

DIPLOMA THESIS

Development and Implementation of a Battery Emulator for Power Hardware-in-the-Loop Simulations

Submitted at the Faculty of Electrical Engineering and Information Technology,
Vienna University of Technology
in partial fulfillment of the requirements for the degree of
Diplom-Ingenieur (equals Master of Sciences)

under supervision of

o.Univ.Prof. Dipl.-Ing. Dr.techn. Dietmar Dietrich
Dipl.-Ing. Mario Faschang

Felix Lehfuß, M.Sc (Austrian Institute of Technology)
Johannes Kathan, M.Sc. (Austrian Institute of Technology)

by

Christian Seidl
Student ID: 0628027
Jagdgasse 28/15, 1100 Vienna

November 22, 2014

Kurzfassung

Im Stromnetz der Zukunft wird Erneuerbare Energie eine große Rolle spielen, um dem Klimawandel entgegen zu wirken und die Emission von Treibhausgasen zu reduzieren. Mit dem vermehrten Einsatz von volatilen Energieträgern wie Wind und Photovoltaik steigt auch der Bedarf an Speichersystemen. Neben effizienteren und kostengünstigeren Batterien ist auch die Entwicklung von dazugehöriger Leistungselektronik, wie zum Beispiel Batteriewechselrichter, essentiell. Das Testen solcher Leistungselektronik ist zeitintensiv und teuer. Batterien müssen zwischen Experimenten langwierig gelad- bzw. entladen werden, was Laborzeit beansprucht und dadurch Geld kostet. Außerdem sind oft unterschiedliche Batterietypen, -spannungen und -kapazitäten notwendig. Eine Lösung dieses Problems ist es, echte Batterien durch emulierte zu ersetzen. Diese bestehen aus einem Echtzeitcomputer, der das Verhalten der Batterie anhand eines internen Batteriemodells berechnet, sowie aus einem Leistungsverstärker, der die berechnete Batteriespannung in eine reale Spannung umwandelt. Damit können verschiedene Batterietypen modelliert, bzw. deren Spannung, Kapazität und der Ladezustand beliebig vorgegeben werden. Diese Arbeit beschäftigt sich mit der Entwicklung eines solchen Batterie-Emulators. Hierfür wird ein passendes Batteriemodell entwickelt, dessen Parametrisierung zuerst diskutiert und danach mithilfe realer Messungen aus einem standardisierten Fahrzyklus validiert. Anschließend wird das Batteriemodell auf ein Echtzeitsystem portiert und der Batterie-Emulator implementiert. Mithilfe einer Kombination aus konstanter Last und programmierbarer Stromquelle werden die Messungen des Fahrzyklus reproduziert und so der Batterie-Emulator verifiziert. Dabei erreicht der batterie-Emulator eine Übereinstimmung von 95% im Vergleich zu den Messungen. Der entwickelte batterie-Emulator kommt am Forschungsprüfstand des Austrian Institute of Technology zum Einsatz und ermöglicht dort schnellere und reproduzierbare Tests von Batteriewechselrichtern, Ladesäulen von Elektrofahrzeugen u.v.m., ohne die jeweilige Batterie kaufen zu müssen.

Abstract

Renewable energy will play an important role in the grid of the future in order to fight climate change and reduce greenhouse gas emissions. The increased use of volatile energy sources like wind and photovoltaic goes along with an enlarged demand of battery energy storage systems. Besides more efficient and cost-effective batteries, the development of battery related power electronics like battery inverters is vital. The test of such power electronics is time-consuming and expensive. Batteries have to be charged and discharged between experiment, a protracted procedure that costs time and hence also money. Furthermore, various battery types with different voltages and capacities are necessary. A solution to this problem is replacing a real with an emulated battery. These consist of a real time system that computes the behavior of the battery on the basis of an internal battery model, and of a power amplifier that converts the calculated battery voltage into a real voltage. That way various battery types can be modeled, as well as whose voltage, capacity and state of charge can be set arbitrary. This work deals with the development of such a battery emulator. Therefore a suitable battery model is developed, its parametrization discussed and afterwards validated by means of measurements from a standardized driving cycle. Then, the battery model is ported onto a real time system and the battery emulator is implemented. The measurements of the driving cycle are reproduced with a combination of a constant load and a programmable current source in order to verify the battery emulator. Here, the battery emulator achieves accuracy of 95% in comparison to the measurements. The developed battery emulator is deployed at the research test bed of the Austrian Institute of Technology and makes faster, more cost-effective and reproducible tests of battery related power electronics possible without the need of buying the particular batteries.

Acknowledgements

Der größte Dank gebührt meiner Familie, insbesondere meinen Eltern Dagmar und Alfons, die mich zeit meines Lebens in vielfältigster Weise unterstützen. Ihnen widme ich diese Arbeit. Außerdem möchte ich ihnen dafür danken, dass sie mich lehrten, was im Leben wichtig ist.

I want to thank my supervisor Mario for his advices, comments, and the huge time and effort he spent to help me and make this thesis possible.

I thank ye, my dear fellows Felix and Johannes. Ye have enlightened me in the area of battery simulations and Power Hardware-in-the-Loop simulations. Ye have aided me, through the beauty of tongues and acts, and oh, those many thoughts we shared, and in detail discussed. But most of all, the thanks to ye I give is for making my time at AIT of the utmost delight.

