Dino Hrvanovic BSc

Aging and State filtering for Lithium-ion cells

MASTERARBEIT

zur Erlangung des akademischen Grades
Diplom-Ingenieur

Masterstudium Elektrotechnik: Automatisierungstechnik und Mechatronik

eingereicht an der

Technischen Universität Graz

Betreuer:
Univ.-Prof. Dipl.-Ing. Dr. techn. Daniel Watzenig
Institut für Elektrische Messtechnik und Messsignalverarbeitung
Institut für Regelungs- und Automatisierungstechnik

Dipl.-Ing. Matthias Scharrer
VIRTUAL VEHICLE Research Center (ViF)

Graz, Juni 2018

Abstract

Due to an environmental, economic and political pressure, we observe undergoing significant changes in automotive industries technology. Electrical energy storage is becoming increasingly important, not only as a part of automotive industries but also for renewable energy sources, like solar and wind power. One of the key technologies nowadays are Lithium-ion batteries, both for automotive industries and renewable energy storage applications. In order to operate battery systems efficiently and safely, besides the control, we need to be able to correctly estimate the battery state. Concerning the safety and efficiency, two states are of particular importance, those are: State of Charge and State of Health of the cell. Unfortunately, we still do not have the technology to measure these states directly, hence we need to estimate them.

In this thesis different approaches are described to cell modeling and cell state estimation algorithms. Firstly, two different approaches to cell modeling are proposed, a comparative study between RC and ARX model is shown, in order to describe cell response to the cell input as precisely as possible. Next, the behavior of these two models is examined, combined with the Extended Kalman filter algorithms for estimating the SoC. Further, the Extended Kalman filter is enlarged into dual Extended Kalman filter (DEKF), with the goal to be able to estimate the SoH. Both filters are tested for a variety of scenarios, load profiles, temperature varying state and parameter initialization, as well as different cell aging levels.

The simulation results show that the ARX model performs better in terms of computation power and accuracy, the reason for this can be found in the fact, that with the ARX-model approach a closed-loop model is obtained, but it is difficult to interpret physical meaning of the parameters. However, one can assume that both models provide satisfactory results and may be used as a cell model. It may be said that both state estimation algorithms perform well in terms of accuracy of SoC estimation and that they easily outperform the common Coulomb counting methods in terms of robustness, flexibility, and accuracy. It turns out that the tuning of EKF parameters seems to be a very important, but also time consuming-task. Finally, the results for SoH estimation, using the DEKF are presented, this turned out to be challenging task in terms of filter tuning. However, the long-term estimation of the nominal capacity and internal resistance provides a much more stable SoC estimation at the different aging levels, but also that the cell SoH estimate is possible, with the acceptable accuracy compared to test data.

Dedication and Acknowledgements

The publication was written at VIRTUAL VEHICL Research Center in Graz, Austria. The authors would like to acknowledge the financial support of the COMET K2 Competence Centers for Excellent Technologies Programme of the Federal Ministry for Transport, Innovation and Technology (bmvit), the Federal Ministry for Digital, Business and Enterprise (bmdw), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG). They would furthermore like to express their thanks to their supporting industrial and scientific project partners, namely Companies AVL, Porsche, Volkswagen and to the Graz University of Technologies. I would also like to acknowledge the whole battery group at the VIRTUAL VEHICLE especially the coworkers on Alice1/Alice2 project, and special thanks to my mentor Matthias Scharrer and my supervisor Prof. Daniel Watzenig.

Diese Arbeit entstand am VIRTUAL VEHICLE Research Center in Graz,Österreich. Die AutorInnen bedanken sich für die Förderung im Rahmen des COMET K2 - Competence Centers for Excellent Technologies Programms des Österreichischen Bundesministeriums f́ur Verkehr, Innovation und Technologie (bmvit), des Österreichischen Bundesministeriums für Digitalisierung und Wirtschaftsstandort (bmdw), der Österreichischen Forschungsförderungsgesellschaft mbH (FFG), des Landes Steiermark sowie der Steirischen Wirtschaftsförderung (SFG). Ebenfalls danken sie den unterstützenden Industriepartnern AVL, Porsche, Volkswagen und dem wissenschaftlichen Partner TU Graz. Ich danke auch der gesamten Batteriegruppe am VIRTUAL VEHICLE , insbesondere den Mitarbeitern im Alice1 / Alice2-Projekt und meinem Mentor Matthias Scharrer, sowie meinem Betreuer Prof. Daniel Watzenig.

Authors Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe

Graz, date ..... Signature .....

Contents

List of Figures

1 Introduction

1.1 Background and Motivation

The history of lithium-ion (Li-ion) batteries began in 1962, it was initially a battery which could not be recharged after a single discharge. The negative electrode had the lithium and the positive electrode was made from manganese dioxide. It was brought on the market by the company Sanyo. The first rechargeable lithium-based battery was developed in 1985 by Moli Energy, the Negative electrode was made of lithium and the positive one was molybdenum sulfide. It had Safety problems due to the lithium on the negative electrode [14].

The next step in the direction of lithium-ion batteries was achieved by the use of materials on both sides of the electrodes, which enabled a storage and removal of lithium, and had a great voltage potential. It was the first rechargeable lithium based battery and was brought to the Market in 1991 by Sony. The active material of the negative electrode was carbon, that of the positive electrode was lithium cobalt dioxide [21]. After that the main development was done in Countries such as South Korea and Japan and found there usage in many applications.

In 1991, lithium-ion batteries have entered the Market with usage in mobile devices, cell phones, and laptops. Since then we have seen an increase in the research field in order to improve the lifetime as well as the performance of those batteries [14]. After that, the lithium-ion batteries have entered the market of electric and hybrid vehicles, due to their light weight and high Lithium density they were the best candidate for this field of application. In order to minimize the effects of the climate change, some organizations estimate that the Electric Vehicles will represent 60% of the total passenger's cars market by 2050 [4]. As the demand for Electric and Hybrid Vehicles grows, so does the diversity in the implementation of the lithium-ion batteries increase. Nowadays we have a wide range of different chemicals and materials used in lithium-ion batteries, some of them are: Lithium Cobalt Oxide (LiC_oO_2), Lithium Manganese Oxide ($LiMn_2o_2$), Lithium Nickel Manganese Cobalt Oxide ($LiNiMnCoO_2$), Lithium Ion Phosphate ($LiFePO_4$) etc. The main difference between them is specific energy, safety, performance, life span and cost [5]. In order to fulfill the automotive demands, the batteries have to improve, they need to have more capacity, power, and energy density, but also an important role is safety and long life. Automotive industry de-

mands battery life for regular usage to be at least ten years. Opposite to the 'other' battery technologies, Li-ion batteries do not die suddenly, instead, they slowly lose performances over their service life [31]. That means that the end of life (EOL) for Li-ion batteries has to be determined by the manufacturers. We can say in the automotive field that the battery has reached the end of life either by losing $20 - 30\%$ of initial capacity or by an increase of the internal resistance which is related to power applications [4]. In conclusion, the Li-ion battery is not dead at the EOL, but it has reached the pre-defined state that is considered to be the End of Life for application in the automotive field.

In the area of electromobility and hybrid technology, the state estimation of lithium-ion batteries is gaining importance, especially the part with the estimating the state of charge and battery Aging [14]. For this purpose, a large number of measurements were carried out at defined state in the course of an extensive aging experiment. These measurements were carried out together with the industrial and scientific partners at the VIRTUAL VEHICLE Research Center, within the project ALICe (Aging Model for Lithium-Ion Cells). The main goal of this Thesis is to devise a model-based state filter for a non-linear system. With the help of this model, it should be possible to estimate the aging conditions of the cells. Within the ALICe projects, battery aging factors were defined. Over more than two years about 150 custom-built battery cells were tested, under different conditions comprising ambient temperature, different charge-discharge factors in order to obtain aging data. The cells were smaller, but with the same chemistry as the ones being used in the Automotive industries.

1.2 Electrochemical cells

We adapted Figure 1.1 from [33], to show basic structure and the operation function of rechargeable Li-ion cell. Between the two electrodes (anode and cathode), there is an ionic conductive electrolyte and a separator. Separator is a permeable membrane placed between two electrodes. During discharge, the migration of Li-ions from anode to cathode may be observed. During charge we apply external electrical power and thus force li-ions to migrate in reverse direction [14].

Depending on the application, one cell or several cells (= battery) are used, which can be connected in series to build a module. Within module according to the required capacity, several in series connected cells can then be connected in parallel. Several modules may be used, to build a battery system [24]. In order to operate in a safe range we need to control these battery systems. For this we use a Battery Management System (BMS), which monitors the operating range of a battery system. Within the BMS we have sensors for determining the cell voltages, temperature, current and allows the connection and disconnection of the battery system [14].

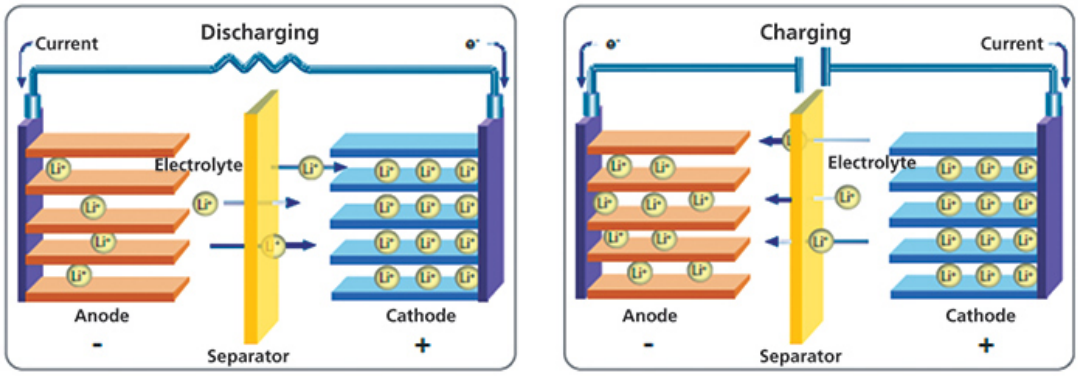


Figure 1.1: Depicted is the operation of a Li-ion battery during discharging and charging. Images: Thermo Fisher Scientific Inc.[33]

We can say that the, advantages of li-ion batteries and there derived systems, are high specific energy, high specific power, high charging and discharging efficiency and low self-discharge[14].

1.2.1 Definitions (Capacity, electrical energy, power and efficiency)

Here we provide a brief introduction to essential parameters, that are important for cell specifications. These are used to characterize a battery cell or a whole battery system.

Capacity is the amount of electrical charge delivered by a power source under specific discharge conditions. It depends on the discharge current, discharge voltage, temperature as well as on the type and amount of active materials in the cell. The unit is Ah [16].

Energy of a cell is calculated according to the product of a capacity and average discharge voltage. The unit is Wh. The specific energy refers to the mass of the battery and has the unit Wh/kg [16].

Power is the product of current and voltage. It has the unit W [16].

Efficiency of li-ion cell is very high, mostly around 95%. It is defined as released energy during discharge, divided by the stored energy during charge [16].

1.2.2 Safety of li-ion batteries

Based on a automotive battery system, we can consider: chemical, electrical, mechanical and functional safety[14]. Chemical safety is determined by the design of battery cell, for example, selection of the corresponding active materials and the structure itself. Electrical safety is achieved by insulating the cables of a battery system and

corresponding housing and sub-components. The mechanical safety may is accomplished by appropriate design, for example special crash boxes. Functional safety is achieved by cell monitoring, battery control units, actuators and corresponding communication interfaces [14].

1.2.3 Lithium-ion battery aging

Over time, the characteristics of a cell/battery system may change. The cells are made of different materials that are in contact with each other and thus can react with each other. At high temperatures, these reactions are accelerated. Subsequently the capacity of the cell may decrease, and the internal resistance may increase, which results in power decrease with time [14].

Degradation mechanisms are different for positive and negative electrode [31]. Negative electrodes are mostly composed of graphite, carbon, titan, or silicone. During the manufacturing a layer is formed on the active material. This is referred to as 'Solid Electrolyte interphase' (SEI) [31]. Its purpose is to protect electrolyte from reduction and the negative electrode form corrosion. Possible degradation mechanisms on a positive electrode are: electrolyte degradation, wear of active mass, electrolyte oxidation and formation of the SEI interaction [4]. Further aging mechanisms are described in [32] in detail.

1.3 Problem Statement

This thesis investigates the approaches of Lithium-ion cell modeling and different monitoring techniques of the cell states that can not be directly measured. It is a state estimation problem. These algorithms need to accurately monitor the cell state of charge and state of health using real measured data. Although conventional methods such as Coulomb counting may fulfill the requirements needed for certain applications, they are not sufficient, if used alone, for the automobile applications.

In this work two different model-based methods i.e. RC- and ARX-model are introduced in order to simulate the Lithium-ion cell response to the input. Kalman filter algorithm is then applied to these model to obtain the robust and stable state of charge and state of health estimation for long and short-term scenarios. The developed methods use real data obtained from more than 400 Lithium-ion cells operating under different cycling profiles.

1.3.1 Battery Test Bench [19]

In order to devise a good model we need adequate measurements. The measurement data that we use was acquired by, VIRTUAL VEHICLE, MEET, and AVL;

- VIRTUAL VEHICLE: 138 channels 5-50A + 8 thermal chambers

- MEET: 4 channels 400A + 4 thermal chambers
- AVL: 4 channels 300A + 1 thermal chambers

Cell tests where divided into three main categories: load-cycling tests, driving cycle tests and calendaric storage tests. In total, 18 calendaric 18650 cells were monitored (current and voltage). The load point settings were designed according to the cell specifications. For the capacity measurement and SoC settings, the cells are operated within a window of 15% to 95% of their nominal voltage limits. The cells that are used had the nominal capacities of $0.32Ah$ for the 18650 cells and $24Ah$ for the PHEV2 cells.

1.3.2 Reference Test Procedure RTP

For our model, we initially used the reference test procedure data which provides the fundamental information for the ongoing cell aging experiment. The programming was divided in three main sections:

- Capacity determination: This test gives insight into the loss of capacity by cycling the cell at $1C$ (Cells are cycled within 15% - 95% SoC range)
- Capacity retention test: This yields insight on the capacity evolution under varying C-rate
- Pulse test: At 7 SoC levels, a number of pulse tests are run, they provide information about a cell's resistance, capacity and parameters for fitting cell models.

1.3.3 Load Points programming

Load points are defined by a certain SoC and the delta-SoC (dSoC), which describes the SoC operation window -see Figure 1.2. The cell is discharged with a defined C-rate from the Design-of-Experiment (DOE) with the Peak Discharge current (PDC) in a certain pulse rest sequence (defined by the frequency) until the minimum SoC has been reached according to the dSoC. Then the cell is charged with a constant C-rate, defined by the constant charge current CC.

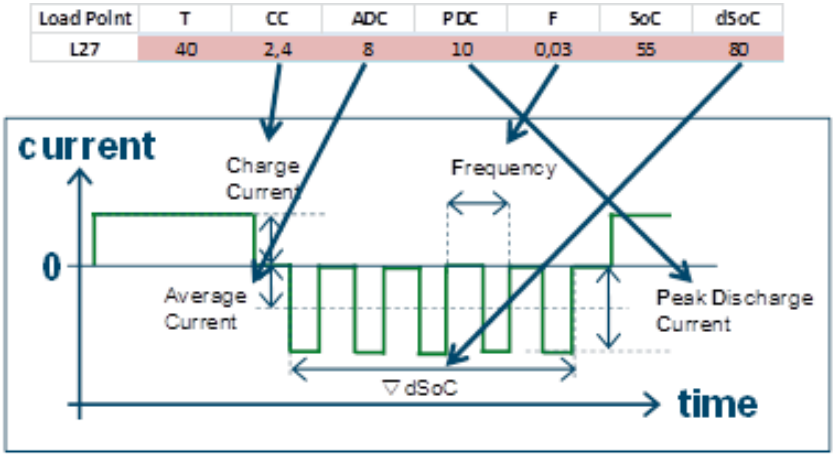


Figure 1.2: Graphical demonstration of DOE parameters in the load point sequence

1.4 Structure of the thesis

The remainder of this work is organized as follows: In Chapter 2 we provide an overview of Lithium-cell models. This is split into two approaches of RC- and ARX-modeling. Further, we present obtained result using both algorithms for 5 RC- and ARX parameters, under different aging stages, SoC and temperature. In Chapter 3 we present the SoC-estimation on the basis of the previously defined models. We describe the EKF-algorithm, as well as the implementation of the same in combination with RC- and ARX-model. In Chapter 4 we combine two EKF-algorithms in order to obtain a DEKF, where one EKF estimate the cell State and the other the cell parameters. Further, we present the longtime SoC estimation combined with the cells capacity and internal resistance tracking, these are related to the SoH estimation.

2 Li-ion Battery Cell Models

This chapter introduces several approaches to model Li-ion battery cells. These models will later on provide a basis for model-based estimation of important quantities, such as; SoC, SoH, available power, and available energy. One way to estimate these quantities is by modelling cell *input output (current/voltage)* dynamics. The system theory classifies models according to their degree of physical interpretability, as *white box*, *grey box* and *black box* model [28].

White box

Also known as physical models, *white box* models are based on the electrochemical and physical partial differential equations (PDEs). This results in slow and numerically sensitive simulation [28]. *White box* models are mostly low-level models with high accuracy. They are designed, to describe the architecture of the materials and illustrate the complex electrochemical phenomena inside the cell, such as thermodynamics, kinetics etc [27]. This is often done with a FEM tool simulation, starting with 0D models, homogenized models, particle models (1D) and homogenized 2D models to 3D models, with the increasing computational effort [27]. The micro structure may be represented in the models either as a reconstruction of a real electrode, obtained by the generic generation of a micro structure with defined parameters or considered as homogenized [28]. Differential equations require, a considerable number of parameters, along with detailed configuration effort to build a physical model. Models like this may offer huge analytical insight, which might be interesting for material scientists. In scope of this work, such insight is not necessary [27].

Grey box

Also known as *abstract models*, these models represent a different but interpretable representation of the battery system, providing a substitute representation of the physical entity. There are different implementations possible, but the most important for this work are the *Equivalent-circuit models* (ECMs). Equivalent-circuit models have been used long before the use of lithium-ion batteries [10]. These models allow us to represent the complex electrochemical processes by a simple electrical circuit [27]. Using this method, the correlation with battery dynamics is preserved, without compromising much of the accuracy [6]. The main disadvantage is that it cannot predict

internal electrochemical state, but it can predict output voltage from current.

In general, equivalent circuit models have a low computational cost and can therefore be applied within real-time environments, as they permit fast and robust simulation. The parametrization of the equivalent circuit models is often performed by fitting to measured impedance spectra, but also a parameter optimization on a current and voltage profile is possible. However, the complexity compared to the physical models is much lower, and look-up tables are still necessary if we want to match experimental data. For example, equivalent circuit models make reasonably good prediction when the cell is operated with currents similar to those used when fitting the parameters (interpolation among the data seen when creating the model), but they tend not to extrapolate well (cell being operated under very different current profiles from the lab-test) [24]. More accurate although more complex models take effects like aging, temperature and nominal capacity fade into consideration. Models like this are commonly used in Kalman filters which are widely used in various forms [28].

Black box

Also known as *Empirical models*, because they are based on the empirical parameters which do not have any physical significance in most cases. Among the black box models we can count artificial neural networks [22]- [29], fuzzy logic [30] and nonlinear autoregressive moving average model with exogenous inputs NARMAX models [11]. These models are mainly used for diagnostic purposes but not for the physical reproduction of the parameters. On the other hand, we have a fully automated parameterization and no necessary understanding of the physio-chemical processes in a battery. The mathematical approaches used to define the transfer function from the input to the output of the *back box* make these models easy to configure, and able to deliver quick responses and predictions [28]. In order to improve the accuracy and the physical insight, we can combine it with a low-level model.

2.1 Simple models

It is possible to use a voltage source, resistors and capacitors in order to predict the behavior of a lithium ion cell. off course the cell itself does not contain internally these electronic components, but its input/output relation can be similarly described by these. Mentioned types of models are largely used in commercial battery packs.

2.1.1 Zero-time-constant Battery Model

In order to realize the objective of this work, an equivalent circuit model is used and thus a *gray box* model chosen. If we want to implement a simple battery model, one way would be a zero-order-constant electrical circuit as shown in Fig. 2.1.

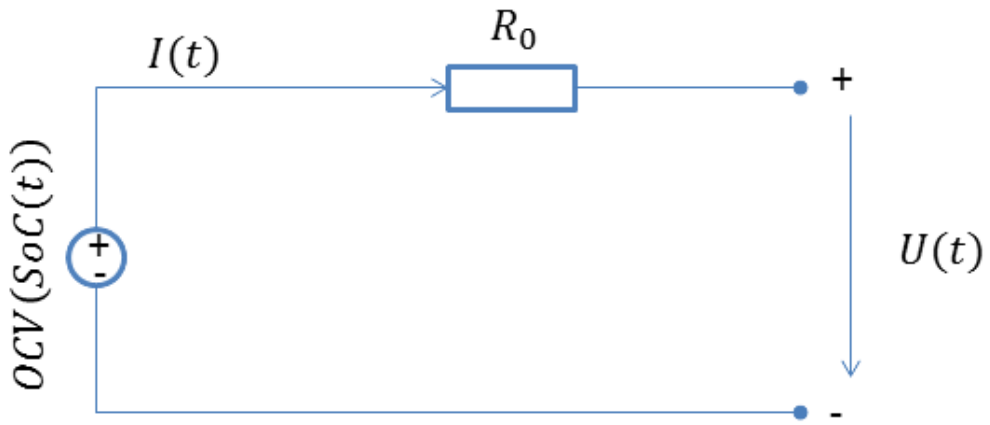


Figure 2.1: Zero-time-constant battery model

Using Kirchhoff's voltage law, we get the following Eq. (2.1).

$$U(t) = OCV(SoC(t)) - R_0 * I(t) \tag{2.1}$$

where $OCV(SoC(t))$ represents the cell's Open Circuit Voltage (OCV). R_0 describes the cell's internal ohmic resistance. Both, OCV and R_0 may depend on SoC, SoH and temperature. $U(t)$ is the battery terminal voltage and $I(t)$ is the output current, positive when discharging and negative when charging [7].

This simple model may be used to describe the static behavior of the cell, but it will not be suitable for describing cell dynamics. OCV is in a direct relationship with the SoC [23]. We should notice that the temperature has a large influence on the parameters, particulary at low temperatures. The main disadvantage of this model is that it does not represent the transient behavior of lithium-ion cells, thus it is not the best solution for accurate assessment of SoC during dynamical operation.

2.1.2 First order RC-model

If we speak of equivalent-circuit models, there are several ways to devise the model and its parameters. The general modeling approach can be divided into two categories: time domain and frequency domain. Frequency domain based models use impedance measurements and Nyquist plots, as shown in Figure 2.2 [15].

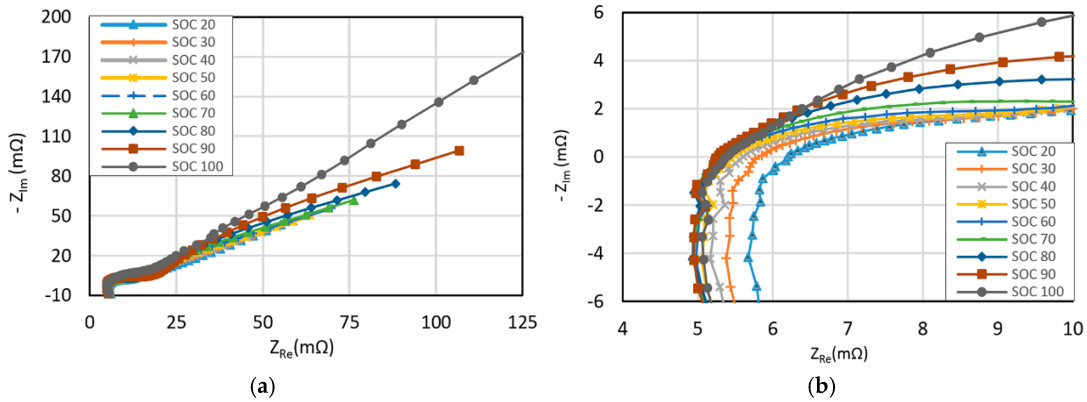


Figure 2.2: Nyquist plot for new battery at various SoC [3]

Another way to do this is the time domain using experimental data current and voltage, as shown in Figure 2.3

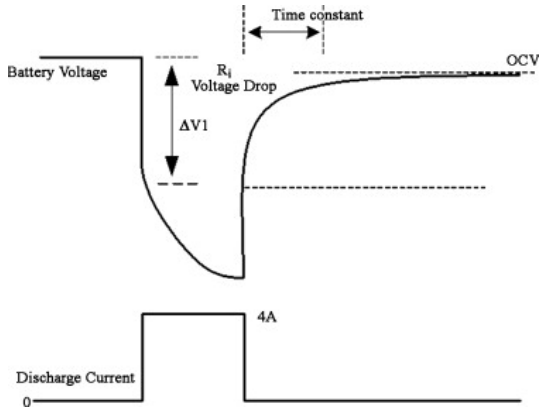


Figure 2.3: Voltage response with respect to current [15]

For the purposes of this work, we are going to adopt the time domain based approach, using the voltage and current analysis. With the help of input/output (current/voltage) information it is possible to obtain the corresponding parameters of the

cell model, as well as the model itself. Figure 2.4 shows a simple cell model that can be used to represent transient behaviour of lithium-ion cells.

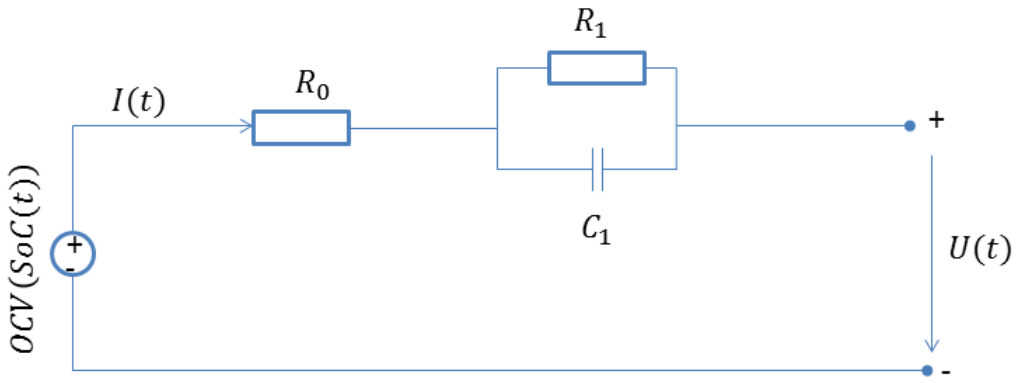


Figure 2.4: First order RC-model

The voltage source OCV is dependent on the SoC. We can describe the relation between these two using Eq. (2.3). The resistor R_0 represents the cell's ohmic Resistance. The resistor-capacitor pair describes the diffusion effects. Adding further resistor-capacitors as shown in Figure 2.5 can improve model fidelity. Although the transient behaviour of the model may be imprecise, we can rely on the fact that the voltage prediction would be reasonable after some time has elapsed. If we consider parameters R_0,R_1,C_1 as a unit, they do describe the physical properties of the diffusion process, but considered individually they do not describe physical property of the cell [24]. The SoC in time domain can be calculated as shown in Eq. (2.2), SoC is the ratio of standard remaining charge to the nominal capacity:

$$SoC(t) = SoC(t_0) - \int_0^t \frac{\eta i(t)}{C} dt. \tag{2.2}$$

Where SoC(t) is state of charge at the time t, $SoC(t_0)$ is the initial value of SoC, $i(t)$ is the current in/out of the cell, and η is the coulomb efficiency. The SoC at discrete time k evolves according to Eq. (2.3)

$$OCV(SoC_{k+1}) = SoC_k - \frac{\eta_k i_k \Delta t}{C_{total}}. \tag{2.3}$$

where i_k is the input/output current at the time k, η is the Coulomb efficiency at time k, Δt is the sample period, and C_{total} is the total capacity of the cell. η and SoC are unitless, that means if we measure current i_k in Amperes and we have Δt in seconds, our C_{total} must be given in Ampere seconds. Again, we can calculate the

output voltage using Kirchhoff's voltage law, as shown in Eq. (2.4)

$$U(t) = OCV(SoC(t)) - U_1(t) - R_0 I(t). \tag{2.4}$$

2.1.3 Second order RC-model

It is known that charging and discharging the battery can be related to a capacity and resistance in terms of behavior. By testing the characteristics of a lithium-ion battery, a distinct polarisation can be observed. It is possible, to simulate the polarisation characteristic with a first order RC-model, however it is impossible to make a clear difference between the concentration polarisation and electrochemical polarisation at the end of charge or discharge [13]. That is one of the reasons we decided to use second order RC-model also known as the dual polarization (DP) model, for the following work, which is shown in Figure 2.5. Using the dual polarization model it is possible to describe the dynamics and the characteristic of the lithium-ion cell.

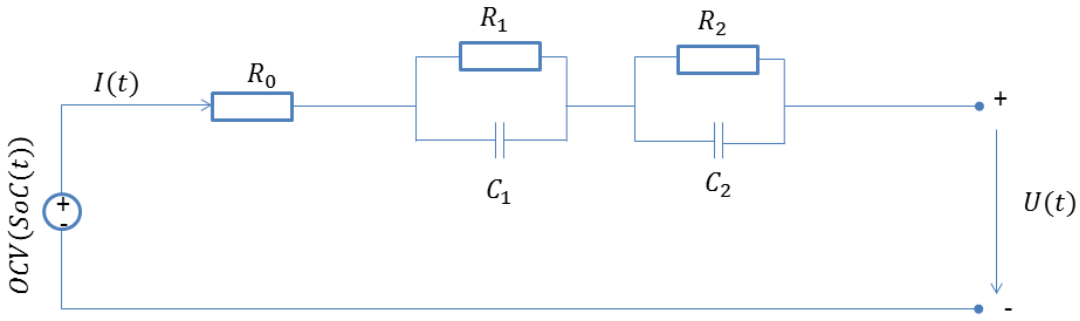


Figure 2.5: Second order RC battery model

$$\begin{aligned} \dot{U}_1(t) &= -\frac{1}{R_1 C_1} U_1(t) + \frac{1}{C_1} I(t) \\ \dot{U}_2(t) &= -\frac{1}{R_2 C_2} U_2(t) + \frac{1}{C_2} I(t) \\ U(t) &= OCV(SoC(t)) - U_1(t) - U_2(t) - R_0 I(t). \end{aligned} \tag{2.5}$$

From Eq.(2.2) and Eq.(2.5), we can write the continuous-time *state-space* equation in time domain.

