficient of the lines.

$$\tau(\text{SoC}, i) = p_1^\tau(i) \text{SoC} + p_2^\tau(i) \quad (8.8)$$

The coefficients p_1^τ and p_2^τ are determined by resolving the LS approximation related to the points referred to a specific current in the corresponding LUTs. Two independent sets of lines are determined for the new and the old batteries. The coefficients are reported in table 8.4 and 8.5 and the corresponding graphs in figure 8.6. The last step is performing another identification batch using the new parameter approximations to find the R . In this way, all the uncertainties from the approximations are reflected on the new R LUT. The problem is the same already defined in equation 8.7. The resulting R is reported in figure 8.7. The LUTs result to be dependent both on current and SoC.

Battery	p_1				
	0.5C	0.75C	1C	1.5C	2C
New	$-2.52 \cdot 10^{-4}$	$-5.31 \cdot 10^{-4}$	$-8.51 \cdot 10^{-4}$	-0.0016	-0.0024
Old	$-1.13 \cdot 10^{-4}$	$-2.28 \cdot 10^{-4}$	$-3.29 \cdot 10^{-4}$	-5.44	-7.75

Table 8.4: Coefficients p_1 of the straight lines approximating current dependent τ

Battery	p_2				
	0.5C	0.75C	1C	1.5C	2C
New	$3.52 \cdot 10^{-4}$	$6.32 \cdot 10^{-4}$	$9.51 \cdot 10^{-4}$	0.0017	0.0025
Old	$2.31 \cdot 10^{-4}$	$3.28 \cdot 10^{-4}$	$4.29 \cdot 10^{-4}$	$6.45 \cdot 10^{-4}$	$8.75 \cdot 10^{-4}$

Table 8.5: Coefficients p_2 of the straight lines approximating current dependent τ

8.2 Analysis of SoH dependency considering SoC_R

This section aims to reduce, where possible, the dependence on SoH. If not possible, the simplest model is considered to obtain sufficient performance.

8.2.1 V_{OCV} approximation

The curves relative to new and old batteries are reported in figure 8.8a. It is possible to notice an almost equivalence between the two straight lines. To simplify the dependency of SoH of the model, a single common characteristic is taken. The coefficient $p_1^{V_{OCV}}$ and $p_2^{V_{OCV}}$ are computed using the mean of the corresponding parameters of the new and the old batteries. The equation is independent of SoH:

$$V_{OCV}(SoC) = p_1^{V_{OCV}} SoC + p_2^{V_{OCV}} \quad (8.9)$$

The resulting characteristic is displayed in figure 8.8b.

8.2.2 R_0 approximation

The curves relative to new and old batteries are reported in figure 8.9a. The angular coefficient of the two lines is almost the same. The graph can be approximated through a not-proper set of straight lines.

$$R_0(SoC, SoH) = p_1^{R_0} SoC + p_2^{R_0}(SoH) \quad (8.10)$$

with $p_2^{R_0}(SoH) = p_1^{p_2, R_0} SoH + p_2^{p_2, R_0}$

The angular coefficient $p_1^{R_0}$ is determined by calculating the mean of the angular coefficient of the original lines. The linear regression to determine $p_2^{R_0}(SoH)$ is

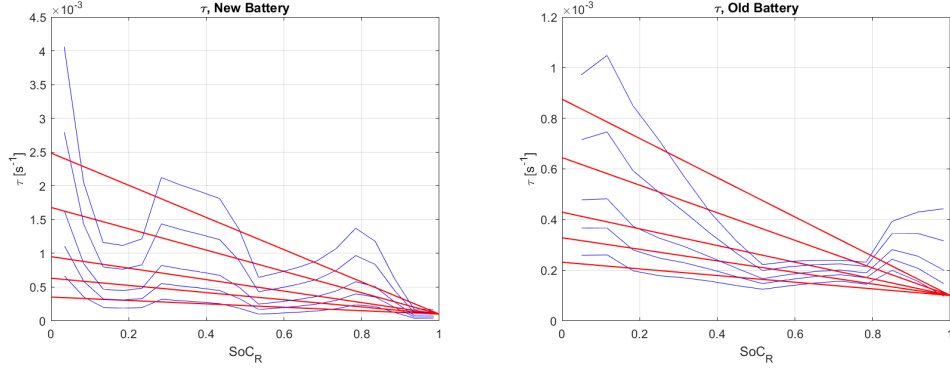


Figure 8.6: Approximation of current dependent τ curves for new and old batteries

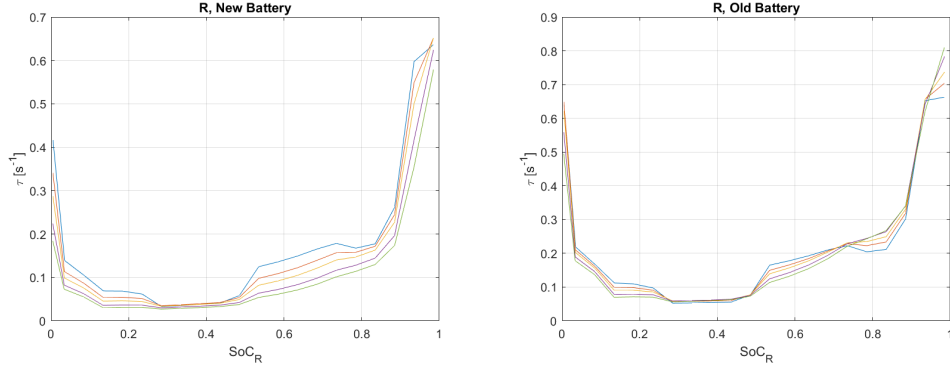


Figure 8.7: Approximation of R with current dependent τ for new and old batteries

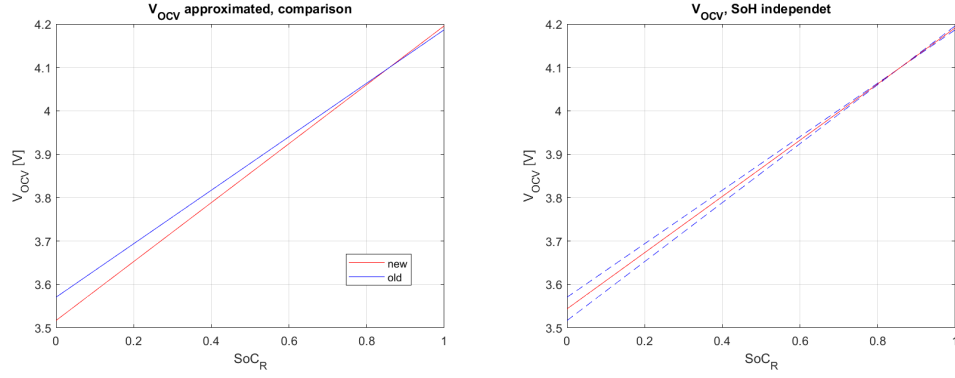
defined by the following problem:

$$\begin{bmatrix} p_2^{R_0}(SoH_1) \\ \vdots \\ p_2^{R_0}(SoH_k) \end{bmatrix} = \begin{bmatrix} SoH_1 & 1 \\ \vdots & \vdots \\ SoH_k & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1^{p_2, R_0} \\ p_2^{p_2, R_0} \end{bmatrix} \quad (8.11)$$

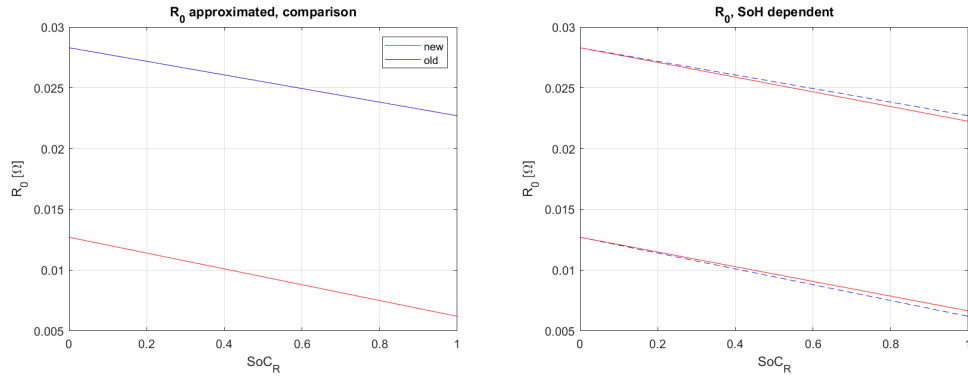
where $1, \dots, k$ are the batteries with different SoH considered in the problem. The result of the approximation is displayed in figure 8.9b.

8.2.3 τ and R approximation

The approximation of τ and R is bounded by the approach followed in the previous section. In figure 8.10 are reported the results obtained for R parameter using both current dependent and independent τ . Considering the data relative to current independent τ , it is not evident relation between the new and the old battery. A



(a) Current independent approximations (b) SoH independent approximation

 Figure 8.8: Approximation of V_{OCV}


(a) Current independent approximations (b) SoH independent approximation

 Figure 8.9: Approximation of R_0

vertical-translating quadratic curve could approximate the data, but it is high in complexity and both introduce big errors. Moreover, the dependency on the current must be analyzed and consequently, the complexity grows up. Moving to the current dependent τ solution, it is possible to notice a lower dependency on the current. The two curves are very similar and they can both be approximated with the same second-order function of SoC.

$$R(\text{SoC}) = p_1^R \text{SoC}^2 + p_2^R \text{SoC} + p_3^R \quad (8.12)$$

R results to be the second parameter with no dependencies on current nor SoH. The result of the approximation is displayed in figure 8.11.

In figure 8.12, the curves relative to current dependent τ are displayed. The lines were designed to cross the same point, in this way the complexity is completely

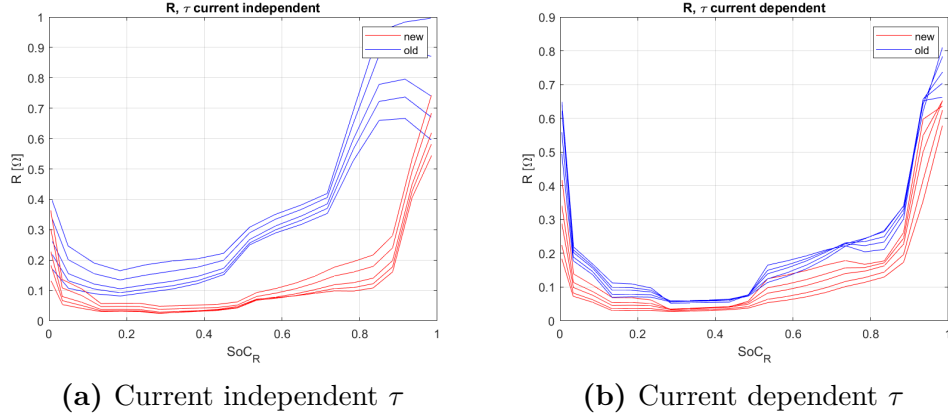


Figure 8.10: Approximation of R

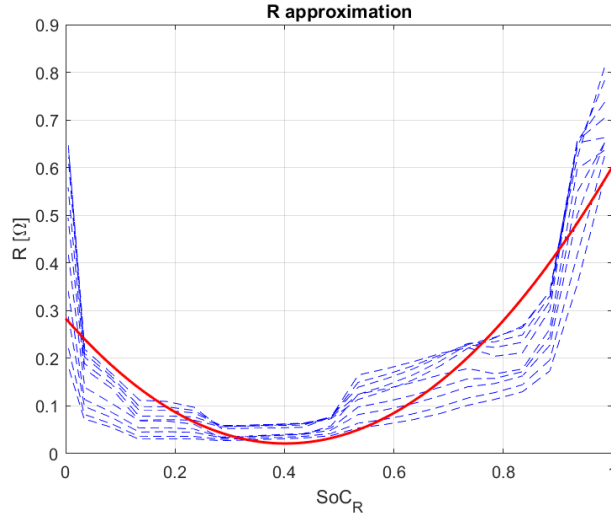


Figure 8.11: R quadratic approximation

assigned to the angular coefficient only.

$$\tau(\text{SoC}, i, \text{SoH}) - \tau_0 = m^\tau(i, \text{SoH})[\text{SoC} - \text{SoC}_0] \quad (8.13)$$

with $\tau_0 = 1 \cdot 10^{-4}$ and $\text{SoC}_0 = 1$. The point $P = (\text{SoC}_0, \tau_0)$ is defined. The m^τ coefficient dependency on current and SoH is represented in figure 8.13. m^τ results to be linearly dependent on current. For each SoH value, the dependency on the current of the coefficients can be written as:

$$m^\tau(i)|_{\text{SoH}_j} = p_1^{m,\tau} i + p_2^{m,\tau} \quad (8.14)$$

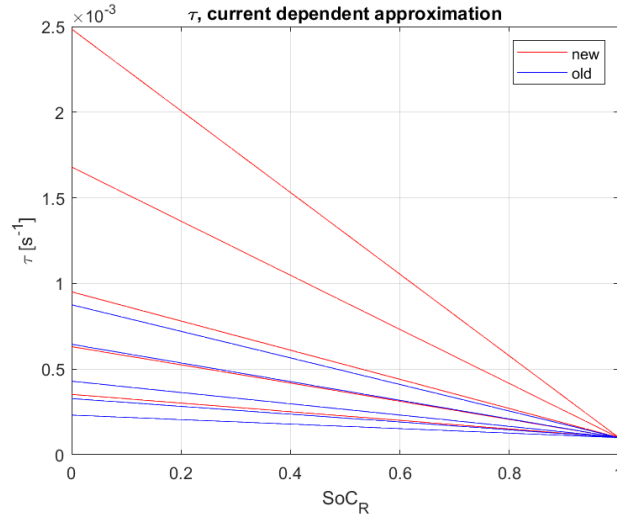


Figure 8.12: Current dependent τ approximation

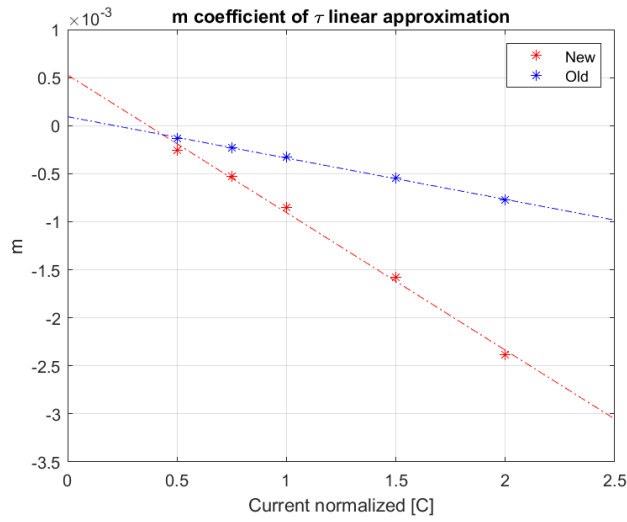


Figure 8.13: m^τ current dependency

The coefficients $p_1^{m,\tau}$ and $p_2^{m,\tau}$ are calculated using LS estimator on the problem:

$$\begin{bmatrix} m^\tau(i_1)|_{SoH_j} \\ \dots \\ m^\tau(i_k)|_{SoH_j} \end{bmatrix} = \begin{bmatrix} i_1 & 1 \\ \dots & \dots \\ i_k & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1^{m,\tau}(SoH_j) \\ p_2^{m,\tau}(SoH_j) \end{bmatrix} \quad (8.15)$$

The results are displayed in figure 8.13. To continue the analysis of SoH dependency, the graph can be approximated as a proper set of straight lines. The angular

coefficient of each line is the same reported in equation 8.15.

$$m^\tau - m_0^\tau = p_1^{m,\tau}(SoH_j)[i - i_0] \quad (8.16)$$

The fixed point (i_0, τ_0) is calculated resolving the following problem:

$$\begin{bmatrix} m^\tau(i_1)|_{SoH_j} - p_1^{m,\tau}(SoH_j)i \\ \dots \\ m^\tau(i_k)|_{SoH_{jj}} - p_1^{m,\tau}(SoH_{jj})i \end{bmatrix} = \begin{bmatrix} -p_1^{m,\tau}(SoH_j) & 1 \\ \dots & \dots \\ -p_1^{m,\tau}(SoH_{jj}) & 1 \end{bmatrix} \cdot \begin{bmatrix} i_0 \\ m_0^\tau \end{bmatrix} \quad (8.17)$$

The point $Q = (i_0, m_0^\tau)$ is defined. Finally, a relation between the coefficients $p_1^{m,\tau}(SoH_j)$ has to be investigated. The simplest way is to perform a LS estimation of the problem:

$$\begin{bmatrix} p_1^{m,\tau}(SoH_1) \\ \dots \\ p_1^{m,\tau}(SoH_j) \end{bmatrix} = \begin{bmatrix} SoH_1 & 1 \\ \dots & \dots \\ SoH_j & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1^{pm,\tau} \\ p_2^{pm,\tau} \end{bmatrix} \quad (8.18)$$

The factors can be computed through:

$$p_1^{m,\tau}(SoH) = p_1^{pm,\tau} SoH + p_2^{pm,\tau} \quad (8.19)$$

The resultant approximation of τ , is reported in figure 8.14. The process of

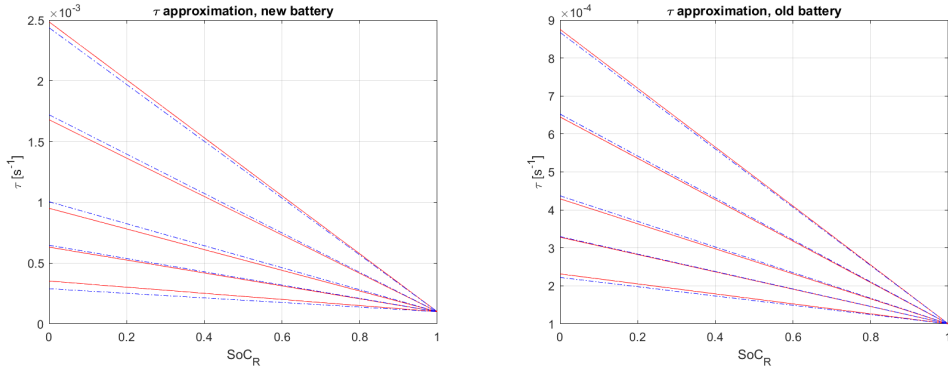


Figure 8.14: Approximation of τ

τ determination is defined starting from SoC, SoH, and current values. The parameters needed to perform the estimation are P , Q , $p_1^{pm,\tau}$ and $p_2^{pm,\tau}$.

$$\begin{cases} p^{m,\tau} = p_1^{pm,\tau} SoH + p_2^{pm,\tau} \\ m^\tau = p^{m,\tau}(i - Q_1) + Q_2 \\ \tau = m^\tau(SoC - P_1) + P_2 \end{cases} \quad (8.20)$$

8.2.4 C_R definition

To complete the model, it is necessary to define the capacity of the battery. Using SoC_R , it is consequent the use of C_R , as explained in chapter 4. The relation between C_R and SoH is derived from the definition itself:

$$C_R = C_N SoH \quad (8.21)$$

8.3 Analysis of SoC and Current dependencies considering SoC_N

The same analysis displayed in the previous section is performed using LUTs generating during *Identification phase* using SoC_N . The starting LUTs are reported in figure 8.15. It could be seen a substantial equivalence with the results obtained with SoC_R , but the parameter characteristics relative to the old battery are translated into a sub-space of the original plane. The approximation process of the

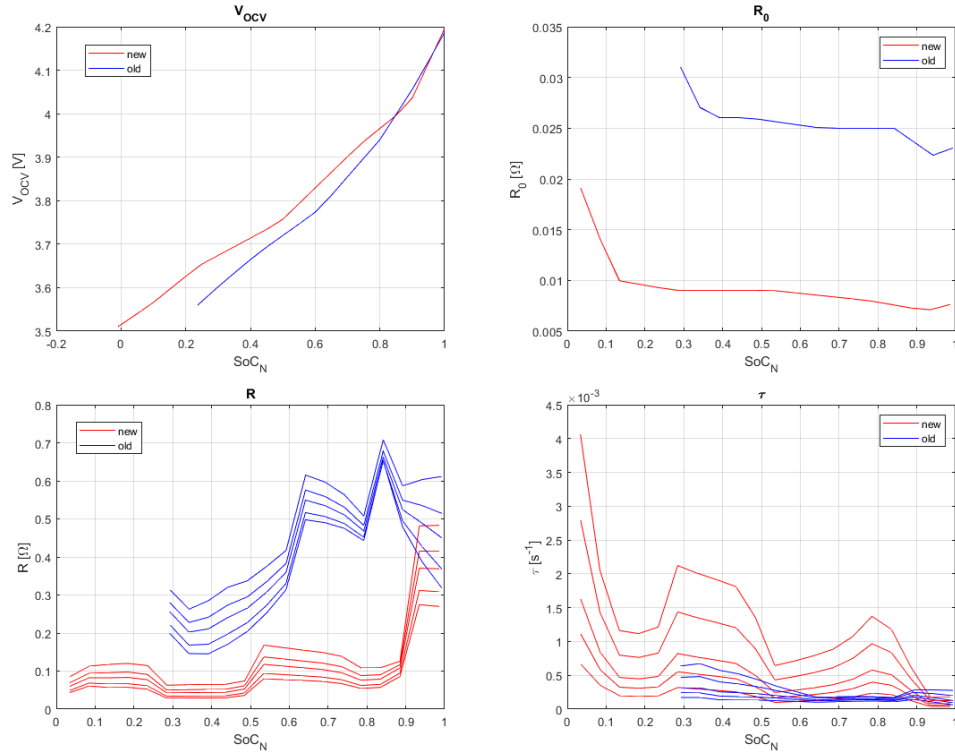


Figure 8.15: Comparison between new and old parameters identified using SoC_N

previous chapter is followed. In figure 8.16 the approximation of V_{OCV} and R_0 are

depicted. As for SoC_R , the approximation of τ and R could follow two approach.