Finally I wanna thank all my friends, especially Alex, Birgit, Dani, Jakob, Marlies, Richard, Steffi and Tara, for their help, support, motivation, inspiration and distraction.

*For a successful technology,
reality must take precedence over public relations,
for nature cannot be fooled.*

Richard Feynman

Table of Contents

1	Introduction	1
1.1	Need of Storage Systems	1
1.2	Replacing Batteries with Emulated Batteries	2
1.3	Power Hardware-in-the-loop Simulations	3
1.4	Problem Statement and Methodology	5
1.5	Terminology and Definitions	8
2	State of the Art Analysis of Battery Modeling	10
2.1	Classification of battery models	10
2.2	PHIL Applications with Battery Models	11
2.3	Electrochemical Models	12
2.4	Models Determining State of Charge and Battery Lifetime	12
2.5	Generic Models	16
2.6	Equivalent Electrical Circuit Models	22
2.7	Comparison of EEC and Generic Models	27
2.8	Summary of the Literature Review	29
3	Parametrization of Battery Models	30
3.1	Parametrization Using Dedicated Measurements	30
3.2	Parametrization Using Data Sheet Values	37
3.3	Parametrization of the Kinetic Battery Model	40
3.4	Scaling a Battery Model	41

4	Validation of the Battery Models	46
4.1	Error Definition	46
4.2	Implementation in SIMULINK	47
4.3	Validation	50
5	PHIL Implementation	60
5.1	OPAL-RT Real Time System	61
5.2	Power Amplifier	62
5.3	Variable Load and Source	67
5.4	Preparation of Measurements and Scaling the Battery Model	69
6	PHIL Simulations	73
6.1	Recording and Post Processing of Measurements	73
6.2	Discharging of a Battery with a Constant Resistor	74
6.3	Simulating the FTP-72 Cycle	75
7	Discussion and Conclusion	80
7.1	Review and Discussion	80
7.2	Future Work	85
7.3	Personal Experiences and Perceptions	87
7.4	Summary and Conclusion	88
A	Used Hardware	93
B	Computed Parameters	97
C	Simulation results	101
D	Measurement Data of Offset Correction	114
	Literature	116
	Internet References	121

Nomenclature & Abbreviations

ADC	Analog to Digital Converter
AIT	Austrian Institute of Technology
BESS	Battery Energy Storage System
CC	Constant Current mode
CP	Constant Power mode
CV	Constant Voltage mode
DAC	Digital to Analog Converter
DoD	Depth of Discharge
EEC	Equivalent Electrical Circuit
eMSM	extended Modified Shepherd Model
EV	Electric Vehicle
FTP	Federal Test Procedure
HIL	Hardware-in-the-Loop
HuT	Hardware under Test
ITM	Ideal Transformer Model
KiBaM	Kinetic Battery Model
LiFePO ₄	Li-ion battery with a lithium iron phosphate cathode
MSM	Modified Shepherd Model
NiCd	Nickel-cadmium battery
NiMH	Nickel-Metal Hydride battery
OCV	Open Circuit Voltage
PHIL	Power Hardware-In-the-Loop
PV	Photovoltaic
Q	Capacity of a battery
RMS	Root Mean Squared
RTS	Real Time System
SoC	State of Charge

1 Introduction

For decades the structure of the electric power supply system had been the same: mainly few big power plants burning coal or gas, fissioning uranium or water power plants, which fed into the transmission grid. But this situation changed a lot in recent years. Renewable energy, like photovoltaic and wind power plants, now plays an important role and will even play a more important role in the energy mix of countries like Germany and Austria, and even China [IEA14, IEC11, Eur13]. Therefore the reasons are various. They could be

- ecological reasons like climate change, the global warming, the 2°C target [Eur13], or the Europe 2020 targets [5],
- political reasons like the nuclear phase out after the Fukushima nuclear disaster [13] as well as the general refusal of nuclear energy in public, or the independence of imports of fossil fuels from other countries with politically unstable conditions,
- or economic reasons, because Renewable energy can be more economic than conventional fossil or nuclear power plants [Fra13, Dep13].

1.1 Need of Storage Systems

With the increased use of volatile energy sources like wind and photovoltaic also comes an increased need of storage systems, because the generation needs to fit the consumption at any time. These storage systems could be mechanical storage like pump storage, flywheels, compressed air energy storage, thermal energy storage, chemical storage like hydrogen storage, power to gas or electrochemical like batteries or supercapacitors [IEA14]. The technologies can be categorized by storage technology, power, energy, discharge duration, number of cycles, response time, efficiency, investment costs per kW, etc. Batteries can be ranked in this classification with power up to a few MW, discharge durations from minutes up to hours, which means that they work both as power storage and energy storage, response time of a few seconds and an efficiency – depending on the cell technology – above 90% [IEA14, Pol08, IEC11].