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\boldsymbol{x}(t)) + \mathbf{D}\boldsymbol{u}(t). \end{aligned} \tag{2.6}$$

where $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the input, $\mathbf{y}(t) \in \mathbb{R}^p$ is the output, and where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector.

$$x(t) = \begin{bmatrix} U_1(t) \\ U_2(t) \\ SoC(t) \end{bmatrix}; \mathbf{A} = \begin{bmatrix} -\frac{1}{R_1 C_1} & 0 & 0 \\ 0 & -\frac{1}{R_2 C_2} & 0 \\ 0 & 0 & 0 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} \frac{1}{C_1} \\ \frac{1}{C_2} \\ -\frac{\eta}{C_{nom} * 3600} \end{bmatrix};$$
$$\mathbf{C} = \begin{bmatrix} -1 & -1 & \frac{\delta OCV}{\delta SoC} \end{bmatrix}; \quad \mathbf{D} = [-R_0] \tag{2.7}$$
$$u(t) = [I(t)]; \quad y(t) = [U(t)].$$

Look at the form of the Eq. (2.6) it can be conclude that it looks similar to a linear state space system, but the dependence of the OCV on SoC makes the equations non-linear, as shown in Fig. 2.6

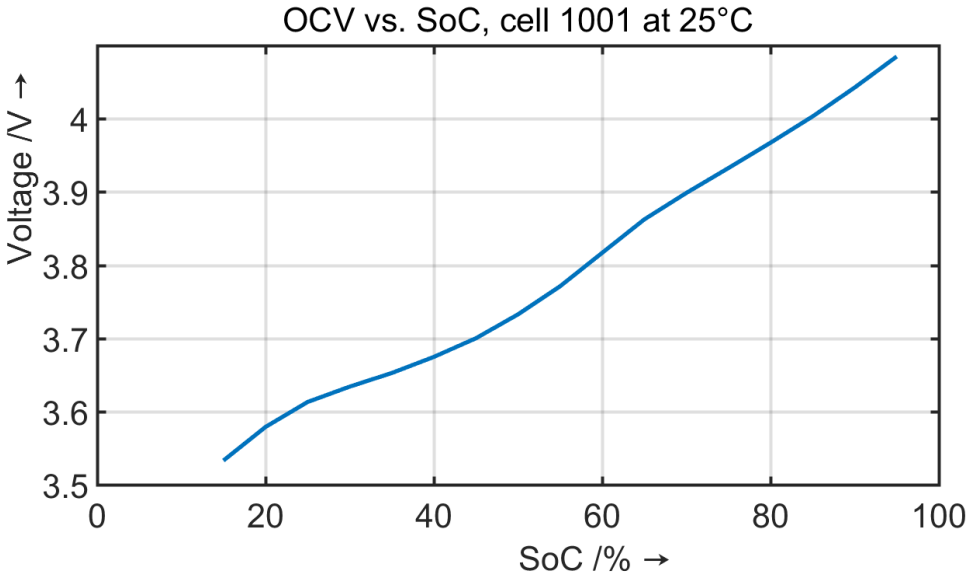


Figure 2.6: OCVe vs. SoC

The continuous-time state-space model is transferred into the discrete-time state-space model. Note, they have a similar form as the continuous-time state-space models. This makes them suitable for numerical evaluation i.e. parameter identification.

$$\begin{aligned} x[k+1] &= \mathbf{A}\boldsymbol{x}[k] + \mathbf{B}\boldsymbol{u}[k] \\ \boldsymbol{y}[k] &= \mathbf{C}\boldsymbol{x}[k] + \mathbf{D}\boldsymbol{u}[k]. \end{aligned} \tag{2.8}$$

Again $\boldsymbol{u}[k] \in \mathbb{R}^m$ is the input, $\boldsymbol{y}[k] \in \mathbb{R}^p$ is the output, and the $x[k] \in \mathbb{R}^n$ is the state vector. Considering continuous or discrete state-space model, the matrices **A**, **B**, **C** and **D** describe the same system. We may convert continuous to discrete state-space models by using:

$$\begin{aligned} \mathbf{A} &= e^{\mathbf{A}_{con.}\Delta t} \\ \mathbf{B} &= \mathbf{A}_{con.}^{-1}(e^{\mathbf{A}_{con.}\Delta t} - I)\mathbf{B}_{con.} \\ \mathbf{C} &= \mathbf{C}_{con.} \\ \mathbf{D} &= \mathbf{D}_{con.}, \end{aligned} \tag{2.9}$$

if the inverse of the Matrix $\mathbf{A}_{con.}^{-1}$ exist. Using this we can rewrite Eq.(2.7) [20]:

$$x[k+1] = \begin{bmatrix} U_1(k+1) \\ U_2(k+1) \\ SoC(k+1) \end{bmatrix}; \mathbf{A} = \begin{bmatrix} e^{\frac{-\Delta t}{R_1 C_1}} & 0 & 0 \\ 0 & e^{\frac{-\Delta t}{R_2 C_2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} R_1(1 - e^{\frac{-\Delta t}{R_1 C_1}}) \\ R_2(1 - e^{\frac{-\Delta t}{R_2 C_2}}) \\ \frac{-\eta \Delta t}{C_{nom} * 3600} \end{bmatrix};$$

$$\mathbf{C} = \begin{bmatrix} -1 & -1 & \frac{\delta OCV}{\delta SoC} \end{bmatrix}; \qquad \mathbf{D} = [-R_0[k]]; \tag{2.10}$$

$$u(t) = [I[k]]; \qquad y(t) = [U[k]]$$

In order to make our model fit as best as possible, the real measurements which have been made on Lithium-ion battery cells are used. The measurements have been made on cells in the Laboratory under different conditions: temperature, charge/discharge current (C-rate), as well as different levels of State of Charge. The SoC is calculated on the basis of the usable capacity of the battery. The parameters depend on the SoC, current and the temperature. With these measurements and the parameter-fitting method we estimated the model structure and unknown parameters. In order to have a battery model with good performance, the model parameters need to be accurately estimated. Figure 2.7 shows the simulated model that represents our lithium-ion cell, the simulation was performed with the help of *Matlab Simulink*.

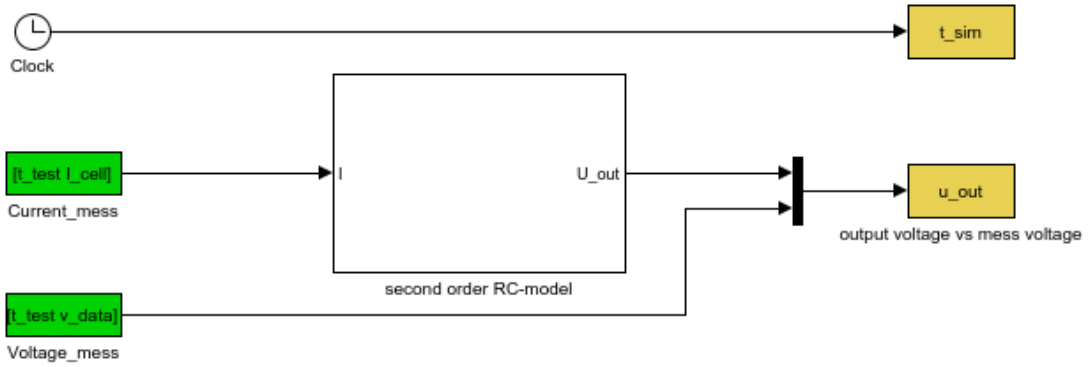


Figure 2.7: *Simulink* model of a cell model

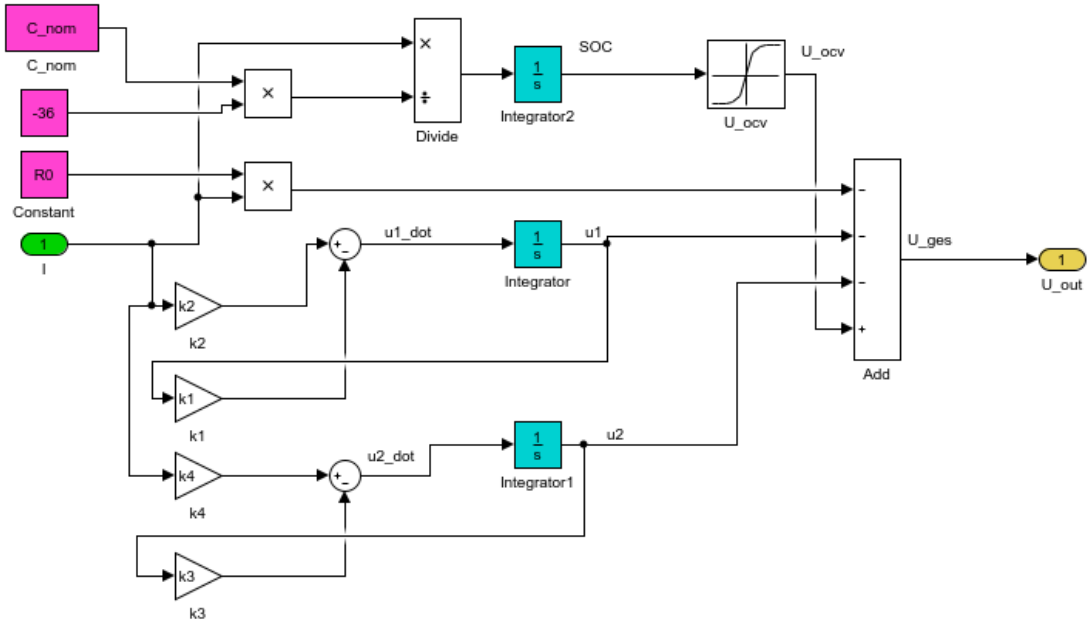


Figure 2.8: Second order RC model in *Simulink*

2.1.3.1 The Least Squares fitting Method

In order to determine the best fit, we used the method of non-linear least squares which is already implemented in MALAB `lsqnonlin`. In this chapter, the short introduction of the mathematics behind this method is conducted. `lsqnonlin` is a non-linear least-square solver, it solves the curve fitting problems of the form:

$$\min_x = ||f(x)||_2^2 = \min_x(f_1(x)^2 + f_2(x)^2 + ... + f_n(x)^2) \tag{2.11}$$

wit optional lower and upper bounds *lb* and *ub* on the components of x [1].

In our case, the parameters are optimized by minimizing the error between the measured data and the simulated results. The accuracy of the parameters depends on the iteration steps and a predefined termination tolerance of x. It does not make sense to make too many iterations, due to computing power and the time needed. Figure 2.9 shows the optimization with 5 iteration steps, and Figure 2.10 optimization with 25 iteration steps. In Table 2.1 we can see the difference within the optimized parameters where:

$$K_1 = \frac{1}{R_1 C_1}; \quad K_2 = \frac{1}{C_1}; K_3 = \frac{1}{R_2 C_2}; \quad K_4 = \frac{1}{C_2}. \tag{2.12}$$

In order to simplify the *Simulink* model we switch to $K_1 ... K_4$ parameters, this way we do not need to do division and multiplication directly in *Simulink*

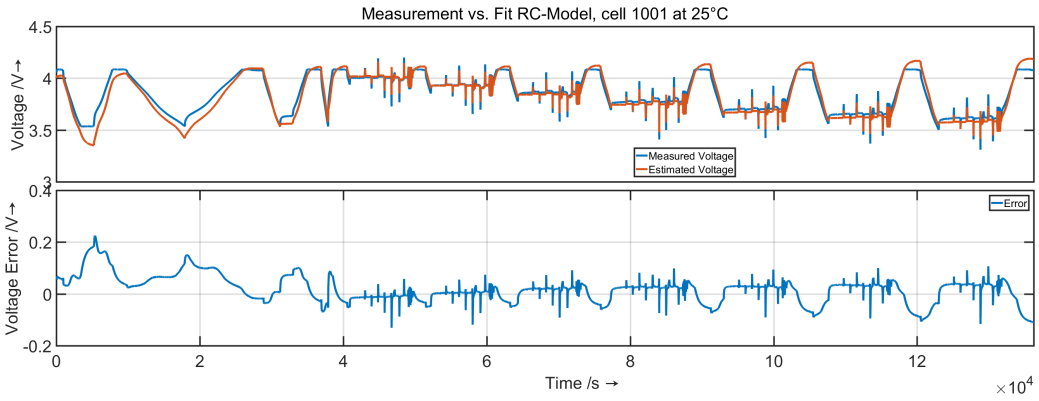


Figure 2.9: Measured data vs Optimized data with 5 iteration steps

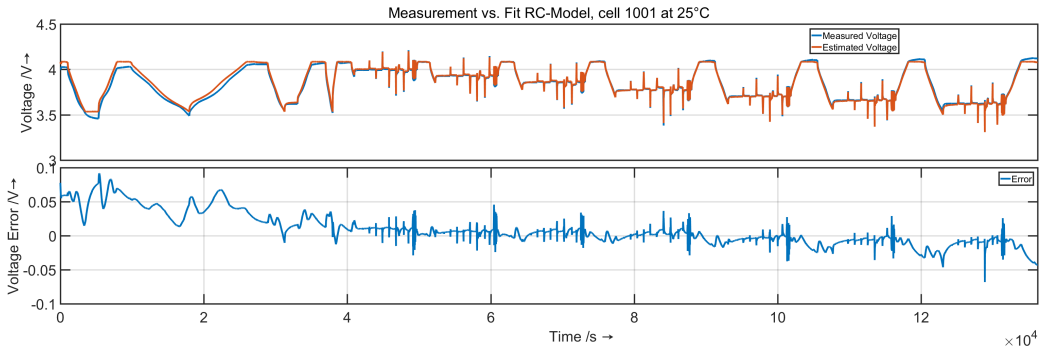


Figure 2.10: Measured data vs Optimized data with 25 iteration steps

Parameters	5-Iteration steps	25-Iteration steps
R_0	0.0878	0.0914
K_1	0.127	0.0633
K_2	0.0083	0.0043
K_3	0.0023	0.0070
K_4	$7.0542 \cdot 10^{-4}$	$6.2372 \cdot 10^{-4}$

Table 2.1: Parameter values for 5-iterations vs 25-iterations

Furthermore, the RTP is divided into 7 steps, each having lower SoC as the one before in order to obtain an even more accurate parameter fit. By dividing it we ensure that we have as much voltage/current diversity as possible (rest, and high dynamic responses). Figure 2.11 shows the fit from one step, SoC at $\approx 65\%$

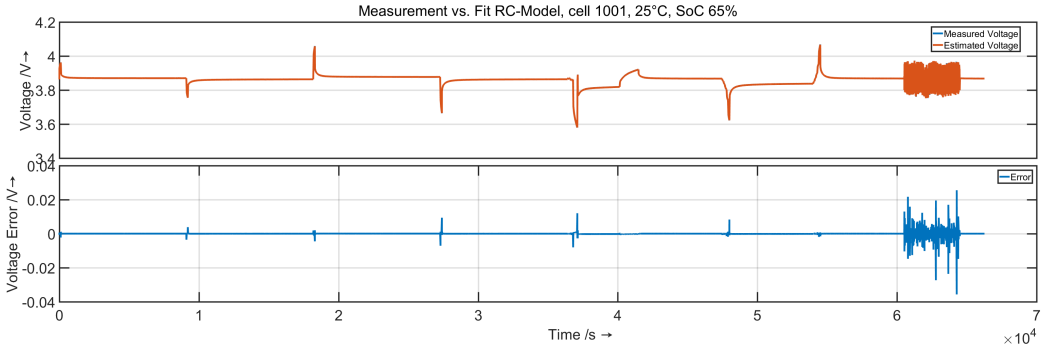


Figure 2.11: Measurement vs Fit 25C at $\approx 65\%$ SoC

These sets of parameters where then used as the initial values, in order to estimate the cell's states. More results are shown in the Chapter 2.2

2.1.4 ARX-Model

As mentioned in 2.1.2. the time domain modeling approach was used. We rely on observed input and output sequences, without knowing characteristics and internal principles of the system. This kind of modeling is widely used in stochastic optimal control and adaptive control applications [34]. The dynamic output of the system -in our case voltage- can be identified as zero-mean sequence if the input signal has zero-mean white noise properties. For this reason, we bring in the concept of time series, taking the current and dynamic voltage as two separated time series sequences [34]. In other words, current and voltage may be modeled as a time-series model. There is a variety of the time series models, that can be selected for cell parameter estimation, such as Box-Jenkins, ARMAX, ARMA ,ARX etc. [18]. For its simplicity, a specific case of ARMAX model, the Auto regressive model with "exogenous" input, so called ARX-mode, was adopted for this work, see Figure 2.12 .

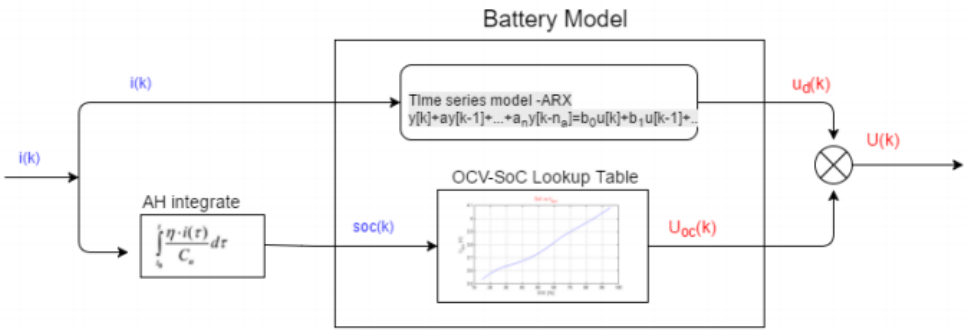


Figure 2.12: ARX battery mode

The most used model structure of the ARX models is the simple linear difference equation which relates the voltage output $y(t)$ to a finite number of past outputs $y(t-k)$ and inputs $u(t-k)$.

$$\begin{aligned} y(t) + a_1 y(t-1) + ... + a_{na} y(t-na) = b_0 u(t) + b_1 u(t-nk) + ... \\ + b_{nb} u(t-nk-nb+1) \end{aligned} \tag{2.13}$$

Equation (2.14) reveals that the additive noise is not the white noise but colored noise that is strongly influenced by the nature of the $A(z^{-1})$ [34].

$$y(k) = \frac{\mathbf{B}(z^{-1})}{\mathbf{A}(z^{-1})} u(k) + \frac{1}{\mathbf{A}(z^{-1})} e(k) \tag{2.14}$$

Further advantages of the ARX-Model:

- Very good approximation for every SISO linear and time invariant system dynamics
- Straightforward parameter identification

Disadvantages:

- Difficult physical interpretation

If we consider our ARX model as a n^{th} order linear time invariant model, we obtain:

$$y[k]+a_1y[k-1]+...+a_{na}y[k-na] = b_0u[k]+b_1u[k-1]+...+b_{nb}u[k-nb]+e[k], \quad (2.15)$$

where $y \in \mathbb{R}^{na}$ is the output sequence in our case voltage, $u \in \mathbb{R}^{nb}$ is the input sequence - in our case current -, and a_i and b_j are the system coefficients to be identified. We have a single-input, single output (SISO) system; ARX models are known for their simplicity to identify such time invariant system dynamics [34]

The structure is entirely defined by integers n_a, n_b where, n_a and n_b represent the model order. n_a is the auto-regression and n_b is the moving average term,. Model coefficients that need to be identified can be writen as $[a_1, ...a_{n_a}, b_0, b_1, ...b_{n_b}]$. We rewrite the above equation as [34]:

$$\begin{aligned} \mathbf{A}(q)y(t) = \ & \mathbf{B}(q)u(t) + \mathbf{e}(t) \\ \text{where :} \quad & \\ & \mathbf{A}(q) = 1 + a_1q^-1 + ... + a_{n_a}q^-n_a \\ & \mathbf{B}(q) = b_0 + b_1q^-1 + ... + b_{n_b}q^-n_b \end{aligned} \quad (2.16)$$

Term $\mathbf{e}(t)$ is the noisy disturbance due to sampling error, current sensor etc. We chose our model order to be the same as in Chapter 2.1.3, i.e. a second order model and we include three recent input samples. This means we have 5 parameters to identify, i.e. $\vartheta = [a_1, a_2, b_0, b_1, b_2]$. To estimate such a model we used a pseudo-linear regression, also known as the *extended least square method* [8]. We recorded the inputs *u(0), u(1),..,u(N)* and the outputs *y(0), y(1),...,y(N)* with $N >> max(n_a, n_b)$ [34] . We introduce the matrix:

$$\Phi(k)^T = [-y(k-1), -y(k-2), ..., -y(k-n_a),\ u(k-1), u(k-2), ..., u(k-n_a)],$$

where:

$$\vartheta = [a_1, ...a_{n_a}, b_0, b_1, ...b_{n_b}]^T$$

Thus we can write the output as:

$$\hat{y}(k, \vartheta) = \Phi(k)^T\vartheta \quad (2.17)$$

We use the iterative method in order to determine the coefficients.

$$\phi^{(i)} = \begin{bmatrix} y(n_a-1) & ... & y(0) & u(n_a-1) & ... & u(n_a-n_b) \\ y(n_a) & ... & y(1) & u(n_a) & ... & u(n_a-n_b+1) \\ . & . & . & . & . & \\ . & . & . & . & . & \\ y(N-1) & ... & y(N-n_a) & u(N-1) & ... & u(N-n_b) \end{bmatrix} \tag{2.18}$$

As described in the paper from (Shifei Yuan 2013) [34] four steps are required to determine the model order and to identify the five parameters.

- Define the vector $y := [y(n_a, ..., y(N))]^T$ and the matrix $\Phi^{(i)}$ with $i = 0$;
- Calculate the unknown $(n_a + n_b)$-dimensional parameter vector $\vartheta^{(0)}$, based on the least square estimation $\hat{\vartheta} = (\Phi^T\Phi)^{-1}\Phi^T y$;
- Calculate the prediction errors $\varepsilon(k, \vartheta^{(0)})$, where $\vartheta = [a_1, ...a_{n_a}, b_0, b_1, ...b_{n_b}]$ and $\varepsilon(k, \vartheta) = y(k) - \hat{y}(k, \vartheta)$;
- Given the prediction errors from an ordinary least-square estimation of the ARX-parameters and the back to step two, until reaching the final termination criterion.

Figure 2.13 shows the first RTP fit with the ARX-model. Again, more results are shown in Chapter 2.2. Compared to the RC-mode we observe, much better accuracy when fitting the voltage. The reason behind is that with the ARX-model we have a closed-loop models, compared to the RC-model where we have the open-loop model. In the open-loop model, the output voltage error can barely be controlled, this may lead to error accumulation, thus provide harmful information. This can be avoided using the closed-loop model [34], which computes a one-step prediction.

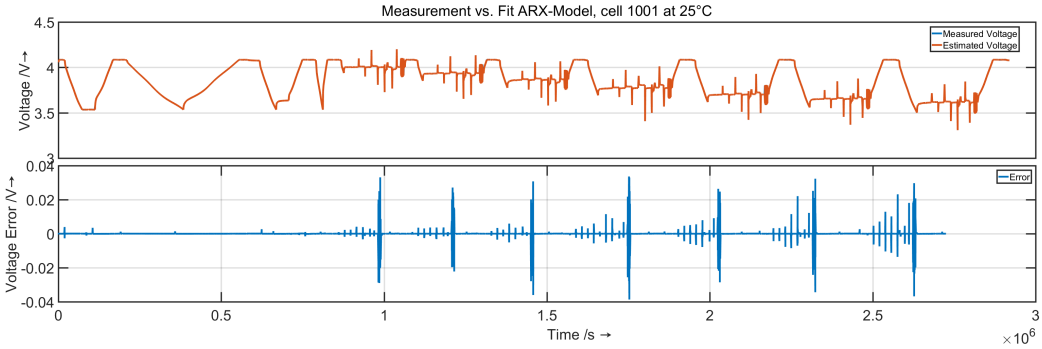
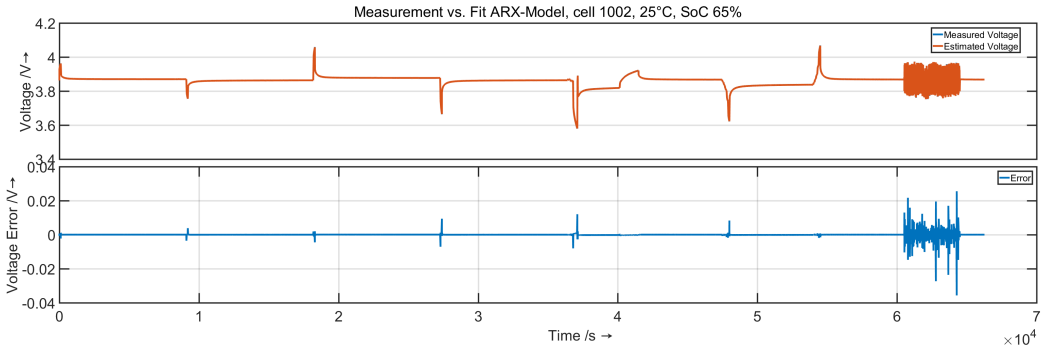
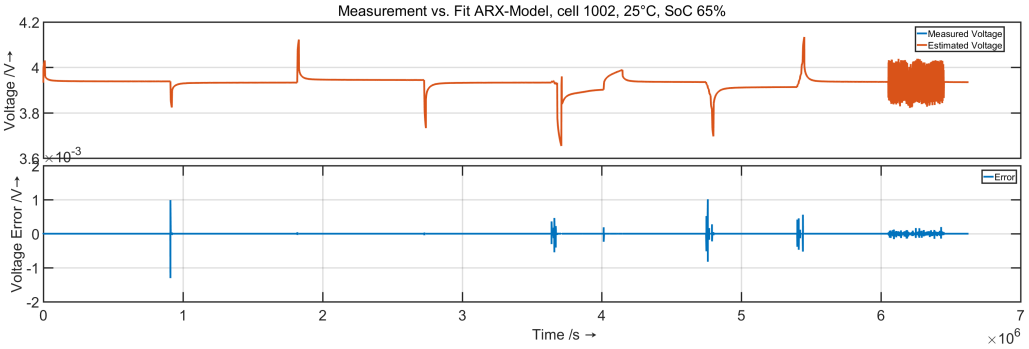


Figure 2.13: Measured data vs Optimized data with ARX model

Figure 2.14 shows parameter fitting for the same cell with different sampling times 0.1s and 0.001s. Here, we can see that the better accuracy is achieved by using the smaller sampling time, but on the other hand, we face an increased demand of computing power. We observe that sampling the data with 0.001s, it took us approximately then times longer in order to obtain the results shown below using *Matlab*. At the cost of computation time, the voltage error using 0.1s is between $\pm$ 0.02V, while using 0.001s between $\pm$ 0.001V.



(a) Measurement vs fit at $35C$ and 65% SoC with sampling time $0.1s$



(b) Measurement vs fit at $35C$ and 65% SoC with sampling time $0.001s$

Figure 2.14: Cell 1002 at same step with different sampling time

We summarize our ARX model as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}i(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}i(k) \end{aligned} \tag{2.19}$$

$$\begin{bmatrix} SoC(k+1) \\ u_d(k+1) \\ u_d(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -a_1 & -a_2 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} SoC(k) \\ u_d(k) \\ u_d(k-1) \end{bmatrix} + \begin{bmatrix} \frac{\eta_i \Delta t}{C_{nom} * 3600} & 0 & 0 \\ 0 & b_1 & b_2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i(k+1) \\ i(k) \\ i(k-1) \end{bmatrix}$$

$$y(k) = OCV(SoC(k)) + u_d(k) + b_0 i(k). \tag{2.20}$$

2.2 Results and Discussions

In this Chapter, the results of the parameter identification are presented, as well as the comparison between the measured voltage and the model output (voltage). As mentioned in Chapter 1.3.1, the measurements have been performed, within 2.5 years. That is why the $''long\text{-}term''$ changes in the parameters have been investigate, in order to draw a conclusion between the parameters of the $''new''$ cells and the $''old''$ cells.

2.2.1 RC-model

After devising the RC model section 2.1.3, the parameters of the model were identified and analysed. We aimed, to see if there is a connection between the cell's aging-state and the changes of the parameters. Five parameters were estimated for each cell. The cells are operated within the window from 15% to 95% of their nominal potential limits. This means that the capacity is always measured over a range of 80% of the dSoC and then divided by 0.8 to obtain the corresponding 100% capacity. At 7 stages of SoC, a series of pulse tests were performed to obtain information about the cells characteristics, such as resistance, capacitance and cell parameters. The initial cell characterization involves RTP (Chapter 1.3.2) testing at five temperatures (5, 15, 25, 35, 45 °C) for all cells. Thereafter, the RTP runs at 25 °C every 21 days and every third RTP is tested at three temperatures (5, 25 and 45 °C) [19]. Figure 2.15 shows input currents and measured voltage (output) of the 18650 mini-cell's ALICe1 (cell No. 1001 at 25 °C). The first parameter estimation was performed for a complete RTP to test the model quality.

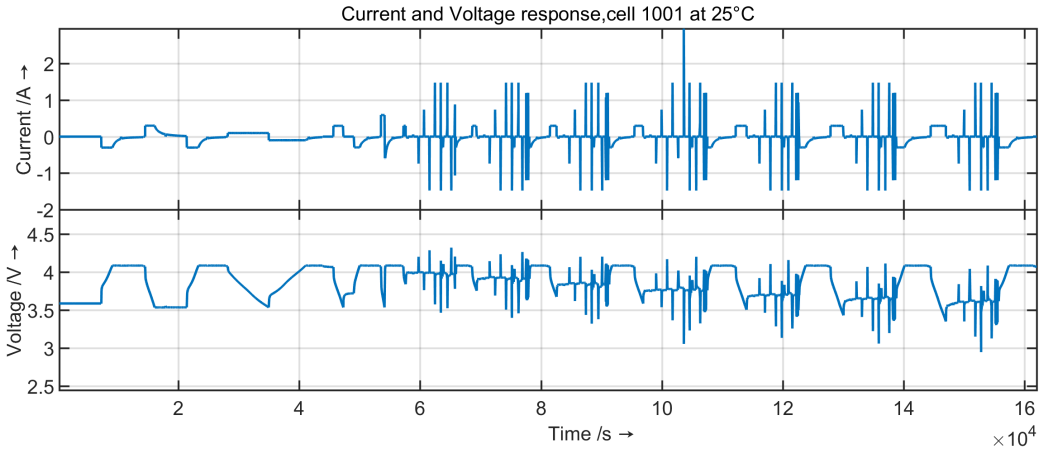


Figure 2.15: 18650 Mini-Cell ALICe1 cell 1001

Figure 2.16 compares the model response estimated voltage to the measured voltage (Data) and the error achieved with this pure feed-forward simulation.

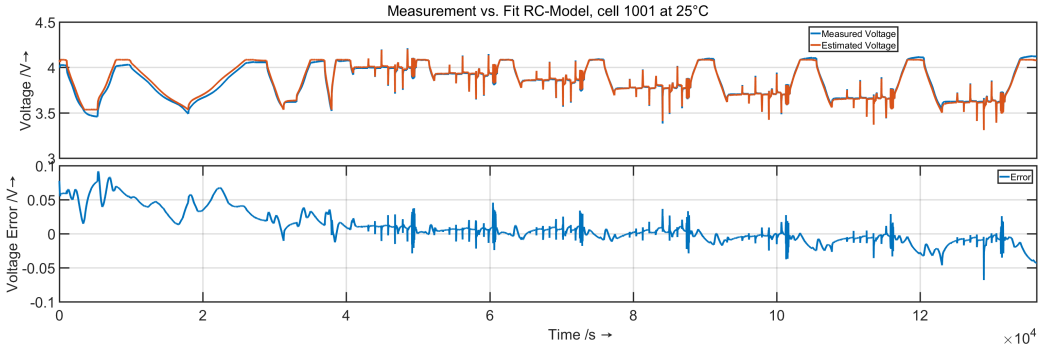
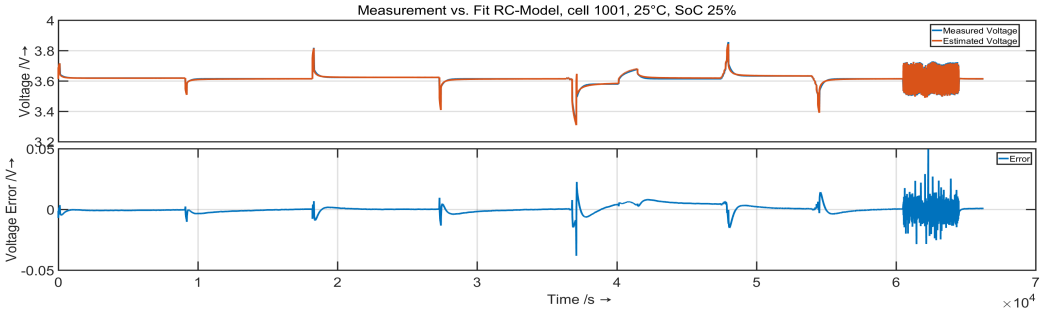
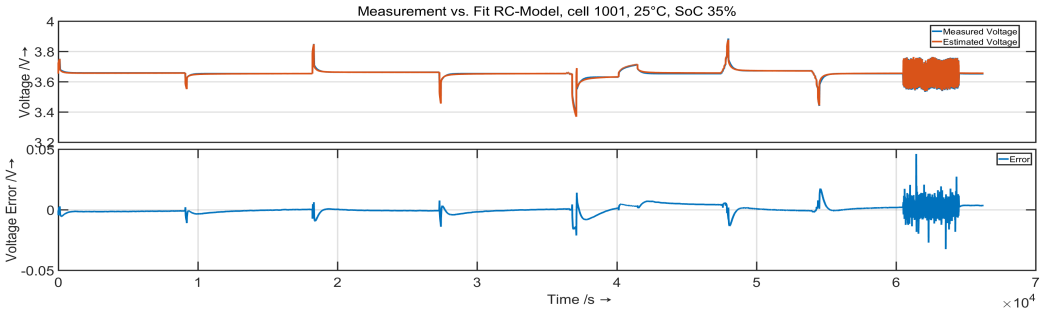


Figure 2.16: Measured vs estimated cell voltage cell 1001 at 25°C

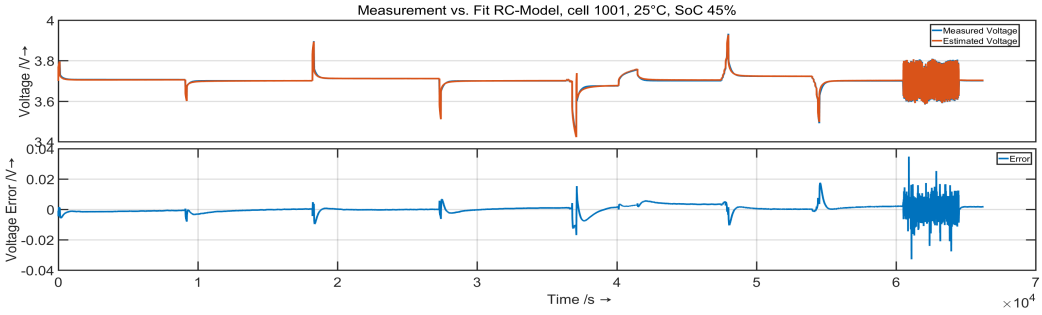
We see that the error stays between ± 0.1V and that the model could reasonably represent the cell in many applications. To get an even more accurate estimate, we divided the RTP into 7 SoC stages (see next Figures) and performed the parameter identification for each step.