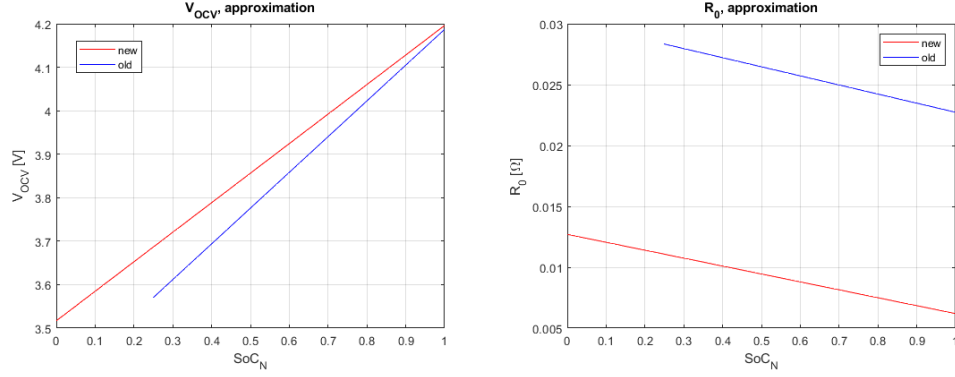


Figure 8.16: Approximation of V_{OCV} and R_0 using SoC_N

The results regarding current independent τ are reported in figure 8.17, while the ones relative to current dependent approximation are depicted in figure 8.18. The

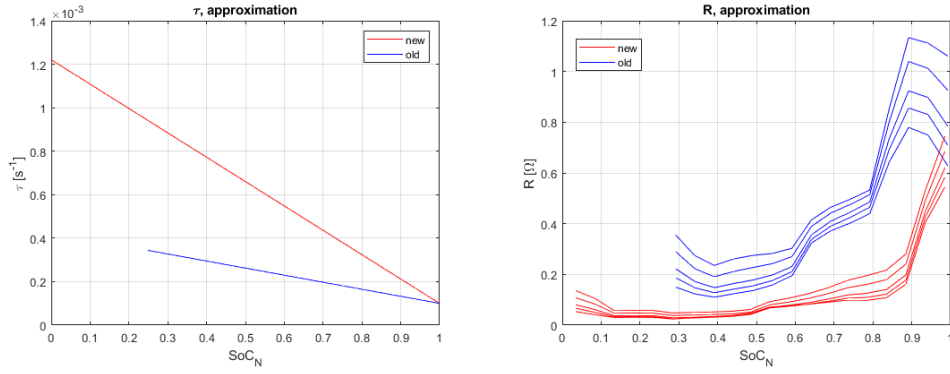


Figure 8.17: Approximation of current independent τ and R using SoC_N

use of SoC_N is a simplification for the use of C_N without any correction terms, but it has some other issues regarding the complexity of R LUTs. For both τ dependent and independent from the current, the R assumes complex transformation with the variation of SoH. For this reason, the SoC_R approach will be preferred in the next sections.

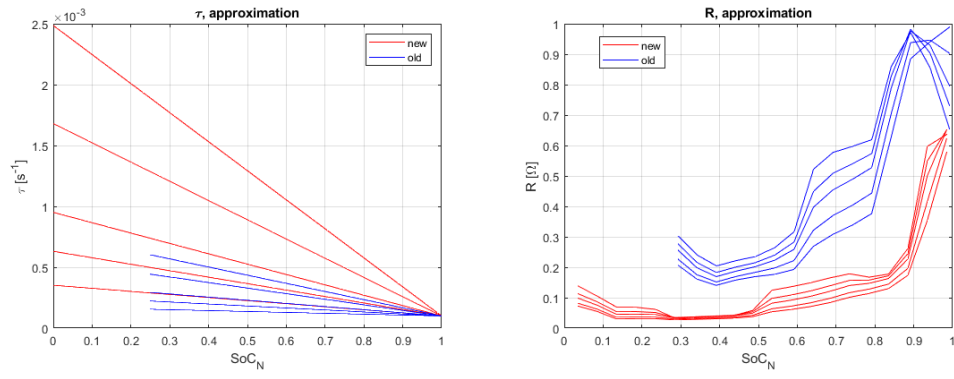


Figure 8.18: Approximation of current dependent τ and R using SoC_N

Chapter 9

ERMES

9.1 Introduction of a new battery

Until this point, all the computations are done based on the two models defined in chapter 5 and called *new* and *old batteries*. To proceed it is useful to add at least one more model, with a different SoH value. From the literature, the model of *medium battery* is provided. The resulting SoH of the battery results to be

$$SoH = 0.875 \quad (9.1)$$

The complete parameter LUTs are reported in figure 9.1.

9.2 Multi-model definition

The multi-model must be updated including the data relatives to *medium battery*. The computation is following the *dependent current* τ approach analyzed in the previous chapter. Remembering that $x_1 = V$, $x_2 = SoC$ and $y = V_T$, the resulting model is:

$$\begin{cases} x_1(k+1) = x_1(k) + [-\tau x_1(k) + \tau Ru(k)]T_s \\ x_2(k+1) = x_2(k) + \frac{T_s}{3600 \cdot Capacity} u(k) \\ x_3(k+1) = [1 - \tau T_s]x_1(k) + [\tau RT_s + R_0]u(k) + V_{OCV} \\ y(k) = x_3(k) \end{cases} \quad (9.2)$$

where:

$$Capacity = C_R = C_N \cdot SoH \quad (9.3)$$

$$V_{OCV}(x_2) = p_1^{V_{OCV}} x_2 + p_2^{V_{OCV}} \quad (9.4)$$

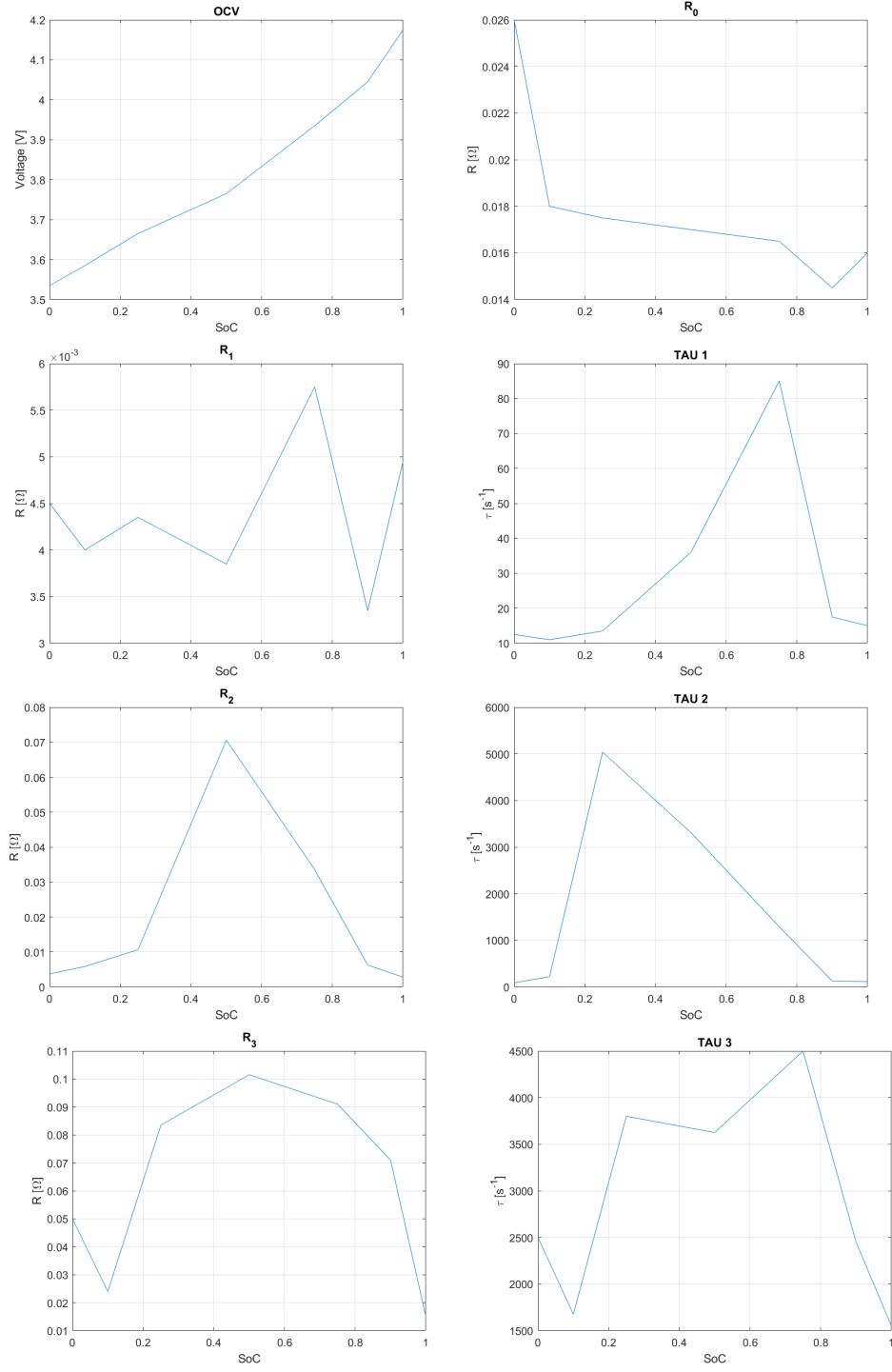


Figure 9.1: Parameter LUTs of *medium battery*

$$R_0(x_2, SoH) = p_1^{R_0} x_2 + p_2^{R_0} (SoH) \quad (9.5)$$

$$\text{with } p_2^{R_0} (SoH) = p_1^{p_2, R_0} SoH + p_2^{p_2, R_0}$$

$$R(x_2) = p_1^R x_2^2 + p_2^R x_2 + p_3^R \quad (9.6)$$

$$\tau(x_2, SoH, u) = m^\tau(x_2 - P_1) + P_2$$

$$\text{with } m^\tau = p^{m, \tau}(i_N - Q_1) + Q_2$$

$$p^{m, \tau} = p_1^{p^{m, \tau}} SoH + p_2^{p^{m, \tau}} \quad (9.7)$$

$$i_N = \frac{u}{C_N}$$

The values of the parameters are reported in table 9.1.

Parameter	Sub-parameter	Value
Capacity	C_N	5.4Ah
V_{OCV}	$p_1^{V_{OCV}}$	0.6479
	$p_2^{V_{OCV}}$	3.5436
R_0	$p_1^{R_0}$	-0.0057
	$p_1^{p_2, R_0}$	-0.0624
	$p_2^{p_2, R_0}$	0.0749
R	p_1^R	1.6255
	p_2^R	-1.3058
	p_3^R	0.2835
τ	$p_1^{p^{m, \tau}}$	-0.0040
	$p_2^{p^{m, \tau}}$	0.0026
	P	$(1, 1 \cdot 10^{-4})$
	Q	$(0.0785, 1.5883 \cdot 10^{-4})$

Table 9.1: Parameters of the multi-model

9.3 ERMES definition

ERMES is an algorithm patented by *Brain technologies*. It exploits a multi-model approach to estimate complex parameters such as SoH. The main idea is a set of observers with a pre-set value of the parameter to be estimated. The data are processed simultaneously by all the observers and the output is compared with

some algorithm. In this application, a simple integration of the squared absolute error is used. In the end, the decision variable is compared and the minimum error model is assumed to be the estimate of the parameter. [31, 36, 37] The logical scheme is provided in figure 9.2. As observers, the EKF illustrated in chapter

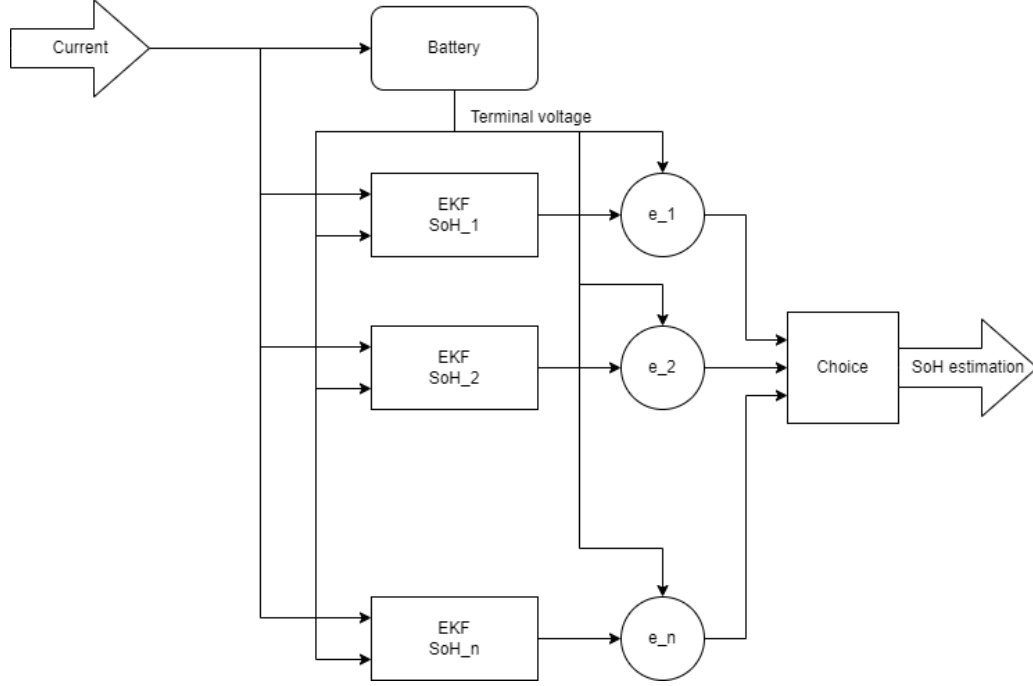


Figure 9.2: ERMES scheme

7 is exploited. With the approximation of the parameters, \mathbf{A} and \mathbf{C} matrices have a different formulation. Considering the model previously defined, the partial derivatives are:

$$\begin{aligned} \frac{\partial f_1}{\partial x_1} &= 1 - \tau T_s \\ \frac{\partial f_1}{\partial x_2} &= -m^\tau T_s x_1 + (2p_1^R x_2 + p_2^R) \tau T_s u + m^\tau R T_s u \\ \frac{\partial f_1}{\partial x_3} &= 0 \end{aligned} \tag{9.8}$$

$$\frac{\partial f_2}{\partial x_1} = 0, \quad \frac{\partial f_2}{\partial x_2} = 1, \quad \frac{\partial f_2}{\partial x_3} = 0 \tag{9.9}$$

$$\begin{aligned}\frac{\partial f_3}{\partial x_1} &= 1 - \tau T_s \\ \frac{\partial f_3}{\partial x_2} &= -m^\tau T_s x_1 + (2p_1^R x_2 + p_2^R) \tau T_s u + m^\tau R T_s u + p_1^{V_{OCV}} + p_1^{R_0} u \\ \frac{\partial f_3}{\partial x_3} &= 0\end{aligned}\tag{9.10}$$

$$\frac{\partial h}{\partial x_1} = 0, \quad \frac{\partial h}{\partial x_2} = 0, \quad \frac{\partial h}{\partial x_3} = 1\tag{9.11}$$

and the consequent Jacobians:

$$A = \begin{bmatrix} 1 - \tau T_s & -m^\tau T_s x_1 + (2p_1^R x_2 + p_2^R) \tau T_s u + m^\tau R T_s u & 0 \\ 0 & 1 & 0 \\ 1 - \tau T_s & -m^\tau T_s x_1 + (2p_1^R x_2 + p_2^R) \tau T_s u + m^\tau R T_s u + p_1^{V_{OCV}} + p_1^{R_0} u & 0 \end{bmatrix}\tag{9.12}$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\tag{9.13}$$

9.4 SoH estimation with correct values set of EKF

The first test to evaluate the performance of ERMES and the defined multi-model approach is using three EKFs, each one with a correct value of SoH. The SoH pre-set to the observers are:

$$SoH \in \{0.75; 0.875; 1\}\tag{9.14}$$

The test is performed with the same validation data sets defined in chapter 7. The datasets are called *D1*. For each EKF, the integral squared error is computed. At each time instant, the model with the lower error is chosen as an approximation of SoH. After a first transient, the algorithm stabilizes the SoH estimation on a single value. To overcome the problem of a slower estimation, a reset to zero must be added to the integral error. The problem of the identification of the best time instant to do this operation has not a solution yet. In this work, it is proposed a periodical reset. In this way, it is not necessary to identify the best spot, while a successive reset will be near enough to the optimal point. It is not an optimal solution, but it permits a good approximation without a complex transient study. The results for the 15 tests are reported from figure 9.3 to 9.17.