A typical application for households is the combination of a photovoltaic system (PV) in combination with a battery energy storage system (BESS) with lead-acid or lithium batteries. These combinations are used to maximize the in-house consumption – the batteries are charged about noon when the sun is shining and the PV produces more than currently consumed, while the batteries are discharged during the evening and night. This makes on the one hand sense in an economical point of view and on the other hand helps the grid to reduce the deviation between

peak and base load [IEC11, Zha13]. The battery inverter of a BESS is one of the most important factors when it comes to the overall efficiency of a battery. A modern inverter dealing with PV and a battery has to meet certain requirements like reducing the power fed in to the grid as a function of the grid frequency, pass fault ride through test, or controlling the reactive power according to the real power [VDE13]. All these requirements have to work with various irradiated PV power, at distinct states of charge of the battery with different loads, where the behavior of the inverter maybe varies for each use case. And in all these cases, the MPPP-tracker (maximum power point) of the PV as well as the other power electronics should work efficient. There are a lot of further use cases a BESS system can be tested for. Some of them are for example

- Can the system recognize islanding?
- Can the BESS operate in islanding mode?
- Is the BESS black start capable? And if yes; under which circumstances?
- Can the battery inverter handle different battery types and voltages and does the efficiency defer for them?

Another system that remained similar for decades and will change in the next years is the individual motor car traffic. Cars with combustion engines have been the primary used technology on the market, but now electric vehicles edge into the market and will play an important role in the future [IEA13, Eur10]. Vehicles and charging stations will also interact with the grid to reduce the strain of the grid. This creates a demand for intelligent charging stations and so intelligent battery related power electronics.

So battery system will play a greater role in the next years and thus also battery related power electronics. These power electronics have to be tested and certified, which is very extensive and time consuming, because there are various test cases. Batteries have to be charged or discharge, either full, or to a certain state of charge after a test before the next test can start. This is time consuming and also increases costs because the laboratory environment is used without any practical benefits. The batteries themselves are expensive and probably various batteries with different technologies, voltages and capacities are necessary. Also the tests are not exactly reproducible because some internal battery procedures are not reversible.

1.2 Replacing Batteries with Emulated Batteries

Power Hardware-in-the-Loop (PHIL) simulations of battery systems could significantly reduce the time and costs of battery inverter and power system testing. The idea behind such PHIL simulations is to replace a real battery with an emulated battery. This emulated battery is built out of a bidirectional power source, controlled by a real time system (RTS) which computes a battery model. Outwards, this setup behaves like a battery, with the advantage that the voltage/current-behavior is freely settable (within the hardware limits of the power source and the RTS). This approach has a few advantages.

- The battery state can be set arbitrary. So the charging and discharging to a desired state of charge can be omitted. This saves time and money.
- The battery voltage and capacity can be set arbitrary too. So there is no need to buy different batteries with various cell technologies, voltages and capacity, which reduces the costs.

- The tests are reproducible. Also tests like short circuit experiments are possible which maybe would destroy a real battery.
- Not only the battery side can be emulated. The research test bed of AIT's Energy Department, where the developed battery emulator will be installed (cf. Section 1.4.2), already has a PV and a grid simulator. So the complete environment of a battery inverter can be emulated.

But as always, there are also some disadvantages.

- First of all, there is a need for a battery model, which implies an additional effort to create, test and validate the model.
- A battery model can never exactly reproduce the behavior of the corresponding real battery. So there is limited accuracy and depending on the area of application, the accuracy of the model can be too low. Also, there are always some effects which are not represented in the model. These can be negligible, because they are not focus of the domain, e.g. long term capacity fading or a temperature rise when the focus is on transient simulations in a range of some minutes. But it is also possible, that there is no existing model at all, which meets the demands.
- Every model has its parameters which have to be determined. This procedure can be time consuming and extensive. A wrong parametrization also reduces the possible maximum accuracy of a model.
- Beside the controlled power source, which converts the set value of the battery model in real voltages adds an additional error.

Although there are some drawbacks, a Power Hardware-in-the-Loop setup to emulate a battery is a promising approach to increase the efficiency of battery related power electronics developing, testing and certifying. Hence, this work identifies a suitable battery model, verify it and then implement a PHIL setup to build an emulated battery. Before defining the problem statement, the concept of Power Hardware-in-the-Loop is examined more detailed.

1.3 Power Hardware-in-the-loop Simulations

Power Hardware-in-the-Loop simulations are a special type of Hardware-in-the-Loop (HIL) simulations. Hence, this term has to be explained before. A HIL simulation is a simulation to test or certify a real hardware (hardware under test, HuT). The rest of the system, where this HuT normally operates, is simulated on a computer (see Fig. 1.1a). Typically, the simulation computer is connected via analog-to-digital and digital-to-analog converter (ADC and DAC), but digital I/Os or buses are also possible. Since the simulation computer and the hardware under test form a loop by each reading the input values, processing them, setting the output values, this kind of setup is called Hardware-in-the-Loop simulation.

A typical application is the test of an airplane's controller, which is then called Controller Hardware-in-the-Loop (CHIL). A dynamic model of the airplane is computed and the simulated values of sensors are the inputs of the flight controller. Using a HIL simulation, the controller's behavior at a stall or an engine failure can be analyzed without having an actual airplane in the air and risking destroying it.