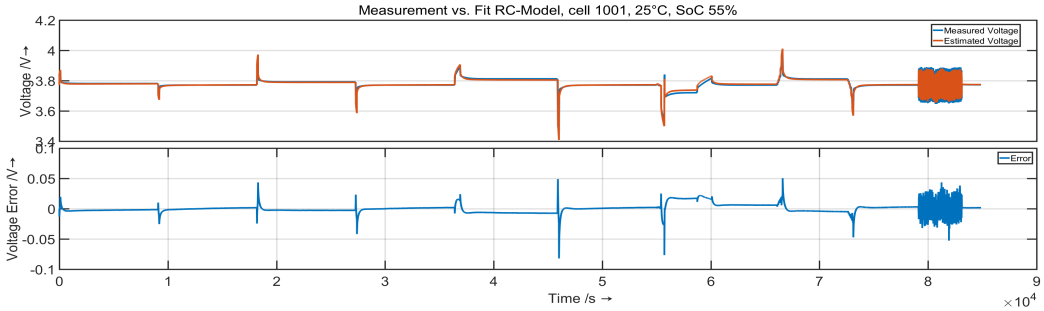
Fit and model error at 25% SoC



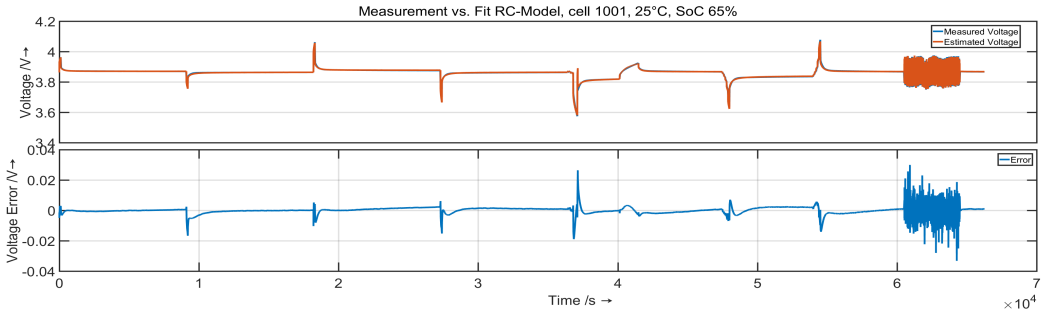
Fit and model error at 35% SoC



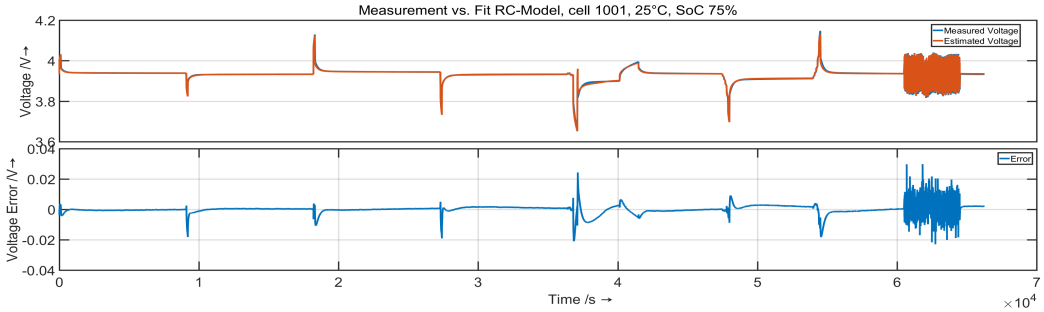
Fit and model error at 45% SoC



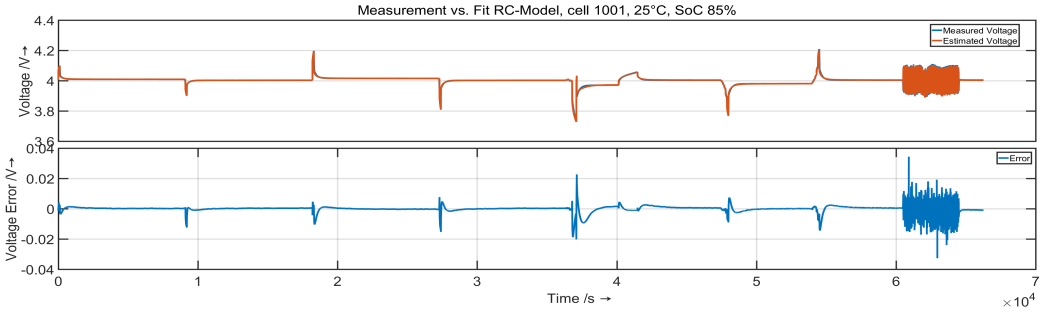
Fit and model error at 55% SoC



Fit and model error at 65% SoC



Fit and model error at 75% SoC



Fit and model error at 85% SoC

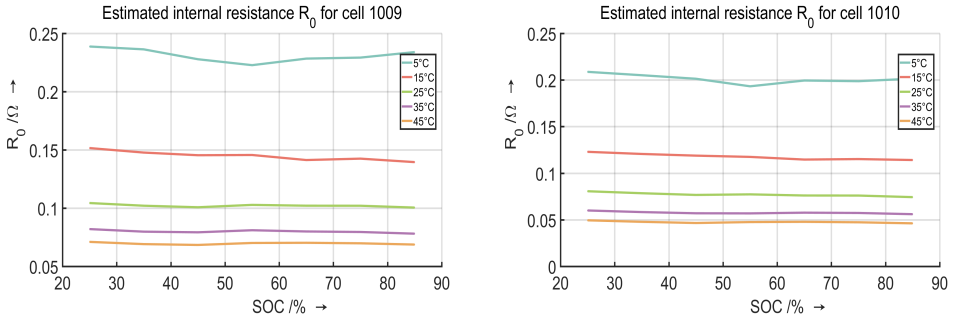
Figure 2.17: Measurement vs estimated cell voltage of cell 2013061248 at 25 °C at SoC = (25%, 35%, 45%, 55%, 65%, 75%, 85%) RC model

The voltage error lies between $-0.03V$ and $0.03V$. It would be possible to make the error even smaller if the sampling time is further reduced. For a reasonable application, the sampling time was set to 0.1s. The parameters depend on the SoC, Temperature and the cell's aging conditions. In order to be able to analyze the parameters as good as possible for different cells, we have divided them into three groups:

- Parameter dependence of temperature and SoC
- Parameter dependence of temperature and aging conditions
- Parameter dependence of SoC and aging conditions

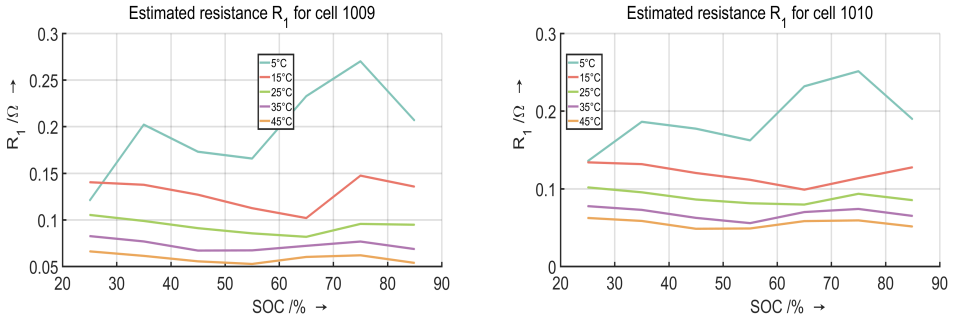
Parameter dependence of temperature and SoC

As assumed, the cells differ from each other due to production variability. This justifies why the cell parameters are different even under almost identical conditions (T, SoC, age) for different cells. Figures 2.18-2.23 show two different cells of the same type (18650) under the same conditions. At the beginning of the test (same age), we see that the internal resistance as well as the capacitance (C_1, C_2) and the resistances (R_1, R_2) have different values. In addition, we can also see that the model parameters are highly dependent on cell temperature and cell SoC. Resistor R_0 represents the internal ohmic resistance. The polarization resistances include R_1 and C_1 to represent the fast kinetic effects and R_2 and C_2 to represent the concentration polarization. As expected, the higher the SoC, the lower the internal resistance. We can also observe an increase in capacity C_1 at higher SoC.



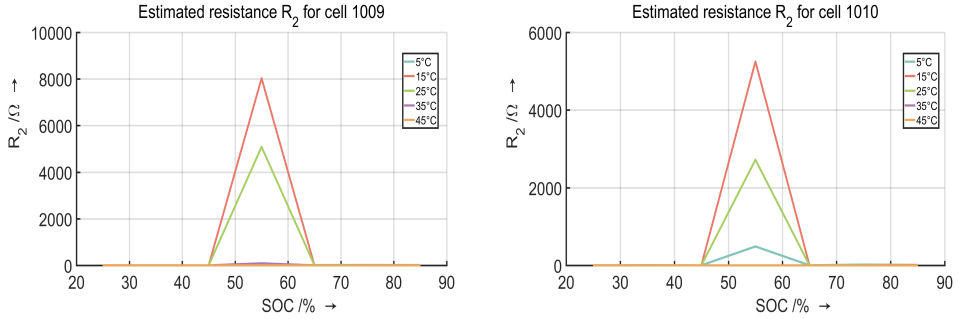
(a) R_0, cell 1009 (b) R_0, cell 1010

Figure 2.18: Estimated internal resistance R_0 for two different cells of the same age



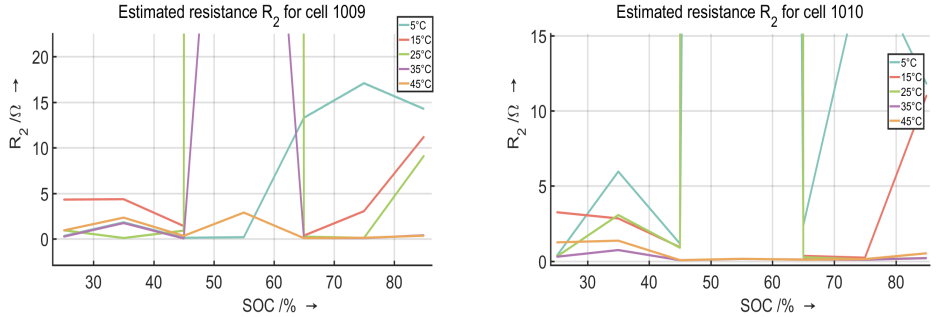
(a) R_1, cell 1009 (b) R_1, cell 1010

Figure 2.19: Estimated resistance R_1 for two different cells of the same age



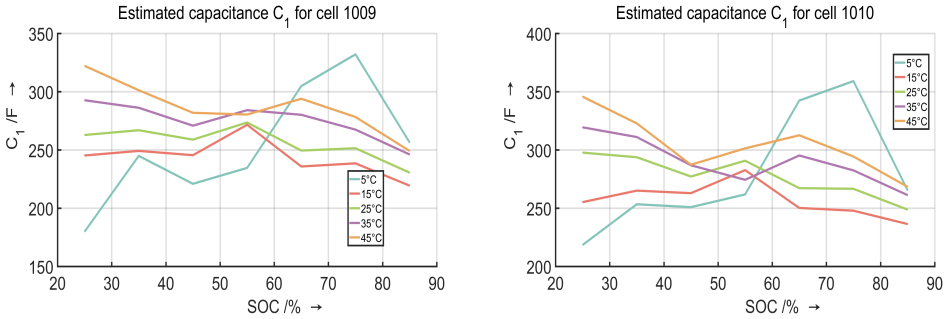
(a) R_2, cell 1009 (b) R_2, cell 1010

Figure 2.20: Estimated resistance R_2 for two different cells of the same age



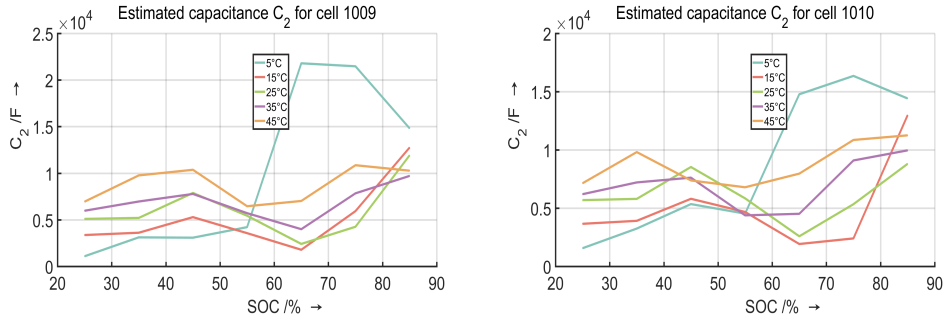
(a) R_2 zoom, cell 1009 (b) R_2 zoom, cell 1010

Figure 2.21: Estimated resistance R_2 for two different cells of the same age



(a) C_1, cell 1009 (b) C_1, cell 1010

Figure 2.22: Estimated capacitance C_1 for two different cells of the same age



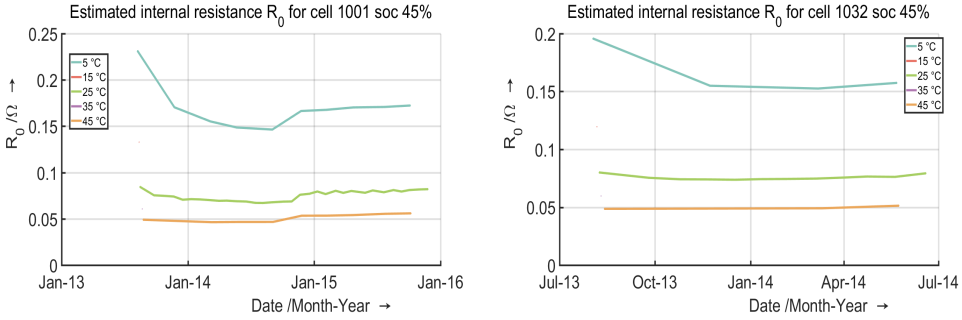
(a) C_2, cell 1009 (b) C_2, cell 1010

Figure 2.23: Estimated capacitance C_2 for two different cells of the same age

From the Figure above, it can be concluded that low temperature, in our case 5 °C, stands out, compared to the higher temperature.

Parameter dependence of temperature and aging conditions

As mentioned earlier, the main objective is to establish a link between the estimated parameters and the aging conditions. Below, the parameter changes as a function of temperature and cell aging are presented. In the figures below we can see different cells at the beginning of the test and after about 1.5 years, in the temperature range from 5 to 45 °C. To see clearly what happens to the internal resistance R_0 over time, we plot the parameters R_0 for the cells 1001 and 1032 at the same temperature 25 °C and SoC = 45%.



(a) R_0,cell 1001 SoC = 45%

(b) R_0, cell 1032 SoC = 45%

Figure 2.24: Estimated resistance R_0 over time and temperature

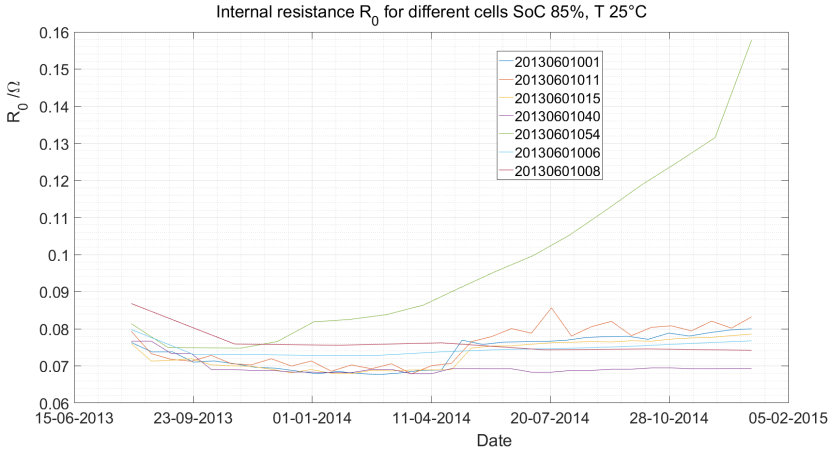
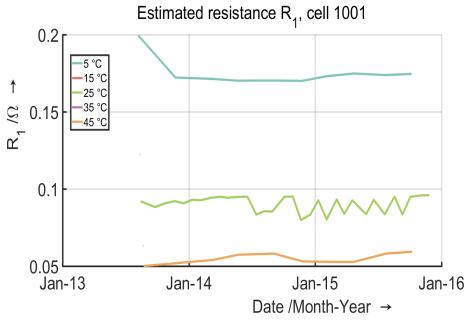


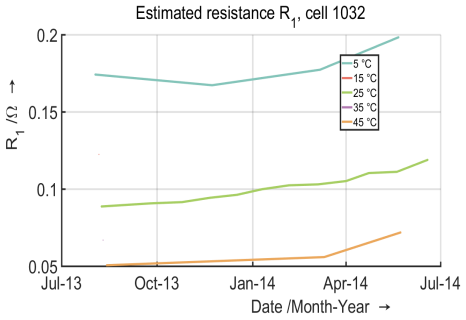
Figure 2.25: Estimated resistance R_0 over time for different cells at constant temperature 25°C and SoC = 85%

Figure 2.25 shows that the cell resistance R_0 slowly decreases over time (here it is

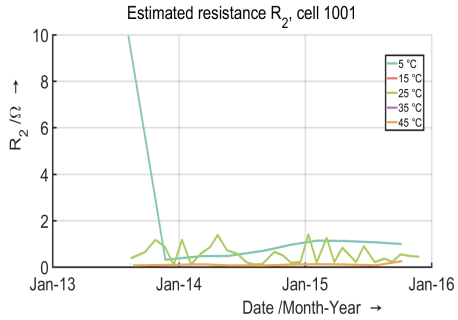
meant that the cell is further cycled), but at some point it starts to increase. Clearly, the resistance also depends on the temperature and SoC.



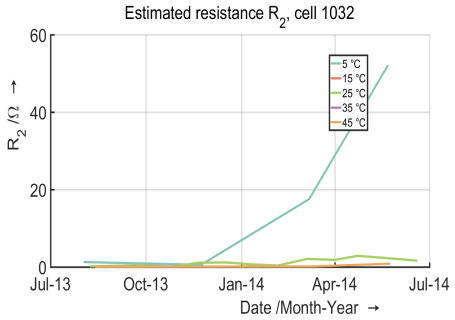
(a) R_1, cell 1001, SoC = 45%



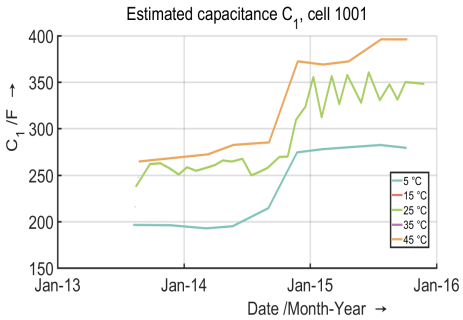
(b) R_1; cell 1032, SoC = 45%



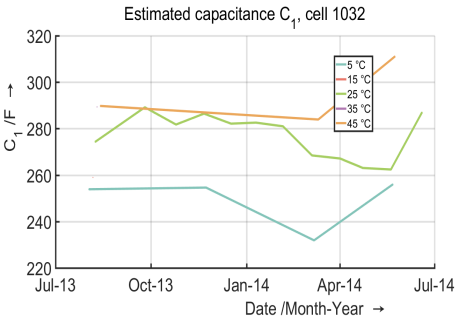
(c) R_2; cell 1001, SoC = 45%



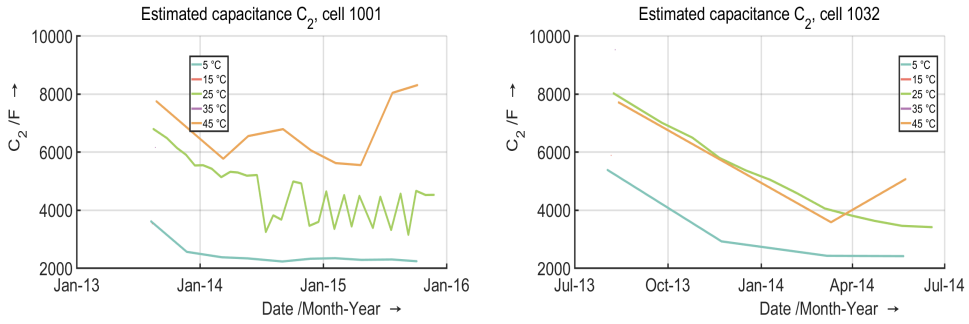
(d) R_2, cell 1032, SoC = 45%



(e) C_1, cell 1001, SoC = 45%



(f) C_1, cell 1032, SoC = 45%



(g) C_2, cell 1001, SoC = 45%

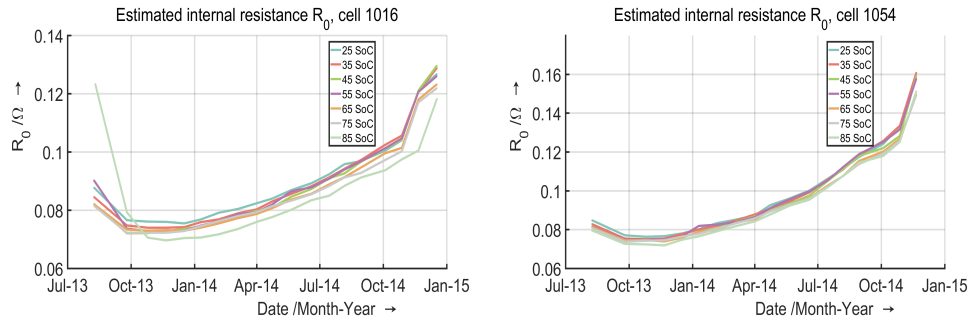
(h) C_2, cell 1032, SoC = 45%

Figure 2.26: Estimated parameters over time for constant SoC = 45%

The figures above show the rest of the model parameters. It is important to note that the measurement data was taken from cell 1032 until about mid-2014, and the cell 1001 data was taken almost until the end of 2015. Cell 1032 was removed from the test bed as planned due to having reached approximately half of its lifetime for detailed chemical analysis.

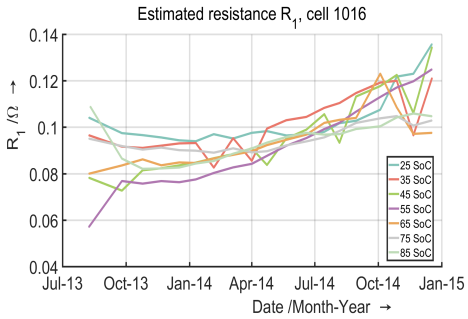
Parameter dependence of SoC and aging conditions

In this section, we show the model parameter dependence of SoC and aging conditions. Approximately 400 cells have undergone 2 years aging process, under different charge and discharge cycles, different temperatures and different SoCs. Figure 2.27 shows two different cells that have undergone the same aging process. As we can see, the internal resistance is increasing with the use of the cell, this can be interpreted as one possible effect of cell aging. The internal resistance is also dependent on the SoC.

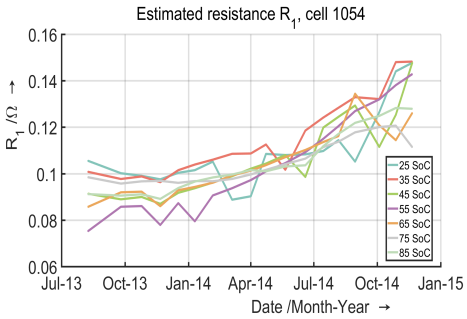


(a) R_0, cell 1016

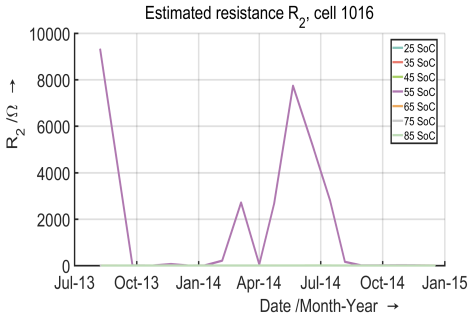
(b) R_0, cell 1054



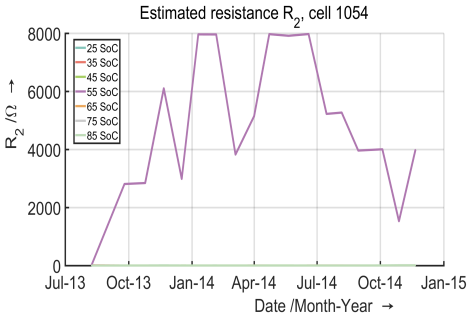
(c) R_1, cell 1016



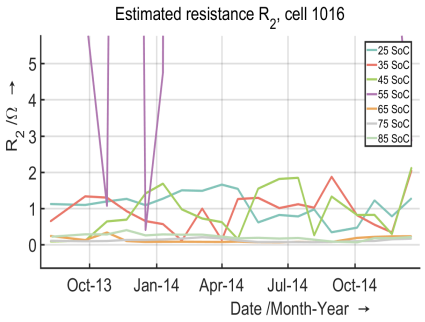
(d) R_1, cell 1054



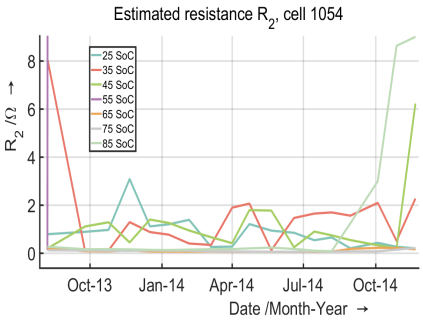
(e) R_2, cell 1016



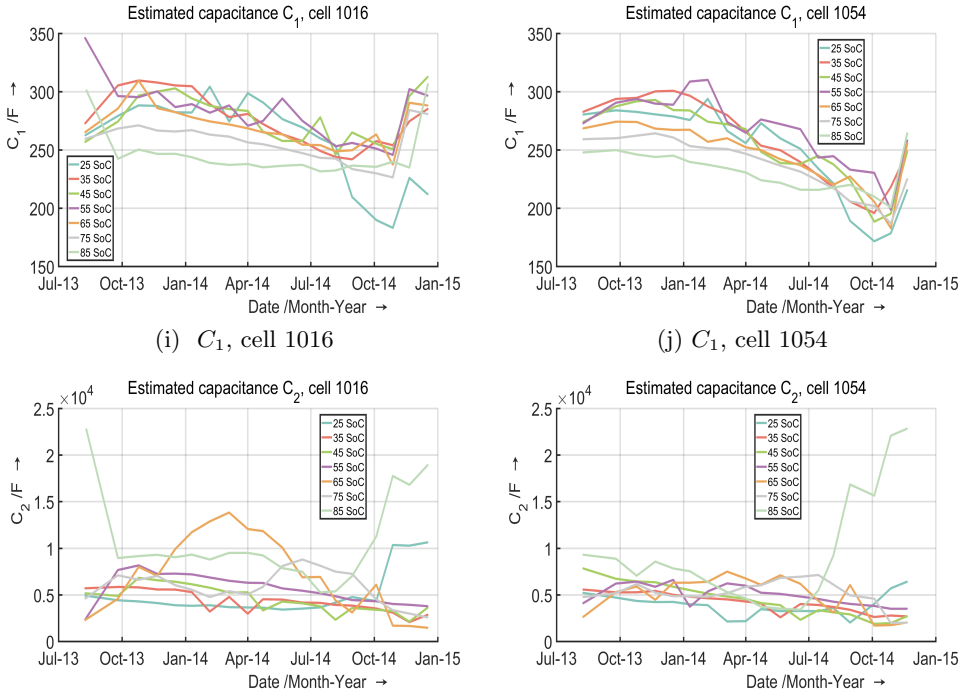
(f) R_2, cell 1054



(g) R_2, zoom, cell 1016



(h) R_2 zoom, cell 1054



(i) C_1, cell 1016 (j) C_1, cell 1054

(k) C_2, cell 1016 (l) C_2, cell 1054

Figure 2.27: Estimated parameters over SoC and time at 25°C

It is important to mention that our measured data is within the span of $5 - 45°C$ and $25\% - 85\% SoC$. Later on, data outside these limits is extrapolated by *Matlab*

2.2.2 ARX-model

In the case of the ARX model, the same procedure was applied as for the RC model. The RTP was divided into 7-Steps in order to obtain the required parameters (test data is same as in the RC model: same current, voltage, and cell type). It could be noticed that the fit residual is much smaller compared to the RC model. With the RC model the voltage error, was between $-0.1V$ and $0.1V$, using the ARX algorithm the voltage difference between measured and estimated voltage lies between $-0.05V$ and $0.05V$, when fitting the whole RTP.

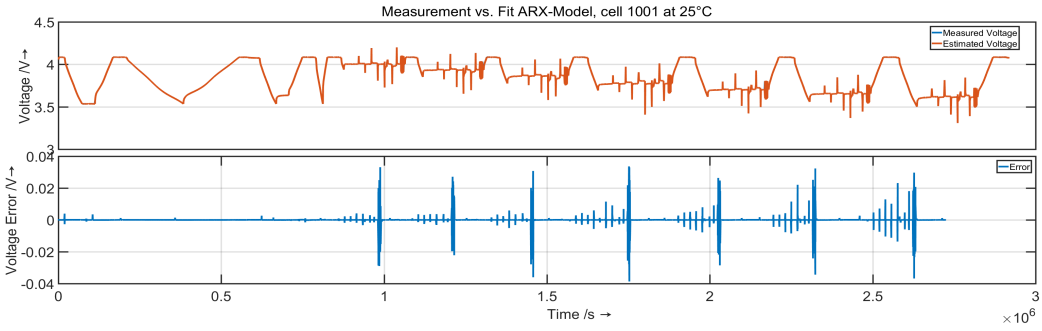
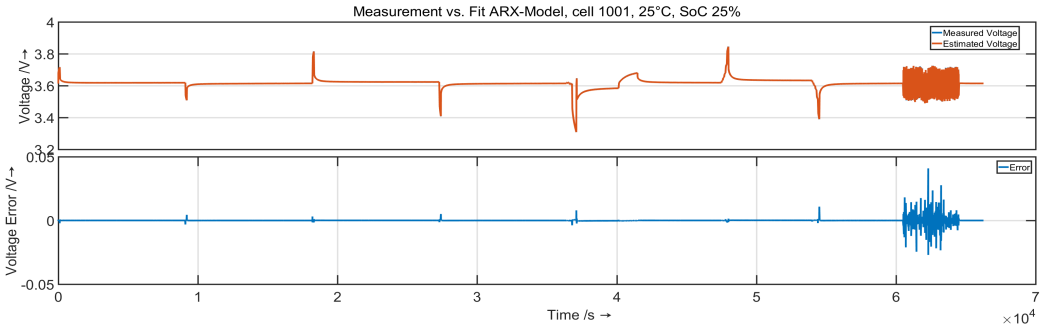
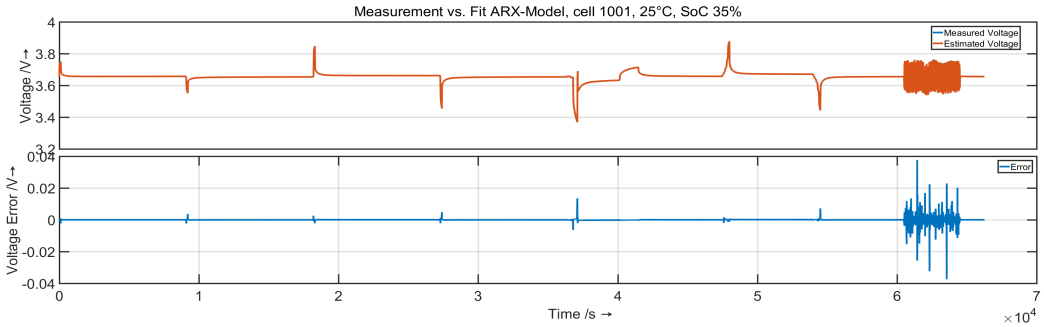


Figure 2.28: Mesured vs estimated cell voltage cell 1001 at 25°C

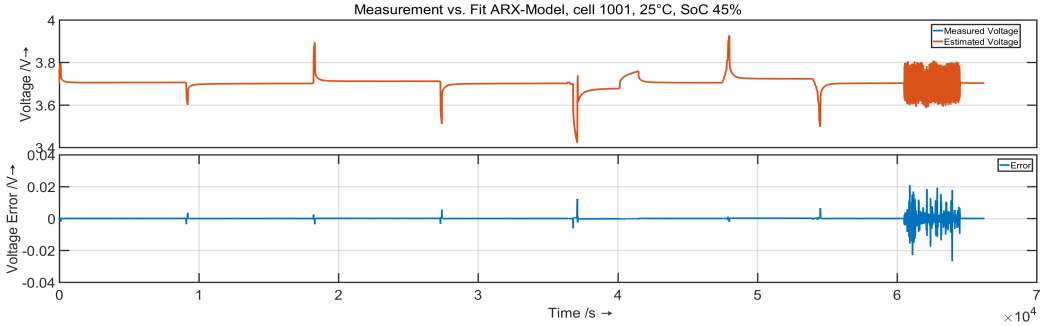
Figure 2.28 shows the fit of the whole RTP as well as the voltage error. And in Figure 2.29 we see the fitted the RTP divided in 7 steps.