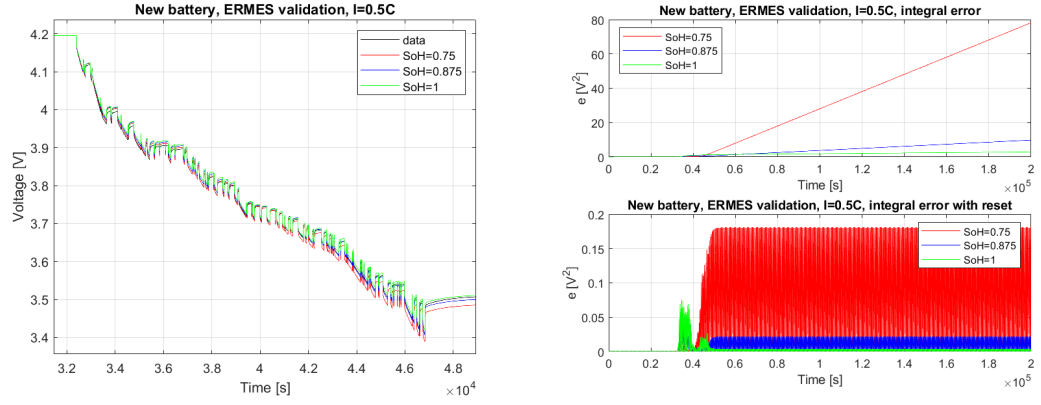


Figure 9.3: New battery, ERMES with correct SoH, $I = 0.5C$

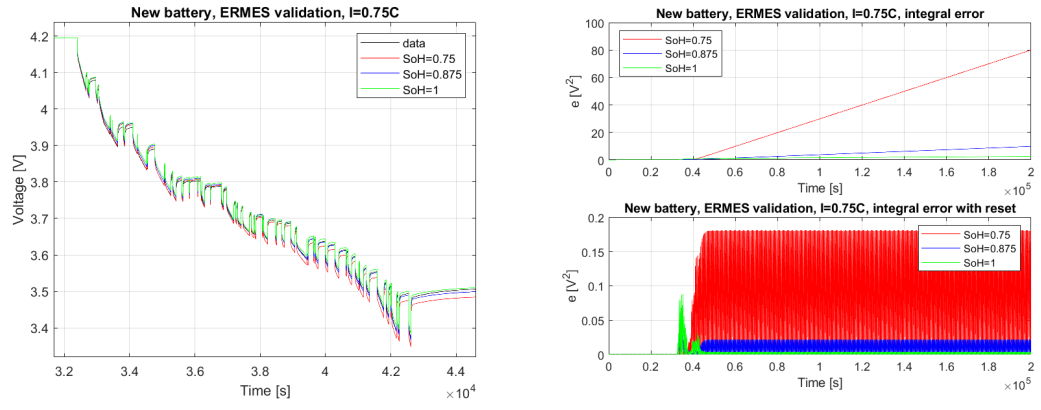


Figure 9.4: New battery, ERMES with correct SoH, $I = 0.75C$

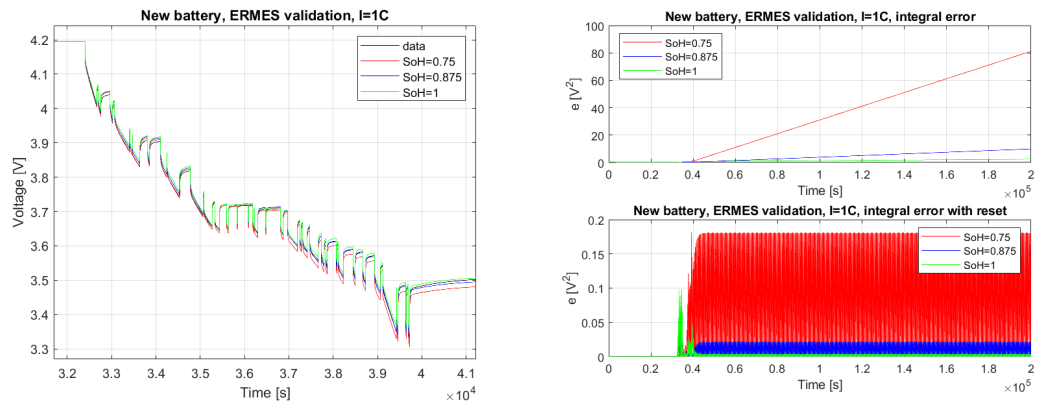


Figure 9.5: New battery, ERMES with correct SoH, $I = 1C$

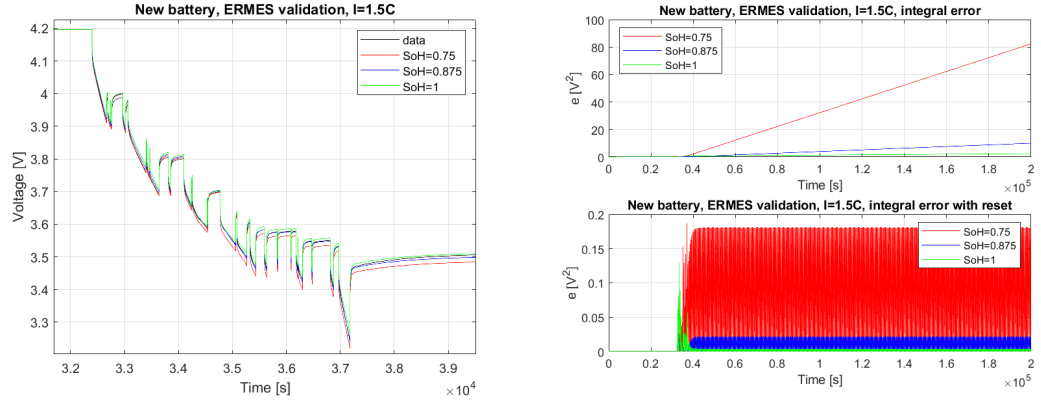


Figure 9.6: New battery, ERMES with correct SoH, $I = 1.5C$

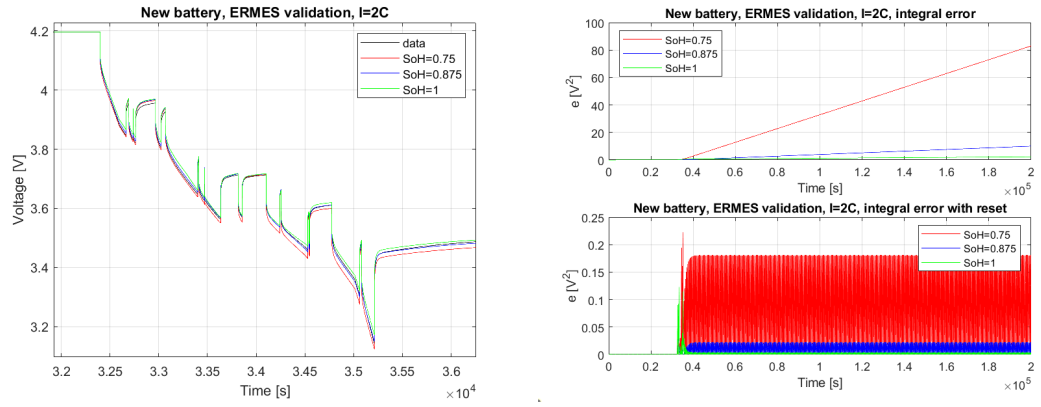


Figure 9.7: New battery, ERMES with correct SoH, $I = 2C$

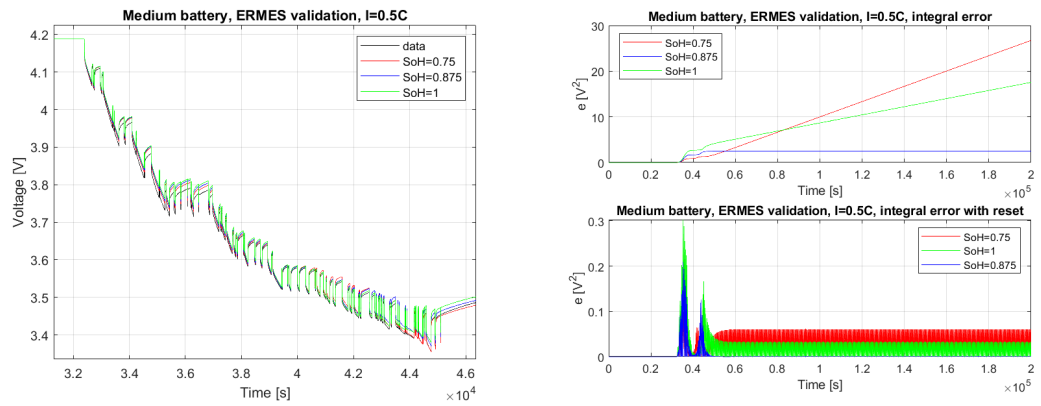


Figure 9.8: Medium battery, ERMES with correct SoH, $I = 0.5C$

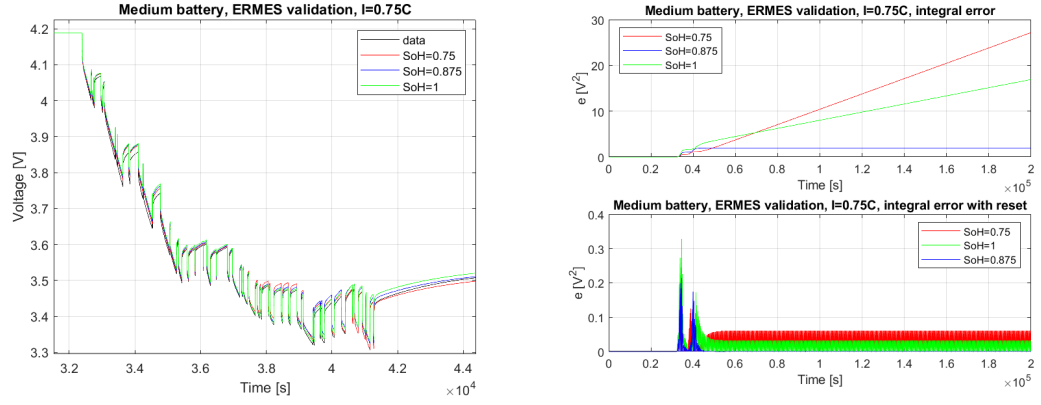


Figure 9.9: Medium battery, ERMES with correct SoH, $I = 0.75C$

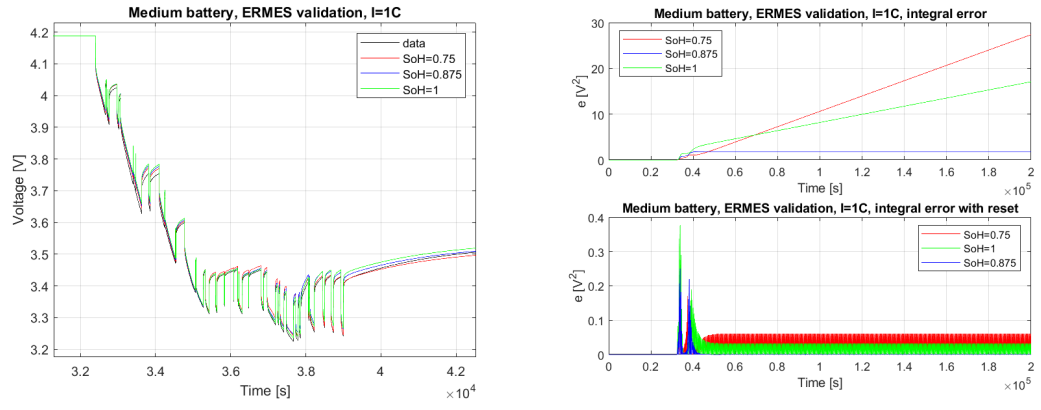


Figure 9.10: Medium battery, ERMES with correct SoH, $I = 1C$

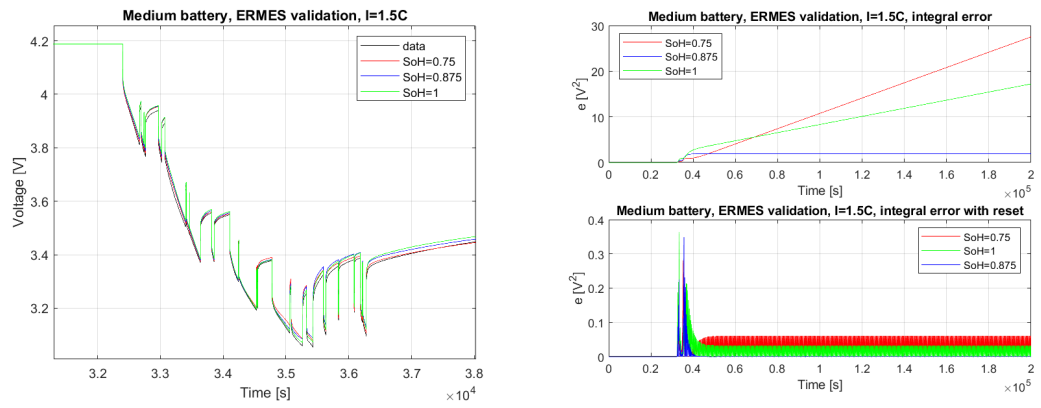


Figure 9.11: Medium battery, ERMES with correct SoH, $I = 1.5C$

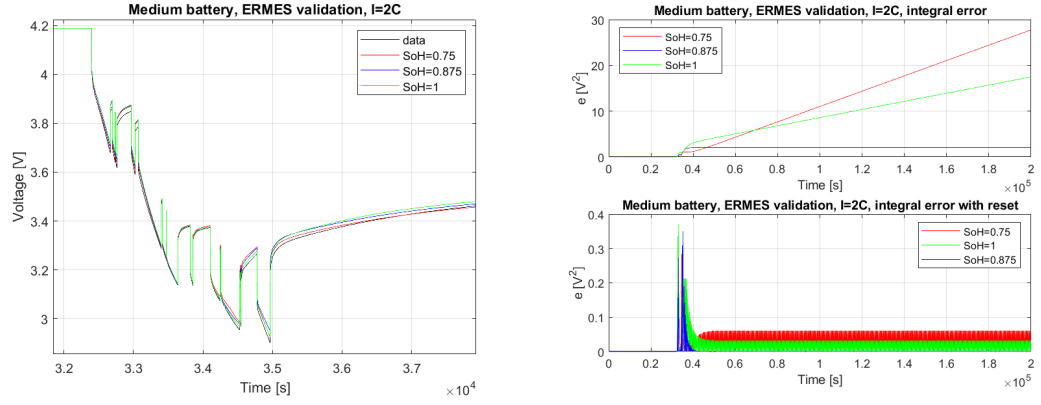


Figure 9.12: Medium battery, ERMES with correct SoH, $I = 2C$

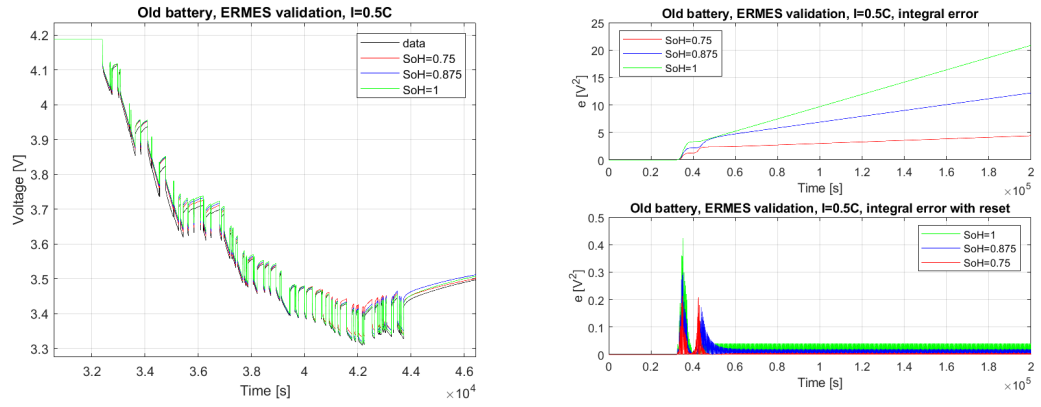


Figure 9.13: Old battery, ERMES with correct SoH, $I = 0.5C$

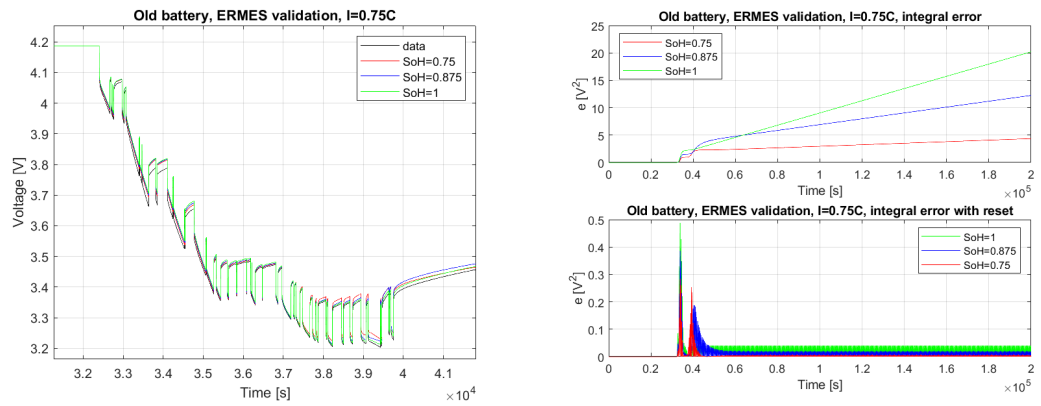


Figure 9.14: Old battery, ERMES with correct SoH, $I = 0.75C$

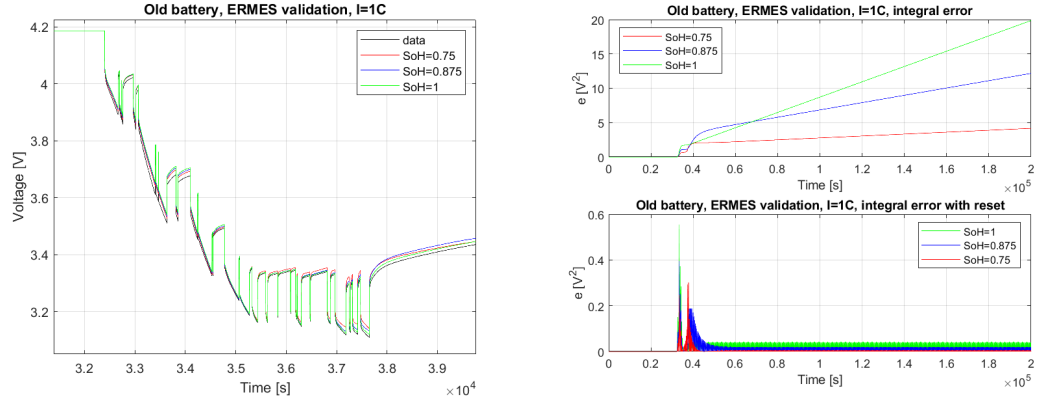


Figure 9.15: Old battery, ERMES with correct SoH, $I = 1C$

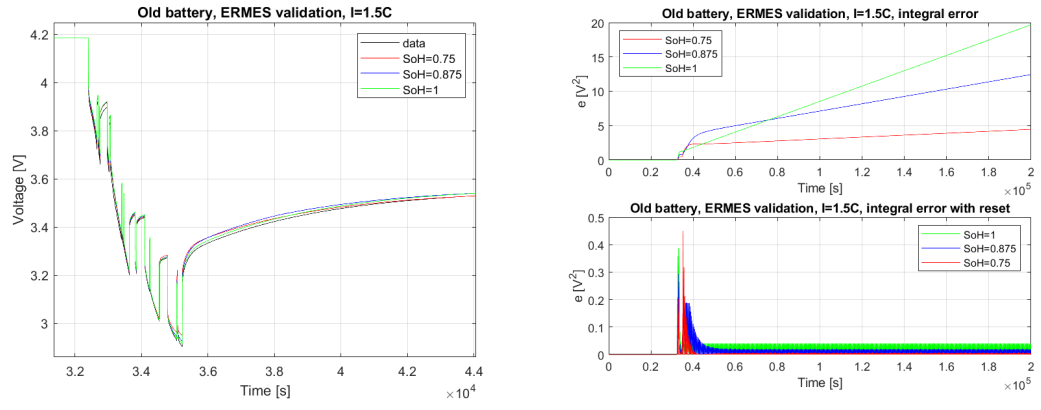


Figure 9.16: Old battery, ERMES with correct SoH, $I = 1.5C$

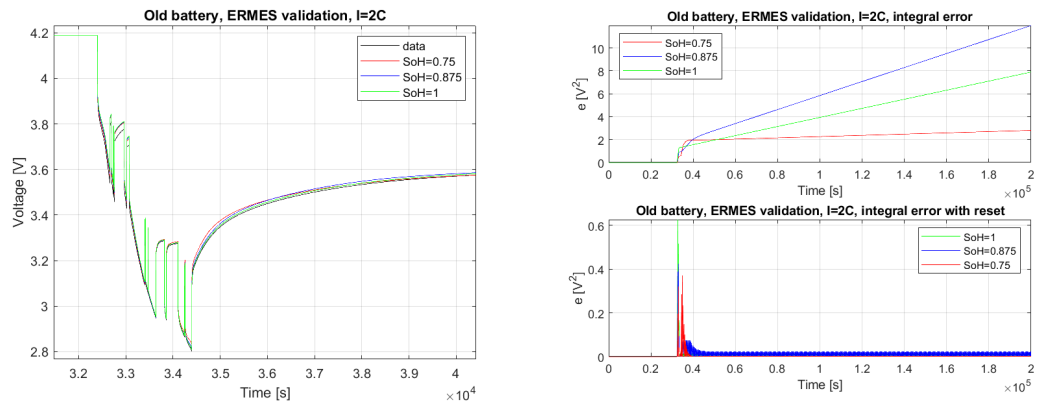


Figure 9.17: Old battery, ERMES with correct SoH, $I = 2C$

It is evident a good separation between the error values. After the first transient, the errors depict a good estimation of the SoH. The error with periodical reset is faster to estimate the correct value. To evaluate the performance of the ERMES algorithm, it is useful to build the boxplot of the estimated SoH values. The boxplot is a simple instrument to visualize the statistical properties of a dataset. The box includes the data inside the twenty 25° and the 75° percentile. The red line is the median. The points outside the boxes are the outliers of the problem. In figure from 9.18 to 9.20 are displayed the boxplot related to the previous validation test. The estimation is statistically significant. At high SoH and low current, the algorithm has some uncertainty due to the similarity in the behavior of the cells. Introducing the periodical reset enhances the performance. In both approaches, the results are completely acceptable.