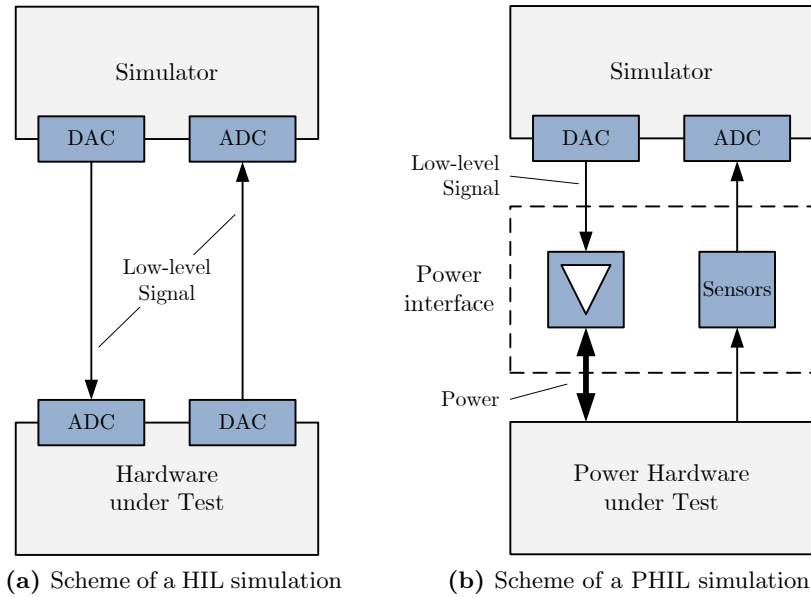


Figure 1.1: Scheme of a HIL and a PHIL simulation [RSB08]

The example of the airplane controller shows that the simulation computer often has to be a real time system. ADCs and DACs are driven with a synchronous clock with the sampling frequency f_s . The simulation has to satisfy the hard real time criteria. This means that reading and processing input values, calculating a model step and writing the output data to the DACs must not take longer than $T_s = \frac{1}{f_s}$. This circumstance is shown in Fig. 1.2.

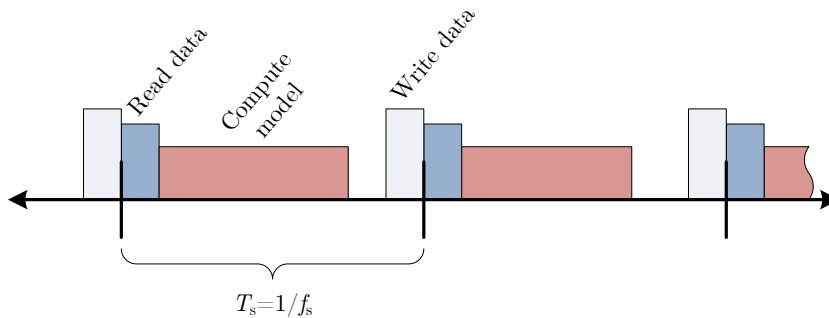


Figure 1.2: Hard real time criteria

At a PHIL simulation, the HuT is not a controller but a power electronic device. This creates the need for a power amplifier and sensors which measure the power signals (see Fig. 1.1b). From a control point of view, the power amplification equals an additional control loop [SKLL14]. This control loop impacts the stability and can cause instabilities which require additional filters [VLF11]. There are several interface algorithms which implement the power interface between the RTS and the Power HuT. A good comparison of different interface algorithms can be found in [RSB08] and [JGV⁺11]. This work uses the ideal transformer model, because it is one of the simplest and showed good performance. Therefore no other interface algorithms were investigated.

Ideal Transformer Model

The ideal transformer model (ITM) is one of the most conventional and straightforward methods in PHIL simulations [RSB08]. The principle of the ITM is shown in Fig. 1.3. The connection between the power HuT and the rest of the system is cut and a combination of a controlled current and a controlled voltage source is inserted. The rest of the system, together with its

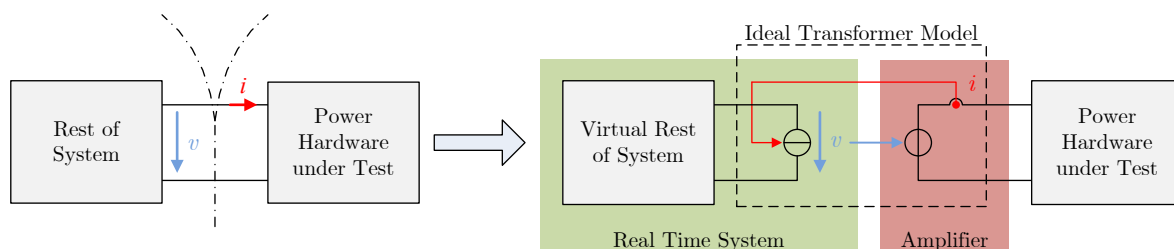


Figure 1.3: Ideal transformer model

connected controlled source is modeled and then simulated on the real time system. A voltage source controlled by the RTS amplifies the set value from the simulation, while the measured current is fed back into the RTS. Depending on the impedances of the virtual rest of the system, the Power HuT, the dead times in the system, and the amplifier, the ITM can cause instabilities. Further information about this issue can be found in [RSB08, VLF11, JGV⁺11].

The configuration shown in Fig. 1.3 is named voltage-ITM, because the HuT is driven by a controlled voltage source and the virtual rest of the system terminated with a controlled current source. In some use cases, the HuT is driven by a current source, while the simulated counterpart is a voltage source. This setup is then called current-ITM [RSB08].

1.4 Problem Statement and Methodology

1.4.1 Problem Statement

The aim of this work is to build a PHIL capable battery emulator. This battery emulator has to consist of a bidirectional power source and a real time system that computes a battery model and controls the power source based on the output of the battery model. The power source should be capable of battery voltages up to a few hundred volts with a power up to 20 kW.