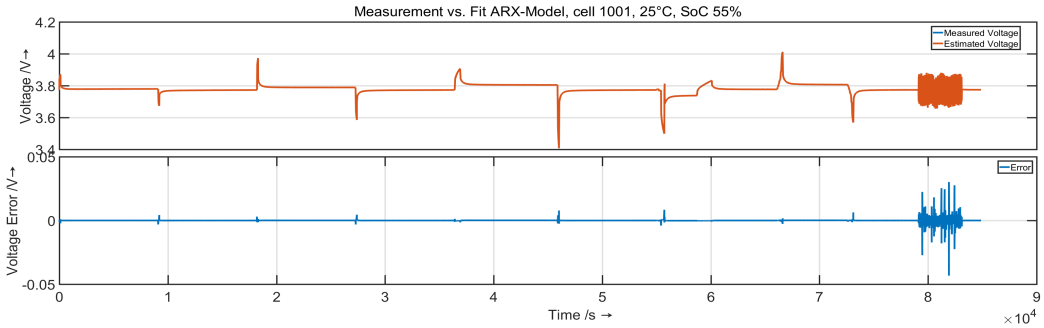
(a) Fit and model error at 25% SoC



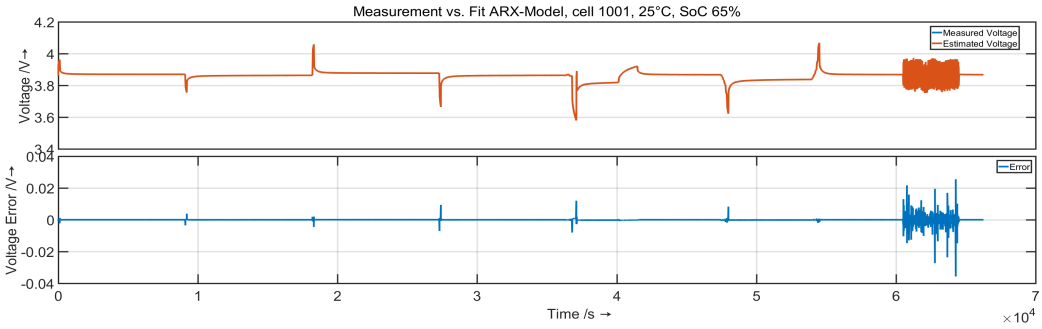
(b) Fit and model error at 35% SoC



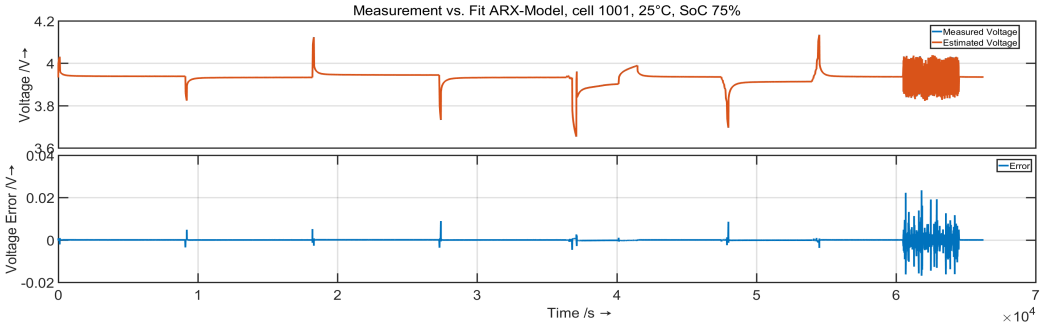
(c) Fit and model error at 45% SoC



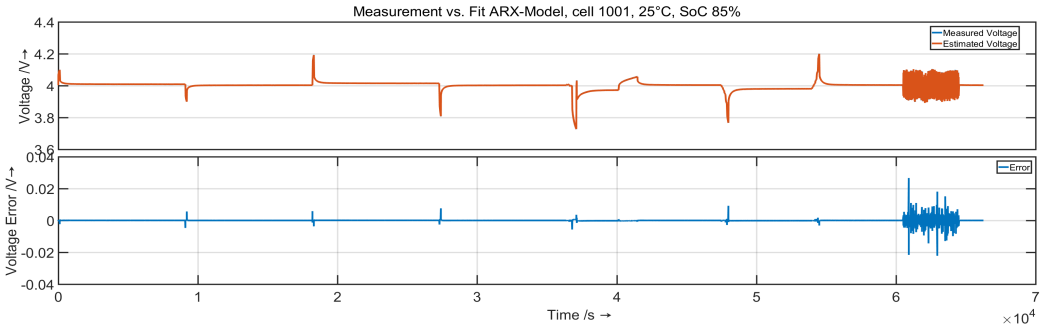
(d) Fit and model error at 55% SoC



(e) Fit and model error at 65% SoC



(f) Fit and model error at 75% SoC



(g) Fit and model error at 85% SoC

Figure 2.29: Measurement vs estimated cell voltage of cell 2013061002 at 25 ° C at SOC = (25%, 35%, 45%, 55%, 65%, 75%, 85%) ARX model

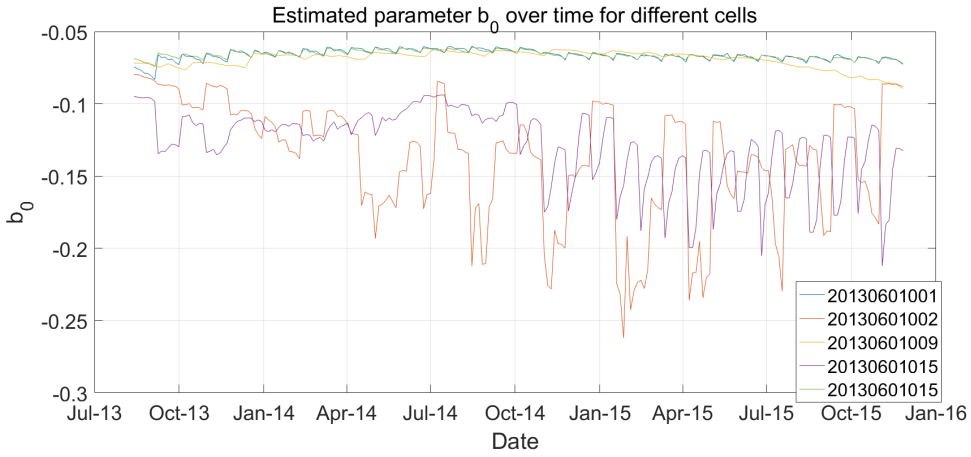
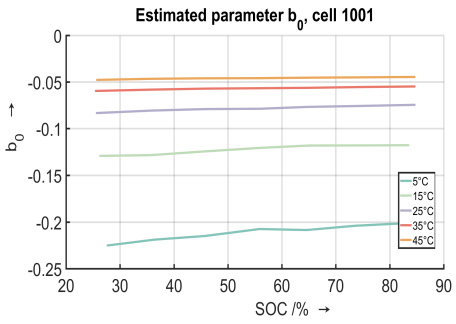


Figure 2.30: Progress of the parameter b_0 over time for different cells

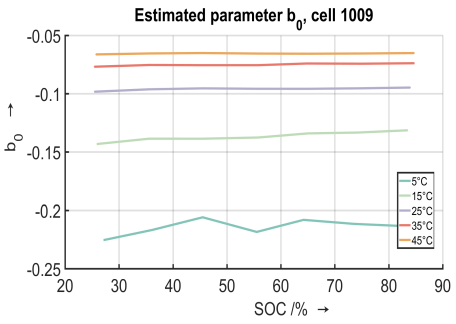
If we look at the mathematical description of the ARX-model output equation in Chapter 2.1.4 it is obvious that the parameter b_0 has the same role, compared to the output equation of the RC-model as the parameter R_0, thus should correspond to the inner resistance of the cell. Figure 2.25 presents the progress of the parameter b_0 over the time.

Parameter dependence of temperature and SoC

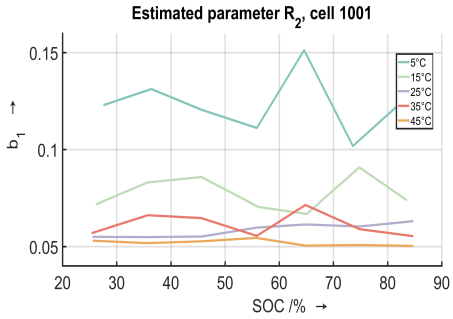
Same as with the RC-model we are going to show the parameters for two different cells under the same conditions, i.e. SoC, age, temperature. Again we can see that the parameters for every cell are different, even though the cells have the same size, capacity, and chemistry ("same cells"). With the ARX-Model it is hard to physically interpret each individual parameter. We can only assume physical interpretation for some parameters.



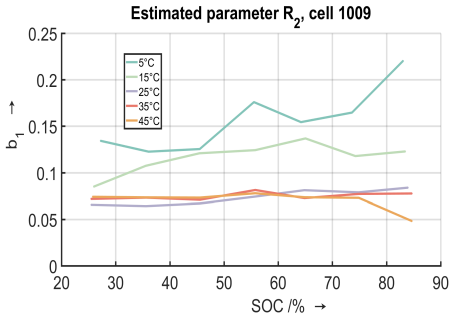
(a) b_0, cell 1001



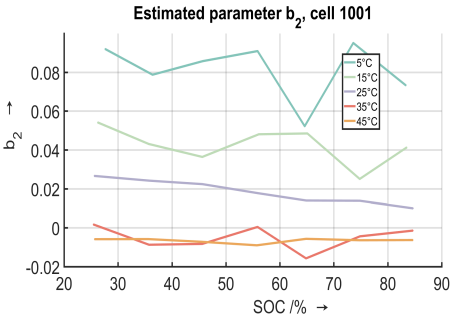
(b) b_0, cell 1009



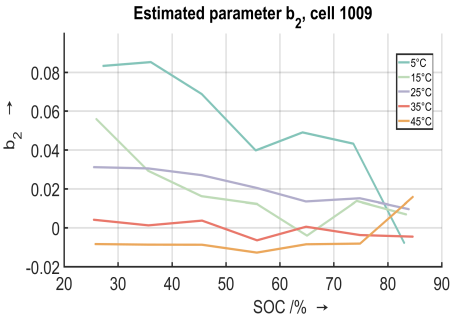
(c) b_1, cell 1001



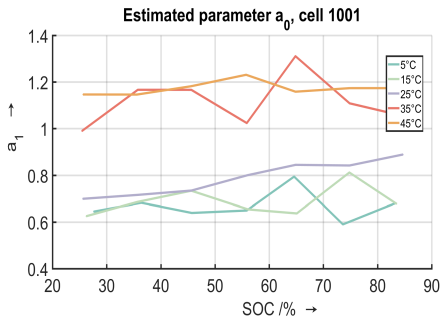
(d) b_1, cell 1009



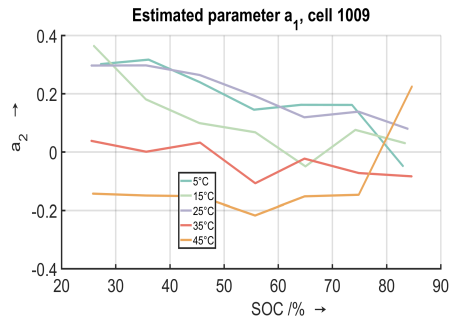
(e) b_2, cell 1001



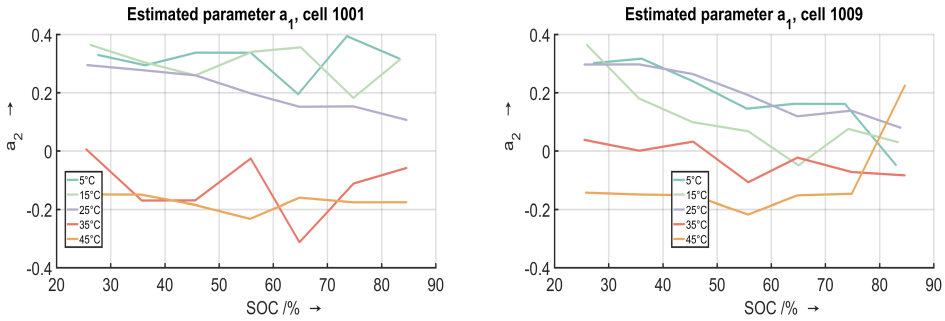
(f) b_2, cell 1009



(g) a_1, cell 1001



(h) a_1, cell 1009



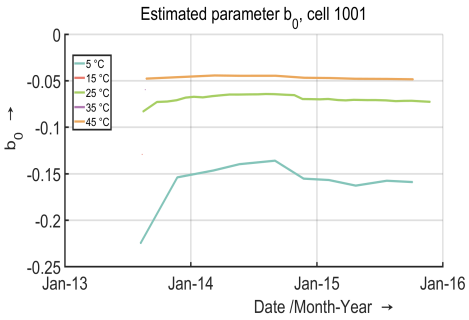
(i) a_2, cell 1001 (j) a_2, cell 1009

Figure 2.31: Estimated parameters over SoC and Temperature

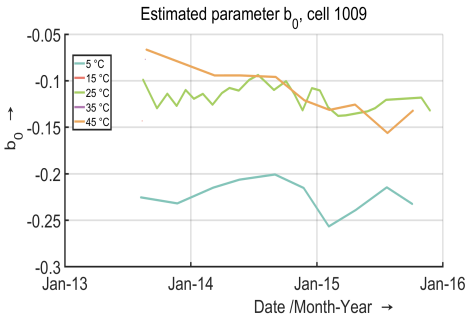
Figure 2.31 shows the progress of individual parameters depending on the temperature and under different SoC levels. The evaluated parameters are in the area, 25% - 85% SoC. A similar progress as shown with the RC- parameters may be observed.

Parameter dependence of temperature and aging conditions

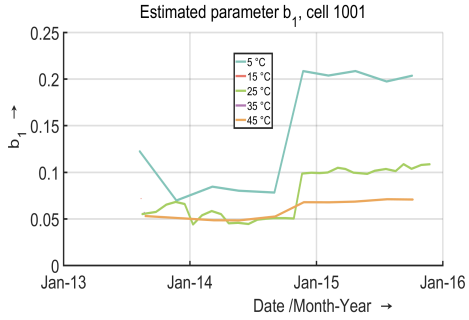
In this section, we show how the model parameters depend on temperature and aging conditions, for two different cells (same cell type) under same conditions see Figure 2.32.



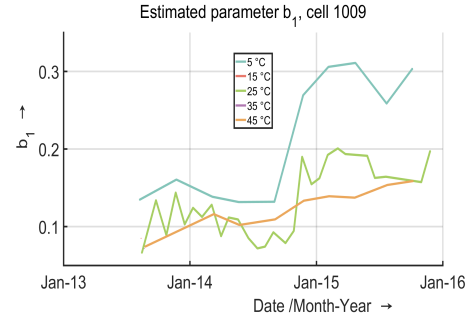
(a) b_0, cell 1001



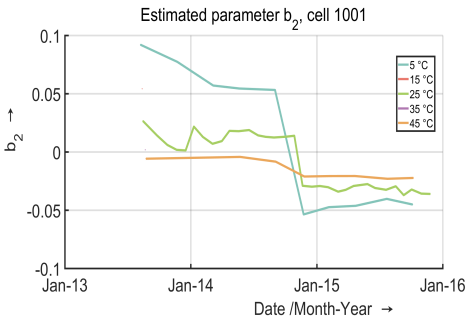
(b) b_0, cell 1009



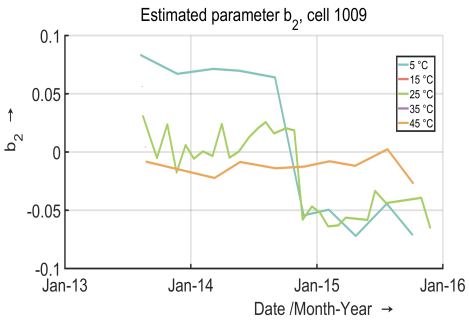
(c) b_1, cell 1001



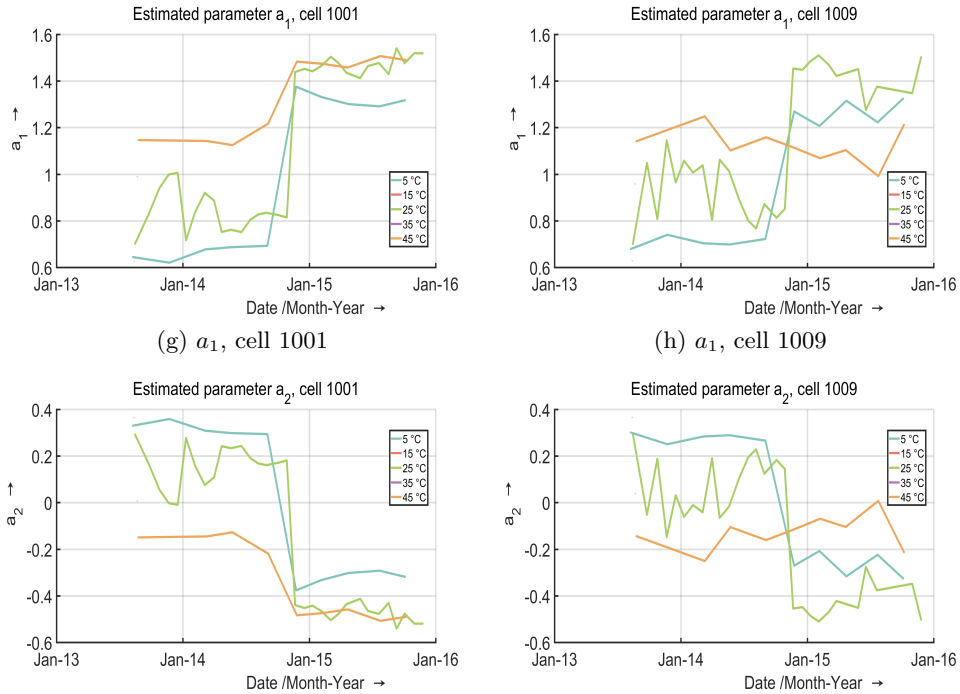
(d) b_1, cell 1009



(e) b_2, cell 1001



(f) b_2, cell 1009



(g) a_1, cell 1001

(h) a_1, cell 1009

(i) a_2, cell 1001

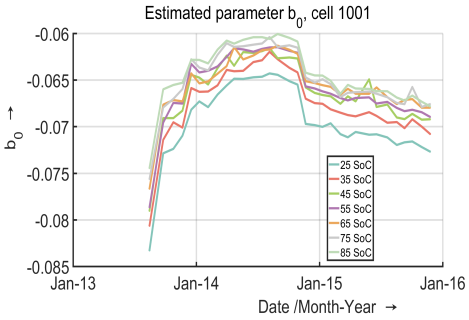
(j) a_2, cell 1009

Figure 2.32: Estimated parameters over Temperature and time, SoC 25%

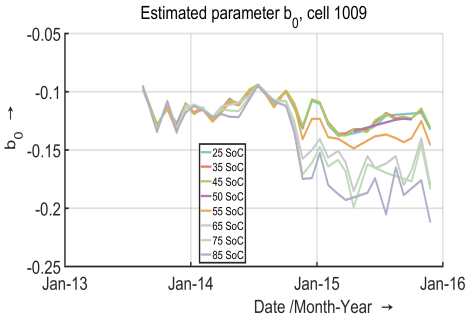
If we look at the individual parameter progress for different cells, it is obvious that even under different temperatures the value of the individual parameters lies in the same area regardless the individual cell. Unfortunately, this was insufficient to draw some obvious conclusions, regarding the parameter dependency and cell aging conditions. We should mention, the RTPs for five temperatures were done at the beginning and the end of the cell tests, that is why the legend in figures shows five temperatures but we may only see the progress of three temperatures, there are only a few single data points for the $15°C$ and $35°C$.

Parameter dependence of SoC and aging conditions

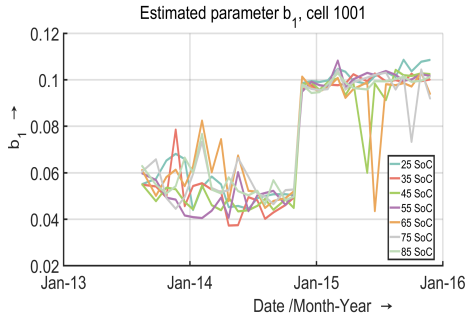
In this section, we show the model parameter dependence of SoC and aging conditions. Figure 2.33 shows two different cells that have undergone the same aging process.



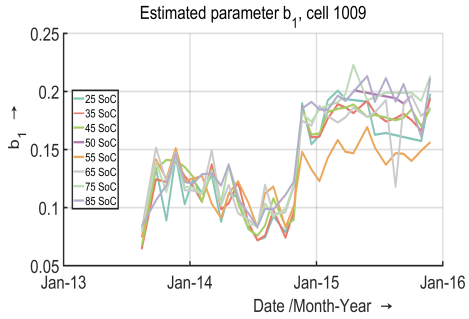
(a) b_0, cell 1001



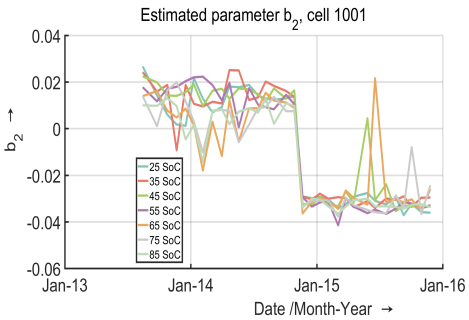
(b) b_0, cell 1009



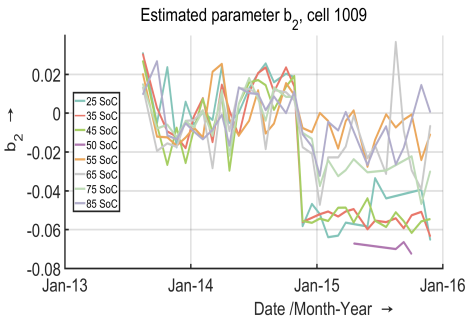
(c) b_1, cell 1001



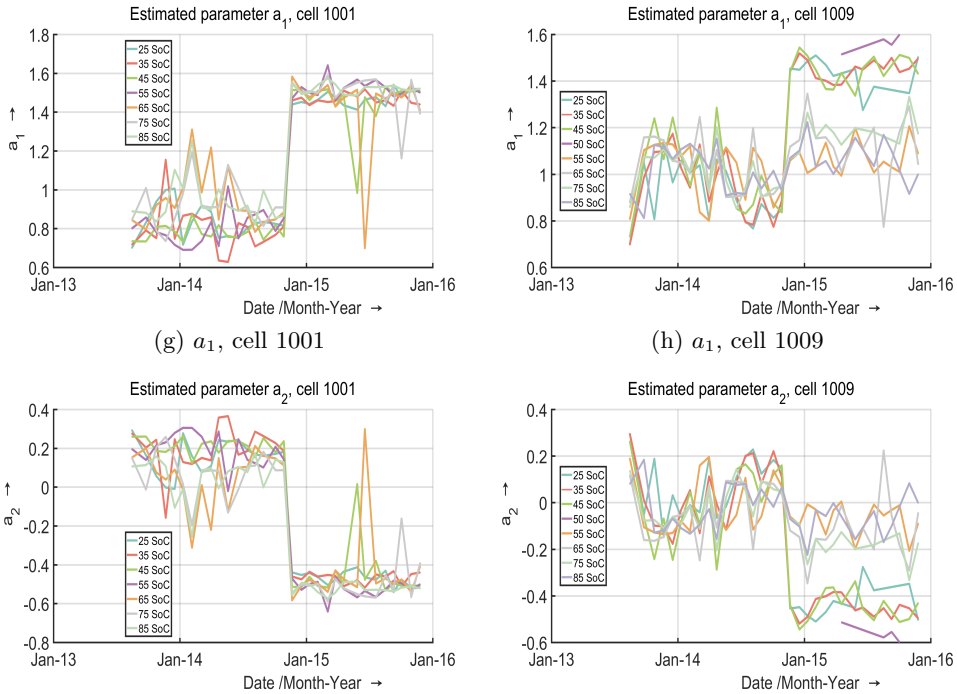
(d) b_1, cell 1009 temp



(e) b_2, cell 1001



(f) b_2, cell 1009



(g) a_1, cell 1001 (h) a_1, cell 1009

(i) a_2, cell 1001 (j) $a2$, cell 1009

Figure 2.33: Estimated parameters over SoC and time at 25 ° C

Again we see that the progress of a individual parameters compared between different cells, of the same type, are of approximately the same magnitude. Compared to the temperature dependency section above, we may say that the SoC has less influence on the parameters. The parameter progress here is kept narrow for different SoC stages.

Conclusion

In this section different approach for battery modeling, i.e. the auto regressive exogenous (ARX) model is presented. The ARX-model with model structure (n_a, n_b) was chosen, where $n_a = 2$ and $n_b = 3$. This means that there were five parameters to be identified. Similar as in the RC-model a least square method for parameter identification was used, same data was used as in RC-models, 18650 cells were used, with the nominal capacity $0.32Ah$ over its cycle lifespan.

2.2.3 Dynamic parameter comparison

In this chapter, we show the course of the estimated internal resistance R_0 fitted together with other degrees of freedom, as well as the progression of the R_0 in case where the other parameters are kept constant, and only internal resistance R_0 is estimated. With this, we want to separate the parameter dependencies from the cell's internal resistance R_0, and find out if model could still describe the data reasonably well. Table 2.34 shows us the course that we did in order to achieve this.

Datum	Temperatur	R_0/b_0	k_1/a_1	k_2/a_2	k_3/b_1	k_4/b_2
07.07.2013	5°	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
	15°	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
	25°	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
	35°	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
	45°	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
24.09.2013	5°	„Fit_2“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
	25°	„Fit_2“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
	…	„Fit_2“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
24.10.2013	5°	„Fit_3“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
	25°	„Fit_3“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“
..	…	„Fit_3“	„Fit_1“	„Fit_1“	„Fit_1“	„Fit_1“

Figure 2.34: Fit with only one degree of freedom[1]

The following Figure 2.35 demonstrate the difference between intern resistance R_0 fitted with and without the rest of the parameters. It has been noticed that there was no distinct difference, in the progress of the internal resistance, estimated along with other parameters and estimated alone. Small changes in the progress have been observed, but the magnitude is approximately the same in both cases. This was done for different aging levels as well as for different temperature and SoC stages.

[1] $K_1 = \frac{1}{(R_1 C_1)}$, $K_2 = \frac{1}{C_1}$, $K_3 = \frac{1}{(R_2 C_2)}$, $K_4 = \frac{1}{C_2}$

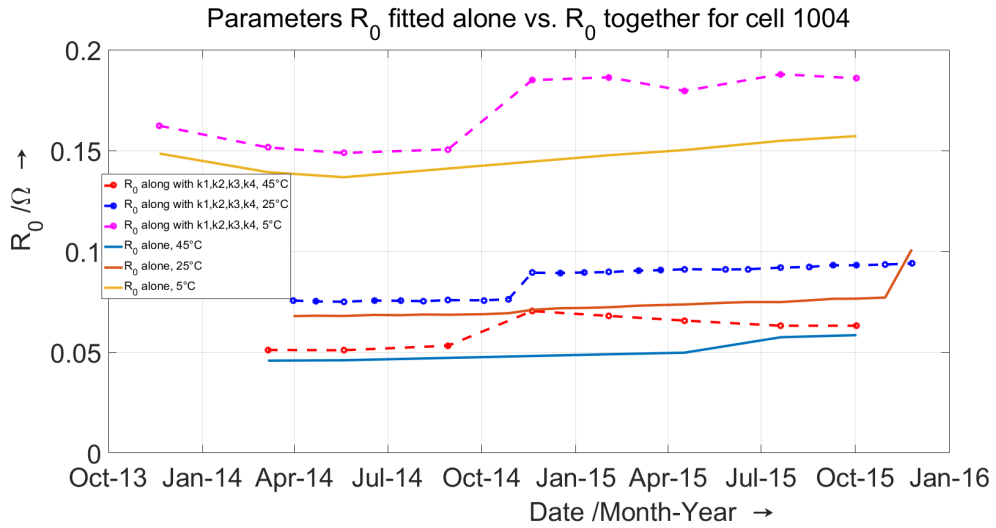


Figure 2.35: Course of internal resistance R_0 fitted allone vs. along with k_1, k_2, k_3, k_4, SoC 45%

The same was then for the different SoC stages. Figure 2.36 shows us the course of R_0 adapted with the parameters k_1, k_2, k_3, k_4, as well as adjusted with constant parameters k_1, k_2, k_3, k_4.

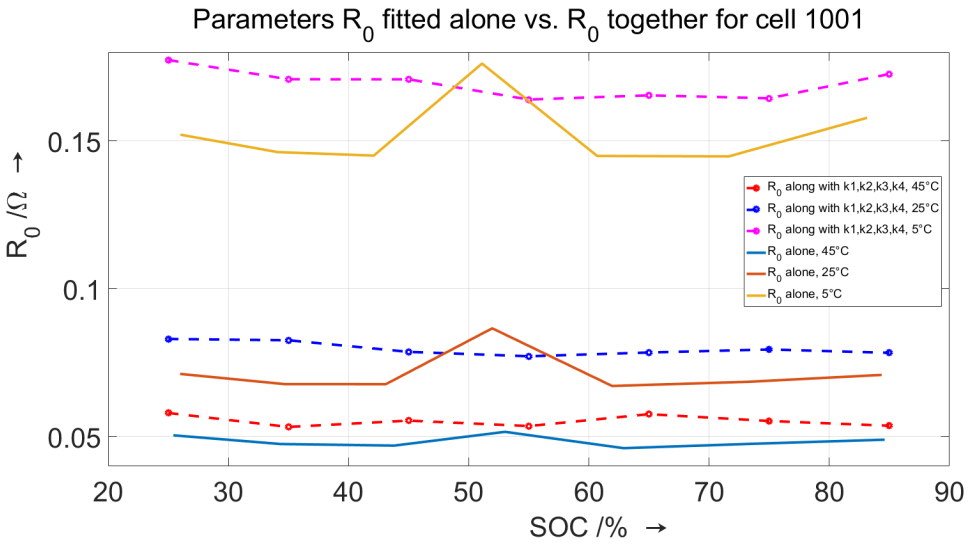


Figure 2.36: Course of internal resistance R_0 fitted allone vs. along with k_1, k_2, k_3, k_4, on 20.07.2015

The same procedure is used for ARX models with parameters a_1, a_2, b_1, b_2. Figure 2.37 shows the estimation of all five parameters.

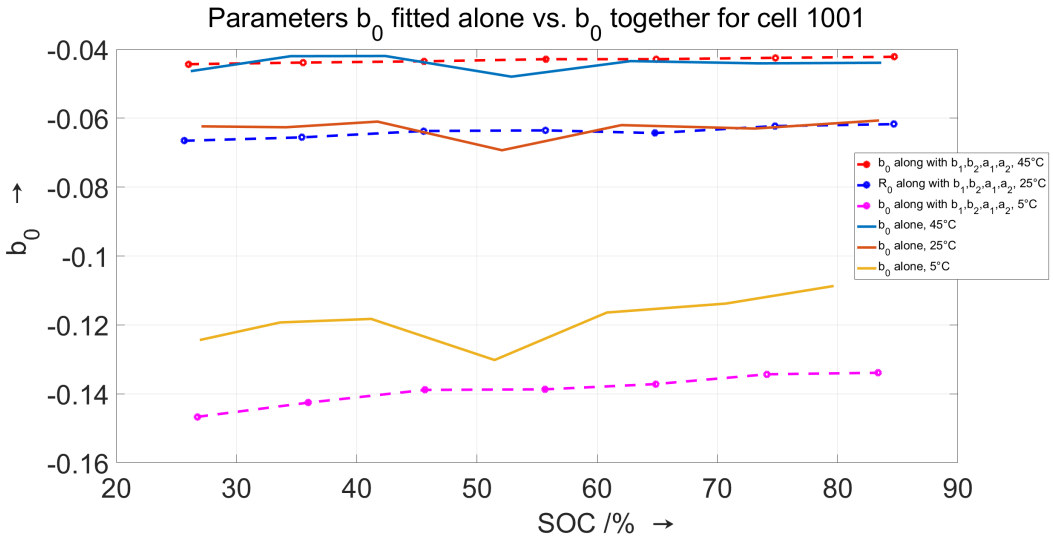


Figure 2.37: Estimated parameters b_0 fitted alone vs. along with a_1, a_2, b_1, b_2 for cell 1001 on 06.03.2014.

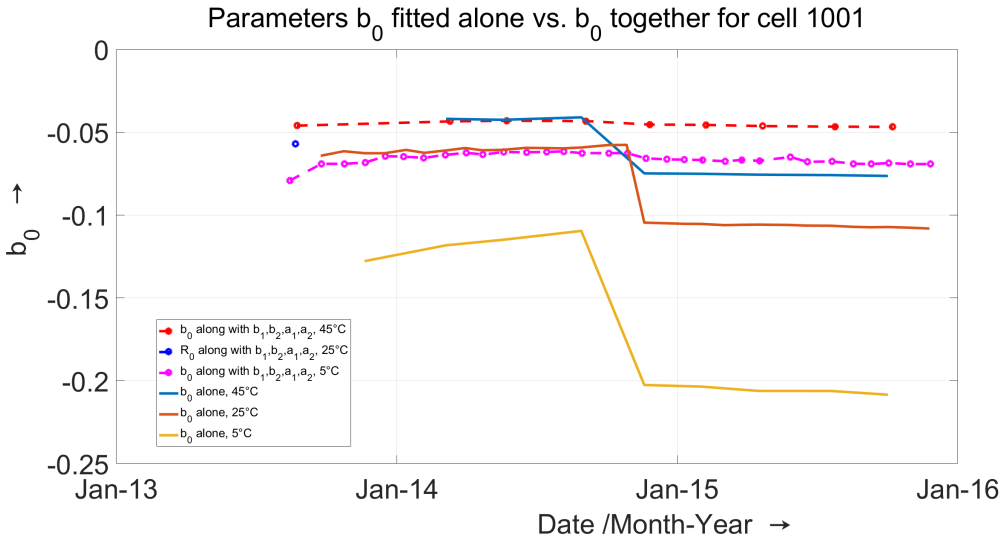


Figure 2.38: Estimated parameters b_0 fitted alone vs. along with a_1, a_2, b_1, b_2 for cell 1001 SoC = 45 %.