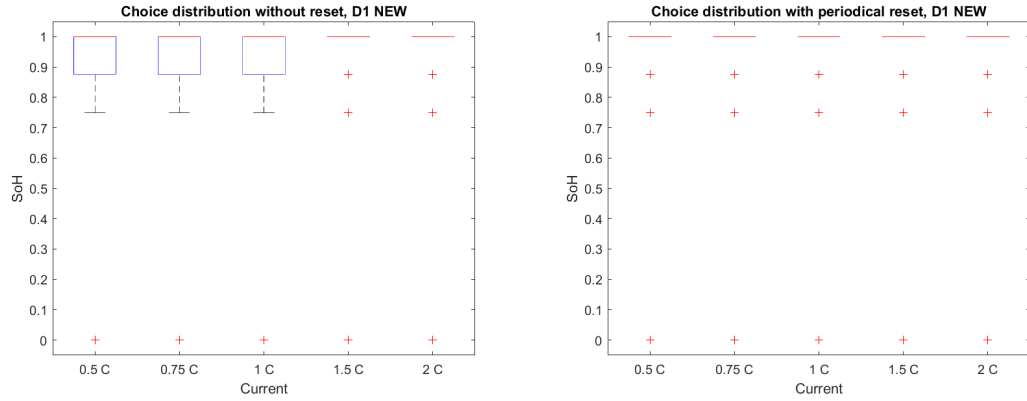


Figure 9.18: New battery D1, SoH estimation boxplot

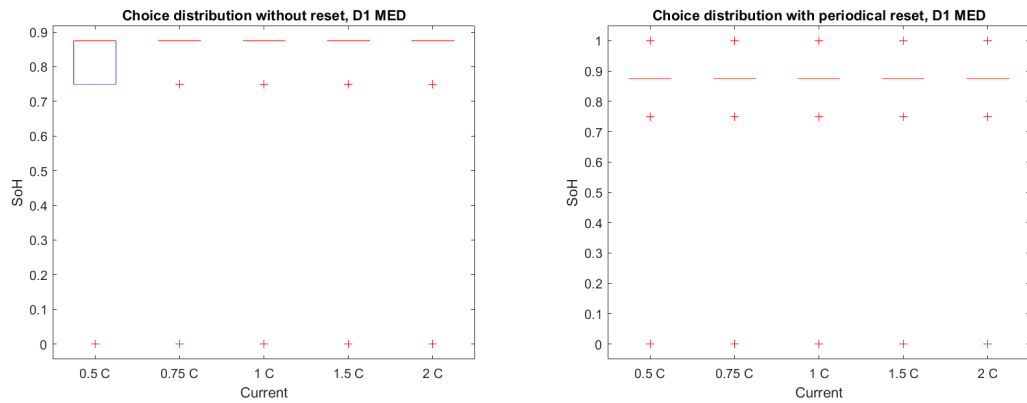


Figure 9.19: Medium battery D1, SoH estimation boxplot

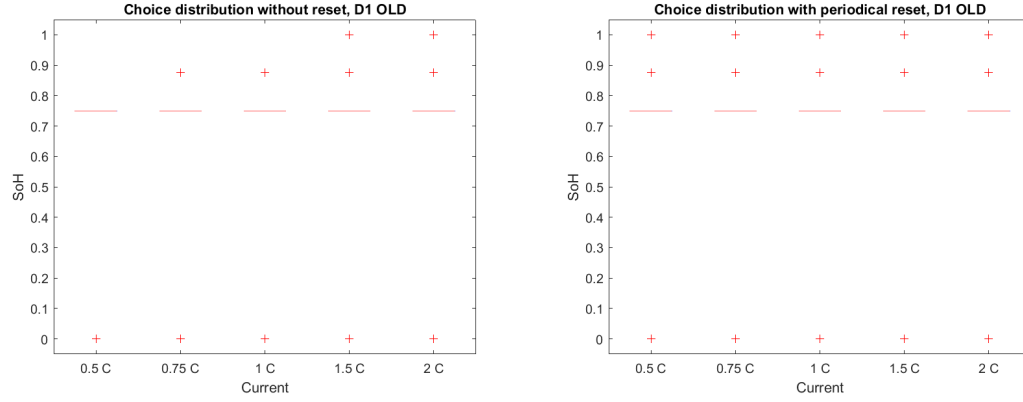


Figure 9.20: Old battery D1, SoH estimation boxplot

The next test is the use of the same EKF set with some validation dataset at constant current discharge. The datasets are called $D2$. Only the boxplots are reported in the following section. The results are displayed in figures from 9.21 to 9.23.

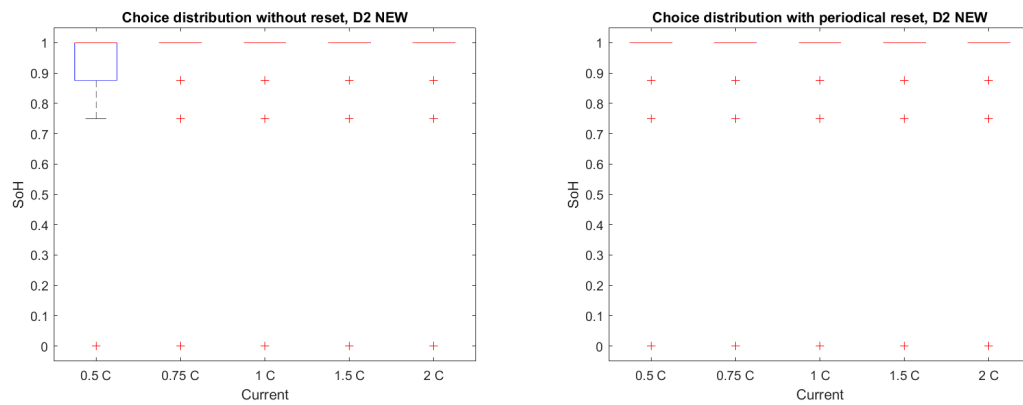


Figure 9.21: New battery D2, SoH estimation boxplot

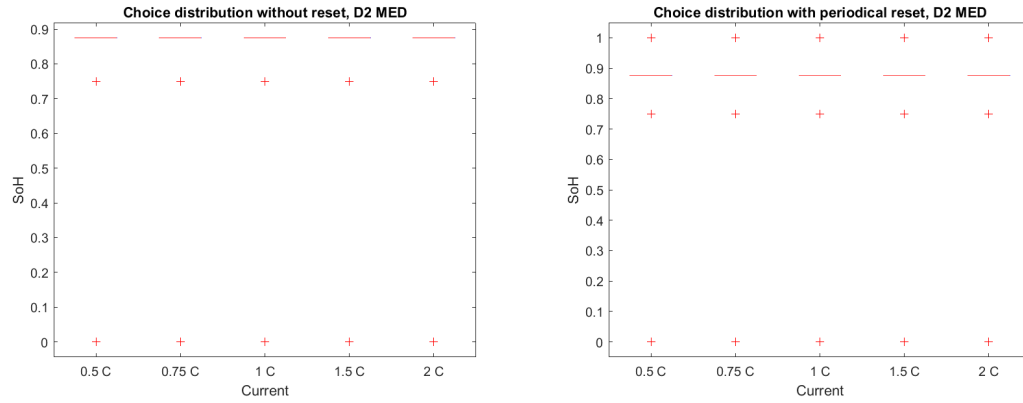


Figure 9.22: Medium battery D2, SoH estimation boxplot

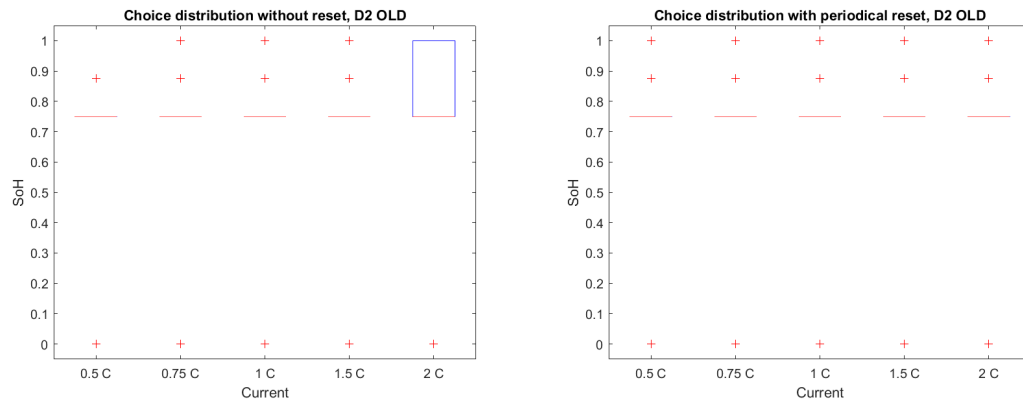


Figure 9.23: Old battery D2, SoH estimation boxplot

The results are consistent between tests performed on *D1* and *D2*. The SoH value is estimated with good statistical performance. The results of the test on *D1* and *D2* are resumed in figure 9.24.

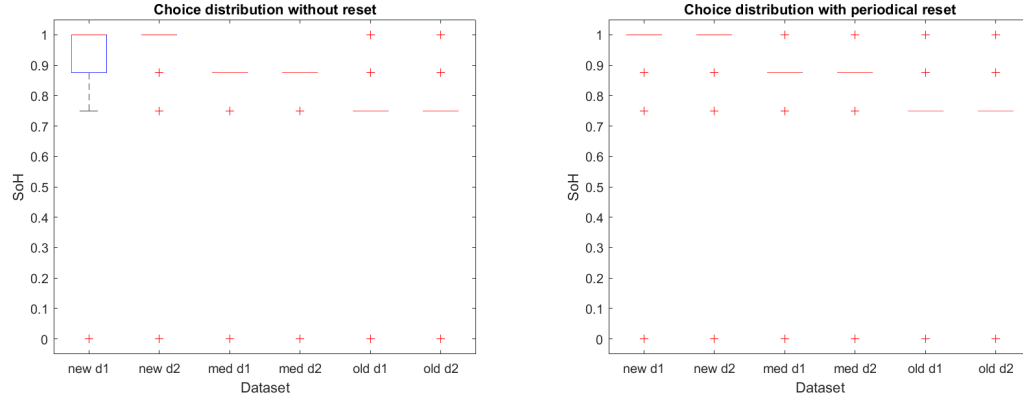


Figure 9.24: Comparison of test $D1$ and $D2$

9.5 SoH estimation with linearly separated values set of EKF

In this section, the EKF bank is pre-set with some linearly separated values of SoH. It is the first idea of the final product of an estimation. The SoH is estimated inside the intervals of the SoH definition. Both tests on $D1$ and $D2$ are performed. The statistical distribution of the error is displayed from figure 9.25 to figure 9.30. There is a clear separation of the errors considering the integral error both with or without reset. From the graphs is possible to consider the integral error as a statistically valid decision variable.

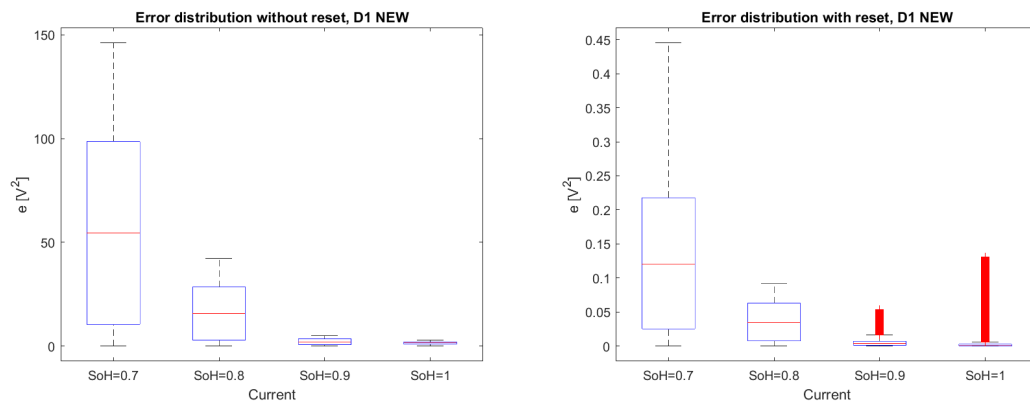


Figure 9.25: New Battery D1, linear SoH values, error boxplot

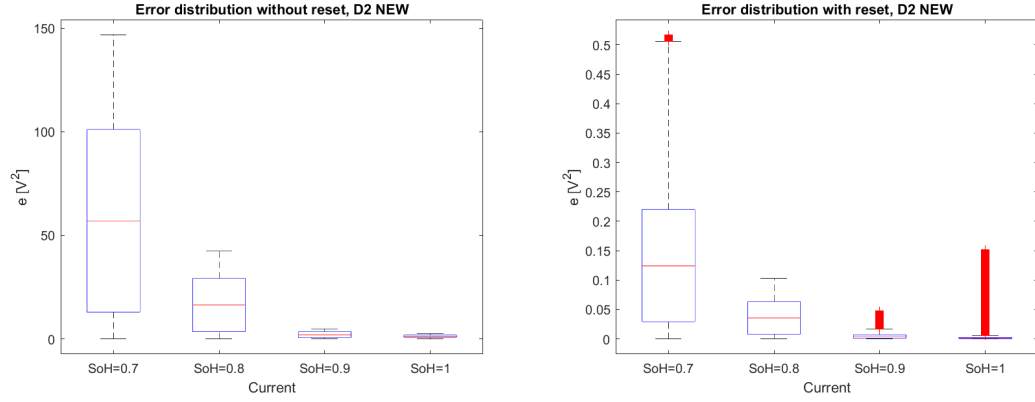


Figure 9.26: New Battery D2, linear SoH values, error boxplot

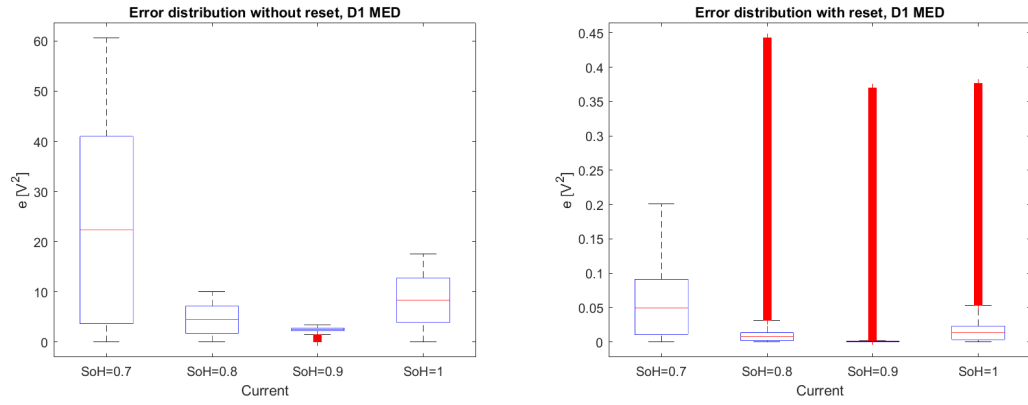


Figure 9.27: Medium Battery D1, linear SoH values, error boxplot

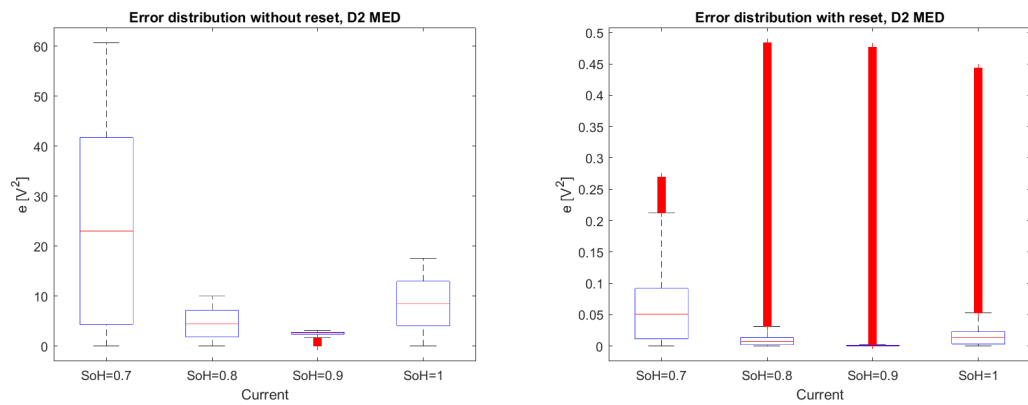


Figure 9.28: Medium Battery D2, linear SoH values, error boxplot

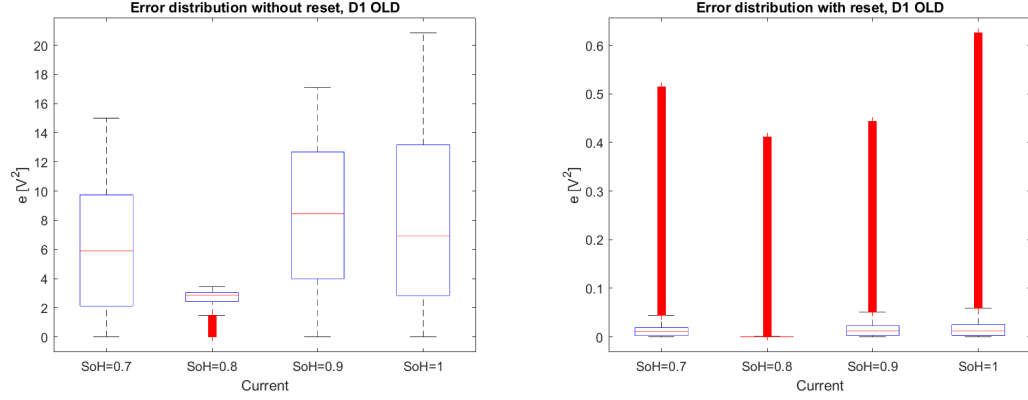


Figure 9.29: Old Battery D1, linear SoH values, error boxplot

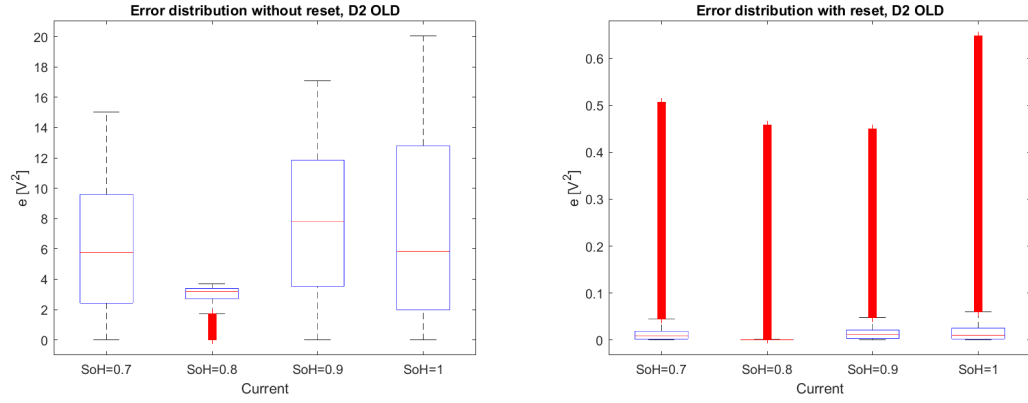


Figure 9.30: Old Battery D2, linear SoH values, error boxplot

In the next pool of graphs, from figure 9.31 to 9.36, the choice of the algorithm is depicted. The estimation of SoH is correct for all the batteries, consistently with the linear resolution of the ERMES set of EKF. In general, the estimation exploiting the periodical reset of the integral error provides better results than the simple integration. The performance of the simple estimator on the error is good enough for the work. The aim is to estimate SoH band values and the results are consistent. The addition of a periodical reset provides a statistically unique estimation. From the collected data, the algorithm tends to overestimate the SoH value when it lies in the middle of an interval between two different pre-set values.