The model's output is the battery voltage, and its input the current of the power source. To calculate the voltage, the battery model also has to determine the battery's state of charge, because the voltage depends on it. The battery model also has to represent the polarization of charge carriers inside the battery. This polarization can be described in good approximation with a PT1-element. Effects like the heating of the battery due to internal losses or long time effects should not be considered since the typical duration of a PHIL simulation in this domain is in the range of seconds to an hour.

The battery model should be easy to parametrize and as generic as possible. At best, only data sheet values and experienced data should be necessary to parametrize a model. So the parametrization effort is minimized, being aware that this will decrease the accuracy.

The existing OPAL-RT real time system (see Section 1.4.2) has to be used for this work, where one core of it is available for calculating the battery model. In future applications the emulated battery will be part of a bigger PHIL simulation with an emulated AC grid. The AC grid emulation requires a step size of $10\ \mu\text{s}$. Hence, the execution time of the battery model must be below $10\ \mu\text{s}$. This is a very crucial requirement.

The work focuses on the modeling and parametrization of a Li-ion polymer battery, because only measurement data of such a battery were available for this work. Further information is given in Section 1.4.3.

1.4.2 Cooperation with the Austrian Institute of Technology

This work is performed in cooperation with the Energy Department of the Austrian Institute of Technology (AIT). The AIT provided all the hardware used in this work. The battery emulator will part of the *Research Test Bed* of the *SmartEST Laboratory* of AIT [2]. A principle structure of the test bed is shown in Fig. 1.4. The test bed can emulate the complete environment of a PV

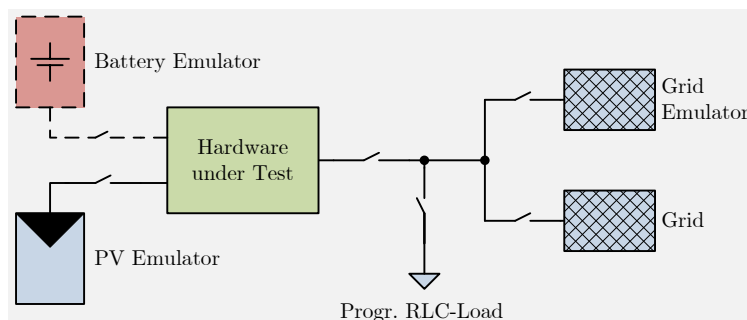


Figure 1.4: Structure of AIT’s *Research Test Bed* with the battery emulator as future extension

inverter with grid emulation, a PV emulator and controllable resistive, capacitive and inductive loads. The battery emulator will extend the existing setup to be able to test also battery inverters or other battery related power electronics like charging stations for electric vehicles.

Out of this work resulted two conference papers which cover the principle ideas and results of this thesis

- *Selection and Implementation of a Generic Battery Model for PHIL Applications* [SKLL13], presented at the 39th “Annual Conference of the IEEE Industrial Electronics Society” (IECON 2013), and
- *Power Hardware-in-the-Loop Implementation and Verification of a Real Time Capable Battery Model* [SKLL14], presented at the 23rd IEEE International Symposium on Industrial Electronics” (ISIE 2014).

1.4.3 Used Data Base

For this work measurements of an *EiG ePLB C020* Li-ion battery were used. This is a Li-ion polymer battery with a $\text{Li}[\text{NiCoMn}]\text{O}_2$ based cathode, a capacity of 20 Ah and a nominal voltage of 3.65 V. Its data sheet can be found in Fig. A.4 on page 96.

The measurements were conducted by the Mobility Department of the Austrian Institute of Technology for the work of [ECKF13]. These are measurements from the repeatedly applied FTP-72 (federal test procedure, [4]) driving cycle on a fully charged battery until the battery was completely discharged. The FTP-72 driving cycle is a standardized driving schedule which simulates the drive through an urban area with acceleration, braking and frequent stops and shown in Fig. 1.5. The maximum speed is 91.25 km/h and the average speed is 31.5 km/h. In [ECKF13]

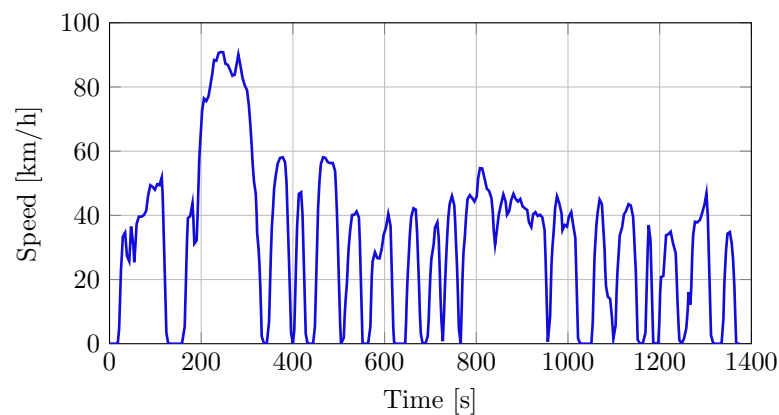


Figure 1.5: FTP-72 driving cycle

battery power values were calculated out of the speed values, which were then transformed into current set values for the battery current. This current profile is shown in Fig. 4.5 in Section 4.3.