The same conclusion as for the RC-model parameters can be drawn. The progress of b_0 does not change significantly, either estimated along with the rest of the parameters, or estimated with the parameters held constant. The sign as well as the magnitude of the b_0 stays approximately the same, we observed some smoother progress but nothing more that that.
Note: in the one degree of freedom estimation, the SoC was also estimated, and thus is the SoC value not exact (e.g. 25% but between 23% 27%). while in the estimation for all five parameters the SoC was not estimated, but held exactly corresponding to the setpoint (for example 25%)

3 SOC Estimation

The main goal of this Master-Thesis is the state estimation of the cells condition. Some of these conditions, such as cells total capacity, internal resistance etc. can be describe as slow changing quantities, these we call the cells Parameters. For example cell resistance may change only few percent in value over the year. On the other hand we have fast changing quantities, which we call states of the cells. These are cells SoC, hysteresis state etc. For example some cells can be emptied in a matter of minutes. The reason way the States and Parameters need to be estimated, is that we can not measure these quantities directly.

In this Chapter we describe state estimation, mainly the SoC estimation, using different methods and algorithms to do so. Estimating the SoC is required in order to perform cell balancing, as well as to calculate the remaining power and energy. We can consider the SoC as a kind of fuel sensor (in combustion engines), that has the value between 0% and 100%, zero being empty and 100% being full. Among others, the main reason for the need of accurate SoC estimation is the cell sensitivity on overcharge and over-discharge. This can lead to fast irreversible cell damage (degradation), and in some cases to thermal runaway (burning or even cell explosion).

3.1 SOC estimation methods in general

There are different methods to determine the SoC of the cell, some of them are more complex and some less. So how do we chose one? Firstly, it depends on the type of the cell and the application for which the cell is used. In this work we are going to focus on the lithium-ion cell's (Li-cell's). These are widely used in the automotive area. First question that we have to ask our self is: what do we need, in order to *calculate/estimate* the cell SoC? We need to somehow combine the measurements and use our knowledge of the cell model in order to estimate SoC. Firstly we are interested in the measured quantities current, voltage, temperature. Secondly, our model has to deliver reasonable results, in order to depict the voltage response as the answer to an input.

3.1.1 SoC estimation using open circuit voltage

As previously shown in chapter 2.1.3 terminal voltage OCV is a function of SoC Equation 2.5. This means if the cell is at rest the hysteresis and the diffusion voltage can be ignored, so we have a relationship as shown in Figure 2.6. It is possible to make a lookup-table with this relationship, i.e. measured terminal voltage vs. SoC. The main problem with this method is that the cell has to be rested and the hysteresis has to be negligible in order for this to be true. In summary we can say that this method may be used in the applications where the cell is rarely under load, and often at rest for a long time. One more use for this method may be to estimate the initial SoC for other methods, e.g. Coulomb counting or Kalman filter.

3.1.2 Coulomb counting

Coulomb counting or ampere hour counting method is the most common method in calculating the SoC. After all the charging and discharging is related to current flow in/out of the cell. For accurate calculation of the SoC, we need starting point of the SoC. This means that we can calculate the SoC, by adding the current [24]

$$SoC = SoC_0 - \frac{\Delta t}{C_{nom}} \sum_{j=0}^{k-1} \eta_j i_j \tag{3.1}$$

If we look at the equation, it measures the amount of charge added to or removed from the cell, normalized with the total capacity. However in real applications the measured current comes with measurement noise, and if the initial value of the SoC is incorrect, there is no mechanism to correct this error over time. This means that the calculated error will sum up as same as we sum the current, and eventually it will not be even close to the real/expected SoC value. Another uncertainty is that we have to estimate the total capacity as well, we can expect capacity loss over time. All of these things influence the SoC deviations from the real value. So, we can conclude that this method is not the best choice to estimate the SoC, especially if its used in applications where long term prediction is needed, and if there is no possibility of resetting the initial values.

3.1.3 Model-based estimation

As we show both current and voltage based estimations have their advantages and disadvantages, do not fulfill the needs in many use cases for estimating the states. So the idea is to combine those two approaches somehow. This can be done by simulating cell input/output (current/voltage) behaviors, within a model-based estimation. In model-based estimation we have to measure input and the output of the cell (current and voltage). One issue that we are aware of, is the fact that the current and

voltage sensors are not ideal, so we have to expect noisy measurements. We model the uncertainty in current as processes noise, this will cause the change in the the cell state, but the change can not be predicted as we don't know the ratio between the true and measured current. The uncertainty in the voltage can be represented as a sensor noise, the uncertainty in the voltage will not effect the true state, but it means that we can not fully trust the accuracy of the voltage measurement [24].

in order for our model-based estimation to work we need a feedback loop, to compare the estimated and the measured voltage. If those two are different we know that our model state estimate is not good and we need to adjust it. If those two are same or the difference is small enough we have indicator that the model state estimation is good. So we used this information in the feedback mechanism to correct our model. Figure 3.1 shows us one possible model based estimation approach.

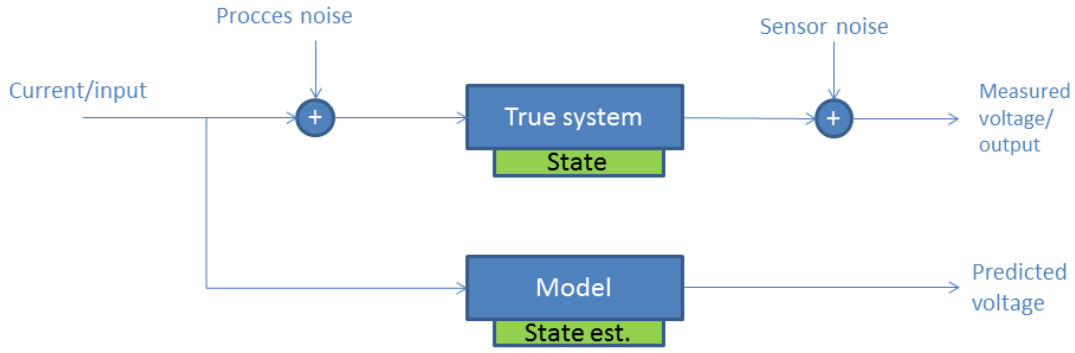


Figure 3.1: Model based estimation approach [24]

3.2 Implementing an EKF using the cell models

As we know in Electric and Hybrid vehicle, a battery management system (BMS) has to estimate values of the battery operating conditions: battery state of charge, power fade, capacity fade and available power. It is possible to use extended Kalman filtering (EKF) to estimate some of these states, on a lithium-ion battery. The HEV environment is very challenging on the BMS and the battery systems, so the estimation of some parameters should improve performance, robustness and useful lifetime of the battery pack. With the help of Kalman filters we can make intelligent (and sometimes optimal) means for estimating the present value of the time-varying "state" of a dynamic system [25].

Why do we use Extended and not Linear Kalman filters? The Kalman filtering algorithm is the optimum state estimator for linear systems, and as we mentioned earlier, we have a non-linear system (non-linearity is given by the OCV-SoC correlation). For non-linear a system linearization process is computed at each time step, to approximate the non-linear system as a linear time varying system (LTV). There are some problems with this approach, but it is still popular and can work well if the systems non-linearites are mild [24]. The EKF makes two simplifying assumptions:

1. When computing output estimates of a non-linear function, EKF assumes that the expected value of a non-linear function, and the unknown state is equal to the same non-linear function evaluated at the expected value of the state.

$$\mathbb{E}[\mathbf{fn}(x)] \approx \mathbf{fn}(\mathbb{E}[x]) \tag{3.2}$$

This is not true in general, it is only true when $\mathbf{fn}(x)$ is linear [24]

2. When computing the covariance estimate, EKF uses Taylor-series expansion to linearize the system equations around an operating point. Higher order terms are to be discarded from the expansion. This is the reason why EKF works best for systems having only mild non-linearities [24].

To derive the EKF equations we follow six general steps, as shown in [24].
Our discrete non-linear state-space model whose states we would like to estimate looks like:

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1}) \\ \mathbf{y}_k &= g(\mathbf{x}_k, \mathbf{u}_k, v_k), \end{aligned} \tag{3.3}$$

where $\mathbf{x}_k$ is the state vector, $\mathbf{u}_k$ is the known input signal (current), w_k is unmeasurable process-noise, v_k is unmeasurable sensor-noise, the system output $\mathbf{y}_k$ is noisy measurement of a cell voltage.

1a: *State – prediction time update*
Using assumption 1. and Eq. 3.3 we can approximate.

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbb{E}[f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1})|\mathbb{Y}_{k-1}] \\ &\approx f(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \bar{w}_{k-1}) \end{aligned} \tag{3.4}$$

Where $\bar{w}_{k-1} = \mathbb{E}[w_{k-1}]$. We assume that $\hat{\mathbf{x}}_{k-1}^+, \bar{w}_{k-1}$ propagate through the state equation, the approximate provides the expected value of the new state.

1b: *Error – covariance time update*

We approximate $\tilde{\mathbf{x}}_k^-$ through:

$$\begin{aligned}\tilde{\mathbf{x}}_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- \\ &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1}) - f(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \bar{w}_{k-1}),\end{aligned} \tag{3.5}$$

using the assumption 2. the first term is defined by $p_{k-1} = (\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \bar{w}_{k-1})$

$$\begin{aligned}\mathbf{x}_k \approx f(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \bar{w}_{k-1}) &+ \underbrace{\left.\frac{df(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1})}{d\mathbf{x}_{k-1}}\right|_{p_{k-1}}}_{Defined\ as\ \hat{\mathbf{A}}_{k-1}} (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+) \\ &+ \underbrace{\left.\frac{df(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1})}{dw_{k-1}}\right|_{p_{k-1}}}_{Defined\ as\ \hat{\mathbf{B}}_{k-1}} (w_{k-1} - \bar{w}_{k-1}),\end{aligned} \tag{3.6}$$

From Eq. 3.6 follows:

$$\tilde{\mathbf{x}}_k^- \approx \hat{\mathbf{A}}_{k-1}\tilde{\mathbf{x}}_{k-1}^+ + \hat{\mathbf{B}}_{k-1}\tilde{w}_{k-1} \tag{3.7}$$

To find the prediction $error - covariance$ we substitute the Eq. 3.5 and we get:

$$\begin{aligned}\mathbf{P}_{\tilde{\mathbf{x}},\mathbf{k}}^- &= \mathbb{E}[(\tilde{\mathbf{x}}_k^-)(\tilde{\mathbf{x}}_k^-)^T] \\ &\approx \mathbb{E}[(\hat{\mathbf{A}}_{k-1}\tilde{\mathbf{x}}_{k-1}^+ + \hat{\mathbf{B}}_{k-1}\tilde{w}_{k-1})(\hat{\mathbf{A}}_{k-1}\tilde{\mathbf{x}}_{k-1}^+ + \hat{\mathbf{B}}_{k-1}\tilde{w}_{k-1})^T] \\ &= \hat{\mathbf{A}}_{k-1}\mathbb{E}[(\tilde{\mathbf{x}}_{k-1}^+)(\tilde{\mathbf{x}}_{k-1}^+)^T]\hat{\mathbf{A}}_{k-1}^T + \hat{\mathbf{B}}_{k-1}\mathbb{E}[\tilde{w}_{k-1}]\mathbb{E}[(\tilde{\mathbf{x}}_{k-1}^+)^T]\hat{\mathbf{A}}_{k-1}^T \\ &+ \hat{\mathbf{A}}_{k-1}\mathbb{E}[\tilde{\mathbf{x}}_{k-1}^+]\mathbb{E}[\tilde{w}_{k-1}^T]\hat{\mathbf{B}}_{k-1}^T + \hat{\mathbf{B}}_{k-1}\mathbb{E}[\tilde{w}_{k-1}\tilde{w}_{k-1}^T]\hat{\mathbf{B}}_{k-1}^T \\ &= \hat{\mathbf{A}}_{k-1}\mathbf{P}_{\tilde{\mathbf{x}},\mathbf{k-1}}^+\hat{\mathbf{A}}_{k-1}^T + \hat{\mathbf{B}}_{k-1}\mathbf{P}_{\tilde{w}}\hat{\mathbf{B}}_{k-1}^T \\ &= \hat{\mathbf{A}}_{k-1}\mathbf{P}_{\tilde{\mathbf{x}},\mathbf{k-1}}^+\hat{\mathbf{A}}_{k-1}^T + Q_{w,k-1},\end{aligned} \tag{3.8}$$

Where we assume that the prediction is uncorrelated with the process noise at the same time index, and the state-prediction error is zero mean [24]. We determine analytically both matrices from Eq. 3.6 as shown below, both matrices are functions of time.

$$\hat{\mathbf{A}}_\mathbf{k} = \begin{bmatrix} \frac{df_1(x_k,u_k,w_k)}{dx_{k,1}} & \frac{df_1(x_k,u_k,w_k)}{dx_{k,2}} & \dots & \frac{df_1(x_k,u_k,w_k)}{dx_{k,n}} \\ \frac{df_2(x_k,u_k,w_k)}{dx_{k,1}} & \frac{df_2(x_k,u_k,w_k)}{dx_{k,2}} & \dots & \frac{df_2(x_k,u_k,w_k)}{dx_{k,n}} \\ . & . & \dots & . \\ . & . & \dots & . \\ \frac{df_n(x_k,u_k,w_k)}{dx_{k,1}} & \frac{df_n(x_k,u_k,w_k)}{dx_{k,2}} & \dots & \frac{df_n(x_k,u_k,w_k)}{dx_{k,n}} \end{bmatrix}$$

We do the same to calculate $\hat{\mathbf{B}}_{k-1}$.

In this case, it can be assumed that the terms $d\mathbf{u}_{k-1}/d\mathbf{x}_{k-1}$ and $dw_{k-1}/d\mathbf{x}_{k-1}$ are zero since the present deterministic input nor the present process noise are functions

of the current system state. Hence that the partial differential is the same as the total differential[24].

$$\frac{df(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1})}{dx_{k-1}} = \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1})}{\partial x_{k-1}}, \tag{3.9}$$

But we will see in Chapter 4, when there is a need for model parameter estimation, that these are not the same. 1c: *Predict system output* $\mathbf{y}_k$
For linear Kalman filters we predict the system output using the model and the prior information.

$$\begin{aligned} \hat{\mathbf{y}}_k &= \mathbb{E}[\mathbf{y}_k|\mathbb{Y}_{k-1}] \\ &= \mathbb{E}[g(\mathbf{x}_k, \mathbf{u}_k, v_k)|\mathbb{Y}_{k-1}], \end{aligned} \tag{3.10}$$

Using the Eq. 3.10 and the assumption 1 we approximate the output by

$$\begin{aligned} \hat{\mathbf{y}}_k &= \mathbb{E}[g(\mathbf{x}_k, \mathbf{u}_k, v_k)|\mathbb{Y}_{k-1}] \\ &\approx g(\hat{\mathbf{x}}_k^-, \mathbf{u}_k, \bar{v}_k), \end{aligned} \tag{3.11}$$

where $\bar{v}_k = \mathbb{E}[v_k]$, usually $\bar{v}_k = 0$
2a: *Estimator gain matrix* $\mathbf{L}_k$
In order to compute the Kalman gain we use the Taylor-series expansion of $\mathbf{y}_k$ around $q_k = \{\hat{\mathbf{x}}_k, \mathbf{u}_k, \bar{v}_k\}$. Same as in the step 1b we show that the total derivatives shown in Eq. 3.12 are equal to partial derivatives.
Output prediction error:

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k = g(\mathbf{x}_k, \mathbf{u}_k, v_k) - g(\hat{\mathbf{x}}_\mathbf{k}^-, \mathbf{u}_k, \bar{v}_k),$$

Taylor-series expansion of $\mathbf{y}_k$:

$$\begin{aligned} \mathbf{y}_k &\approx g(\hat{\mathbf{x}}_\mathbf{k}^-, \mathbf{u}_k, \bar{v}_k) + \underbrace{\frac{dg(\mathbf{x}_k, \mathbf{u}_k, v_k)}{d\mathbf{x}_k}\Big|_{q_k}}_{Defined\ as\ \hat{\mathbf{C}}_k}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \\ &\quad + \underbrace{\frac{dg(\mathbf{x}_k, \mathbf{u}_k, v_k)}{dv_k}\Big|_{q_k}}_{Defined\ as\ \hat{\mathbf{D}}_k}(v_k - \bar{v}_k) \\ \tilde{\mathbf{y}}_k &\approx \hat{\mathbf{C}}_k\tilde{\mathbf{x}}_k^- + \hat{\mathbf{D}}_k\tilde{v}_k, \end{aligned} \tag{3.12}$$

From 1b follows:

$$\begin{aligned} \frac{dg(\mathbf{x}_k, \mathbf{u}_k, v_k)}{d\mathbf{x}_k} &= \frac{\partial g(\mathbf{x}_k, \mathbf{u}_k, v_k)}{\partial \mathbf{x}_k} \\ \frac{dg(\mathbf{x}_k, \mathbf{u}_k, v_k)}{dv_k} &= \frac{\partial g(\mathbf{x}_k, \mathbf{u}_k, v_k)}{\partial v_k}, \end{aligned}$$

Using this we can compute the Kalman gain:

$$\mathbf{P}_{\tilde{\mathbf{y}},k} \approx \hat{\mathbf{C}}_k \mathbf{P}^-_{\tilde{x},k} \hat{\mathbf{C}}^T_k + \underbrace{\hat{\mathbf{D}}_k \mathbf{P}_{\tilde{v}} \hat{\mathbf{D}}^T_k}_{\mathbf{R}}$$

$$\mathbf{P}_{\tilde{\mathbf{y}}\tilde{\mathbf{x}},k} = \mathbf{P}^-_{\tilde{x},k} \hat{\mathbf{C}}^T_k \tag{3.13}$$

$$\mathbf{L}_k = \mathbf{P}^-_{\tilde{\mathbf{x}},k} \hat{\mathbf{C}}^T_k [\hat{\mathbf{C}}_k \mathbf{P}^-_{\tilde{x},k} \hat{\mathbf{C}}^T_k + \mathbf{R}]^{-1},$$

2b: *State − estimate measurement update*
We use the estimator gain L_k and the innovation $(\mathbf{y}_k - \hat{\mathbf{y}}_k)$ to compute the state estimate. Same as with linear Kalman filter:

$$\hat{\mathbf{x}}^+_k = \hat{\mathbf{x}}^-_k + \mathbf{L}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k), \tag{3.14}$$

2c: *Error − covariance measurement update*
We compute the covariance update as:

$$\mathbf{P}^+_{\tilde{\mathbf{x}},k} = (\mathbf{I} - \mathbf{L}_k \mathbf{C}_k) \mathbf{P}^-_{\tilde{\mathbf{x}},k}, \tag{3.15}$$

Figure 3.2 shows the data flow within the EKF

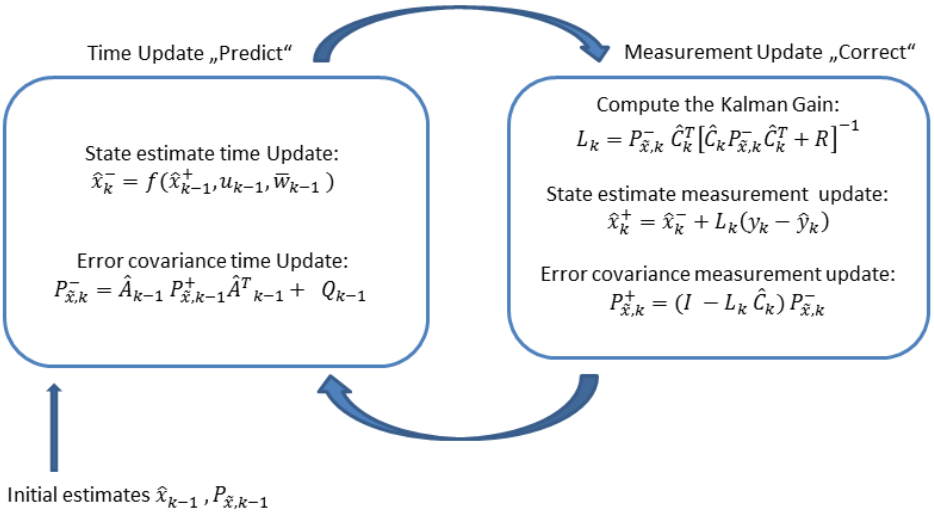


Figure 3.2: Interaction within the EKF

A summary of the entire EKF algorithm is given in Appendix A.

3.2.1 EKF for Second order RC-model

We apply the algorithm shown in section 3.2 on our second order RC-model described in section 2.1.3. Our goal is to accurately estimate the cell state, mainly the SoC of the cell, using the equivalent-circuit model and the EKF. By accurate SoC estimation we can prevent over-charging or over-discharging the battery witch can lead to damage or reduced lifetime of a battery, it also can make the Performance of a battery better in terms of using the entire capacity. In order to do so we need to compute the components of the state equation $\hat{\mathbf{A}}_k$ and $\hat{\mathbf{B}}_k$, as well as the components of the output equation $\hat{\mathbf{C}}_k$ and $\hat{\mathbf{D}}_k$. Using the discrete battery model described in section 2.1.3, we calculate the Jacobian using the partial derivatives from the equations 2.10 with respect to state. The Jacobian $\hat{\mathbf{A}}_k$ remains constant, its states are independent. But we need to compute the state dependent component $\hat{\mathbf{C}}_k$. This can be done by approximating the partial derivative. For this we need a SoC vector with evenly spaced points, and the corresponding open-circuit voltage vector OCV Figure 2.6.

$$\hat{\mathbf{A}}_k = \begin{bmatrix} e^{\frac{-\Delta t}{R_1 C_1}} & 0 & 0 \\ 0 & e^{\frac{-\Delta t}{R_2 C_2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \hat{\mathbf{C}}_k = \begin{bmatrix} -1 & -1 & \frac{\partial OCV(soc_k)}{\partial SoC_k} \end{bmatrix} \Big|_{soc_k = \hat{soc}_k^-} \tag{3.16}$$

We compute the matrix $\hat{\mathbf{B}}_k$ and $\hat{\mathbf{D}}_k$ as shown below:

$$\hat{\mathbf{B}}_k = \begin{bmatrix} R_1(1 - e^{\frac{-\Delta t}{R_1 C_1}}) \\ R_2(1 - e^{\frac{-\Delta t}{R_2 C_2}}) \\ \frac{-\eta \Delta t}{C_{nom} * 3600} \end{bmatrix}; \quad \hat{\mathbf{D}}_k = [R_0] \tag{3.17}$$

We simplify the model by making $\eta = 1$, and assuming that the measured current and voltage already contain the Gaussian process-noise.

Having reduced the error between the measured data and the outputs from our model to our satisfaction, we can proceed with the implementation of the extended Kalman filter. The algorithm compares the output data obtained from the model under load, with the measured cell voltage. The error is used to match the state of the cell model to the measured cell voltages. It is important to note that the OCV curve of lithium-ion cells usually has non-linearities. To cope with these non-linearities, the EKF algorithm was designed. Figures (3.3-3.5) show the implementation of our extended Kalman filter using *Matlab Simulink*. The input/output quantities, current and voltage are measured and estimated/simulated values. Currents and voltages were measured on 18650 Li-ion cells within the project.

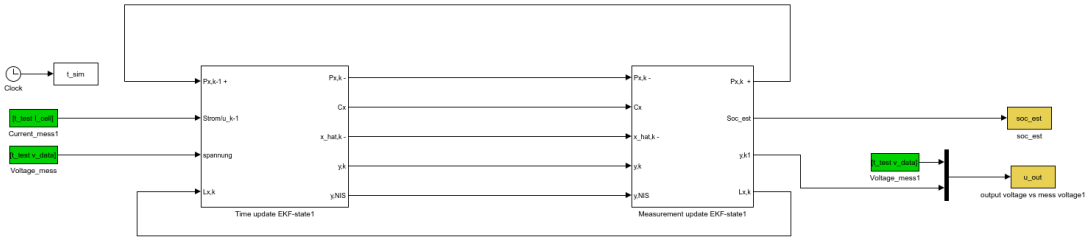


Figure 3.3: Extended Kalman filter *Simulink* model

As the output of the EKF we get the estimated voltage and the estimated SoC, the voltage is compared with the measured data, and the reference for the SoC is the calculated SoC with the help of Coulomb counting method.

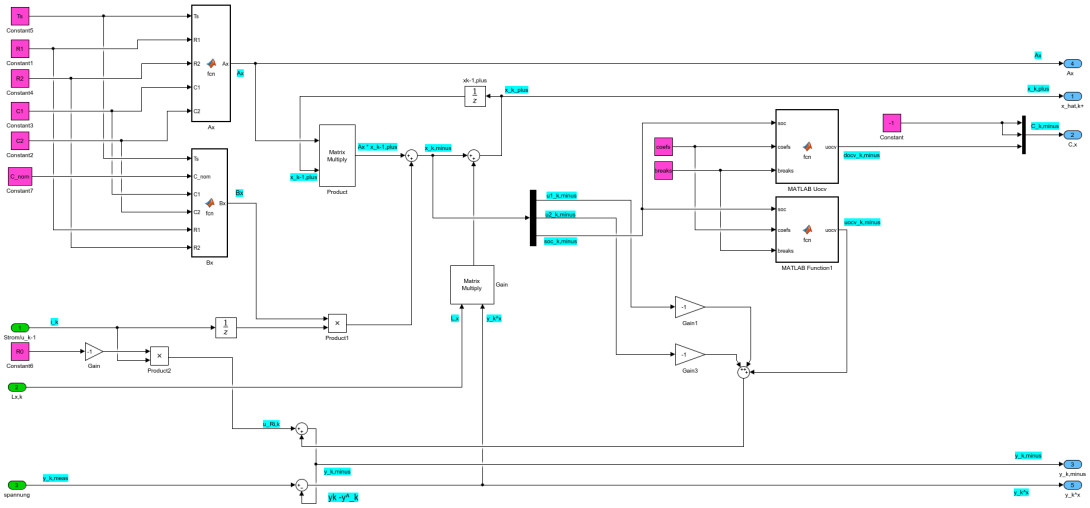


Figure 3.4: Extended Kalman filter *Simulink* subsystem time update

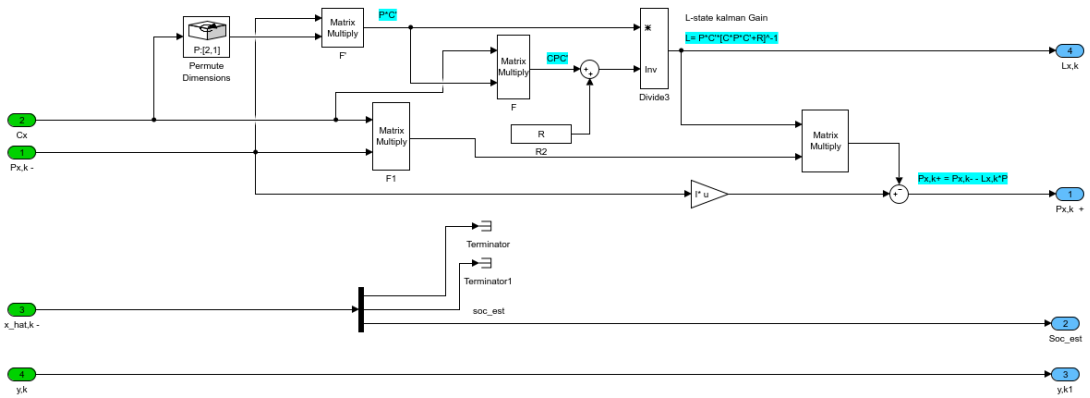


Figure 3.5: Extended Kalman filter *Simulink* subsystem measurement update

The sample results and discussions for RTPs and Loadpoints using the EKF algorithm is shown in the chapter 3.3.

3.2.2 EKF for ARX-Model

The equivalent-circuit models are widely used in real applications, however we need to consider some questions. By increasing the model order, the model precision is improved but at the cost of model complexity. The second question is about discretization and implementation. That was one of the reasons we consider a more straightforward approach, which was the ARX-model. Here we adopt the extended Kalman filter algorithm in order to improve the proposed battery model. Figure 3.6 shows us the proposed ARX-model together with the EKF.

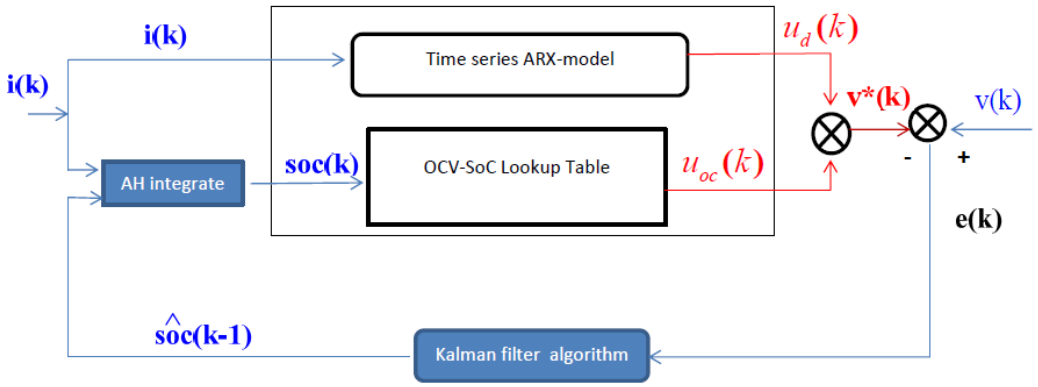


Figure 3.6: ARX battery model with EKF

Where:

- $i(k)$ Measured current
- $u_d(k)$ Transient voltage
- $u_{ocv}(k)$ Open-circuit voltage
- $V^*(k)$ Estimated voltage
- $V(k)$ Measured voltage

ARX models are very popular for discrete systems with single input and single output (SISO). Main disadvantage is that we can not assign physical value to the estimated parameters, Depending on the error between the measured and estimated voltage, the EKF adapts the SoC in order to change the output voltage from the model, and thus minimize the voltage error. After some iterations, the error converges towards optimum, and the SoC converges towards its optimal value [34]. In contrast to the EKF with RC-model, the implementation of the ARX-model as well as the EKF algorithm was done using *Matlab -script*, without *Simulink*. We used the same general six steps as described in the section 3.2, our state space equation can be expressed as:

$$\begin{aligned} x_{k+1} &= f(x_k, i_k) + w_k \\ y_k &= g(x_k, i_k) + v_k \end{aligned} \tag{3.18}$$

where x_k is the $[soc(k)\ u_d(k)\ u_d(k-1)]$, $g(x_k, i_k)$ is the output equation, k is the time index, w_k is a process white noise with covariance $Q_k = diag(Q_{soc(k)},\ Q_{u_d(k)},\ Q_{u_d(k-1)})$, and v_k is observation white noise with covariance R_k. The data flow is here the same as in Figure 3.2

3.3 EKF Results and Discussion

RTPs are used to characterize the cells before starting the long-term testing of the load cycles or driving cycles, as well as to characterize the cells every three weeks and for the final characterization. Duration of each RTP is ca. 38 hours. Figure 3.7 shows an example RTP on the cell 1002.

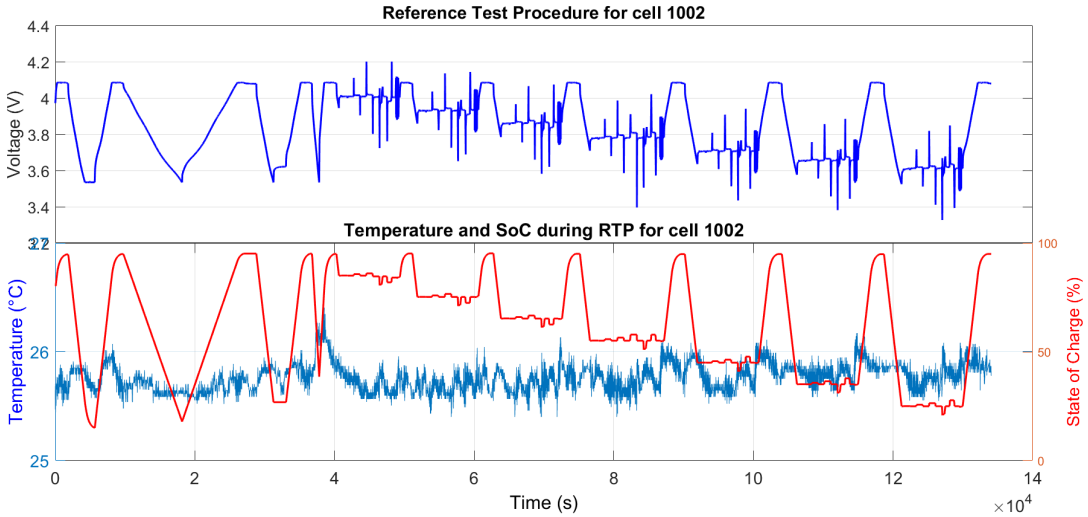


Figure 3.7: RTP for the cell 1002 at ≈ 25°C

Different load cycles are tested on the different cells in the long term. The cells are kept in the operating area according to their specification. For this purpose, a basic version for the operation of the load points was written and individually adjusted for each load point according to the influencing factors (profile parameters). Duration of each load cycle is ca. 3 weeks. Figure 3.8 shows an example of the load cycle.