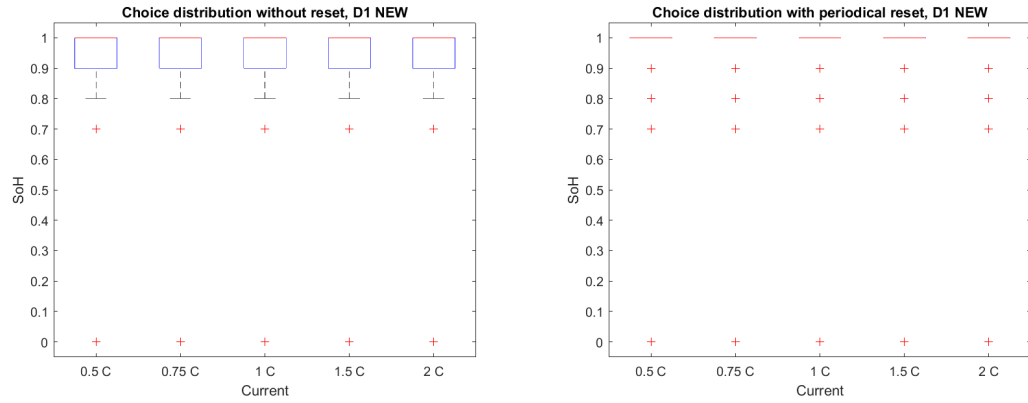


Figure 9.31: New Battery D1, linear SoH values, SoH estimation boxplot

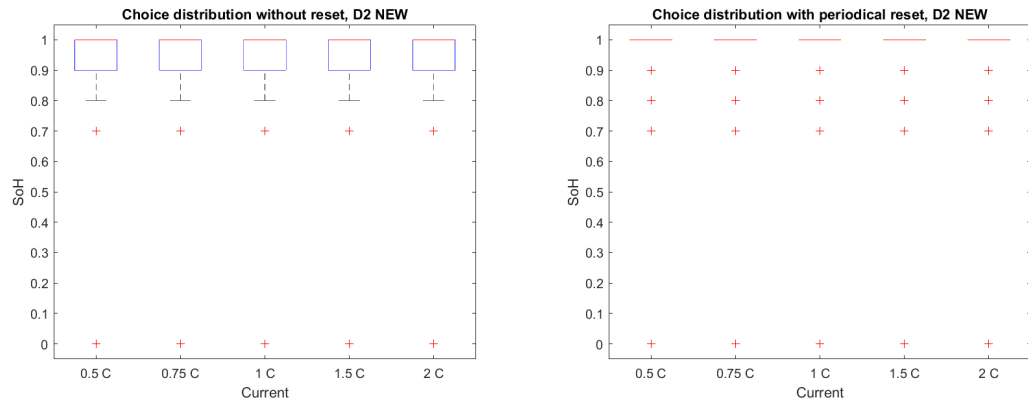


Figure 9.32: New Battery D2, linear SoH values, SoH estimation boxplot

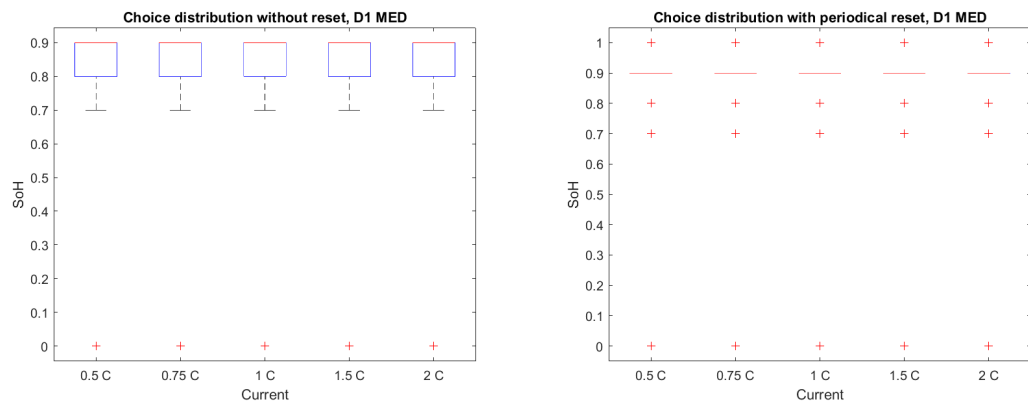


Figure 9.33: Medium Battery D1, linear SoH values, SoH estimation boxplot

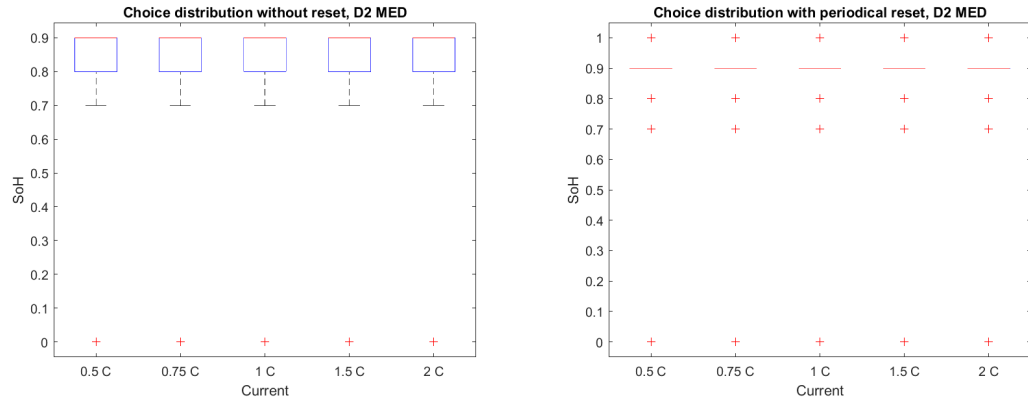


Figure 9.34: Medium Battery D2, linear SoH values, SoH estimation boxplot

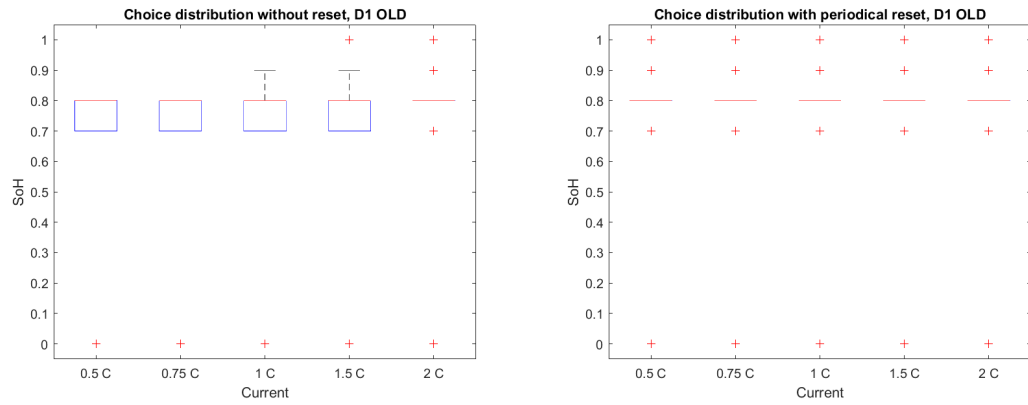


Figure 9.35: Old Battery D1, linear SoH values, SoH estimation boxplot

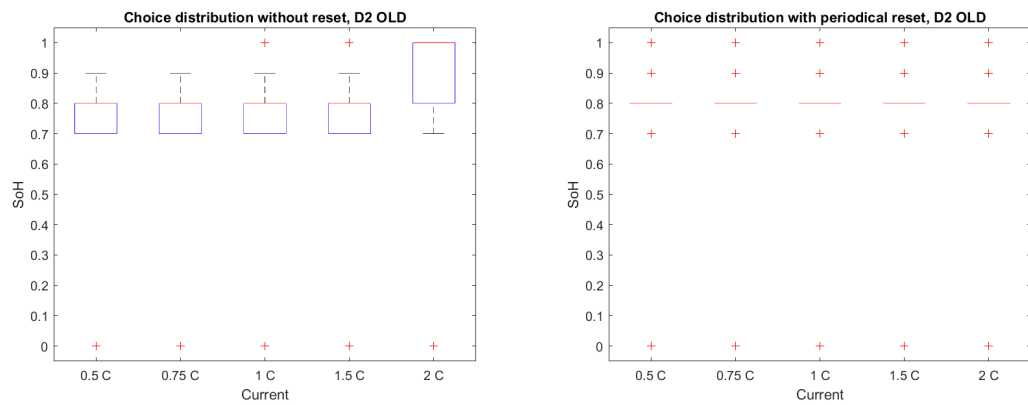


Figure 9.36: Old Battery D2, linear SoH values, SoH estimation boxplot

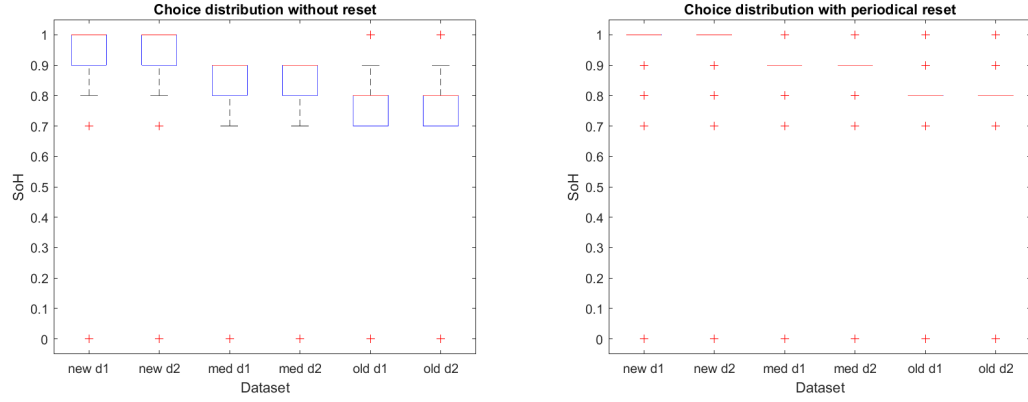


Figure 9.37: Summary of the test, SoH estimation boxplot

Finally, in figure 9.37 a summary of the test is reported. In all the 30 tests performed, varying the current and SoH of the datasets, the estimation of SoH is correct. The performance of the algorithm with a periodical reset is higher than the simple one.

Chapter 10

Conclusions and Future Works

10.1 Conclusions

The thesis aims to investigate a new methodology of simulation and parameter estimation in the Li-ion batteries field.

Firstly a depth analysis of the literature is performed to address the research. The most common methods are useful to build a digital twin for simulation purposes, but there are no real efforts to provide a robust method for wide-spectrum applications. The most critical factor is the complexity of the processes involved inside a Li-ion cell and the consequent side phenomena. Until the first research, it is clear that a simple and flexible method could be the key to resolving estimation problems. The previous solutions involve more complex tools at each iteration of the problem, each of them fits perfectly with the analyzed battery but is completely useless for the study of an unseen cell.

The solution proposed in this work exploits the simplest model possible with a dynamic transient to approximate the behavior of a real cell. The use of a lower complexity model with the data generated by a more complex model gives a good generalization of the methodology used. During the data generation and identification phases, a rigid energetic framework is defined to collect repeatable results. The definition of *SoC*, *empty* or *full battery*, *SoH*, and *Capacity* could influence significantly the results of the performed test. All the possible ways are investigated at the same time and only the best is chosen for the successive steps.

The identification phase is guided by two performance indexes: the closed-loop simulation error compared to the original data and the qualitative shapes of the resultant parameter LUTs. Instead of researching the direct correlation between data, as done with machine learning techniques, the collection of data is driven to

some intuitive results. This approach might lead to bias-errors in the approach, but the issue can be avoided by designing a significant validation phase. The process leads to a new framework, where all the parameters are approximated to almost-linear functions dependent on the same quantities. The parameters of different SoH values of the same cell are seen as values of the same function, dependent on some quantities. This vision is innovative and not deeply studied in the literature. The functions are determined from the data and they do have not to relate to physical meaning. The closed-loop approach for simulation guarantees good accuracy in the fitting of different validation datasets, overcoming the issues derived by the low-grade model and the approximations done on the parameters.

The last step is the introduction of SoH estimation using the ERMES algorithm. The traditional way to estimate SoH is the use of an augmented states EKF. It is a complex solution with some issues in terms of convergence. ERMES exploits the previous characterization of the cell to build a multi-model estimator with different pre-set values of SoH. The estimation is done based on which model fits better the original data. The integral of the squared error is used as a decision variable for the choice of the best model. This approach is simpler, faster, and computationally less expensive than the traditional way. The estimation is performed online and does not require a specific test to be achieved, so the algorithm could be introduced in real-time monitoring systems.

10.2 Future works

This thesis shows the potential of the application of ERMES algorithm and a more general multi-model approach to the SoH estimation problem of a Li-ion cell.

The first successive step is the experimental validation of the framework defined in this work. The data have to be collected following the procedures exploited in the simulation performed on the mathematical models. The inclusion of more cells, SoH values, and battery chemistry is an important step to generalize the method to a wider spectrum of applications.

Another important improvement is in ERMES definition. As seen in the thesis, a constrained version of the EKF is needed. To not impact the performance of the system, the projection of the solution into a feasible sub-space is chosen, but it is a sub-optimal solution. Future works could investigate other solutions, such as the use of different observers from EKF.

Regarding ERMES, a deep statistical analysis of the residuals is needed to confirm the results obtained in this work. Considering the EKFs, fine-tuning the internal parameters can improve the speed of the correct estimation. While better performance is achieved with the periodical reset approach, the impact of the period length is not investigated in the thesis, so research in this field could improve the

results.

Appendix A

Battery model for data generation

In figure A.1, the model used for data generation is reported. The block scheme is performed on Simulink and Matlab platform. In figure A.2, it is reported the

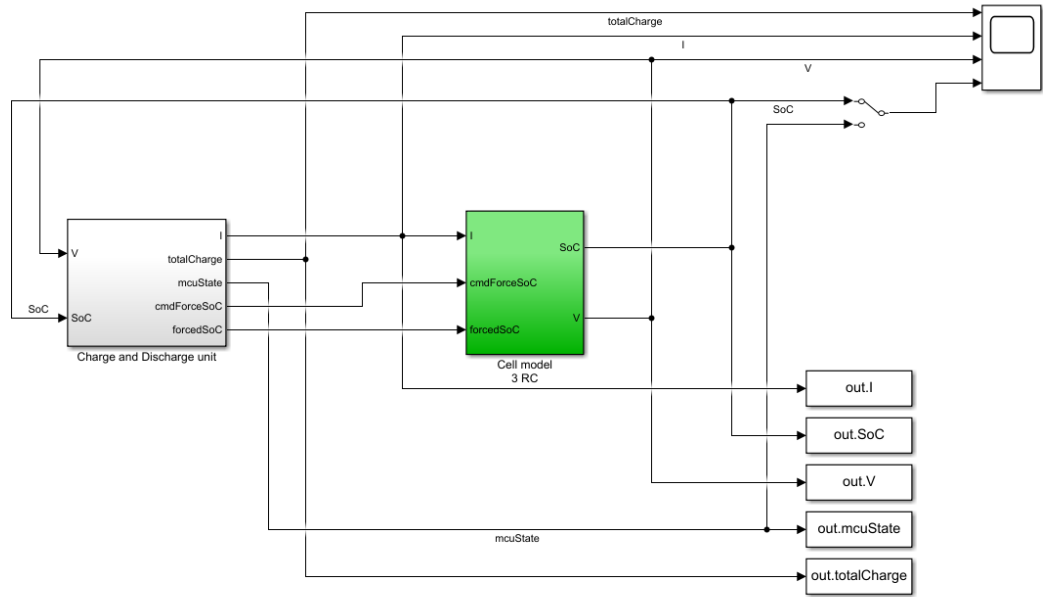


Figure A.1: Simulink model for data generation

details of the cell model block. The function *ModelCell* is listed at the end of the section.