Beside the FTP-72 driving cycle measurements, also a measured open circuit voltage curve and a parametrization of an equivalent electrical circuit (EEC) model (cf. Section 2.6) was available. With this data the EEC model can be validated against the measurements and compared with other models.

Unfortunately no other measurement data were available and also there was no possibility to perform own measurements on the same or other batteries. So the work can only rely on this data and is therefore focused on the modeling of Li-ion batteries. A possible extension to lead-acid batteries is always kept in mind.

1.4.4 Methodology

To accomplish the above demands, several steps have to be done in this work

1. First, a literature research about battery modeling is done in order to find a suitable PHIL battery model.
2. Then the suitable battery models are parametrized.
3. Afterwards, the models are implemented in SIMULINK and validated in a pure software simulation against the measurement.

4. The next step is to build the hardware setup and perform all the preparatory work so that the battery model can be executed on the real time system, which processes the current measurement and sets the terminal voltage of the power source.
5. Finally the PHIL simulations are conducted and their results discussed.
6. The work will finish with a conclusion and an outlook.

Before the work will continue with a state of the art analysis of battery modeling, some terms are defined.

1.5 Terminology and Definitions

Battery

A battery consists of one or more chemical cells, connected in series (to increase voltage) or in parallel (to increase capacity). Batteries can be divided in primary batteries and secondary batteries, where primary batteries can only be discharged once, whereas secondary batteries can be discharged and charged several times. Typical secondary battery technologies are lead-acid, Li-ion, NiMH (Nickel metal hydride) and so forth. In this work only secondary batteries are covered, so the term battery is used as a synonym for a secondary battery.

State of Charge

The state of charge (SoC) is defined as the ratio of the currently available charge $Q_{\text{available}}$ of the battery to total charge Q (total capacity) of the battery (in percent or p.u.). So $SoC = 1$ for a completely charged battery and $SoC = 0$ for an empty battery. A few models to calculate the SoC can be found in Section 2.4.

Current direction

In this work the battery is seen as a source, so the sign of the current is positive when the current flows out of the battery (discharge) and vice versa.

Battery Lifetime

In this work the battery lifetime is defined as the time to discharge a fully charged battery (typically some hours). So the State of Charge decreases from one to zero. It has to be mentioned that the term also has other usages, e.g. the time until a battery is at the end of its life cycle because of too many charge and discharge cycle (which can be more than 1000 for modern Li-ion polymer batteries, cf. Fig A.4 on page 96).

C-rate

The C-rate describes the battery current normalized to its capacity. This gives the possibility to compare batteries with different capacities. A current of $1C$ is the current which discharges a battery in one hour. For example, for a battery with a capacity of 15 Ah, $1C$ equates to 15 A, $2C$ to 30 A, $0.5C$ to 7.5 A and so on.

Rate Capacity Effect

The usable capacity of a battery depends on the discharge current and is lower for higher currents. For more details see Section 2.4.1.

Recovery Effect

During periods of very low or no discharge, a battery can regain the not usable capacity which was “lost” due to the Rate Capacity Effect. This effect is called Recovery Effect [JH08]. Its principle is shown in Fig. 1.6. The electro-active species needs time to diffuse to the electrodes.

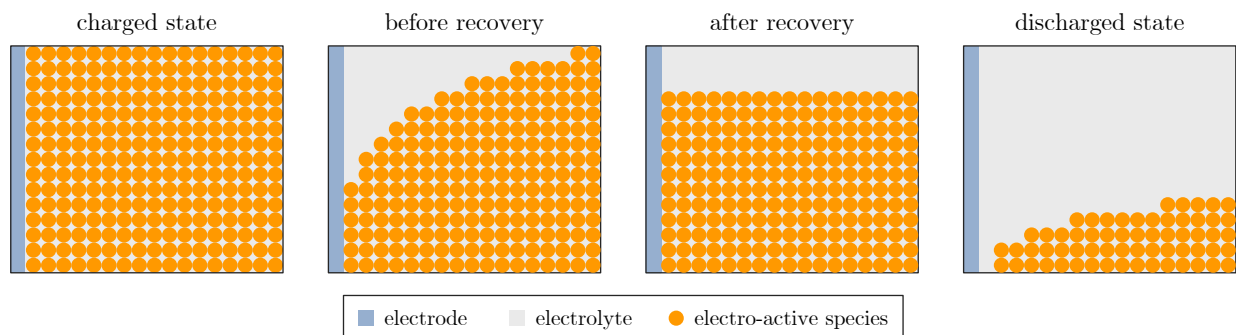


Figure 1.6: Recovery Effect illustrated by the model of Rakhmatov and Vrudhula [JH09, RV01]

If the discharge current is too high, they cannot diffuse in time, the voltage breaks down and the battery seems to be empty. After the recovery time the electro-active species is uniform distributed again, the battery voltage rose and the battery can be discharged further on.

2 State of the Art Analysis of Battery Modeling

Beginning with a classification of different types of battery models, this chapter will give a short discussion about battery modeling related to Hardware-in-the-Loop (HIL) and Power Hardware-in-the-Loop (PHIL) simulations. Since there are no satisfying papers about this topic, the chapter continues with a broad literature review about battery modeling and discusses the models with the focus on the needs and boundary conditions of a generic PHIL battery model.