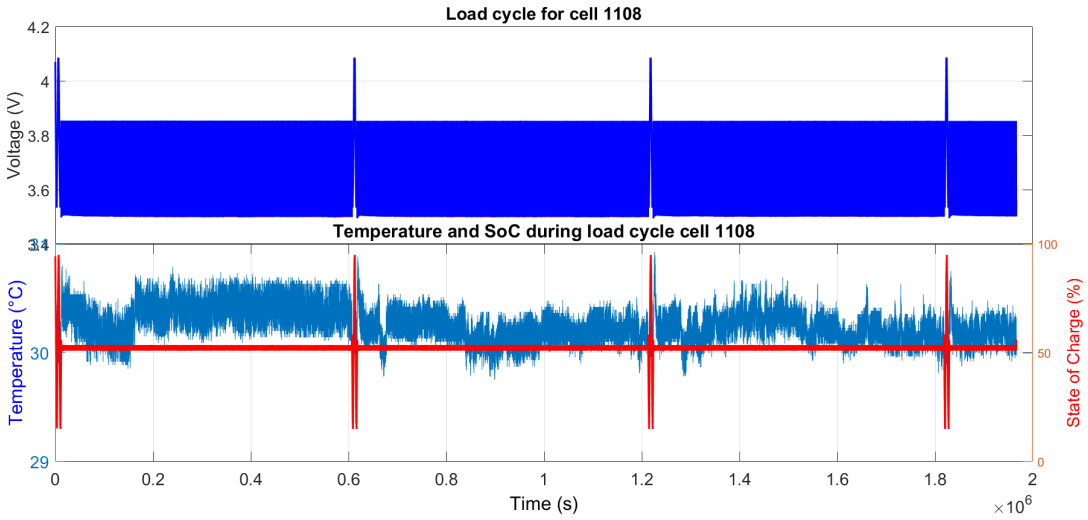


Figure 3.8: Example of the load cycle

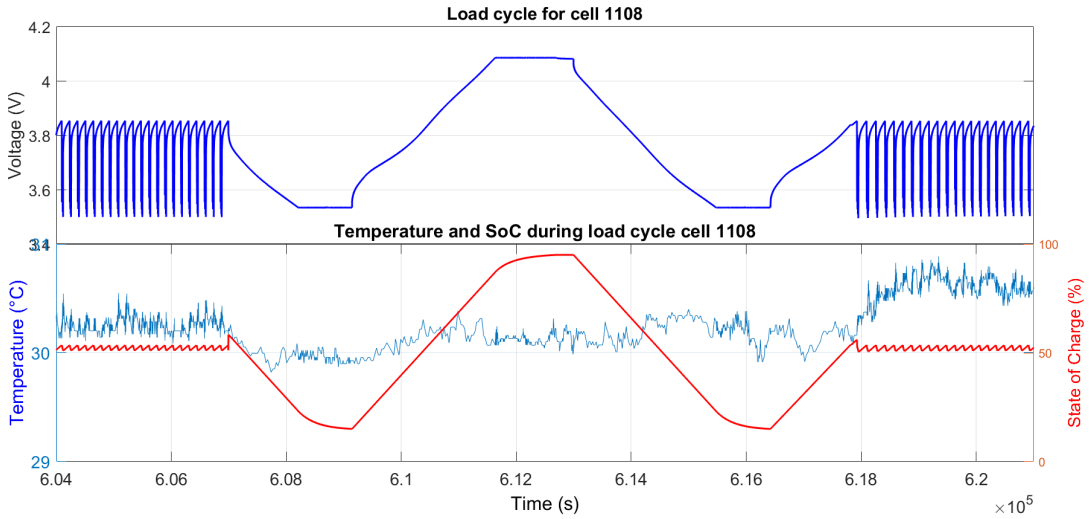


Figure 3.9: Same example of the load cycle zoomed in

Due to the simple feed-forward SoC calculation based on device-logged, charge offsets, without feedback to the measured cell voltages, all SoC data is still untrustworthy. At low dSoC (very short pulses), a systematic drift can also be detected [6].

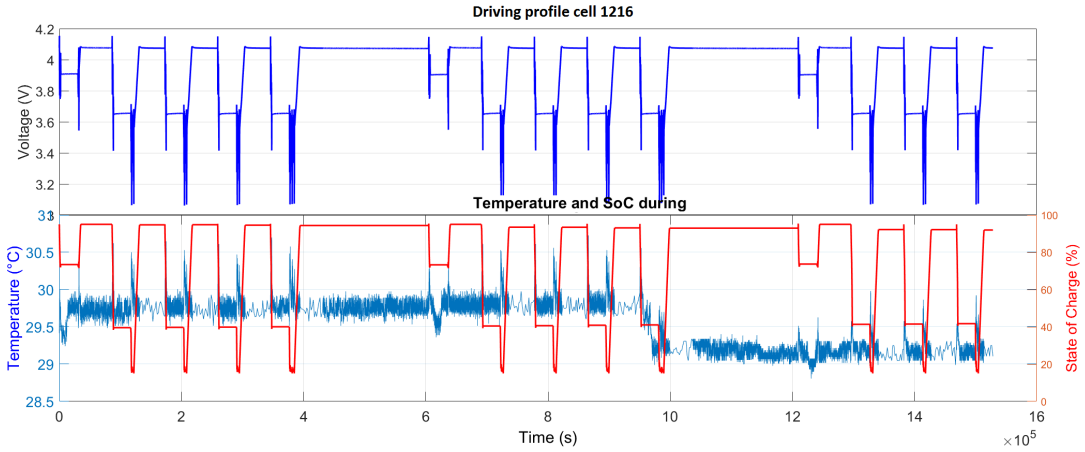


Figure 3.10: Example of the driving cycle

Figure 3.10 shows the driving cycle, it covers one week with the aim to represent real driving conditions from Monday to Friday and rest during the weekend. On Monday two current profiles are run with a rest of $8h$ in between. From Tuesday to Thursday tree profiles are run on each day with 8 hours rest between first and second. Friday routine is basically identical to the previous days, except that the last current simulation file is repeated.

3.3.1 EKF for Second order RC-model

Here we present the results, and the differences between the measured and estimated data using the EKF. The tests were performed on the RTPs, load cycles and driving cycle, for individual cycles of three weeks, as well as coupled together.

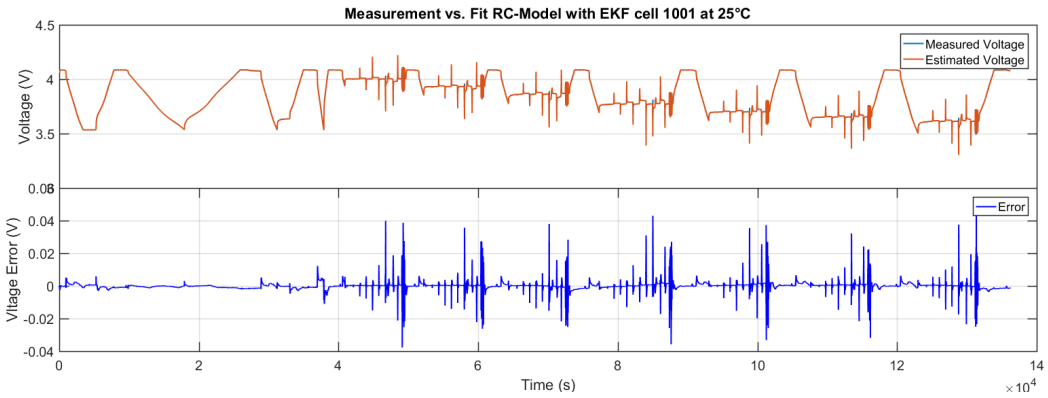


Figure 3.11: Estimated vs Measured voltage using EKF cell 1001

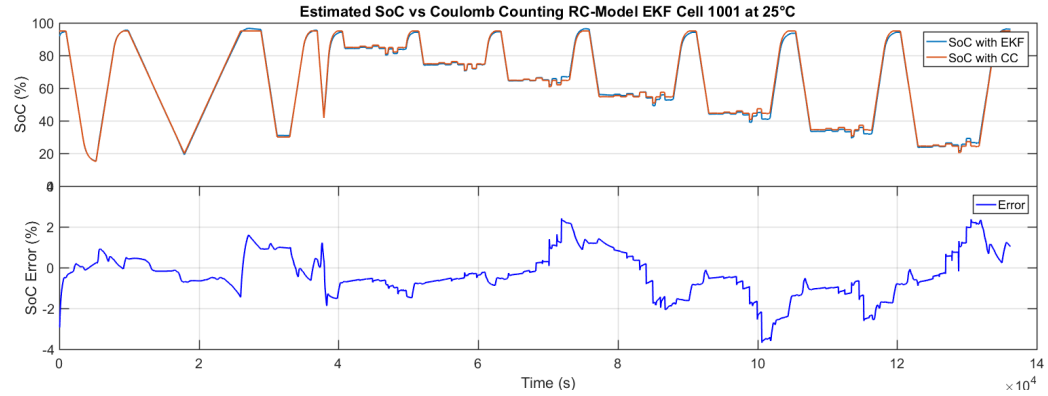


Figure 3.12: Estimated vs calculated SoC and the error between them for cell 1001

Figure 3.12 shows the estimated SoC using the EKF versus the calculated SoC using the coulomb counting method for RTP, as well as the SoC uncertainty between these two. In the top frame we see that the SoC tracking by EKF, observed in a absolute sense is quite good. The bottom frame shows the SoC error. The root-mean-square (RMS) for this example was 1.106%.

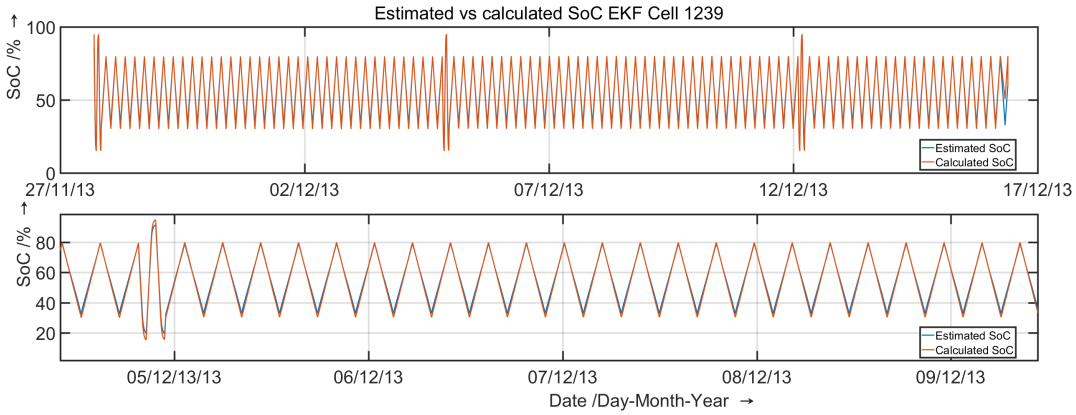


Figure 3.13: Estimated vs calculated SoC for cell 1239

Figure 3.13 shows load profile started at 27.11.2013 and was finished approximately at 16.12.2013. The top frame shows, the progress of a calculated and estimated SoC, for the whole load profile, in bottom frame we see the zoom on the same plot.

As mentioned earlier, we have RTPs between the load cycle, and there might have been some couple of hours breaks in the cell cycling, due to time needed to change the cycling protocol. In Figure 3.14 we can see load cycles coupled together with RTPs. This, as well as all further long time estimation, was done without breaks in simulation between load cycle and RTPs. The finale states at the end of the load cycle/RTPs were applied as an initial value of the next load cycle/RTP, the transferred states are state vector x_k and covarance matrix P_k. The OCV-curve and the RC-parameters are kept constant, although they are diverse for different temperatures. This is the main reason, this implausible peaks in SoC values for 5°C RTPs (see fig. 3.14 occur.)

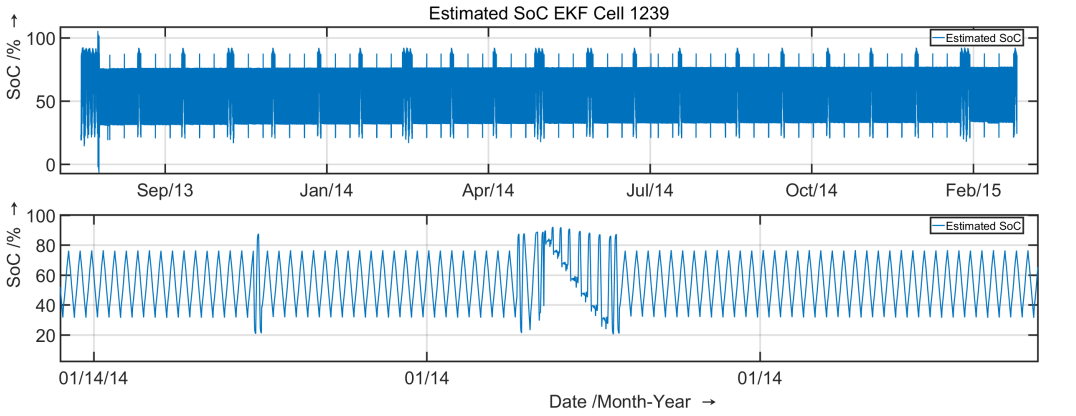


Figure 3.14: Estimated SoC for cell 1239

Temperature during the load cycle was set to 20°C, while for RTPs we have 4 different temperatures see section 2.2.1. We noticed that there might be slight drift in the temperature, calculated mean for load cycles is 20.3°C.

We can see from Figure 3.15 that even during operation errors in SoC calculations may occur, and that some implausible values in SoC can be expected. This can be related to systematic measurement error. Figure 3.15 shows the calculated and estimated SoC, only for load cycles, now plotted over each other. We see that SoC tracking is quite good.

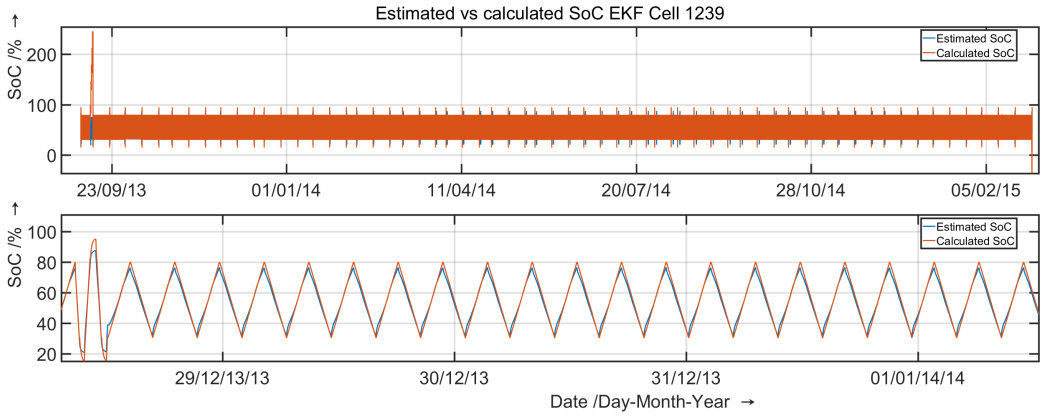


Figure 3.15: Estimated vs Calculated SoC for cell 1239 load cycles

Figure 3.16 shows, the estimated and calculated SoC for the driving cycles, again for single driving cycles, as well coupled together. The SoC uncertainty is slightly greater compared to load cycles, reason for that can be found in fact that we used same set of parameters $P^{+}_{\tilde{x},0}$, Q_w,R as used for load cycles. But now we have different current stimulation on the cell. This was one thing we encountered during the EKF tuning, we could find one parameters set which would work very well for one driving profile, but we would have bigger deviation in SoC estimation for different driving profile. So we could say that, we need different parameters sets for EVs and HEVs. Of course it would be possible to find one parameter set for both, but we would have to take higher SoC deviation into account.

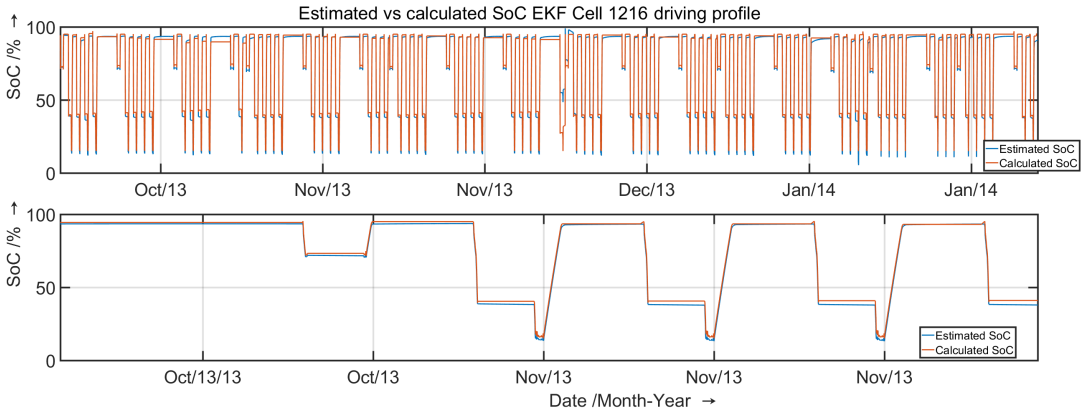


Figure 3.16: Estimated vs calculated SoC for cell 1126 (taken on 03-2014,04-2014,08-2014)

Figure 3.17 shows estimated SoC for coupled driving cycles together with RTPs.

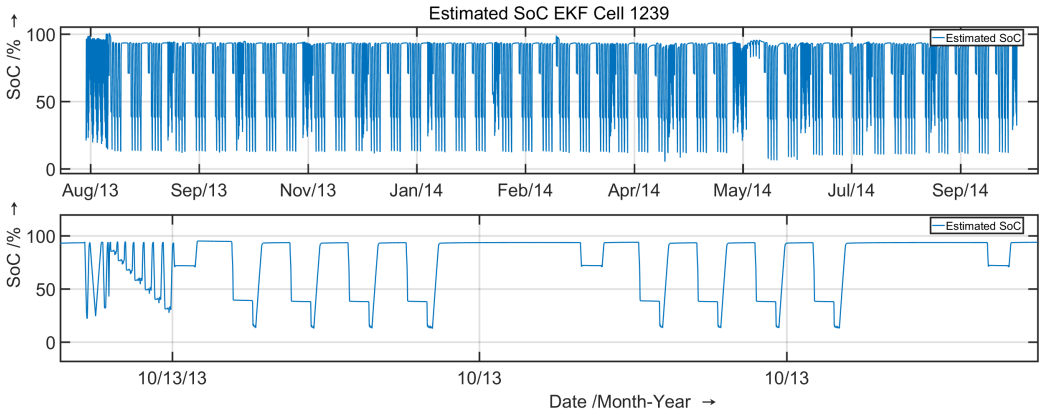


Figure 3.17: Estimated SoC for cell 1126 and RTPs

In addition we performed evaluations on new Panasonic cells, of same dimensions (18650 cells) with a rated capacity 3.25Ah. Measurements were obtaind within second project, the cells went under a similar cycling procedure, with some small adjustment in DoE. Again we have RTPs, load profiles, and different driving profiles. We have noticed that the EKF algorithm was much more robust, in terms of temperature dependency, and adjusting the OCV/SoC curves and RC-parameters for single cells, as well as finding the right parameter set for the EKF. One of the reasons for this may lie in the bigger cell capacity, but it can also be related to the different cell chemistry. Here we used slightly different EKF parameters (see Chapter 6), but again only one

parameter set for different cycling profiles. Figure 3.18 shows a load cycle coupled together for cell 070, without RTPs in-between, which yields satisfying results.

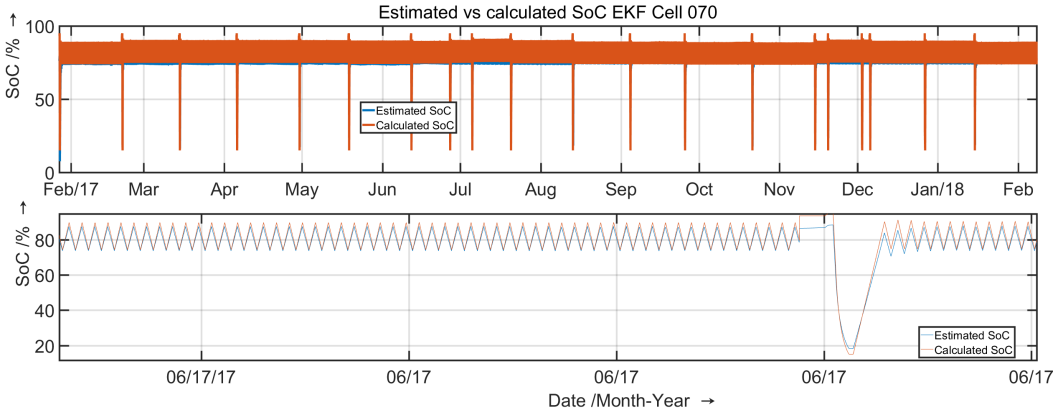


Figure 3.18: Estimated vs. calculated SoC for cell 070

As we can see from Figure 3.19, for smaller dSoC we have larger deviation in SoC tracking. Again we have to notice, that we used one parameter set for EKF, as well as one OCV/SoC curve, for different cycle profiles.

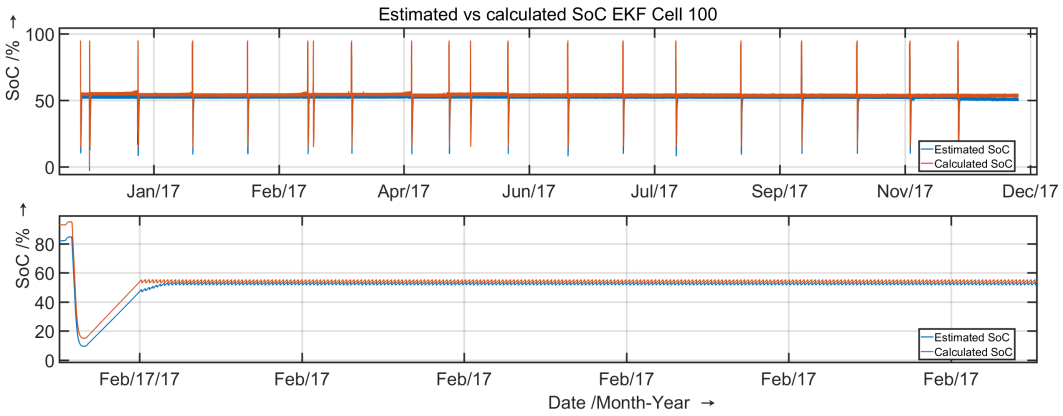


Figure 3.19: Estimated vs. calculated SoC for cell 100

Conclusion

Some time was spend adjusting the initial parameters $P^+_{\tilde{x},0}$, Q_w,R in order to obtain satisfactory results. Some example parameter sets are shown in the Appendix A. The aim was to find a single parameter set, which would be acceptable in terms of results for all data-sets. This can be done by hand which is the case here, or in an optimization program loop [24]. There is also some literature on the adaptive version of EKF, but the experiences using this methods are mixed, since these seem to work very well for some application, but may have problems with others. Theoretically, we could find an expression for $P^+_{\tilde{x},0}$ using the initial voltage uncertainty and the SoC vs. OCV curve uncertainty, the value of the Q_w could be derived from the variance of the current-signal, and R could be derived based on the voltage-signal variance. However our EKF assumptions described in section 3.2 are *not* fully fulfilled. That is why, derived quantities can be only used as a starting point in the parameter adjustments.

At last we found one parameter set that was partly satisfactory, this was used in above shown results. We saw that in some cases with the low temperatures, and certain data-sets, the SoC tracking was slightly off. But we also saw that we can expect some uncertainty in the calculated SoC, that is one of the reason we did not plot boundarys, as this might lead to false conclusions. As expected, using the test data collected at warmer temperatures did yield better estimation results, while for the data sets collected at low temperatures, in order to get a satisfactory results we had to use the appropriate SoC/OCV curve, and change the initial parameter sets. We witnessed, that using the same SoC/OCV curve and fitted parameters R_0,R_1,R_2,C_1,C_2 for different cells, did not provide adequate results.

3.3.2 EKF for ARX-model

In this section the results for estimated SoC, obtained with EKF for ARX-model are shown. The data-sets are the same as shown in section 3.3.1. Again we aimed to find a single satisfactory set of parameters $P^{+}_{\tilde{x},0}$, Q_w,R, these are shown in the Appendix A. Figure 3.20 shows fitted and measured voltage-response using the ARX-based EKF.

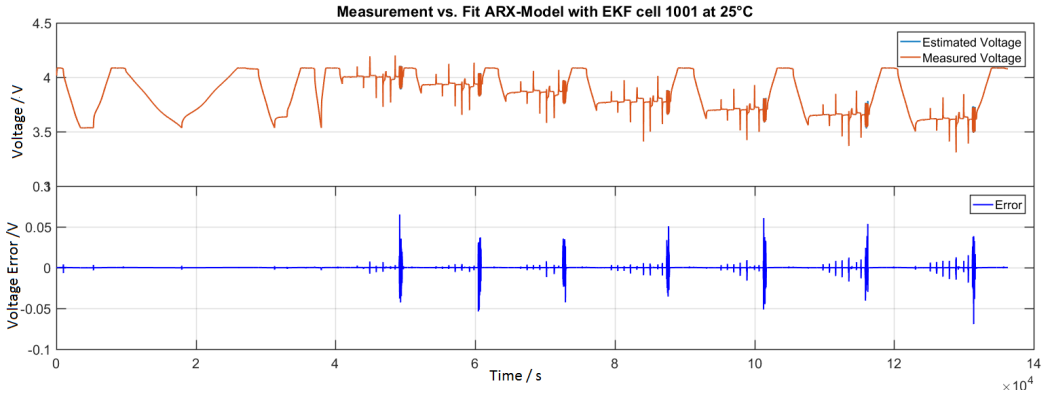


Figure 3.20: Estimated vs Measured voltage using ARX-based EKF cell 1001

Next we show the estimated and calculated SoC trend for cell 1001 at 25°C, as well as the SoC deviation between those two.

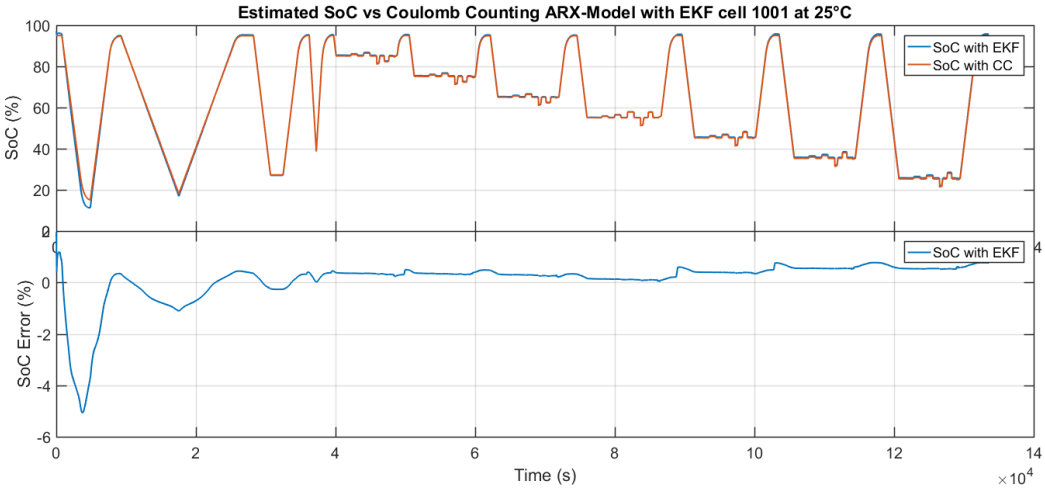


Figure 3.21: Estimated vs calculated SoC using ARX-based EKF cell 1001

In Figure 3.22 we see the estimated SoC compared with calculated SoC, for the

single load cycle on cell 1239, at the beginning of cell testing 04.09.2013. Figure 3.23 shows the SoC course for the same cell, at a different time 04.02.2015.

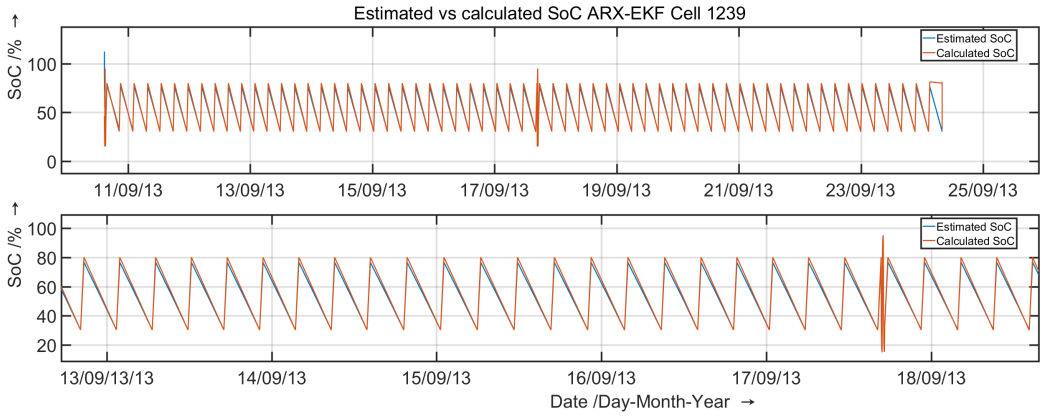


Figure 3.22: Estimated vs calculated SoC using ARX-based EKF cell 1239 load cycle

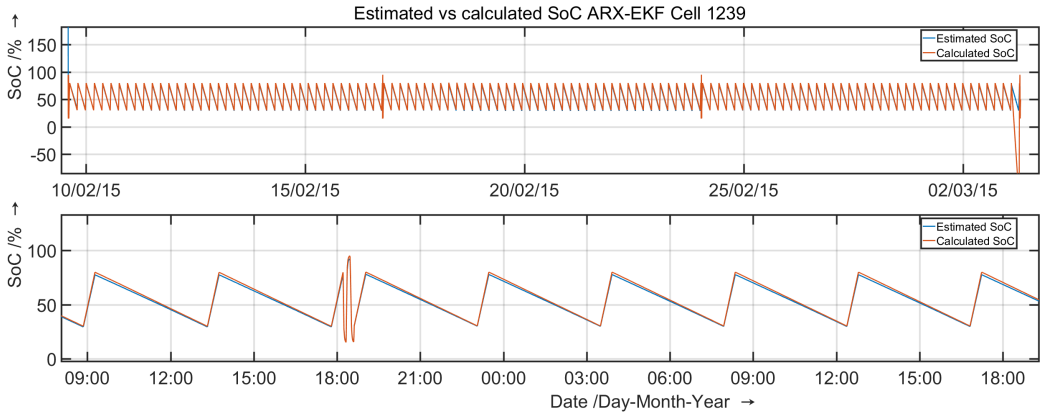


Figure 3.23: Estimated vs Measured SoC using ARX-based EKF cell 1239 load cycle

Further we see the coupled data- set, where we estimated the SoC for RTPs and load cycles together, using one set of parameters for $P^{+}_{\tilde{x},0}$, Q_w,R. Values for parameters a_1,a_2,b_0,b_1,b_2 are randomly picked from a spreadsheet, that we obtained after fitting all cells under different aging stages and different conditions, such as temperature, SoC etc. We can see that even for long-term estimation the SoC progress is stable, we observe some peaks, but need to say that we did not spend as much time tuning the EKF for ARX-model as we did for RC-model.

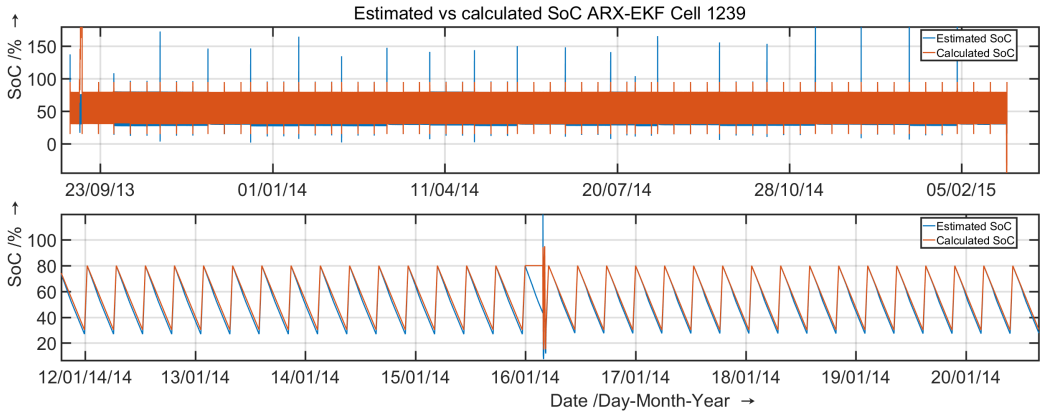


Figure 3.24: Estimated vs calculated SoC using ARX-based EKF cell 1239 load cycle

Next we show SoC course for the driving cycle, same as in section 3.3.1. Again the same set of EKF parameters was used as for load cycles and RTPs, nevertheless the SoC progress may be taken as trustworthy, except some peaks in the estimation, which could be dealt with SoC limits and extend filter tuning.