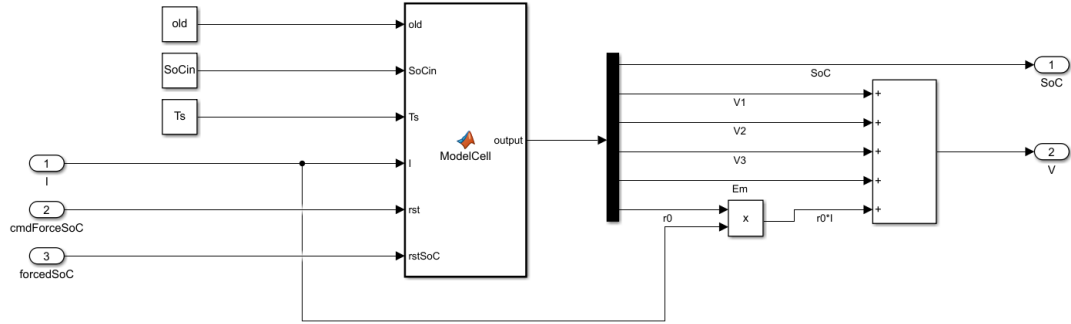


Figure A.2: Detail of the cell model block

```

1 function output = ModelCell(old, SoCin, Ts, I, rst, rstSoC)
2
3 persistent SoC;
4 if isempty(SoC)
5     SoC = SoCin;
6 end
7 persistent V1;
8 if isempty(V1)
9     V1 = 0;
10 end
11 persistent V2;
12 if isempty(V2)
13     V2 = 0;
14 end
15 persistent V3;
16 if isempty(V3)
17     V3 = 0;
18 end
19
20 SOC_LUT = [0 0.1 0.25 0.5 0.75 0.9 1];
21
22 if old == 0 % new
23
24     Capacity = 5.4;
25     Em_LUT = [3.51 3.56 3.65 3.75 3.93 4.02 4.18];
26     R0_LUT = [0.02 0.01 0.009 0.009 0.008 0.007 0.008];
27
28     R1_LUT = [0.006 0.003 0.0035 0.0032 0.004 0.0027 0.0029];
29     t1 = [10 12 15 12 20 15 12];
30     C1_LUT = t1 ./ R1_LUT;

```

```

31
32 R2_LUT = [0.0025 0.0017 0.0013 0.0012 0.0021 0.0025 0.0026];
33 t2 = [25 40 75 125 80 100 110];
34 C2_LUT = t2 ./ R2_LUT;
35
36 R3_LUT = [0.025 0.013 0.007 0.003 0.007 0.012 0.005];
37 t3 = [1000 1250 1100 850 1000 1400 1100];
38 C3_LUT = t3 ./ R3_LUT;
39
40 elseif old == 1 % old
41
42 Capacity = 4.05;
43 Em_LUT = [3.56 3.61 3.68 3.78 3.94 4.07 4.17];
44 R0_LUT = [0.032 0.026 0.026 0.025 0.025 0.022 0.024];
45
46 R1_LUT = [0.003 0.005 0.0052 0.0045 0.0075 0.004 0.007];
47 t1 = [15 10 12 60 150 20 18];
48 C1_LUT = t1 ./ R1_LUT;
49
50 R2_LUT = [0.005 0.01 0.02 0.14 0.065 0.01 0.003];
51 t2 = [150 400 10000 6500 2500 150 120];
52 C2_LUT = t2 ./ R2_LUT;
53
54 R3_LUT = [0.075 0.035 0.16 0.2 0.175 0.13 0.025];
55 t3 = [4000 2100 6500 6400 8000 3500 2000];
56 C3_LUT = t3 ./ R3_LUT;
57
58 else % medium
59
60 Capacity = 4.725;
61 Em_LUT = [3.535 3.585 3.665 3.765 3.935 4.045 4.175];
62 R0_LUT = [0.026 0.018 0.0175 0.017 0.0165 0.0145 0.016];
63
64 R1_LUT = [0.045 0.004 0.0043 0.0039 0.0057 0.0034 0.0049];
65 t1 = [12.5 11 13.5 36 85 17.5 15];
66 C1_LUT = t1 ./ R1_LUT;
67
68 R2_LUT = [0.0037 0.0059 0.0106 0.0706 0.0336 0.0063 0.0028];
69 t2 = [0.0875 0.22 5.037 3.322 1.29 0.125 0.115];
70 C2_LUT = t2 ./ R2_LUT;
71
72 R3_LUT = [0.05 0.024 0.0835 0.1015 0.091 0.071 0.015];
73 t3 = [0.05 0.024 0.0835 0.1015 0.091 0.071 0.015];
74 C3_LUT = t3 ./ R3_LUT;
75
76 end
77
78 r0 = interp1(SOC_LUT, R0_LUT, SoC, 'linear', 'extrap');
79 r1 = interp1(SOC_LUT, R1_LUT, SoC, 'linear', 'extrap');

```

```

80 r2 = interp1(SOC_LUT, R2_LUT, SoC, 'linear', 'extrap');
81 r3 = interp1(SOC_LUT, R3_LUT, SoC, 'linear', 'extrap');
82 c1 = interp1(SOC_LUT, C1_LUT, SoC, 'linear', 'extrap');
83 c2 = interp1(SOC_LUT, C2_LUT, SoC, 'linear', 'extrap');
84 c3 = interp1(SOC_LUT, C3_LUT, SoC, 'linear', 'extrap');
85 Em = interp1(SOC_LUT, Em_LUT, SoC, 'linear', 'extrap');
86
87 V1 = V1 + (-(1/(r1*c1))*V1 + (1/c1)*I)*Ts;
88 V2 = V2 + (-(1/(r2*c2))*V2 + (1/c2)*I)*Ts;
89 V3 = V3 + (-(1/(r3*c3))*V3 + (1/c3)*I)*Ts;
90
91 SoC = SoC + I*Ts/(Capacity*3600);
92
93 if rst == 1
94     SoC = rstSoC;
95 end
96
97 output = [SoC, V1, V2, V3, Em, r0];
98
99 end

```

Appendix B

Matlab codes: Identification

B.1 Static identification

```
1 close all
2 clear
3 clc
4
5 load Idataset2new
6
7 Ts = 0.1;
8
9 [h, 1] = size(OCV);
10
11 for ih = 1:h
12
13     for il = 1:l
14
15         % Find start and finish of each current pulse
16         cnt = 1;
17         found = 0;
18         for i = 1:length(s_current{ih,il})
19             if s_current{ih,il}(i)<0 && found==0
20                 i_current{ih,il}(cnt, 1) = i;
21                 found = 1;
22             elseif s_current{ih,il}(i)==0 && found==1
23                 found = 0;
24                 i_current{ih,il}(cnt, 2) = i-1;
25                 cnt = cnt + 1;
26             end
27         end
28
29         % Extract data related to current pulses
```

```

30     cnt = 1;
31     j = 1;
32     for i = 1:length(s_current{ih,il})
33         if i>=i_current{ih,il}(cnt,1)+1 && i<=i_current{ih,il}(
cnt,2)
34             Idata{ih,il}(cnt,j) = s_current{ih,il}(i);
35             Vdata{ih,il}(cnt,j) = s_voltage{ih,il}(i);
36             SOCdata{ih,il}(cnt,j) = s_soc{ih,il}(i);
37             j = j + 1;
38         end
39
40         if i==i_current{ih,il}(cnt,2) && cnt<length(i_current{ih,
il})
41             cnt = cnt + 1;
42             j = 1;
43         end
44     end
45
46     % Extract SOC/OCV relation from the data
47     OCV_LUT{ih,il} = OCV{ih,il}(1,:);
48     SOC_OCV_LUT{ih,il} = SOCn{ih,il}(1,:);
49
50     [n, nn] = size(SOCdata{ih,il});
51     N0 = 700;
52
53     for i = 1:n
54         cnt = 1;
55         beta = [];
56         soc = [];
57         for j = N0:nn
58             ocv = interp1(SOC_OCV_LUT{ih,il}, OCV_LUT{ih,il},
SOCdata{ih,il}(i,j), 'linear', 'extrap');
59             Y = Vdata{ih,il}(i,j) - ocv;
60             X = Idata{ih,il}(i,j);
61             beta(cnt) = Y / X;
62             soc(cnt) = SOCdata{ih,il}(i,j);
63             cnt = cnt + 1;
64         end
65         R_LUT{ih,il}(i) = mean(beta);
66         SOC_R_LUT{ih,il}(i) = mean(soc);
67     end
68
69     SOC_R_LUT{ih,il}(end) = [];
70     R_LUT{ih,il}(end) = [];
71
72     for t = 1:length(s_time{ih,il})
73         ocv = interp1(SOC_OCV_LUT{ih,il}, OCV_LUT{ih,il}, s_soc{
ih,il}(1,t), 'linear', 'extrap');

```

```

74         r = interp1(SOC_R_LUT{ih,il}, R_LUT{ih,il}, s_soc{ih,il}
75         }(1,t), 'linear', 'extrap');
76         VsimpleR{ih,il}(1,t) = ocv + r*s_current{ih,il}(1,t);
77     end
78     figure, plot(s_time{ih,il}, s_voltage{ih,il}, 'k', ...
79     s_time{ih,il}, VsimpleR{ih,il}, 'r')
80
81 end
82
83 end
84
85 save('ID1_parametersStatic_Idataset2new.mat', ...
86 'SOC_OCV_LUT', 'OCV_LUT', ...
87 'SOC_R_LUT', 'R_LUT', ...
88 'VsimpleR')

```

B.2 Identification from static parameters

```

1 close all
2 clear
3 clc
4
5 load Idataset2new
6 load ID1_parametersStatic_Idataset2new
7
8 SOC_Rs_LUT = SOC_R_LUT;
9 Rs_LUT = R_LUT;
10 clear SOC_R_LUT R_LUT
11
12 Ts = 0.1;
13
14 [h, l] = size(OCV);
15
16 for ih = 1:h
17     for il = 1:l
18
19         % Find start and finish of each current period
20         cnt = 1;
21         found = 0;
22         flag = 0;
23         for i = 1:length(s_current{ih,il})
24             if s_current{ih,il}(i)<0 && found==0
25                 if cnt>1 && flag==0

```



```

27         i_current{ih,il}(cnt-1, 2) = i-1;
28     end
29     i_current{ih,il}(cnt, 1) = i;
30     found = 1;
31     flag = 0;
32     elseif found==1 && s_state{ih,il}(i)~=3
33         found = 0;
34         i_current{ih,il}(cnt, 2) = i-1;
35         cnt = cnt + 1;
36         flag = 1;
37     elseif s_current{ih,il}(i)==0 && found==1
38         found = 0;
39         cnt = cnt + 1;
40     end
41 end
42
43 % Extract data related to current pulses
44 cnt = 1;
45 j = 1;
46 for i = 1:length(s_current{ih,il})
47     if i>=i_current{ih,il}(cnt,1)+1 && i<=i_current{ih,il}(
cnt,2)
48         Idata{ih,il}(cnt,j) = s_current{ih,il}(i);
49         Vdata{ih,il}(cnt,j) = s_voltage{ih,il}(i);
50         SOCdata{ih,il}(cnt,j) = s_soc{ih,il}(i);
51         Tdata{ih,il}(cnt,j) = s_time{ih,il}(i);
52         OCVdata{ih,il}(cnt,j) = interp1(SOC_OCV_LUT{ih,il},
OCV_LUT{ih,il}, SOCdata{ih,il}(cnt,j), 'linear', 'extrap');
53         j = j + 1;
54     end
55
56     if i==i_current{ih,il}(cnt,2) && cnt<length(i_current{ih,
il})
57         cnt = cnt + 1;
58         j = 1;
59     end
60 end
61
62 figure()
63 [n,nn] = size(i_current{ih,il});
64 for i = 1:n
65     plot(s_time{ih,il}(i_current{ih,il}(i,1):i_current{ih,il}
}(i,2)), ...
66         s_voltage{ih,il}(i_current{ih,il}(i,1):i_current{ih,
il}(i,2)))
67     hold on
68 end
69
70 % Generate the SOC mean for each current pulse

```

```

71 [n, nn] = size(SOCdata{ih, il});
72 for i = 1:n
73     dSOC(i,1) = SOCdata{ih, il}(i,1);
74     dSOC(i,2) = SOCdata{ih, il}(i,end);
75     SOCmean{ih, il}(i) = (dSOC(i,1)+dSOC(i,2))/2;
76 end
77
78 % Extract SOC/OCV relation from the data
79 OCV_LUT{ih, il} = OCV{ih, il}(1,:);
80 SOC_OCV_LUT{ih, il} = SOCn{ih, il}(1,:);
81
82 % Generate the SOC mean for each current pulse
83 [n, nn] = size(SOCdata{ih, il});
84 for i = 1:n
85     dSOC(i,1) = SOCdata{ih, il}(i,1);
86     dSOC(i,2) = SOCdata{ih, il}(i,end);
87     SOCmean{ih, il}(i) = (dSOC(i,1)+dSOC(i,2))/2;
88 end
89
90 figure()
91 [n,nn] = size(i_current{ih, il});
92 for i = 1:n
93     plot(s_time{ih, il}(i_current{ih, il}(i,1):i_current{ih, il}
94         }(i,2)), ...
95         s_voltage{ih, il}(i_current{ih, il}(i,1):i_current{ih,
96         il}(i,2)))
97         hold on
98 end
99
100 % All togheter
101
102 [n, nn] = size(SOCdata{ih, il});
103
104 for i = 1:n
105     for j = 2:nn
106         OCVk = interp1(SOC_OCV_LUT{ih, il}, OCV_LUT{ih, il},
107             SOCdata{ih, il}(i,j), 'linear', 'extrap');
108         OCVkp1 = interp1(SOC_OCV_LUT{ih, il}, OCV_LUT{ih, il},
109             SOCdata{ih, il}(i,j-1), 'linear', 'extrap');
110
111         Y{ih, il}(j-1,i) = Vdata{ih, il}(i,j) - Vdata{ih, il}(i,
112             j-1) - OCVk + OCVkp1;
113
114         X1{ih, il}(j-1,i) = Idata{ih, il}(i,j);
115         X2{ih, il}(j-1,i) = Idata{ih, il}(i,j-1);
116         X3{ih, il}(j-1,i) = OCVkp1 - Vdata{ih, il}(i,j-1);
117
118     end
119
120 end

```

```

115     X = [X1{ih, il}(:,i) X2{ih, il}(:,i) X3{ih, il}(:,i)];
116     b = X\Y{ih, il}(:,i);
117     alpha{ih, il}(i) = b(1);
118     beta{ih, il}(i) = b(2);
119     gamma{ih, il}(i) = b(3);
120
121     Rs{ih, il}(i) = interp1(SOC_Rs_LUT{ih, il}, Rs_LUT{ih, il},
SOCdata{ih, il}(i,j), 'linear', 'extrap');
122
123     tau{ih, il}(i) = gamma{ih, il}(i) / Ts;
124     R01{ih, il}(i) = beta{ih, il}(i) / (tau{ih, il}(i)*Ts-1);
125     R02{ih, il}(i) = (alpha{ih, il}(i) - Rs{ih, il}(i)*tau{ih, il}
}(i)*Ts)/(1-tau{ih, il}(i)*Ts);
126     R1{ih, il}(i) = Rs{ih, il}(i) - R01{ih, il}(i);
127     R2{ih, il}(i) = Rs{ih, il}(i) - R02{ih, il}(i);
128     Crc{ih, il}(i) = 1 / (tau{ih, il}(i)*R1{ih, il}(i));
129
130     end
131
132     end
133
134 end
135
136 figure()
137 subplot(2,2,1)
138 cnt = 1;
139 for ih = 1:h
140     for il = 1:l
141         plot(SOCmean{ih, il}, R01{ih, il}, 'r', ...
SOCmean{ih, il}, R02{ih, il}, 'b')
142         hold on
143         cnt = cnt + 1;
144     end
145 end
146 xlabel('SoCn'), ylabel('R_0 [\Omega]'), grid on, ...
147 title('R_0 vs SOCn')
148
149
150 subplot(2,2,2)
151 cnt = 1;
152 for ih = 1:h
153     for il = 1:l
154         plot(SOCmean{ih, il}(1,:), R1{ih, il}(1,:), 'r', ...
SOCmean{ih, il}(1,:), R2{ih, il}(1,:), 'b')
155         hold on
156         cnt = cnt + 1;
157     end
158 end
159 xlabel('SoCn'), ylabel('R [\Omega]'), grid on, ...
160 title('R vs SOCn')
161

```

```

162 subplot(2,2,3)
163 cnt = 1;
164 for ih = 1:h
165     for il = 1:l
166         plot(SOCmean{ih,il}(1,:), tau{ih,il}(1,:), '*')
167         hold on
168         legendInfo{cnt} = ['SoCin=' num2str(SoCinVect(ih)) ', Iamp='
169         num2str(abs(IampVect(il)/5.4)) 'C'];
170         cnt = cnt + 1;
171     end
172 end
173 legend(legendInfo), xlabel('SoCn'), ylabel('\tau [s]'), grid on, ...
174     title('\tau vs SOCn')
175
176 subplot(2,2,4)
177 cnt = 1;
178 for ih = 1:h
179     for il = 1:l
180         plot(SOCmean{ih,il}(1,:), R01{ih,il}(1,:)+R1{ih,il}(1,:), 'r'
181         , ... SOCmean{ih,il}(1,:), R02{ih,il}(1,:)+R2{ih,il}(1,:), 'b')
182         hold on
183         cnt = cnt + 1;
184     end
185 end
186 xlabel('SoCn'), ylabel('C [F]'), grid on, ...
187     title('C vs SOCn')
188
189 [SOC_R0_LUT, R0_LUT] = deleteDuplicates(SOCmean, R01);
190 [SOC_TAU_LUT, TAU_LUT] = deleteDuplicates(SOCmean, tau);
191 [SOC_R_LUT, R_LUT] = deleteDuplicates(SOCmean, R1);
192 [SOC_C_LUT, C_LUT] = deleteDuplicates(SOCmean, Crc);
193 [SOC_OCV_LUT, OCV_LUT] = deleteDuplicates(SOC_OCV_LUT, OCV_LUT);
194
195 save('ID1_parametersMode32_Idataset2new.mat', ...
196     'SOC_OCV_LUT', 'OCV_LUT', ...
197     'SOC_R0_LUT', 'R0_LUT', ...
198     'SOC_TAU_LUT', 'TAU_LUT', ...
199     'SOC_R_LUT', 'R_LUT', ...
200     'SOC_C_LUT', 'C_LUT')

```

B.3 Direct dynamical model identification

```

1 close all

```

```

2 clear
3 clc
4
5 load Idataset2new
6
7 Ts = 0.1;
8
9 [h, 1] = size(OCV);
10
11 for ih = 1:h
12
13     for il = 1:l
14
15         % Extract SOC/OCV relation from the data
16         OCV_LUT{ih, il} = OCV{ih, il}(1,:);
17         SOC_OCV_LUT{ih, il} = SOCr{ih, il}(1,:);
18
19         % Find start and finish of each current period
20         cnt = 1;
21         found = 0;
22         flag = 0;
23         for i = 1:length(s_current{ih, il})
24             if s_current{ih, il}(i)<0 && found==0
25                 if cnt>1 && flag==0
26                     i_current{ih, il}(cnt-1, 2) = i-1;
27                 end
28                 i_current{ih, il}(cnt, 1) = i;
29                 found = 1;
30                 flag = 0;
31             elseif found==1 && s_state{ih, il}(i)~=3
32                 found = 0;
33                 i_current{ih, il}(cnt, 2) = i-1;
34                 cnt = cnt + 1;
35                 flag = 1;
36             elseif s_current{ih, il}(i)==0 && found==1
37                 found = 0;
38                 cnt = cnt + 1;
39             end
40         end
41
42         % Extract data related to current pulses
43         cnt = 1;
44         j = 1;
45         for i = 1:length(s_current{ih, il})
46             if i>=i_current{ih, il}(cnt,1)+1 && i<=i_current{ih, il}(
cnt,2)
47                 Idata{ih, il}(cnt, j) = s_current{ih, il}(i);
48                 Vdata{ih, il}(cnt, j) = s_voltage{ih, il}(i);
49                 SOCdata{ih, il}(cnt, j) = s_soc{ih, il}(i);