2.1 Classification of battery models

Battery modeling is always a trade off between four criteria [RVR03]:

- *Accuracy*: How accurate can the model predict the battery variables of interest in comparison to measured quantities and experimental data?
- *Computation complexity*: How long does it take to calculate one model cycle and how much memory is used?
- *Configuration effort*: How many parameters does the model have, which effort is needed to parametrize it and does the model require knowledge of the battery chemistry?
- *Analytical insight*: Does the model provide qualitative understanding of the battery internals and behavior and is this insight useful for the specific application?

According to this classification the wanted battery model can be characterized as follows:

- *Computation complexity*: The computation time for one step should be below 10 μs at the present OPAL RT real time system. This is a very strict requirement and thus stands on first position.
- *Configuration effort*: The battery model should be as generic as possible and in the best case only data sheet values should be necessary to parametrize the model. This work will discuss other ways for parametrization (see Chapter 3), but using only data sheet values is the preferred method.

- *Accuracy*: The above two criteria limit the accuracy of the battery model. Of course, the model should be as accurate as possible, but the low computation time is more important. It is enough to have a maximum relative error of about 10%.
- *Analytical insight*: An analytical insight is nice to have, but not important. Chemical insights like charge carrier density distribution are not necessary. More important is that it is easy to roughly guess the model output for a given input.

There are various types of battery models modeling diverse characteristics with different complexity according to varying applications. Literature reviewed showed, that there are various classifications of battery models (e.g. [RVR03, TSK12, LK11, JH09]). Derived from these classifications, this work has its own, which is a mix of those in the literature, because none of the existing ones matched the focus of this work enough. This works distinguish between

- *electrochemical models* which model the chemical processes in the battery, such as the charge carrier density,
- models to calculate the *state of charge* respectively the *battery lifetime*,
- *generic models*, which are mathematical models without modeling the chemical internals of a battery,
- *equivalent electrical circuit (EEC) models* which uses a network of electrical circuit components to model the battery.

An overview of some generic battery models is shown in [HB11]. Beside of generic models, a good overview of several types of battery models can be found in [JH09] and [RVR03].

2.2 PHIL Applications with Battery Models

While there is a vast amount of literature about battery modeling, battery models with a focus on Hardware-in-the-Loop and especially Power Hardware-in-the-Loop have been rarely discussed in the literature before. This was one of the main motivations of this work. Still there are a few papers which deal with topics that are related to the focus of this work.

- He et al. [HLK10] used a HIL system to develop and verify a *SoC* prediction of embedded battery algorithms. They used a real cell and compared it with the results of an EEC model. So their approach is different to this work, where a battery should be emulated. He et al. also did not discuss the computation effort and execution time.
- Edrington et al [EVH⁺10] built a PHIL setup and used a simplified version of the Shepherd Model (see Section 2.5.3) which was simplified in order to avoid an algebraic loop. Their paper has in principle the same focus than this work, but it is more a proof of concept. The battery model is too simple and issues like accuracy and computation time are not discussed.
- Yoo et al. [YCC⁺12] uses a HIL simulation of a DC microgrid with a multi-agent system as hardware. The microgrid is simulated with the SIMULINK SimPowerSystems toolbox including its battery model. The authors neither discuss the battery model, nor mention the step time of the HIL simulation.

- Song et al. [SKJ⁺13] used a HIL simulation to validate their equivalent electrical circuit model (see Section 2.6). The authors used the control area network bus (CAN) to charge and discharge the battery in a HIL simulation and then compared the measurements to a battery model.
- Pebriyanti [Peb13] had a quite similar focus to the one of this work. He developed an equivalent electrical circuit model to test a battery management system. The battery model is implemented in SIMULINK and executed on an xPC target computer with an execution time of $\approx 12 \mu\text{s}$. The model was parametrized with an extensive series of impedance spectroscopies from 1 mHz to 1 Hz at different states of charge.

2.3 Electrochemical Models

Electrochemical (also called physical) models are the most exact models above the examined models. Typically six coupled differential equations, with some of them partial differential equations, are used to model characteristics like current density or charge carrier concentration. There is a freely available Fortran implementation called DUALFOIL [7], which is often used as a reference to determine how accurate another battery model is [RVR03].

Electrochemical models have a high computational effort, which is too high for the real time conditions with a step size below $10 \mu\text{s}$ and the computation power of current real time hardware. Furthermore, electrochemical models have a lot of parameters (more than 50 according to [RVR03]), so their configuration effort is also high and exact knowledge of the battery is needed. This does not fulfill the constraint of a easy configurable battery model. Thus, electrochemical models are no longer considered in this work. Further information about electrochemical battery models can be found in [CKC⁺10, DVM12, DDFS08, GWDW02, LCP12, Mar11, RBAI11, SW06, SM13, SRW07].

2.4 Models Determining State of Charge and Battery Lifetime

Exact information about the current state of charge and remaining battery lifetime is important for a lot of applications, especially for mobile ones. This section discusses models which calculate the battery lifetime or the state of charge.