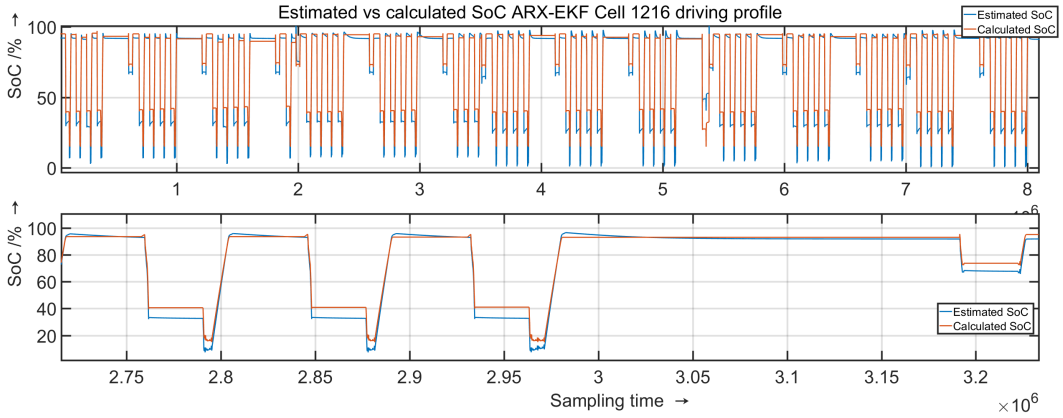


Figure 3.25: Estimated vs calculated SoC using ARX-based EKF cell 1216 load cycle

Conclusion

In this section, one way for cell modeling using the ARX model combined with EKF algorithm is presented. The approach for estimating the SoC for Li-ion cells are presented as well as obtained results using this method. It is shown that the ARX-model in combination with EKF algorithm provides a decent performance and robustness in SoC estimation, and it can significantly boost the accuracy of the model output voltage. If we want to improve the convergence rate, we could adjust the sampling rate to smaller time intervals. Shorter sampling time interval leads to more iterations and more feedback, this will be favorable to the SoC convergence, on the other hand it requires a greater computing power. This approach should be applicable in real-time applications.

4 SoH Estimation

The state of health (SoH) is the measured condition of the cell at the moment, compared to a fresh cell. Depending on the definition, it summarizes as capacity retention and possibly increase of the inner resistance. Depending on the cells usage in a battery pack, the cells will age and it will come to performance degradation. Eventually, the cell will reach the state where it no longer meets the performance requirements, this might be considered as the end of life of the cell or battery pack. In comparison to other battery technologies, Li-ion batteries do not undergo so called 'sudden-death' (this can happen to lead-acid batteries) [9]. Rather they go through a gradual decrease in performance over their life. According to literature, battery packs for the xEV[1] application are expected to operate between 10 and 15 years, or between 20000 and 30000 cycles, respectively [2]. One way to fulfill these expectations is through over-sizing of battery packs by designing, this may lead to unwanted cost increase, storage space decrease and increased vehicle weight.

We need to make sure that we have the needed knowledge regarding the cells degradation, between the beginning and the end of life, in order to be able to accurately calculate the SoC, available energy and available power. It is important to note that 'normal' aging may be one cause of failure, but there can be many more failures, such as design faults, impurities in the materials during manufacturing process (such cells may appear normal at the beginning of life but later degrade rapidly), abuse etc. The method described in this chapter, can not prevent these conditions but it may help us track the SoH of the cell and maximize the safety if the cells are near there End of Life (EoL).

[1]Here we refer to any vehicle having electric drive-train components (HEV, PHEV, EV)

4.1 Approach to SoH estimation

The most used term in literature for cell aging is SoH. One indicator of SoH, is capacity loss. During our cell test we determine the capacity by the RTPs under different temperatures and discharge rates, these we use as a reference for our capacity estimation methods. For better comparison of capacity loss, we can calculate the relative cell capacity as Eq. 4.1 [17] - [12]:

$$C_{relative} = \frac{C_n}{C_{n0}} \tag{4.1}$$

Where C_n is the capacity of the current cells, C_{n0} is the initial capacity of cell before cycle test, and $C_{relative}$ is the relative capacity of current cells. Generally we consider that the cell has reached the EoL when its capacity fades to 80% of its initial value. There are different methods to estimate the aging level, the following Table 4.1 is adapted from [17] and shows us cells aging estimation methods and their performances:

	Adaptation	Precision	Real-time	Prediction	Operate without data
Physical measurement	Excellent	Excellent	Very poor	Very poor	Excellent
equivalent circuit model	Very poor	Fair	Good	Fair	Good
electro-chemical model	Very poor	Excellent	Fair	Fair	Fair
Semi-empirical model	Very poor	Good	Fair	Good	Poor
Analiytical model	Very poor	Good	Poor	Poor	Poor
Statistical model	Fair	Good	Good	Good	Very poor

Table 4.1: Methods for cells aging estimation, performance comparison [17]

Following Figure 4.1 shows capacity evolution during the cell aging. We calculate the points for RTPs and interpolate between them. This was done for all test cells, Figure 4.1 shows us two random cells.

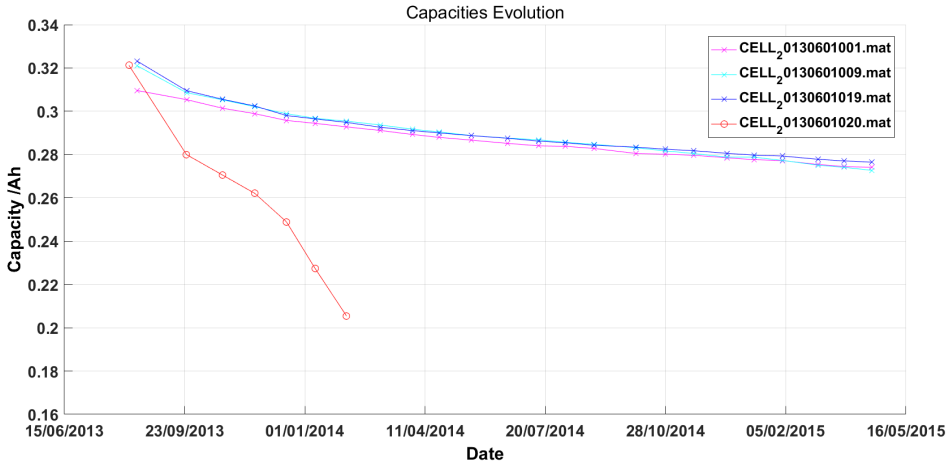


Figure 4.1: Capacities evolution for cells 1001, 1009, 1019, 1020

As the cell ages, beside the capacity degradation other cell model parameters will change as well see chapter 2.2 such as cells OCV relationship due to side reactions and capacity lost [24]; however, we do not track all of these, we are interested in the internal resistance and the total capacity because these have the leading impact on the BMS (Battery management system) performance. Degradation of Li-ion cells is a complex process, that depends strongly on the chemistry used in the cell production, but we rely on the fact that dominant outcomes can be well described by changes in cell total capacity and internal resistance. In chapter 3 it is shown that using EKF we are able to estimate the fast time-varying states of the cell, in this chapter the possibility of using the EKF in order to estimate the slow time-varying parameter quantities is examined. It will be shown that that it is possible to estimate the internal resistance with a reasonable grade of accuracy using these methods, on the other hand finding a good total capacity estimate might be a challenging task.

4.1.1 Internal resistance estimation

To estimate internal resistance we used the model output equation together with the random walk model, and formulate it as follows:

$$\begin{aligned} R_{k+1} &= R_k + r_k \\ y_k &= OCV(SoC_k) - U_{1,k} - U_{2,k} - i_k R_k + e_k. \end{aligned} \tag{4.2}$$

R_k represents the cell internal resistance, we can see that the R_k is modeled as constant value. In order to allow the adaptation of R_k we add a fictitious noise process r_k. Again y_k is the estimate of cells voltage, i_k is the cell current, $U_{1,k}$ and $U_{2,k}$ are voltage drop off on the parallel circuit, and e_k is the measurement error. We use an SoC estimation described in Chapter 3 and we apply an EKF to this model in order to estimate the cell internal resistance. We compare the prediction of y_k with the measured cell voltage and use the difference to adapt R_{k+1} [26].

4.1.2 Nominal capacity estimation

Again we use the model to create a capacity estimation, this we formulate as following:

$$\begin{aligned} C_{nom,k+1} &= C_{nom,k} + r_k \\ 0 &= SoC_k - SoC_{k+1} + \frac{-\eta \Delta t}{C_{nom} * 3600} + e_k \end{aligned} \tag{4.3}$$

We compare the right-hand side of the second equation, and compare it to zero during the computation, we used the difference to update the nominal capacity estimation. Again we need good estimation of the present and previous SoC [26].

4.2 Implementing DEKF using the cell model

In order to ensure a correct SoH estimation of a cell/battery pack, it is necessary to obtain information about the cell SoC and the cell resistance as well as, ideally, about possible changes in cell dynamics. For this purpose, we use a Kalman filter again, which can estimate the model parameters based on measurements with low dynamics.

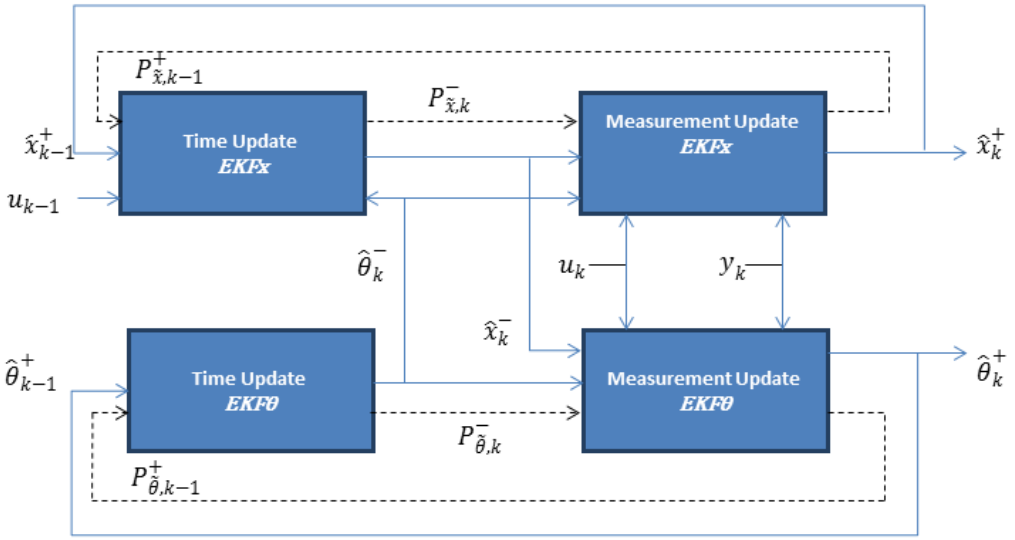


Figure 4.2: Parameters and state estimation in the form of a cascaded system

In order to map the long-term aging estimation and short-term state estimation. We cross two systems as shown in Figure 4.2 [25]. Two separated extended Kalman filters are used, to estimate the state on the one hand and the parameters set on the other hand. Cell states such as SoC, diffusion magnitude and hysteresis, are changing rapidly, while others, such as capacity, internal resistance and SoH, are changing very slowly. This is the reason why we used two different sampling times, when implementing the DEKF. One sampling time was used for the state and inner resistance estimate, and the other ten times slower, for the capacity estimate. In Figure 4.2 we see the dual estimation process, where EKF_x represents the state estimate, and the EKF_Θ is the parameter estimate.

As already mentioned, using the extended Kalman filter, it is possible to estimate the parameters of a dynamic system using noisy input/output measurements if the state is known. Since the SoH of a battery cell is summarized by the cell model

parameters values, this approach seems very promising [24]. We assume that the parameter values change very slowly, so we model the current parameter vector as equal to the previous parameter vector plus a small perturbation r_k. This means the unknown parameters will be modeled as random walk Eq. 4.4.

$$\Theta_k = \Theta_{k-1} + r_{k-1} \tag{4.4}$$

However, we do not assume small zero-mean white-noise error for input r_k, this is used only to force input for the slow change in the parameter values. Instead we use the covariance matrix $Q_{\tilde{r}}$ to set how fast we believe the parameter values may change, and for the Kalman filter to adjusts how fast the parameter value estimates are updated.

$$d_k = g(x_k, u_k, \Theta_k, e_k) \tag{4.5}$$

We have the same six steps of 'sequential-probabilistic inference' that we introduce in section 3.2.

EKF_Θ 1a: *Parameter prediction time update.*

$$\hat{\Theta}_k^- = \hat{\Theta}_{k-1}^+ \tag{4.6}$$

The predicted parameter vector for this time step is equal to the estimated parameter vector calculated at the end of the previous time step. We see that the changes in the parameter vector due to the degradation are detected only via the measurement feedback since the prediction step does not predict any deterioration of the values.

EKF_Θ 1b: *Error covariance time update.*

$$P_{\tilde{\theta},k}^- = P_{\tilde{\theta},k-1}^+ + Q_{\tilde{r}} \tag{4.7}$$

After assuming that the zero-mean fictional noise is uncorrelated with the parameter prediction error [24], mathematically, the time update parameter error covariance is equal to the covariance before the time update, but with additional uncertainty due to the notional noise r_k, which is assumed to drive the change of the parameter values.

EKF_Θ 1c: *Output prediction*

$$\hat{d}_k = g(x_k, u_k, \Theta, e_k) \approx g(x_k, u_k, , \hat{\Theta}_k^-, \bar{e}_k) \tag{4.8}$$

The measurable system output is based on the parameter model prediction of Eq. 4.8 and EKF assumption 1. see section 3.2

EKF_Θ 2a: *Estimator gain matrix*

$$L_k^\Theta = P_{\tilde{\Theta},k}^- (\hat{C}_k^\Theta)^T [\hat{C}_k^\Theta P_{\tilde{\Theta},k}^- (\hat{C}_k^\Theta)^T + R_\Theta]^{-1} \tag{4.9}$$

We obtain the Kalman gain matrix by linearizing the output d_k with the help of the Taylor series expansion, and the EKF assumption 2. see section 3.2. Here we have to be very careful when computing $\hat{C}_k^{\Theta}$. This requires a total differential expansion, we use:

$$\begin{aligned}
\hat{C}_k^{\Theta} &= \left.\frac{dg(\hat{x}_k^-, u_k, \Theta)}{d\Theta}\right|_{\Theta=\hat{\Theta}_k^-} \\
\frac{dg(\hat{x}_k^-, u_k, \Theta)}{d\Theta} &= \frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial \Theta} + \frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial \hat{x}_k^-}\frac{d\hat{x}_k^-}{d\Theta} \\
\frac{d\hat{x}_k^-}{d\Theta} &= \frac{\partial f(\hat{x}_{k-1}^+, u_{k-1}, \Theta)}{\partial \Theta} + \frac{\partial f(\hat{x}_{k-1}^+, u_{k-1}, \Theta)}{\partial \hat{x}_{k-1}^+}\frac{d\hat{x}_{k-1}^+}{d\Theta} \\
\frac{d\hat{x}_{k-1}^+}{d\Theta} &= \frac{d\hat{x}_{k-1}^-}{d\Theta} - L_{k-1}^x \frac{dg(\hat{x}_{k-1}^-, u_{k-1}, \Theta)}{d\Theta}
\end{aligned} \tag{4.10}$$

If we want to obtain $\hat{C}_k^{\Theta}$, we need to calculate all of the partial derivatives in order to find the total derivative. We initialize total derivatives $\frac{dg}{d\Theta}, \frac{d\hat{x}_{k-1}^-}{d\Theta}$ to be zero at $k = 0$ and compute recursively[24].

EKF_Θ 2b: Parameter estimate measurement update

The parameter estimate is updated by the forecast of the estimator gain and the predicted output error.

$$\hat{\Theta}_k^+ = \hat{\Theta}_k^- + L_k^{\Theta}(d_k - \hat{d}_k) \tag{4.11}$$

EKF_Θ 2c: Error covariance measurement update

Finally, we calculate the parameter estimation error covariance as:

$$P_{\tilde{\Theta},k}^+ = (I - L_k^{\Theta} C_k^{\Theta}) P_{\tilde{\Theta},k}^- \tag{4.12}$$

Note: the superscript notations Θ is used to keep parameter estimation matrices different from state estimation matrices. DEKF is summarized in Appendix A.

In order to summarize the cell/battery pack SoH we need the total available capacity and the internal resistance of the individual cells. An accurate estimate of the total capacity and internal resistances makes it possible to reliably calculate the total energy and available power for the battery pack over its lifetime.

4.2.1 DEKF for Second order RC-model

Reason for DEKF implementation lies in the fact, that in order to determine the SoH we need to somehow estimate the cells state and the cell parameters. That is why we use one EKF for state estimation and a second EKF for parameter estimation, with their signals mixed. Our first idea was to implement a full DEKF, where we could track the progress of all six parameters R_0,R_1,R_2,C_1,C_2,C_{nom}. This was done, but after some testing the results were inconclusive, in terms that some parameter progress was plausible, and others not as much. As we experienced with parameter tuning for EKF, that it was time consuming task, here was clear from start that we need much more time in order to get the satisfactory results, on the other hand these are computationally intensive methods. For this reason we decided that the full cell parameters estimation is not necessary. Figure 4.3 shows DEKF implementation to determine cell state, cell capacity, and cell internal resistance using *Matlab Simulink*.

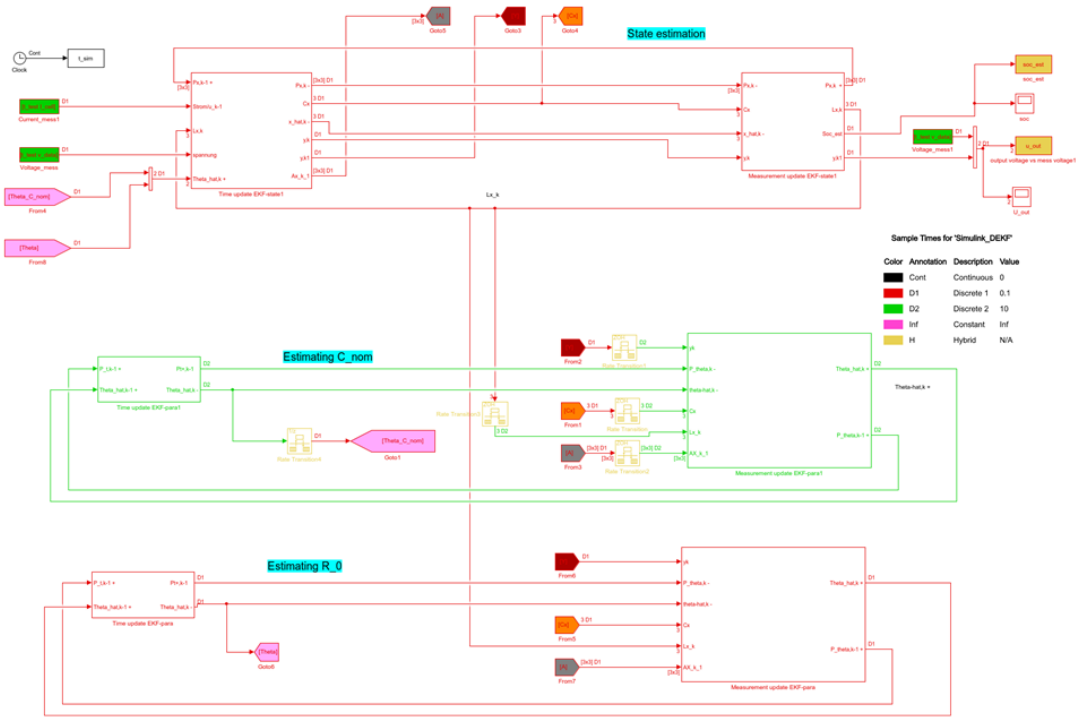


Figure 4.3: *Simulink* model of DEKF

Different approaches have been conducted, with the sampling rate, starting with the same sampling rate for booth EKF. Obtained results were inconclusive, either state estimation or the parameter estimation was good, it was hard to get both estimations to be plausible. Here we speak in terms of adjusting the ratio of parameters Q_w and R in a way that the system may be influenced to put more trust in the measured data or the model data [2]. It is known that if we compare the state and parameters changes, it is obvious that the state changes are much faster than the parameters changes. The change in SoC may be visible in a matter of minutes, while SoH change should be visible in matter of months. As shown in Figure 4.3 the sampling rate for state estimation and internal resistance estimation is 0.1, and for nominal capacity estimation sampling rate is 10. We have summarized the method in Appendix A.

4.3 DEKF Results and Discussions

In section 3.3 we have shown examples of the state estimation using 18650 cells. We have also described the different approaches to cell testing and characterization RTP, load cycle and driving cycle. We use the same data sets as in chapter 3. Here, we focus on dual estimation, state and parameters, mainly internal resistance and nominal capacity. Again the tuning of the parameters was done by hand, we also tested a new parameter tuning method like normalized innovation square method. This seems to work very well for some applications, but in our case it was only for short data sets possibly to follow χ^2 distribution. It can be conclude that tuning of Kalman filters is a very important task, and that adjustments need to conducted if faced with different cell properties.

4.3.1 DEKF for RC model

In this section, obtained results for SoC, capacity and internal resistance estimation at the different aging levels, using DEKF are presented. Firstly, we show the SoC estimation compared with the calculated SoC, same as in section 3.3.1. Then we move on to the the capacity estimation, which we compare to the calculated capacity for RTPs at different aging levels. Finally, the progress of the cells internal resistance estimation is shown.

It should be noted that using DEKF the requirements of initial covariance matrices was more critical and sensitive compared to the EKF, especially when starting values of several connected parameters, SoC and cell capacity, were uncertain. Here, we observe filter tracking the capacity and SoC to the plausible values despite poor initial values of model parameters and OCV(SoC) curve. We also found that the DEKF was

[2] Choosing larger R compared to Q leads to more trust in the model and vice versa.

able to plausibly estimate the internal resistance progress, and quickly converges to the actual value.

Please note that the temperature dependence for different aging levels, as well as for different RTPs was not taken into account. After comparing the estimated quantities with the calculated one, we may conclude that the filter is converging to the correct value. We do not adjust our temperature dependent parameters to OCV/SoC curve, model parameters, initial capacity and internal resistance. These were set for one temperature -mostly the set temperature of a load cycle or driving cycle. Unfortunately, we have no way of absolutely validating the estimated quantities, but the advantages of DEKF over the EKF are clearly illustrated, especially for long-term estimations.

4.3.1.1 SoC estimation using DEKF

In this section we present the results obtained from DEKF, for SoC estimation. We also show the accuracy for the cell's terminal voltage estimation, this has improved especially at later aging levels, due to the fact that the model parameters are updated. First, we shows the estimation values for a RTP as well as the error between the estimated and measured/calculated quantity. We see that the error between estimated and measured voltage is small.

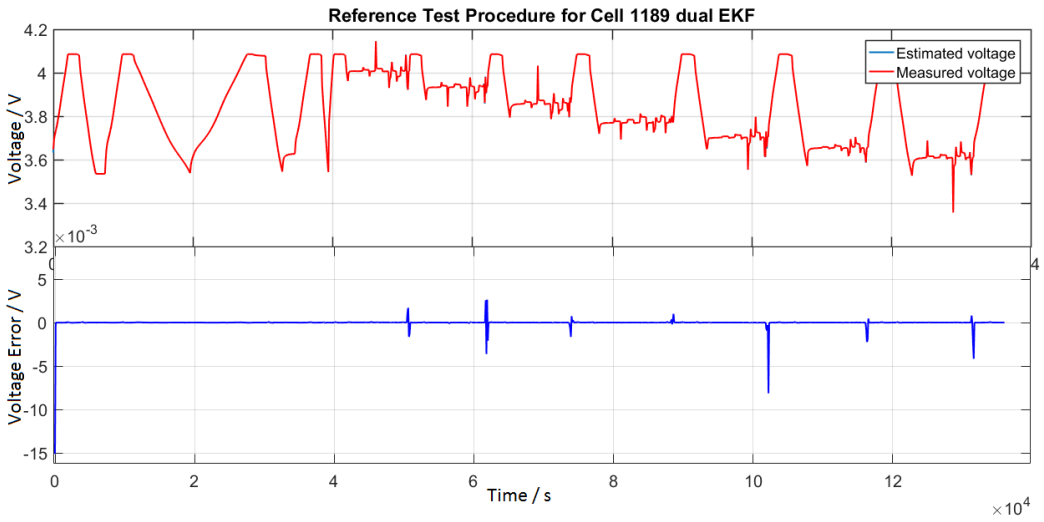


Figure 4.4: Voltage for RTP cell 1189

Next we show the SoC progress for a single load cycle. Here we expect slightly better SoC estimation results, especially at later aging levels, compared to the EKF

due to the fact, that we adapt the new values for the internal resistance and nominal capacity. The important reason in order for SoC estimation not to drift from expected value is the nominal capacity, as the cell ages the cells nominal capacity will reduce. This means that SoC level for new cell and aged cell will not be at the same level if the nominal capacity reduces.

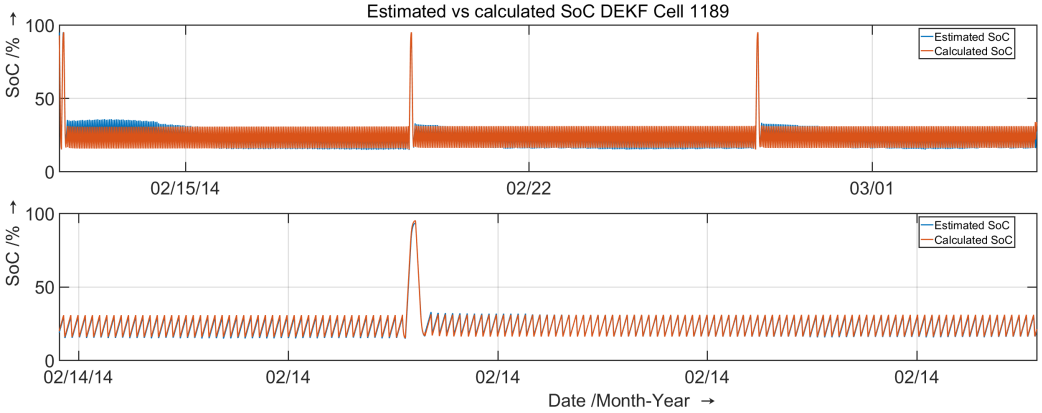


Figure 4.5: Estimated vs calculated SoC for load cycle

Figure 4.6 shows calculated vs. estimated SoC for load cycles at the different aging levels plotted over each other. Here, we see that the estimated SoC corresponds to the calculated value and that we can be satisfied with the obtained results.

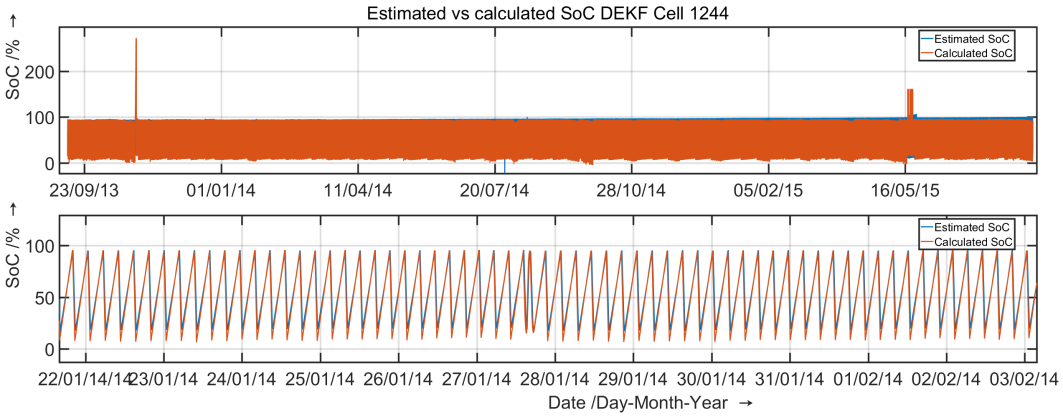


Figure 4.6: Estimated vs calculated SoC for load cycle at different aging levels

We conducted the experiment on different profiles, Figure 4.7 shows the SoC

progress for a driving cycle.

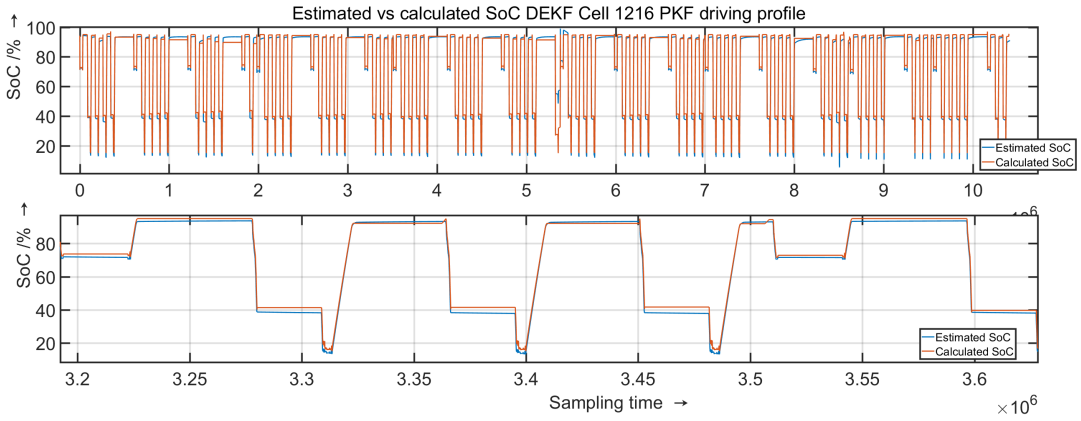


Figure 4.7: Estimated vs calculated SoC for a driving cycle at different aging levels.

Again one set of EKF parameters is used for load cycles as well for RTPs and different driving profiles, in Figure below we see the estimated SoC for driving cycle coupled together with RTP' for different aging levels.

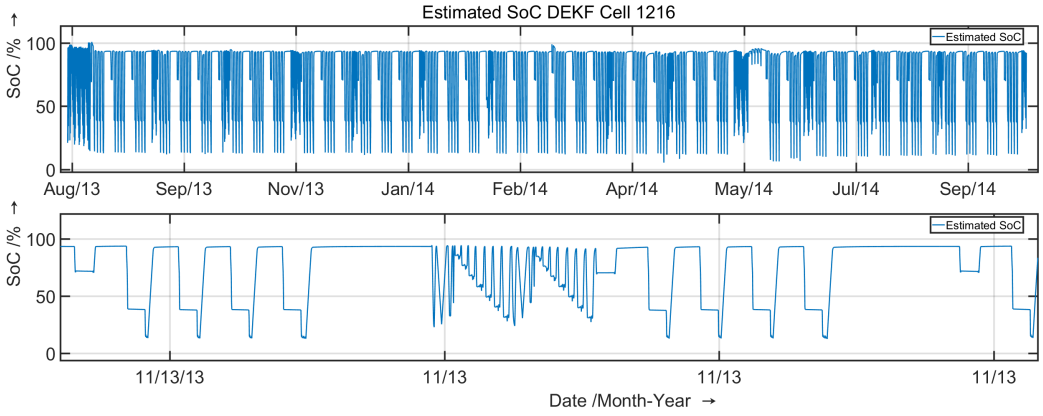


Figure 4.8: Estimated vs calculated SoC for load cycle at different aging levels.

4.3.1.2 Capacity estimation using DEKF

In this section we show the capacity degradation at different aging levels. We compare the estimated with calculated capacity, also conducted at different aging levels. It should be noted that the capacity was calculated for RTPs at 25°C, and the data used for estimating the capacity vary in the temperature as described in section 2.2.1 As mentioned earlier, the nominal capacity is strongly dependent on the temperature, thus jumps in estimated capacity may be related to the temperature.