```

```

50         Tdata{ih,il}(cnt,j) = s_time{ih,il}(i);
51         OCVdata{ih,il}(cnt,j) = interp1(SOC_OCV_LUT{ih,il},
OCV_LUT{ih,il}, SOCdata{ih,il}(cnt,j), 'linear', 'extrap');
52         j = j + 1;
53     end
54
55     if i==i_current{ih,il}(cnt,2) && cnt<length(i_current{ih,
il})
56         cnt = cnt + 1;
57         j = 1;
58     end
59 end
60
61 figure()
62 [n,nn] = size(i_current{ih,il});
63 for i = 1:n
64     plot(s_time{ih,il}(i_current{ih,il}(i,1):i_current{ih,il
}(i,2)), ...
65         s_voltage{ih,il}(i_current{ih,il}(i,1):i_current{ih,
il}(i,2)))
66     hold on
67 end
68
69 % Generate the SOC mean for each current pulse
70 [n, nn] = size(SOCdata{ih,il});
71 for i = 1:n
72     dSOC(i,1) = SOCdata{ih,il}(i,1);
73     dSOC(i,2) = SOCdata{ih,il}(i,end);
74     SOCmean{ih,il}(i) = (dSOC(i,1)+dSOC(i,2))/2;
75 end
76
77 figure()
78 [x,xx] = size(Tdata{1,il});
79 for i = 1:x
80     plot(Tdata{1,il}(i,:), Vdata{1,il}(i,:), 'r', ...
81         Tdata{1,il}(i,:), OCVdata{1,il}(i,:), 'b')
82     hold on
83 end
84
85 % All togheter
86 for i = 1:n
87     for j = 2:nn
88         OCVk = interp1(SOC_OCV_LUT{ih,il}, OCV_LUT{ih,il},
SOCdata{ih,il}(i,j), 'linear', 'extrap');
89         OCVkp1 = interp1(SOC_OCV_LUT{ih,il}, OCV_LUT{ih,il},
SOCdata{ih,il}(i,j-1), 'linear', 'extrap');
90         Y{ih,il}(j-1,i) = Vdata{ih,il}(i,j) - Vdata{ih,il}(i,
j-1) - OCVk + OCVkp1;
91         X1{ih,il}(j-1,i) = Idata{ih,il}(i,j);

```

```

192         X2{ih, il}(j-1,i) = Idata{ih, il}(i, j-1);
193         X3{ih, il}(j-1,i) = OCVkp1 - Vdata{ih, il}(i, j-1);
194     end
195
196     X = [X1{ih, il}(:,i) X2{ih, il}(:,i) X3{ih, il}(:,i)];
197     b = X\Y{ih, il}(:,i);
198     alpha{ih, il}(i) = b(1);
199     beta{ih, il}(i) = b(2);
200     gamma{ih, il}(i) = b(3);
201
202     tau{ih, il}(i) = gamma{ih, il}(i) / Ts;
203     R0{ih, il}(i) = beta{ih, il}(i) / (tau{ih, il}(i)*Ts-1);
204     R{ih, il}(i) = (alpha{ih, il}(i)-R0{ih, il}(i)) / (tau{ih, il}
    }(i)*Ts);
205     Crc{ih, il}(i) = 1 / (tau{ih, il}(i)*R{ih, il}(i));
206
207     end
208
209     end
210
211 end
212
213 figure()
214 subplot(2,2,1)
215 cnt = 1;
216 for ih = 1:h
217     for il = 1:l
218         plot(SOCmean{ih, il}, R0{ih, il}, '*')
219         hold on
220         legendInfo{cnt} = ['SoCin=' num2str(SoCinVect(ih)) ', Iamp='
221 num2str(abs(IampVect(il)/5.4)) 'C'];
222         cnt = cnt + 1;
223     end
224 end
225 legend(legendInfo), xlabel('SoCn'), ylabel('R_0 [\Omega]'), grid on,
226 ...
227     title('R_0 vs SOCn')
228
229 subplot(2,2,2)
230 cnt = 1;
231 for ih = 1:h
232     for il = 1:l
233         plot(SOCmean{ih, il}(1,:), R{ih, il}(1,:), '*')
234         hold on
235         legendInfo{cnt} = ['SoCin=' num2str(SoCinVect(ih)) ', Iamp='
236 num2str(abs(IampVect(il)/5.4)) 'C'];
237         cnt = cnt + 1;
238     end
239 end
240 end

```

```

137 legend(legendInfo), xlabel('SoCn'), ylabel('R [\Omega]'), grid on,
    ...
138     title('R vs SOCn')
139
140 subplot(2,2,3)
141 cnt = 1;
142 for ih = 1:h
143     for il = 1:l
144         plot(SOCmean{ih,il}(1,:), tau{ih,il}(1,:), '*')
145         hold on
146         legendInfo{cnt} = ['SoCin=' num2str(SoCinVect(ih)) ', Iamp='
num2str(abs(IampVect(il)/5.4)) 'C'];
147         cnt = cnt + 1;
148     end
149 end
150 legend(legendInfo), xlabel('SoCn'), ylabel('\tau [s]'), grid on, ...
151     title('\tau vs SOCn')
152
153 subplot(2,2,4)
154 cnt = 1;
155 for ih = 1:h
156     for il = 1:l
157         plot(SOCmean{ih,il}(1,:), Crc{ih,il}(1,:), '*')
158         hold on
159         legendInfo{cnt} = ['SoCin=' num2str(SoCinVect(ih)) ', Iamp='
num2str(abs(IampVect(il)/5.4)) 'C'];
160         cnt = cnt + 1;
161     end
162 end
163 legend(legendInfo), xlabel('SoCn'), ylabel('C [F]'), grid on, ...
164     title('C vs SOCn')
165
166 [SOC_R0_LUT, R0_LUT] = deleteDuplicates(SOCmean, R0);
167 [SOC_TAU_LUT, TAU_LUT] = deleteDuplicates(SOCmean, tau);
168 [SOC_R_LUT, R_LUT] = deleteDuplicates(SOCmean, R);
169 [SOC_C_LUT, C_LUT] = deleteDuplicates(SOCmean, Crc);
170 [SOC_OCV_LUT, OCV_LUT] = deleteDuplicates(SOC_OCV_LUT, OCV_LUT);
171
172 save('ID1_parametersMode2_Idataset2new.mat', ...
173     'SOC_OCV_LUT', 'OCV_LUT', ...
174     'SOC_R0_LUT', 'R0_LUT', ...
175     'SOC_TAU_LUT', 'TAU_LUT', ...
176     'SOC_R_LUT', 'R_LUT', ...
177     'SOC_C_LUT', 'C_LUT')

```


Appendix C

Matlab codes: Validation

C.1 Open-loop validation

```
1 close all
2 clear
3 clc
4
5 load Vdataset1new
6 load ID1_parametersMode2_Idataset2new
7
8 Ts = 0.1;
9
10 [h,l] = size(s_state);
11
12 i_discharge = cell(h,1);
13 Idata = cell(h,1);
14 Vdata = cell(h,1);
15 SOCdata = cell(h,1);
16 Tdata = cell(h,1);
17 simV = cell(h,1);
18 simSOC = cell(h,1);
19
20 for ih = 1:h
21     for il = 1:l
22
23         % Identify discharging fraction of time
24         cnt = 1;
25         found = 0;
26         for i = 1:length(s_state{ih,il})
27             if s_state{ih,il}(i)==3 && found==0
28                 i_discharge{ih,il}(cnt, 1) = i;
29                 found = 1;
```

```

30         elseif s_state{ih,il}(i)~=3 && found==1
31             found = 0;
32             i_discharge{ih,il}(cnt, 2) = i-1;
33             cnt = cnt + 1;
34         end
35     end
36
37     [n,nn] = size(i_discharge{ih,il});
38     for i = 1:n
39         Idata{ih,il}(i,:) = s_current{ih,il}(i_discharge{ih,il}(i
40 ,1):i_discharge{ih,il}(i,2));
41         Vdata{ih,il}(i,:) = s_voltage{ih,il}(i_discharge{ih,il}(i
42 ,1):i_discharge{ih,il}(i,2));
43         SOCdata{ih,il}(i,:) = s_soc{ih,il}(i_discharge{ih,il}(i
44 ,1):i_discharge{ih,il}(i,2));
45         Tdata{ih,il}(i,:) = s_time{ih,il}(i_discharge{ih,il}(i,1)
46 :i_discharge{ih,il}(i,2));
47     end
48
49     for i = 1:n
50         simV{ih,il}(i,1) = Vdata{ih,il}(i,1);
51         simSOC{ih,il}(i,1) = SOCdata{ih,il}(i,1);
52         Vrc = 0;
53         Vocv = interp1(SOC_OCV_LUT{ih,il}, OCV_LUT{ih,il}, simSOC
54 {ih,il}(1,1), 'linear', 'extrap');
55         for k = 2:length(Idata{ih,il}(i,:))
56             simSOC{ih,il}(i,k) = simSOC{ih,il}(i,k-1) + Idata{ih,
57 il}(i,k)*Ts/(Capacity*3600);
58             R0 = interp1(SOC_R0_LUT{ih,il}, R0_LUT{ih,il}, simSOC
59 {ih,il}(i,k), 'linear', 'extrap');
60             R = interp1(SOC_R_LUT{ih,il}, R_LUT{ih,il}, simSOC{ih
61 ,il}(i,k), 'linear', 'extrap');
62             tau = interp1(SOC_TAU_LUT{ih,il}, TAU_LUT{ih,il},
63 simSOC{ih,il}(i,k), 'linear', 'extrap');
64             Vocv_p = Vocv;
65             Vocv = interp1(SOC_OCV_LUT{ih,il}, OCV_LUT{ih,il},
66 simSOC{ih,il}(i,k), 'linear', 'extrap');
67             Vrc_p = Vrc;
68             Vrc = Vrc_p - tau*Ts*Vrc_p + R*tau*Ts*Idata{ih,il}(i,
69 k);
70             simV{ih,il}(i,k) = Vocv + R0*Idata{ih,il}(i,k) + Vrc;
71             simR{ih,il}(i,k) = R;
72             simR0{ih,il}(i,k) = R0;
73             simC{ih,il}(i,k) = 1 / (R*tau);
74         end
75     end
76 end
77

```

```

68 % Plots
69 for ih = 1:h
70     for il = 1:l
71         figure()
72         subplot(1,2,1)
73         [n,nn] = size(Vdata{ih,il});
74         for j = 1:n
75             plot(Tdata{ih,il}(j,:), Vdata{ih,il}(j,:), '-r')
76             hold on
77             plot(Tdata{ih,il}(j,:), simV{ih,il}(j,:), '-b')
78         end
79         title('Terminal voltage'), xlabel('V [V]'), ylabel('t [s]'),
80         grid on
81         subplot(1,2,2)
82         [n,nn] = size(SOCdata{ih,il});
83         for j = 1:n
84             plot(Tdata{ih,il}(j,:), SOCdata{ih,il}(j,:), '-r')
85             hold on
86             plot(Tdata{ih,il}(j,:), simSOC{ih,il}(j,:), '-b')
87         end
88         title('SoC'), xlabel('SoC'), ylabel('t [s]'), grid on
89     end
90 end
91 save('ID1_Vsim_Idataset2_mode2.mat', ...
92     'Idata', 'Vdata', 'SOCdata', 'Tdata', ...
93     'simV', 'simSOC', 'simR', 'simR0', 'simC')

```

C.2 Closed-loop validation

```

1 close all
2 clear
3 clc
4
5 load Vdataset1new
6 load ID1_parametersMode4_Idataset2new
7
8 Ts = 0.1;
9
10 [h,l] = size(s_state);
11
12 i_discharge = cell(h,l);
13 Idata = cell(h,l);
14 Vdata = cell(h,l);
15 SOCdata = cell(h,l);

```

```

16 Tdata = cell(h,1);
17 simV = cell(h,1);
18 simSOC = cell(h,1);
19
20 N0 = 15;
21
22 for ih = 1:h
23     for il = 1:l
24
25         SOC_R_LUT{ih,il}(end) = [];
26         R_LUT{ih,il}(end) = [];
27         SOC_R0_LUT{ih,il}(end) = [];
28         R0_LUT{ih,il}(end) = [];
29         SOC_TAU_LUT{ih,il}(end) = [];
30         TAU_LUT{ih,il}(end) = [];
31
32         Vdata{ih,il} = s_voltage{ih,il}(N0:end);
33         Tdata{ih,il} = s_time{ih,il}(N0:end);
34         Idata{ih,il} = s_current{ih,il}(N0:end);
35         SOCdata{ih,il} = s_soc{ih,il}(N0:end);
36
37         Xh(:,1) = [Vdata{ih,il}(1); SOCdata{ih,il}(1)];
38         V1 = diag([0.1 0.05]);
39         P{1} = V1;
40         V2 = 0.1;
41
42         Ph{1} = V1;
43         Pf{1} = V1;
44         Xf(:,1) = [0; SOCdata{ih,il}(1)];
45         Yf(1) = Vdata{ih,il}(1);
46
47         for t = 2:length(s_voltage{ih,il}(N0:end))
48
49             Vocv = interp1(SOC_OCV_LUT{ih,il}, OCV_LUT{ih,il},
SOCdata{ih,il}(t), 'linear', 'extrap');
50             R0 = interp1(SOC_R0_LUT{ih,il}, R0_LUT{ih,il}, SOCdata{ih
,il}(t), 'linear', 'extrap');
51             R = interp1(SOC_R_LUT{ih,il}, R_LUT{ih,il}, SOCdata{ih,il
}(t), 'linear', 'extrap');
52             tau = interp1(SOC_TAU_LUT{ih,il}, TAU_LUT{ih,il}, SOCdata
{ih,il}(t), 'linear', 'extrap');
53
54             Y(t) = Vdata{ih,il}(t);
55             u(t) = Idata{ih,il}(t);
56
57             % Predictor
58             A = [1-tau*Ts 0; 0 1];
59             C = [1 0];
60

```

```

61 % Filter
62 Xh(1,t) = (1-tau*Ts)*Xf(1,t-1) + R+tau*Ts*u(t);
63 Xh(2,t) = Xf(2,t-1) + Ts/(Capacity*3600)*u(t);
64
65 Ph{t} = A*Pf{t-1}*A' + V1;
66 Yh(t) = Vocv + Xh(1,t) + R0*u(t);
67 e = Y(t) - Yh(t);
68 K{t} = Ph{t}*C'*inv(C*Ph{t}*C' + V2);
69
70 Xf(:,t) = Xh(:,t) + K{t}*e;
71 Yf(t) = Vocv + Xf(1,t) + R0*u(t);
72
73 Pf{t} = (eye(2) - K{t}*C)*Ph{t};
74
75 simVf{ih,il}(t) = Yf(t);
76 simVrcf{ih,il}(t) = Xf(1,t);
77 simSOCf{ih,il}(t) = Xf(2,t);
78
79
80 end
81
82 end
83 end
84
85 % Plots
86 for ih = 1:h
87     for il = 1:l
88
89         figure()
90         subplot(1,2,1)
91         [n,nn] = size(Vdata{ih,il});
92         plot(Tdata{ih,il}, Vdata{ih,il}, '-k')
93         hold on
94         plot(Tdata{ih,il}, simVf{ih,il}, '-r')
95         legend('Original data', 'Kalman')
96         title('Terminal voltage'), ylabel('Voltage [V]'), xlabel('
Time [s]')
97         grid on, ylim([2.5 4.5])
98
99         subplot(1,2,2)
100        plot(Tdata{ih,il}, SOCdata{ih,il}, '-r')
101        hold on
102        plot(Tdata{ih,il}, simSOCf{ih,il}, '-b')
103        legend('Original data', 'Kalman')
104        title('SoC'), ylabel('SoC'), xlabel('Time [s]')
105        ylim([-0.1 1.1])
106
107    end
108 end

```

```
109 |
110 | save( 'ID1_Vsim_Vdataset1new_mode4Kalman.mat', ...
111 |      'Idata', 'Vdata', 'SOCdata', 'Tdata', ...
112 |      'simVf', 'simSOCf')
```

Appendix D

Matlab codes: Parameters approximation

D.1 τ current independent

```
1 close all
2 clear
3 clc
4
5 load ID1_parametersMode2_Idataset2new
6
7 Capacity = 5.4;
8
9 [h, l] = size(C_LUT);
10
11 for il = 1:l
12
13     C_LUT{il}(end) = [];
14     SOC_C_LUT{il}(end) = [];
15
16     R_LUT{il}(end) = [];
17     SOC_R_LUT{il}(end) = [];
18
19     R0_LUT{il}(end) = [];
20     SOC_R0_LUT{il}(end) = [];
21
22     TAU_LUT{il}(end) = [];
23     SOC_TAU_LUT{il}(end) = [];
24
25 end
26
```

```

27 %% TAU
28
29 cnt = 1;
30 X = [];
31 P = [1 1e-4];
32 for il = 1:l
33     for j = 1:length(SOC_TAU_LUT{il})
34         X(cnt,1) = SOC_TAU_LUT{il}(j);
35         Y(cnt,1) = TAU_LUT{il}(j);
36         cnt = cnt + 1;
37     end
38 end
39 X = X - P(1);
40 Y = Y - P(2);
41 pTAU(1) = X \ Y;
42 pTAU(2) = P(2) - pTAU(1)*P(1);
43
44 soc = linspace(0,1,100)';
45 tau = [soc ones(100,1)] * [pTAU(1); pTAU(2)];
46
47 figure()
48 for il = 1:l
49     plot(SOC_TAU_LUT{il}, TAU_LUT{il}, 'b')
50     hold on
51 end
52 plot(soc, tau, 'r', 'LineWidth', 1)
53 title('TAU, New Battery'), xlabel('SoC_R'), ylabel('\tau [s^{-1}]'),
    grid on, xlim([0 1])
54
55 %% R0
56
57 cnt = 1;
58 X = [];
59 for il = 1:l
60     for j = 1:length(SOC_R0_LUT{il})
61         X(cnt,1) = SOC_R0_LUT{il}(j);
62         Y(cnt,1) = R0_LUT{il}(j);
63         cnt = cnt + 1;
64     end
65 end
66 X(:,2) = ones(length(X(:,1)),1);
67 pR0 = X \ Y;
68
69 soc = linspace(0,1,100)';
70 R0 = [soc ones(100,1)] * [pR0(1); pR0(2)];
71
72 figure()
73 for il = 1:l
74     plot(SOC_R0_LUT{il}, R0_LUT{il}, 'b')

```



```

75     hold on
76 end
77 plot(soc, R0, 'r', 'LineWidth', 1)
78 title('R_0, New Battery'), xlabel('SoCr'), ylabel('R_0 [\Omega]'),
    grid on, xlim([0 1])
79
80 %% OCV
81
82 m = (OCV_LUT{1}(end)-OCV_LUT{1}(1))/(SOC_OCV_LUT{1}(end)-SOC_OCV_LUT
    {1}(1));
83 pOCV(1) = m;
84 pOCV(2) = OCV_LUT{1}(1) - m*SOC_OCV_LUT{1}(1);
85
86 soc = linspace(0,1,100)';
87 OCV = [soc ones(100,1)] * [pOCV(1); pOCV(2)];
88
89 figure()
90 for il = 1:l
91     plot(SOC_OCV_LUT{il}, OCV_LUT{il}, 'b')
92     hold on
93 end
94 plot(soc, OCV, 'r', 'LineWidth', 1)
95 title('OCV, New Battery'), xlabel('SoC_R'), ylabel('OCV [V]'), grid
    on, xlim([0 1])
96
97 clearvars -except pTAU pR0 pOCV
98
99 return
100
101 %% R
102
103 load Idataset2new
104
105 Ts = 0.1;
106
107 ih = 1;
108 [h,l] = size(s_voltage);
109 for il = 1:l
110
111     % Find start and finish of each current period
112     cnt = 1;
113     found = 0;
114     flag = 0;
115     for i = 1:length(s_current{ih,il})
116         if s_current{ih,il}(i)<0 && found==0
117             if cnt>1 && flag==0
118                 i_current{ih,il}(cnt-1, 2) = i-1;
119             end
120             i_current{ih,il}(cnt, 1) = i;