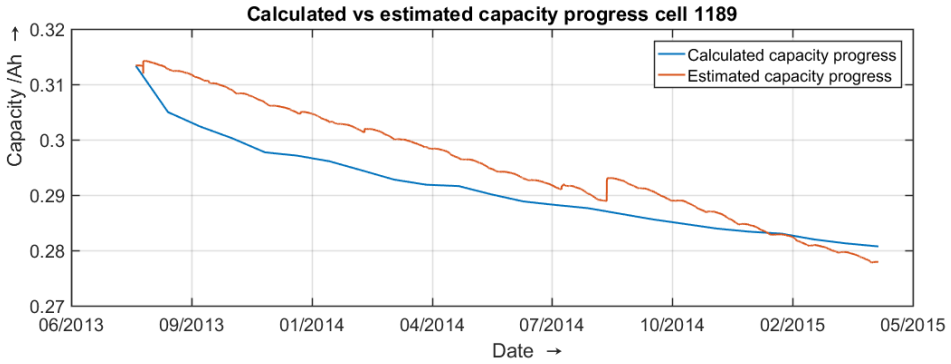


Figure 4.9: Estimated vs measured capacity at different aging levels cell 1189

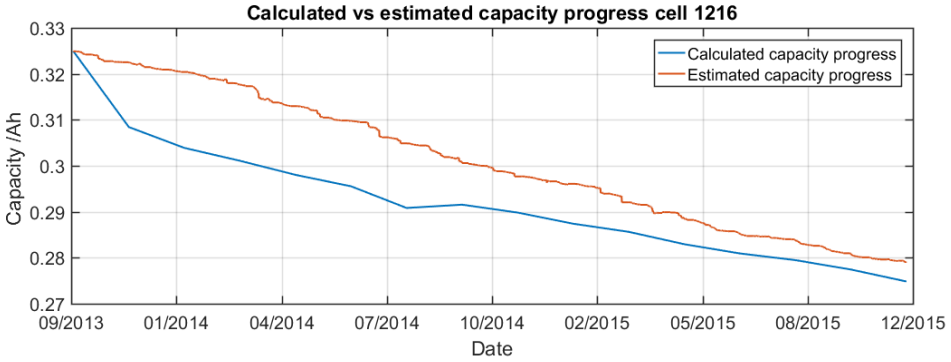


Figure 4.10: Estimated vs measured capacity at different aging levels cell 1126 driving cycle

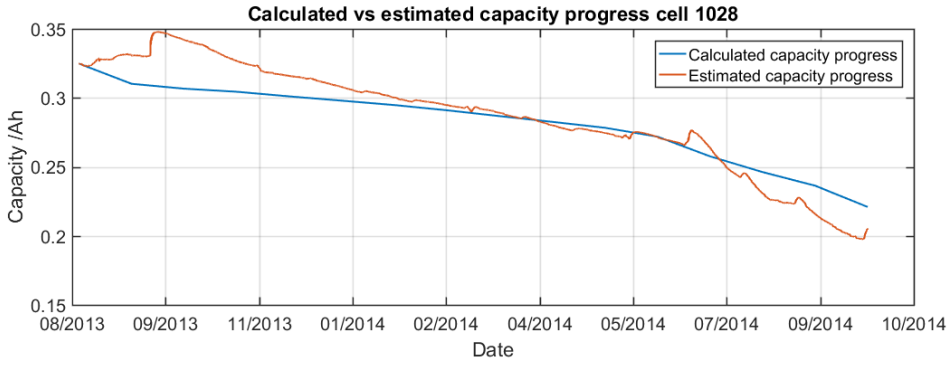


Figure 4.11: Estimated vs measured capacity at different aging levels cell 1028

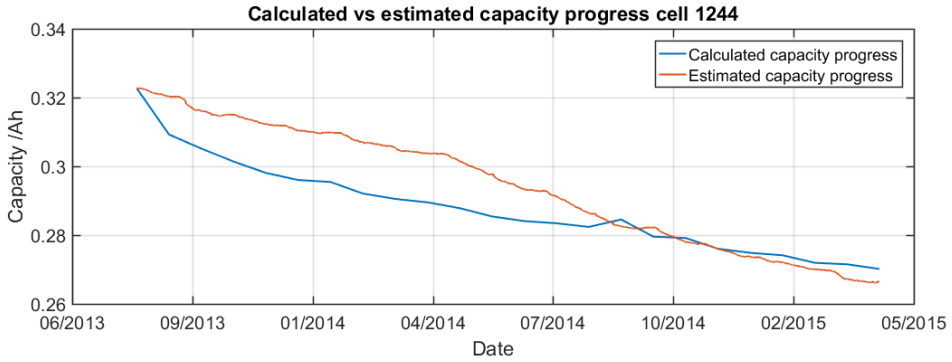


Figure 4.12: Estimated vs measured capacity at different aging levels cell 1244

4.3.1.3 Internal resistance estimation using DEKF

In following figures we show the internal resistance estimation using the DEKF at the different aging levels. Compared with the calculated internal resistance, again the internal resistance was calculated for RTPs at 25°C, and different aging levels (Interpolated between RTPs). Again we can relate the jump in the estimation, to the temperature and more data points.

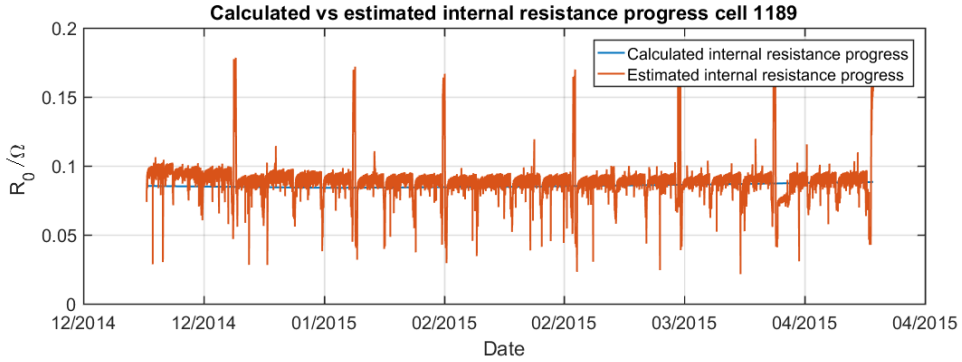


Figure 4.13: Estimated vs measured internal resistance at different aging levels cell 1189

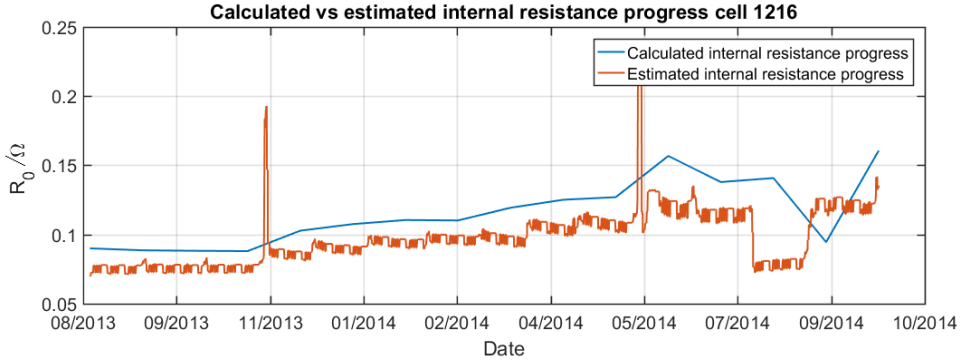


Figure 4.14: Estimated vs measured internal resistance at different aging levels cell 1216

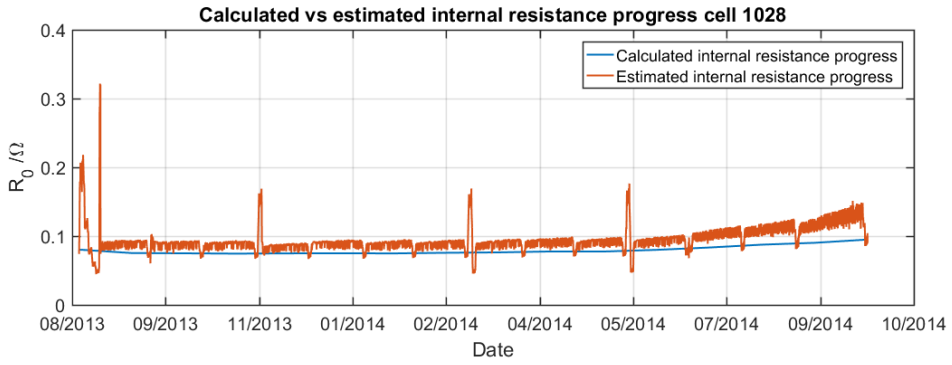


Figure 4.15: Estimated vs measured internal resistance at different aging levels cell 1216

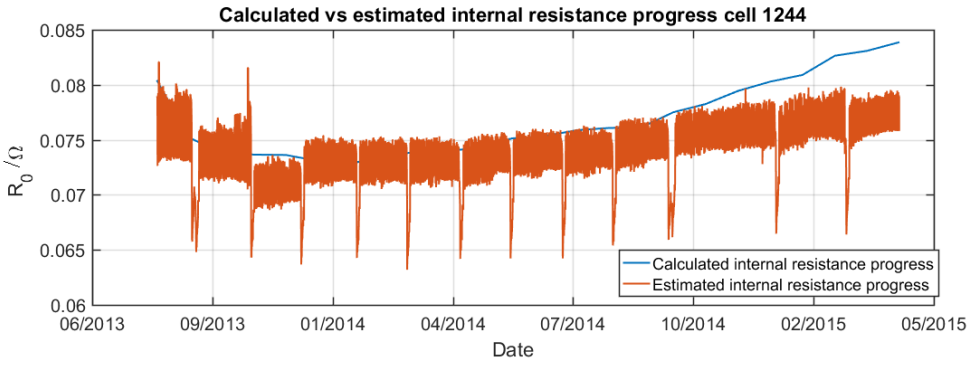


Figure 4.16: Estimated vs measured internal resistance at different aging levels cell 1028

In addition, the state estimation for new cells was accomplished, these were cycled within second project. It can be seen that the SoC progress looks trustworthy, but also that nominal capacity and internal resistance estimation compared to measurement provide satisfactory results. It has to be noted, that one OCV/SoC curve as well as RC-parameters are used for different cells, as well as for different temperatures. For some cycle profiles we need to adjust the covariance parameters that is responsible for the nominal capacity drop (see Appendix A).

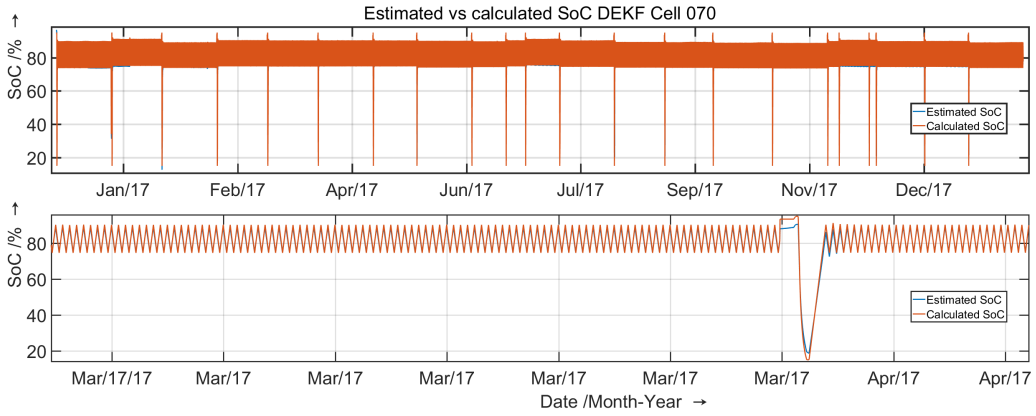


Figure 4.17: Estimated vs measured SoC at different aging levels cell 070

In Figure 4.17 we can see the comparison between estimated SoC using DEKF algorithm and calculated SoC using Coulomb counting method, for different aging levels. Figure 4.18 below shows us only estimated SoC, but now from beginning of cell testing (6.12.2016) till end (19.04.2018), test data is coupled together load cycles and different RTPs. In top frame we see the whole progress, in bottom frame a zoom-in, as we can see one RTP peak is out off plausible region above 100% SoC. This is related to the RTP temperature. As said earlier we did not adapt OCV/SoC curve to the temperature this could be easily done.

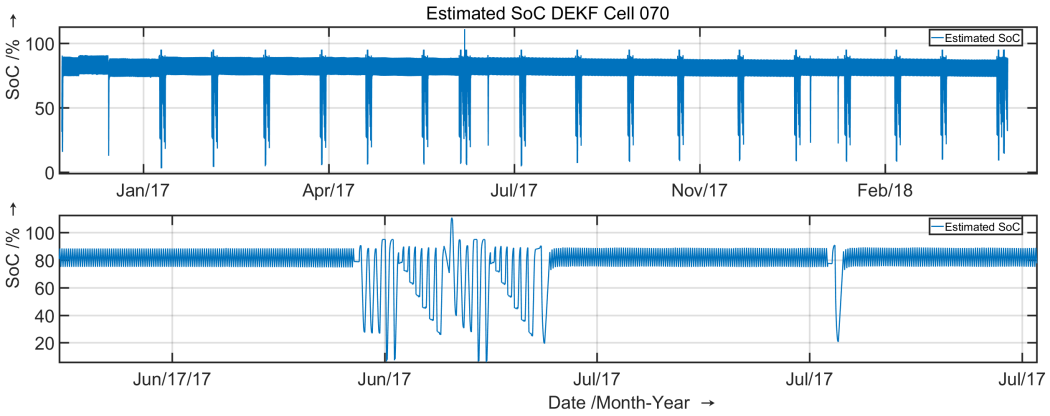


Figure 4.18: Estimated SoC at different aging levels cell 070

Further, we show nominal capacity progress estimated vs. measured. As mentioned earlier the covariance parameter may be adjusted if we think the cell degradation is more rapid or more slow. This means that for any new cell-chemistry we need a long

term testing in order to truly adapt this parameter.

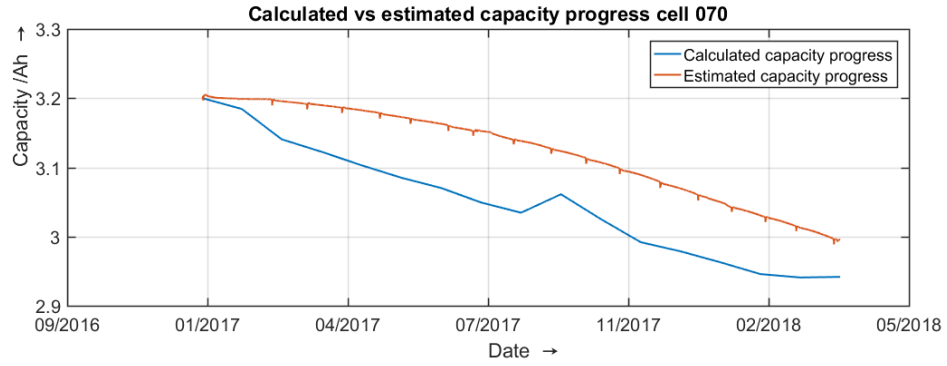


Figure 4.19: Estimated vs measured capacity at different aging levels cell 070

Figure 4.20 shows progress of the internal resistance R_0 at different aging levels.

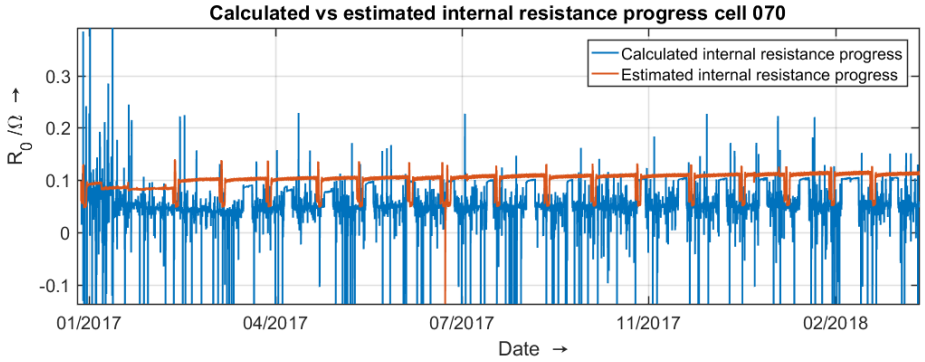


Figure 4.20: Estimated vs measured internal resistance at different aging levels cell 070

Conclusion

In this chapter we have shown, that the DEKF applications may be used in order to estimate the SOC and SOH of the cell, combined with the RC-cell-model. According to literature there are couple of different approaches in order to improve some disadvantages of the DEKF.

We have shown that the tracking of the cell degradation in terms of capacity fade and internal resistance increase, during the cells aging levels, provides satisfactory results. It should be noted that requirements of initial covariance matrices was more

critical and sensitive compared to the EKF, especially when starting values of several connected parameters, SoC, cell capacity and internal resistance, were uncertain. Here, we observe the filter tracking the capacity and SoC to plausible values despite poor initial values. We also found that the DEKF was able to plausibly estimate the internal resistance progress, and may quickly converge to the actual value. Again, we need to mention that even the calculated data in some cases may not be reliable, particularly in cases of SoC and internal resistance calculations. Measured capacity shown in Figures 4.9 4.12 is taken for single RTPs and interpolated in between.

5 Conclusion

This thesis main contribution lies in the testing and development of two different state-space models of Li-ion cells combined with KF algorithms, in order to robustly estimate the SoC and SoH, for long and short-term scenarios. Used data as well as the obtained RC- and ARX model parameters are examined, in order to find a correlation between the behavior of the model and cell aging levels. Unfortunately, the clear correlation could not be concluded. Nevertheless the model parameter dependencies on different temperatures, SoC-levels and different aging stages is presented. In order to test the model parameter dependencies between each other, and to see if the model will keep the same behavior and accuracy, the experiment with one degree of freedom was conducted. The parameter linked to internal resistance (R_0 and b_0) are identified together with the rest of the model parameters and alone, where the rest of the parameters are kept constant at different aging stages. Although, the parameter dependency between themselves (when identifying all five parameters together) is clear, a distinct difference could not be noticed. Small changes in the parameter progress are observed, but the magnitude is approximately the same in both cases. The experiment was conducted for different aging levels, temperature and SoC stages.

The implementation process started with the state EKF algorithm, obtained results are compared with the calculated states for different test and driving profiles as well as different aging stages. It can be concluded that with suitable cell data, the robust and stable offline SoC estimation for Li-ion cells using EKF is possible, and that the estimation accuracy of the EKF depends on finding the proper temperature dependent OCV / SoC curve. Additionally it was noticed, that cell nominal capacity plays an important role for these algorithms, due to the systematic measurement error of current.

Next, for the DEKF implementation again obtained results are presented and compared to the references for different profiles and aging stages. It was clear that due to the fact that DEKF was able to adapt to the parameter variations, it outperforms the EKF especially for long-term scenarios, where parameter variations are noticeable. Choosing the correct parameter set for Q and R was challenging task as it was not know whether to put more trust into model or into data. It can be noted that the ratio of Q to R need to be adjusted for different cycling profiles in order to obtain constant accuracy. At the end, one parameter set was found that works sufficiently for all our cycling profiles, but at the cost that the estimation state accuracy deviate. It has to be noted that without long-term cell test and capacity observation (reference),

tuning a DEKF part responsible for capacity degradation correctly would be almost impossible. After concluding different test scenarios it can be concluded that tuning a KF algorithm is a critical task with respect to the performance of cell state estimation algorithms.

Future Work

In order to extend or improve this work a list of suggestions is done: The KF algorithms are tested and developed using the simulations and offline data hence, a future validation of online state estimation using the same techniques on the micro-controller would give us an adequate overview of possible advantages and disadvantageous, especially if considering micro-controllers constraints on memory and computation power. It would also be interesting to compare the obtained results with the different and more demanding KF applications, like unscented KF and Sigma-point KF, and see if the complexity and computation cost is worthy.

Bibliography

[1] Mathworks. (2018). global optimization toolbox, 2018.

[2] Victor Agubra and Jeffrey Fergus. Lindens handbook of batteries, fourth edition. 6(4):1310–1325, mar 2013.

[3] Can Aksakal and Altug Sisman. On the compatibility of electric equivalent circuit models for enhanced flooded lead acid batteries based on electrochemical impedance spectroscopy. *Energies*, 11(1):118, jan 2018.

[4] Anthony Barré, Benjamin Deguilhem, Sébastien Grolleau, Mathias Gérard, Frédéric Suard, and Delphine Riu. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, 241:680–689, nov 2013.

[5] I. Buchmann and Cadex Electronics Inc. *Batteries in a Portable World: A Handbook on Rechargeable Batteries for Non-engineers*. Cadex Electronics, 2001.

[6] Matthias K. Scharrer Christiane Essel, Rasch Bernhard. Alterungsmodell für lithiumionenzellen 2 (alice2), projektbericht, VIRTUAL VEHICLE research center in Graz. 2018.

[7] Davide Cittanti, Alessandro Ferraris, Andrea Airale, Sabina Fiorot, Santo Scavuzzo, and Massimiliana Carello. Modeling li-ion batteries for automotive application: A trade-off between accuracy and complexity. In *2017 International Conference of Electrical and Electronic Technologies for Automotive*. IEEE, jun 2017.

[8] William R. Cluett. Principles of system identification: Theory and practice [bookshelf]. *IEEE Control Systems*, 37(2):181–184, apr 2017.

[9] Saft Batterien GmbH. Lithium-ion battery life. 2014.

[10] R. Gretsch. *Ein Beitrag zur Gestaltung der elektrischen Anlage in Kraftfahrzeugen*. Friedrich Alexander Universität, Erlangen -Nürnberg, 1979.

[11] Guifang Guo, Xiaolan Wu, Shiqiong Zhuo, Peng Xu, Gang Xu, and Binggang Cao. Prediction state of charge of ni-MH battery pack using support vector

machines for hybrid electric vehicles. In *2008 IEEE Vehicle Power and Propulsion Conference*. IEEE, sep 2008.

[12] Xuebing Han, Minggao Ouyang, Languang Lu, and Jianqiu Li. A comparative study of commercial lithium ion battery cycle life in electric vehicle: Capacity loss estimation. *Journal of Power Sources*, 268:658–669, dec 2014.

[13] Hongwen He, Rui Xiong, and Jinxin Fan. Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies*, 4(12):582–598, mar 2011.

[14] Reiner Korthauer, editor. *Handbuch Lithium-Ionen-Batterien*. Springer Berlin Heidelberg, 2013.

[15] Seongjun Lee, Jonghoon Kim, Jaemoon Lee, and B.H. Cho. State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge. *Journal of Power Sources*, 185(2):1367–1373, dec 2008.

[16] Stephan Leuthner. Übersicht zu lithium-ionen-batterien. In *Handbuch Lithium-Ionen-Batterien*, pages 13–19. Springer Berlin Heidelberg, 2013.

[17] Cheng Lin, Aihua Tang, and Wenwei Wang. A review of SOH estimation methods in lithium-ion batteries for electric vehicle applications. *Energy Procedia*, 75:1920–1925, aug 2015.

[18] Lennart Ljung. System identification: Theory for the user, 12 1999.

[19] Martin Cifrain Matthias Scharrer, Franz Pichler. Ageing model for lithium ion cells alice1. projektbericht, VIRTUAL VEHICLE research center in Graz, 12 07.2015.

[20] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, jan 2003.

[21] Kazunori Ozawa, editor. *Lithium Ion Rechargeable Batteries*. Wiley-VCH Verlag GmbH & Co. KGaA, jun 2009.

[22] Jinchun Peng, Yaobin Chen, and R. Eberhart. Battery pack state of charge estimator design using computational intelligence approaches. In *Fifteenth Annual Battery Conference on Applications and Advances (Cat. No.00TH8490)*. IEEE.

[23] Andreas Jossen Peter Keil. Aufbau und parametrierung von batteriemodellen. 2012.

[24] G.L. Plett. *Battery Management Systems, Volume II: Equivalent-Circuit Methods:*. Number Bd. 2 in Artech House power engineering series. Artech House, 2015.

[25] Gregory L. Plett. Extended kalman filtering for battery management systems of LiPB-based HEV battery packs. *Journal of Power Sources*, 134(2):252–261, aug 2004.

[26] Gregory L. Plett. Dual and joint ekf for simultaneous soc and soh estimation, in. In *Proceedings of the 21st Electric Vehicle Symposium (EVS21*, 2005.

[27] Fida Saidani, Franz X. Hutter, Rares-George Scurtu, Wolfgang Braunwarth, and Joachim N. Burghartz. Lithium-ion battery models: a comparative study and a model-based powerline communication. *Advances in Radio Science*, 15:83–91, sep 2017.

[28] Jan Philipp Schmidt. Verfahren zur charakterisierung und modellierung von lithium-ionen zellen. 2013.

[29] Yanqing Shen. Adaptive online state-of-charge determination based on neuro-controller and neural network. *Energy Conversion and Management*, 51(5):1093–1098, may 2010.

[30] Pritpal Singh, Ramana Vinjamuri, Xiquan Wang, and David Reisner. Design and implementation of a fuzzy logic-based state-of-charge meter for li-ion batteries used in portable defibrillators. *Journal of Power Sources*, 162(2):829–836, nov 2006.

[31] J. Vetter, P. Novák, M.R. Wagner, C. Veit, K.-C. Möller, J.O. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, and A. Hammouche. Ageing mechanisms in lithium-ion batteries. *Journal of Power Sources*, 147(1-2):269–281, sep 2005.

[32] J. Vetter, M. Winter, and M. Wohlfahrt-Mehrens. Seopndary batteries – Lithium rechargeable systems – Lithium-ion | aging mechanisms. In *Encyclopedia of Electrochemical Power Sources*, pages 393–403. Elsevier, 2009.

[33] Paul Voelker, Thermo Fisher Scientific, and Calif Sunnyvale. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *RD-magazine*, april 2014.

[34] Shifei Yuan, Hongjie Wu, and Chengliang Yin. State of charge estimation using the extended kalman filter for battery management systems based on the ARX battery model. *Energies*, 6(12):444–470, jan 2013.

6 Appendix A

Summary of the non − linear extended Kalman filter [24]

1: $x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$

2: $y_k = g(x_k, u_k, v_k)$

Definitions :

3: $\hat{A}_k = \left.\frac{df(x_k, u_k, w_k)}{dx_k}\right|_{x_k=\hat{x}_k^+}$ $\hat{B}_k = \left.\frac{df(x_k, u_k, w_k)}{dw}\right|_{w_k=\bar{w}_k}$

$\hat{C}_k = \left.\frac{df(x_k, u_k, v_k)}{dx_k}\right|_{x_k=\hat{x}_k^-}$ $\hat{D}_k = \left.\frac{df(x_k, u_k, v_k)}{dv}\right|_{v_k=\bar{v}_k}$

Initialization :

For $k = 0$

4: $\hat{x}_0^+ = \mathbb{E}[x_0]$

5: $P_{\tilde{x},0}^+ = \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$

Computation : **for** $k = 1, 2, ..N$

State − pred. time update

6: $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, \bar{w_{k-1}})$

7: $P_{\tilde{x},k}^- = \hat{A}_{k-1} P_{\tilde{x},k-1}^+ \hat{A}_{k-1}^T + Q_{w,k-1}$

State filter meas. update

8: $L_k = P_{\tilde{x},k}^- \hat{C}_k^T [\hat{C}_k P_{\tilde{x},k}^- \hat{C}_k^T + R]^{-1}$

9: $\tilde{y}_k = y_k - g(\hat{x}_k^-, u_k, \bar{v}_k)$

10: $\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \tilde{y}_k)$

11: $P_{\tilde{x},k}^+ = (I - L_k C_k) P_{\tilde{x},k}^-$

end for

Summary of the non-linear dual extended Kalman filter[24]

1: $x_k = f(x_{k-1}, u_{k-1}, \Theta_{k-1}, w_{k-1}), \quad \Theta_{k+1} = \Theta_k + r_k$

2: $y_k = g(x_k, u_k, \Theta_k, v_k), \quad d_k = g(x_k, u_k, \Theta_k, e_k)$

Definitions :

3: $\hat{A}_k = \left.\frac{df(x_k, u_k, w_k)}{dx_k}\right|_{x_k=\hat{x}_k^+} \quad \hat{C}_k = \left.\frac{df(x_k, u_k, v_k)}{dx_k}\right|_{x_k=\hat{x}_k^-}$

$\hat{C}_k^\Theta = \left.\frac{dg(\hat{x}_k^-, u_k, \Theta, e_k)}{d\Theta}\right|_{\Theta=\hat{\Theta}_k^-} \quad \hat{D}_k^\Theta = \left.\frac{dg(\hat{x}_k^-, u_k, \Theta, e_k)}{de_k}\right|_{\Theta=\hat{\Theta}_k^-}$

Initialization : *For* k = 0

4: $\hat{\Theta}_0^+ = \mathbb{E}[\Theta_0], \quad P_{\tilde{\Theta},0}^+ = \mathbb{E}[(\Theta_0 - \hat{\Theta}_0^+)(\Theta_0 - \hat{\Theta}_0^+)^T]$

5: $\hat{x}_0^+ = \mathbb{E}[x_0], \quad P_{\tilde{x},0}^+ = \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$

Computation : **for** $k = 1, 2, ..N$

Param/State − pred. time update

6: $\hat{\Theta}_k^- = \hat{\Theta}_{k-1}^+$

7: $P_{\tilde{\theta},k}^- = P_{\tilde{\theta},k-1}^+ + Q_{\tilde{r}}$

8: $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, \hat{\Theta}_k^-, w_{k-1}^-)$

9: $P_{\tilde{x},k}^- = \hat{A}_{k-1} P_{\tilde{x},k-1}^+ \hat{A}_{k-1}^T + Q_{w,k-1}$

State filter meas. update

10: $L_k = P_{\tilde{x},k}^- \hat{C}_k^T [\hat{C}_k P_{\tilde{x},k}^- \hat{C}_k^T + R]^{-1}$

11: $\tilde{y}_k = y_k - g(\hat{x}_k^-, u_k, \hat{\Theta}_k^-, \bar{v})$

12: $\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \tilde{y}_k)$

13: $P_{\tilde{x},k}^+ = (I - L_k C_k) P_{\tilde{x},k}^-$

Param. − est. meas. update

14: $L_k^\Theta = P_{\tilde{\Theta},k}^- (\hat{C}_k^\Theta)^T [\hat{C}_k^\Theta P_{\tilde{\Theta},k}^- (\hat{C}_k^\Theta)^T + R]^{-1}$

15: $\tilde{y}_k^\Theta = y_k - g(\hat{x}_k^-, u_k, \hat{\Theta}_k^-, \bar{e}_k)$

16: $\hat{\Theta}_k^+ = \hat{\Theta}_k^- + L_k^\Theta \tilde{y}_k^\Theta$

17: $P_{\tilde{\Theta},k}^+ = (I - L_k^\theta C_k^\theta) P_{\tilde{\Theta},k}^-$

end for

Computing total derivative for dual EKF

$$\hat{C}_k^{\Theta} = \left.\frac{dg(\hat{x}_k^-, u_k, \Theta)}{d\Theta}\right|_{\Theta=\hat{\Theta}_k^-}$$

$$\frac{dg(\hat{x}_k^-, u_k, \Theta)}{d\Theta} = \frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial \Theta} + \frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial \hat{x}_k^-}\frac{d\hat{x}_k^-}{d\Theta}$$

$$\frac{d\hat{x}_k^-}{d\Theta} = \frac{\partial f(\hat{x}_{k-1}^+, u_{k-1}, \Theta)}{\partial \Theta} + \frac{\partial f(\hat{x}_{k-1}^+, u_{k-1}, \Theta)}{\partial \hat{x}_{k-1}^+}\frac{d\hat{x}_{k-1}^+}{d\Theta}$$

$$\frac{d\hat{x}_{k-1}^+}{d\Theta} = \frac{d\hat{x}_{k-1}^-}{d\Theta} - L_{k-1}^x \frac{dg(\hat{x}_{k-1}^-, u_{k-1}, \Theta)}{d\Theta}$$

Initialization at k=0 and updates:

$$\Theta = [R_0 \; C_{nom}];$$

1: $$\frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial \Theta} = \left[\frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial R_0}, \frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial C_{nom}}, \right] = [-i_k, 0]$$

2: $$\frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial \hat{x}_k^-} = \left[\frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial U_1}, \frac{\partial g(\hat{x}_k^-, u_k, \Theta)}{\partial U_2}, \frac{dg(\hat{x}_k^-, u_k, \Theta)}{\partial SoC}\right]$$
$$= [-1 \;\; -1 \;\; \frac{\partial OCV(soc_k)}{\partial SoC_k}]$$

3: $$\frac{\partial f(\hat{x}_{k-1}^+, u_{k-1}, \Theta)}{\partial \Theta} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{\eta \Delta t}{C_{nom}^2} i_{k-1} \end{bmatrix}$$

4: $$\frac{\partial f(\hat{x}_{k-1}^+, u_{k-1}, \Theta)}{\partial \hat{x}_{k-1}^+} = \begin{bmatrix} e^{\frac{-\Delta t}{R_1 C_1}} & 0 & 0 \\ 0 & e^{\frac{-\Delta t}{R_2 C_2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5: $$\frac{d\hat{x}_{k-1}^-}{d\Theta} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \text{at k} = 0.$$

Tuning parameters and initialization values for Kalman filtering algorithms

1: *RC − model with EKF algorithm*

Tuning parameter	Alice1 [1]	Alice2 [2]
Q	$diag[10^{-12}, 10^{-12}, 10^{-12}]$	$diag[10^{-8}, 10^{-8}, 10^{-8}]$
R	10^{-6}	10^{-6}
P	$diag[4, 10^{-6}, 10^{-6}]$	$diag[10^{-8}, 10^{-8}, 10^{-8}]$

2: *ARX − model with EKF algorithm*

Tuning parameter	Alice1	Alice2
Q	$diag[10^{-9}, 10^{-6}, 10^{-3}]$	$diag[10^{-9}, 10^{-6}, 10^{-1}]$
R	10^{-1}	10^{-4}
P	$diag[4, 10^{-6}, 10^{-6}]$	$diag[4, 10^{-6}, 10^{-6}]$

3: *RC − model with dual EKF algorithm*

Tuning parameter	Alice1	Alice2
Q	$diag[10^{-4}, 10^{-8}, 10^{-8}]$	$diag[10^{-4}, 10^{-8}, 10^{-8}]$
R	10^{-6}	10^{-6}
Q_Θ	$diag[10^{-8}, 5 \cdot 0^{-3}]$	$diag[10^{-8}, 5 \cdot 10^{-3}]$[3]
R_Θ[4]	$R_{\Theta,r} = 10^{-1}$, $R_{\Theta,c} = 10^{-3}$	$R_{\Theta,r} = 10^{-1}$, $R_{\Theta,c} = 10^{-3}$
P_Θ	$diag[10^{-4}, 20^{-4}, 2.8]$	$diag[10^{-4}, 20^{-4}, 2.8]$

[1] Alice1 - cell's nominal capacity $\approx 0.35Ah$

[2] Alice2 - cell's nominal capacity $\approx 3.35Ah$ different cell chemistry compared to Alice1

[3] In some test cases where we observe a rapid capacity drop, wee need to set this parameters to $5 \cdot 10^{-2}$ in order to follow the measurement.

[4] We split the R_Θ in two, because different sampling time for R_0 and C_{nom}