```

```

121         found = 1;
122         flag = 0;
123     elseif found==1 && s_state{ih,il}(i)~=3
124         found = 0;
125         i_current{ih,il}(cnt, 2) = i-1;
126         cnt = cnt + 1;
127         flag = 1;
128     elseif s_current{ih,il}(i)==0 && found==1
129         found = 0;
130         cnt = cnt + 1;
131     end
132 end
133
134 % Extract data related to current pulses
135 cnt = 1;
136 j = 1;
137 for i = 1:length(s_current{ih,il})
138     if i>=i_current{ih,il}(cnt,1)+1 && i<=i_current{ih,il}(cnt,2)
139         Idata{ih,il}(cnt,j) = s_current{ih,il}(i);
140         Vdata{ih,il}(cnt,j) = s_voltage{ih,il}(i);
141         SOCdata{ih,il}(cnt,j) = s_soc{ih,il}(i);
142         Tdata{ih,il}(cnt,j) = s_time{ih,il}(i);
143         j = j + 1;
144     end
145
146     if i==i_current{ih,il}(cnt,2) && cnt<length(i_current{ih,il})
147         cnt = cnt + 1;
148         j = 1;
149     end
150 end
151
152 % Generate the SOC mean for each current pulse
153 [n, nn] = size(SOCdata{ih,il});
154 for i = 1:n
155     dSOC(i,1) = SOCdata{ih,il}(i,1);
156     dSOC(i,2) = SOCdata{ih,il}(i,end);
157     SOCmean{ih,il}(i) = (dSOC(i,1)+dSOC(i,2))/2;
158 end
159
160 % All togheter
161 for i = 1:n
162
163     for j = 2:nn
164         OCVk = pOCV(1)*SOCdata{ih,il}(i,j) + pOCV(2);
165         OCVkp1 = pOCV(1)*SOCdata{ih,il}(i,j-1) + pOCV(2);
166         R0 = pR0(1)*SOCdata{ih,il}(i,j) + pR0(2);
167         tau = pTAU(1)*SOCdata{ih,il}(i,j) + pTAU(2);
168         Y{ih,il}(j-1,i) = Vdata{ih,il}(i,j) - Vdata{ih,il}(i,j-1)
169     end
170 end

```

```

169         OCVk + OCVkp1 - R0*Idata{ih,il}(i,j) + R0*Idata{ih,il
    }(i,j-1) + ...
170         tau*Ts*(Vdata{ih,il}(i,j-1) - OCVkp1 - R0*Idata{ih,il
    }(i,j-1));
171         X{ih,il}(j-1,i) = tau*Ts*Idata{ih,il}(i,j);
172         end
173
174         R{ih,il}(i) = X{ih,il}(:,i)\Y{ih,il}(:,i);
175
176     end
177
178 end
179
180 subplot(2,2,4)
181 cnt = 1;
182 for ih = 1:h
183     for il = 1:l
184         plot(SOCmean{ih,il}(1,:), R{ih,il}(1,:), '—*')
185         hold on
186         cnt = cnt + 1;
187     end
188 end
189 xlabel('SoCn'), ylabel('R [\Omega]'), grid on, title('R')
190
191 SOC_R_LUT = SOCmean;
192 R_LUT = R;
193
194 save('ID1_appParameters_linearConstTau_new.mat', ...
195     'pR0', 'pTAU', 'pOCV', 'SOC_R_LUT', 'R_LUT', 'Capacity', 'Ts');

```

D.2 τ current dependent

```

1 close all
2 clear
3 clc
4
5 load ID1_parametersMode2_Idataset2new
6
7 Capacity = 5.4;
8
9 [h,l] = size(C_LUT);
10
11 for il = 1:l
12
13     C_LUT{il}(end) = [];

```

```

14 SOC_C_LUT{il}(end) = [];
15
16 R_LUT{il}(end) = [];
17 SOC_R_LUT{il}(end) = [];
18
19 R0_LUT{il}(end) = [];
20 SOC_R0_LUT{il}(end) = [];
21
22 TAU_LUT{il}(end) = [];
23 SOC_TAU_LUT{il}(end) = [];
24
25 end
26
27 %% TAU
28
29 soc = linspace(0,1,100)';
30
31 P = [1 1e-4];
32 for il = 1:l
33     cnt = 1;
34     X = [];
35     for j = 1:length(SOC_TAU_LUT{il})
36         X(cnt,1) = SOC_TAU_LUT{il}(j);
37         Y(cnt,1) = TAU_LUT{il}(j);
38         cnt = cnt + 1;
39     end
40     X = X - P(1);
41     Y = Y - P(2);
42     pTAU{il}(1) = X \ Y;
43     pTAU{il}(2) = P(2) - pTAU{il}(1)*P(1);
44
45     tau{il} = [soc ones(100,1)] * pTAU{il}';
46
47 end
48
49 figure()
50 % subplot(2,2,1)
51 for il = 1:l
52     plot(SOC_TAU_LUT{il}, TAU_LUT{il}, 'b')
53     hold on
54     plot(soc, tau{il}, 'r', 'LineWidth', 1)
55 end
56 title('\tau, New Battery'), xlabel('SoC_R'), ylabel('\tau [s^{-1}]'),
    grid on, xlim([0 1])
57
58 %% R0
59
60 cnt = 1;
61 X = [];

```

```

62 for il = 1:l
63     for j = 1:length(SOC_TAU_LUT{il})
64         X(cnt,1) = SOC_R0_LUT{il}(j);
65         Y(cnt,1) = R0_LUT{il}(j);
66         cnt = cnt + 1;
67     end
68 end
69 X(:,2) = ones(length(X(:,1)),1);
70 pR0 = X \ Y;
71
72 soc = linspace(0,1,100)';
73 R0 = [soc ones(100,1)] * [pR0(1); pR0(2)];
74
75 figure
76 subplot(2,2,2)
77 for il = 1:l
78     plot(SOC_R0_LUT{il}, R0_LUT{il})
79     hold on
80 end
81 plot(soc, R0, 'r', 'LineWidth', 2)
82 title('R_0'), xlabel('SoCr'), ylabel('R_0 [\Omega]'), grid on, xlim
    ([0 1])
83
84 %% OCV
85
86 m = (OCV_LUT{1}(end)-OCV_LUT{1}(1))/(SOC_OCV_LUT{1}(end)-SOC_OCV_LUT
    {1}(1));
87 pOCV(1) = m;
88 pOCV(2) = OCV_LUT{1}(1) - m*SOC_OCV_LUT{1}(1);
89
90 soc = linspace(0,1,100)';
91 OCV = [soc ones(100,1)] * [pOCV(1); pOCV(2)];
92
93 subplot(2,2,3)
94 for il = 1:l
95     plot(SOC_OCV_LUT{il}, OCV_LUT{il})
96     hold on
97 end
98 plot(soc, OCV, 'r', 'LineWidth', 2)
99 title('OCV'), xlabel('SoCr'), ylabel('OCV [V]'), grid on, xlim([0 1])
100
101 clearvars -except pTAU pR0 pOCV
102
103 return
104
105 %% R
106
107 load Idataset2new
108

```

```

109 Ts = 0.1;
110
111 ih = 1;
112 [h,l] = size(s_voltage);
113 for il = 1:l
114
115     % Find start and finish of each current period
116     cnt = 1;
117     found = 0;
118     flag = 0;
119     for i = 1:length(s_current{ih,il})
120         if s_current{ih,il}(i)<0 && found==0
121             if cnt>1 && flag==0
122                 i_current{ih,il}(cnt-1, 2) = i-1;
123             end
124             i_current{ih,il}(cnt, 1) = i;
125             found = 1;
126             flag = 0;
127         elseif found==1 && s_state{ih,il}(i)~=3
128             found = 0;
129             i_current{ih,il}(cnt, 2) = i-1;
130             cnt = cnt + 1;
131             flag = 1;
132         elseif s_current{ih,il}(i)==0 && found==1
133             found = 0;
134             cnt = cnt + 1;
135         end
136     end
137
138     % Extract data related to current pulses
139     cnt = 1;
140     j = 1;
141     for i = 1:length(s_current{ih,il})
142         if i>=i_current{ih,il}(cnt,1)+1 && i<=i_current{ih,il}(cnt,2)
143             Idata{ih,il}(cnt,j) = s_current{ih,il}(i);
144             Vdata{ih,il}(cnt,j) = s_voltage{ih,il}(i);
145             SOCdata{ih,il}(cnt,j) = s_soc{ih,il}(i);
146             Tdata{ih,il}(cnt,j) = s_time{ih,il}(i);
147             j = j + 1;
148         end
149
150         if i==i_current{ih,il}(cnt,2) && cnt<length(i_current{ih,il})
151             cnt = cnt + 1;
152             j = 1;
153         end
154     end
155
156     % Generate the SOC mean for each current pulse
157     [n, mn] = size(SOCdata{ih,il});

```

```

158     for i = 1:n
159         dSOC(i,1) = SOCdata{ih,il}(i,1);
160         dSOC(i,2) = SOCdata{ih,il}(i,end);
161         SOCmean{ih,il}(i) = (dSOC(i,1)+dSOC(i,2))/2;
162     end
163
164 % All togheter
165     for i = 1:n
166
167         for j = 2:nn
168             OCVk = pOCV(1)*SOCdata{ih,il}(i,j) + pOCV(2);
169             OCVkp1 = pOCV(1)*SOCdata{ih,il}(i,j-1) + pOCV(2);
170             R0 = pR0(1)*SOCdata{ih,il}(i,j) + pR0(2);
171             tau = pTAU{il}(1)*SOCdata{ih,il}(i,j) + pTAU{il}(2);
172             Y{ih,il}(j-1,i) = Vdata{ih,il}(i,j) - Vdata{ih,il}(i,j-1)
173             - ...
174             OCVk + OCVkp1 - R0*Idata{ih,il}(i,j) + R0*Idata{ih,il}
175             (i,j-1) + ...
176             tau*Ts*(Vdata{ih,il}(i,j-1) - OCVkp1 - R0*Idata{ih,il}
177             (i,j-1));
178             X{ih,il}(j-1,i) = tau*Ts*Idata{ih,il}(i,j);
179         end
180     end
181
182 end
183
184 subplot(2,2,4)
185 cnt = 1;
186 for ih = 1:h
187     for il = 1:l
188         plot(SOCmean{ih,il}(1,:), R{ih,il}(1,:), '-*')
189         hold on
190         cnt = cnt + 1;
191     end
192 end
193 xlabel('SoCn'), ylabel('R [\Omega]'), grid on, title('R')
194
195 SOC_R_LUT = SOCmean;
196 R_LUT = R;
197
198 save('ID1_appParameters_linearVarTau_new.mat', ...
199     'pR0', 'pTAU', 'pOCV', 'SOC_R_LUT', 'R_LUT', 'Capacity', 'Ts');

```

Appendix E

ERMES scheme

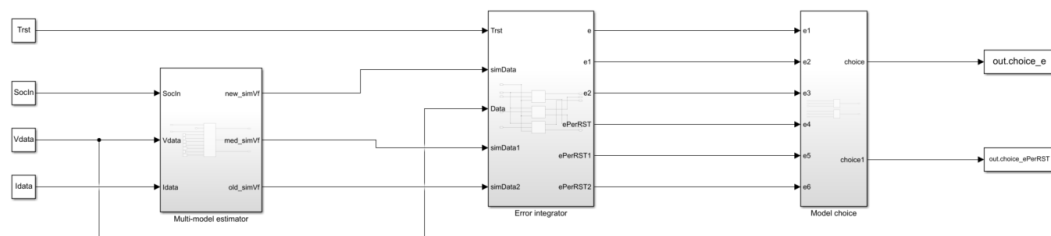


Figure E.1: Simulink ERMES block scheme

Bibliography

- [1] Scrosati and Bruno. «History of lithium batteries». In: *Journal of solid state electrochemistry* 15.7 (2011), pp. 1623–1630 (cit. on p. 3).
- [2] Nitta, Naoki, and et al. «Li-ion battery materials: present and future.» In: *Materials today* 18.5 (2015), pp. 252–264 (cit. on pp. 5, 6).
- [3] Speirs, Jamie, and et al. «The future of lithium availability for electric vehicle batteries.» In: *Renewable and Sustainable Energy Reviews* 35 (2014), pp. 183–193 (cit. on p. 5).
- [4] Madhavi, S., and et al. «Effect of Cr dopant on the cathodic behavior of LiCoO₂.» In: *Electrochimica acta* 48.3 (2002), pp. 219–216 (cit. on p. 5).
- [5] Rougier, A., P. Gravereau, and C. Delmas. «Optimization of the composition of the Li_{1-z}Ni_{1+z}O₂ electrode materials: structural, magnetic, and electrochemical studies.» In: *Journal of The Electrochemical Society* 143.4 (1996), p. 1168 (cit. on p. 5).
- [6] Yan-Bing He, Ming Liu, Zhen-Dong Huang, Biao Zhang, Yang Yu, Baohua Li, Feiyu Kang, and Jang-Kyo Kim. «Effect of solid electrolyte interface (SEI) film on cyclic performance of Li₄Ti₅O₁₂ anodes for Li ion batteries». In: *Journal of Power Sources* 239 (2013), pp. 269–276 (cit. on p. 6).
- [7] Ma, Shuai, and et al. «Temperature effect and thermal impact in lithium-ion batteries: A review.» In: *Progress in Natural Science: Materials International* 28.6 (2018), pp. 653–666 (cit. on pp. 6–8).
- [8] Ji, Yan, Yancheng Zhang, and Chao-Yang Wang. «Li-ion cell operation at low temperatures.» In: *Journal of The Electrochemical Society* 160.4 (2013), A636 (cit. on p. 6).
- [9] Liu, Huaqiang, and et al. «Thermal issues about Li-ion batteries and recent progress in battery thermal management systems: A review.» In: *Energy conversion and management* 150 (2017), pp. 304–330 (cit. on p. 7).
- [10] Bodenes, Lucille, and et al. «Lithium secondary batteries working at very high temperature: Capacity fade and understanding of aging mechanisms.» In: *Journal of Power Sources* 236 (2013), pp. 265–275 (cit. on p. 7).

- [11] Finegan, Donal P., and et al. «In-operando high-speed tomography of lithium-ion batteries during thermal runaway.» In: *Nature communications* 6.1 (2015), pp. 1–10 (cit. on p. 8).
- [12] Atalay, Selcuk, and et al. «Theory of battery ageing in a lithium-ion battery: Capacity fade, nonlinear ageing and lifetime prediction.» In: *Journal of Power Sources* 478 (2020), p. 229026 (cit. on p. 9).
- [13] Spitthoff, Lena, Paul R. Shearing, and Odne Stokke Burheim. «Temperature, ageing and thermal management of lithium-ion batteries.» In: *Energies* 14.5 (2021), p. 1248 (cit. on p. 9).
- [14] Baghdadi, Issam, and et al. «Lithium battery aging model based on Dakin’s degradation approach.» In: *Journal of Power Sources* 325 (2016), pp. 273–285 (cit. on p. 9).
- [15] Ren, Dongsheng, and et al. «A comparative investigation of aging effects on thermal runaway behavior of lithium-ion batteries.» In: *ETransportation* 2 (2019), p. 100034 (cit. on p. 9).
- [16] Galatro, Daniela, and et al. «Thermal behavior of lithium-ion batteries: Aging, heat generation, thermal management and failure.» In: *Frontiers in Heat and Mass Transfer (FHMT)* 14 (2020) (cit. on pp. 9, 10, 18).
- [17] Naseri, Farshid, and et al. «An enhanced equivalent circuit model with real-time parameter identification for battery state-of-charge estimation.» In: *IEEE Transactions on Industrial Electronics* 69.4 (2021), pp. 3743–3751 (cit. on pp. 12, 13, 56).
- [18] Madani, Seyed Saeed, Erik Schaltz, and Søren Knudsen Kær. «An electrical equivalent circuit model of a lithium titanate oxide battery.» In: *Batteries* 5.1 (2019), p. 31 (cit. on p. 12).
- [19] Huria, Tarun, and et al. «Simplified extended kalman filter observer for soc estimation of commercial power-oriented lfp lithium battery cells.» In: *SAE Technical Paper* (2013), pp. 01–1544 (cit. on p. 12).
- [20] Orcioni, Simone, and et al. «Lithium-ion battery electrothermal model, parameter estimation, and simulation environment.» In: *Energies* 10.3 (2017), p. 375 (cit. on pp. 12, 13).
- [21] Huria, Tarun, and et al. «High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells.» In: *IEEE International Electric Vehicle Conference*. 2012 (cit. on p. 12).
- [22] Gholizadeh, Mehdi, and Alireza Yazdizadeh. «Systematic mixed adaptive observer and EKF approach to estimate SOC and SOH of lithium-ion battery.» In: *IET Electrical Systems in Transportation* 10.2 (2020), pp. 135–143 (cit. on p. 13).

- [23] Huang, Shyh-Chin, and et al. «An online SOC and SOH estimation model for lithium-ion batteries.» In: *Energies* 10.4 (2017), p. 512 (cit. on p. 14).
- [24] Barré, Anthony, and et al. «A review on lithium-ion battery ageing mechanisms and estimations for automotive applications.» In: *Journal of Power Sources* 24 (2013), pp. 680–689 (cit. on pp. 14, 16).
- [25] Balagopal, Bharat, Cong Sheng Huang, and Mo-Yuen Chow. «Effect of calendar ageing on SEI growth and its impact on electrical circuit model parameters in Lithium ion batteries.» In: *IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*. 2018 (cit. on p. 14).
- [26] Motapon, Souleman N., and et al. «A generic cycle life model for lithium-ion batteries based on fatigue theory and equivalent cycle counting.» In: *IEEE Open Journal of the Industrial Electronics Society* 1 (2020), pp. 207–217 (cit. on p. 14).
- [27] Xia, Zhiyong, and Jaber A. Abu Qahouq. «Lithium-ion battery ageing behavior pattern characterization and state-of-health estimation using data-driven method.» In: *IEEE Access* 9 (2021), pp. 98287–98304 (cit. on p. 15).
- [28] Qian, Kun, and et al. «State-of-health (SOH) evaluation on lithium-ion battery by simulating the voltage relaxation curves.» In: *Electrochimica Acta* 303 (2019), pp. 183–191 (cit. on p. 15).
- [29] Tian, Jiaqiang, and et al. «Capacity attenuation mechanism modeling and health assessment of lithium-ion batteries.» In: *Energy* 221 (2021), p. 119682 (cit. on p. 15).
- [30] Osara, Jude A., and Michael D. Bryant. «A thermodynamic model for lithium-ion battery degradation: application of the degradation-entropy generation theorem.» In: *Inventions* 4.2 (2019), p. 23 (cit. on p. 16).
- [31] S. Pregnolato. «Real-Time Battery Conditions Estimations». Master’s Degree Thesis: Mechatronic Engineering. Torino, Italy: Politecnico, 2019 (cit. on pp. 19, 81).
- [32] Ahmed, Ryan, and et al. «Model-based parameter identification of healthy and aged li-ion batteries for electric vehicle applications.» In: *SAE International Journal of Alternative Powertrains* 4.2 (2015), pp. 233–237 (cit. on p. 22).
- [33] Shen, Weixiang, Thanh Tu Vo, and Ajay Kapoor. «Charging algorithms of lithium-ion batteries: An overview.» In: *7th IEEE conference on industrial electronics and applications (ICIEA)*. 2012 (cit. on pp. 25, 27).
- [34] Ungarala, Sridhar, Eric Dolence, and Keyu Li. «Constrained extended Kalman filter for nonlinear state estimation.» In: *IFAC Proceedings Volumes* 40.5 (2007), pp. 63–68 (cit. on p. 59).

- [35] Gupta, Nachi, and Raphael Hauser. «Kalman filtering with equality and inequality state constraints.» In: *arXiv preprint* 0709.2791 (2007) (cit. on p. 59).
- [36] M. Colabella. «Identification and Predictive Analysis of Storage System». Master's Degree Thesis: Mechatronic Engineering. Torino, Italy: Politecnico, 2019 (cit. on p. 81).
- [37] D. Faverato. «Virtual Sensing for the Estimation of the State of Health of Batteries». Master's Degree Thesis: Mechatronic Engineering. Torino, Italy: Politecnico, 2020 (cit. on p. 81).