2.4.1 Peukert Model

A simple model to calculate the battery lifetime L is the Peukert Model [DS06]. It is named after its explorer Wilhelm Peukert, who found in the beginning of the 20th century the empirical equation

$$L = \frac{a}{I^b} \cdot (1 \text{ A})^{b-1}, \quad (2.1)$$

with I as a constant battery current and the two parameters a and b . The parameter a is Q for an ideal battery and close to Q for a real battery. For a real battery the parameter b is always greater than 1 which implies that the lifetime and also the usable capacity $L \cdot I$ depends on the current and is lower for higher discharge current. This effect is called *Rate Capacity Effect* or Peukert Effect. The term $(1 \text{ A})^{b-1}$ is only a correction term to achieve a time as unit for L . Li-ion

batteries have a b close to the ideal value 1, while b is around 1.1 – 1.25 for gel and 1.2 – 1.6 for flooded lead-acid batteries [3].

One has to mention that the term $L \cdot I$ is only the *usable* capacity and the battery is not really empty, but has an “unavailable capacity”. The battery still can be discharged, but with very small currents. This can be explained by the Kinetic Battery Model (see Section 2.4.4).

The Peukert equation can be extended for variable currents i [JH09] to

$$L = \frac{a}{\left(\frac{1}{L} \int_0^L i(\tau) d\tau\right)^b}. \quad (2.2)$$

Beside the fact that this is a nonlinear implicit equation in L and hence difficult to solve, Equation 2.2 would imply that all current waveforms with the same mean value would have the same lifetime. This is amongst other things not valid due to the Recovery Effect (see Section 1.5). So the Peukert model is only suitable for general estimations of the battery lifetime considering the Rate Capacity Effect.

2.4.2 Diffusion Model of Rakhmatov and Vrudhula

Unlike the Peukert model, Rakhmatov and Vrudhula’s model [RV01] is not empirical but deduced from Fick’s laws of diffusion. Their model equation is

$$\alpha = \int_0^L \frac{i(\tau)}{\sqrt{L-\tau}} d\tau + 2 \sum_{m=1}^{\infty} \int_0^L \frac{i(\tau)}{\sqrt{L-\tau}} \exp\left(-\frac{\beta^2 m^2}{L-\tau}\right) d\tau \quad (2.3)$$

with the two parameters $\alpha = \nu F A \sqrt{\pi D C^*} \rho(L)$ and $\beta = \frac{w}{\sqrt{D}}$. L is the lifetime, D is the diffusion constant of the diffusion process in the cell, ν the number of electrons involved in the electrochemical reaction at the electrode, F the Faraday’s constant, A the area of the electrode surface, w the length of the one-dimensional diffusion, C^* is the concentration at $t = 0$ and $\rho(L) = 1 - \frac{C(x=0,L)}{C^*}$. The model equation describes a relationship between battery current and lifetime and covers both Rate Capacity as well as Recovery Effect. In addition, Equation 2.3 can be transformed in an equation describing the *SoC* [JH09].

However, the equation cannot be used in practice because it contains an infinite sum. This issue can be solved by discretising the integral and replacing the infinite sum with a finite one. Rakhmatov and Vrudhula did this in their paper until $m = 10$ at a staircase-shaped current which is implicitly discretised. The average error of their model was smaller than 1% in comparison to a simulation with DUALFOIL. Since the parametrization of the model is difficult and the model is too computational expensive to use it for calculating the *SoC*, it is not further used in this work.

2.4.3 Coulomb Counting

In good approximation the *SoC* can be calculated via the integral over the battery current i :

$$SoC = SoC_{\text{init}} - \frac{1}{Q} \int_0^t i(\tau) d\tau \quad (2.4)$$

This method is called Coulomb Counting and the most common way of determining the state of charge. There are some extensions for this simple method. One of them is the Kinetic Battery Model which is described in Section 2.4.4. To consider self discharge, Equation 2.4 is extended to

$$SoC = SoC_{\text{init}} - \int_0^t (i(\tau) - i_{\text{selfdisc}}(\tau)) d\tau. \quad (2.5)$$

When charging a lead-acid battery not all of the current is used for charging. A part of it fissions water molecules into hydrogen and oxygen. This effect is called gassing. Gassing is modeled similar to self discharge

$$SoC = SoC_{\text{init}} - \int_0^t (i(\tau) - i_{\text{gassing}}(\tau)) d\tau. \quad (2.6)$$

Of course these two effects can be combined. Another possibility is to use a charge and discharge efficiency. Therefore an efficiency η is introduced:

$$SoC = SoC_{\text{init}} - \int_0^t \eta i(\tau) d\tau. \quad (2.7)$$

η is less or equal 1 and usually dependent on the current direction (charging or discharging), but can also be a function of the SoC or the temperature [GLD02].

2.4.4 Kinetic Battery Model

The Kinetic Battery Model (KiBaM) is an enhanced model to calculate the SoC of a battery [MM93]. The name is derived from its use of a kinetic process to describe the battery behavior. It was developed as an abstract model, but Jongerden and Haverkort [JH09] proved, that it is a first order approximation of the Diffusion Model of Rakhmatov and Vrudhula (see Section 2.4.2 and [RV01]). The KiBaM covers the Rate Capacity Effect as well as the Recovery Effect. The KiBaM originally was developed as a discharge model and extended by [WC10] for charging.

The basic idea of the KiBaM is to separate the total charge Q of the battery into two charge wells: one for the available charge y_1 and one for the bound charge y_2 (see Fig. 2.1). Two differential

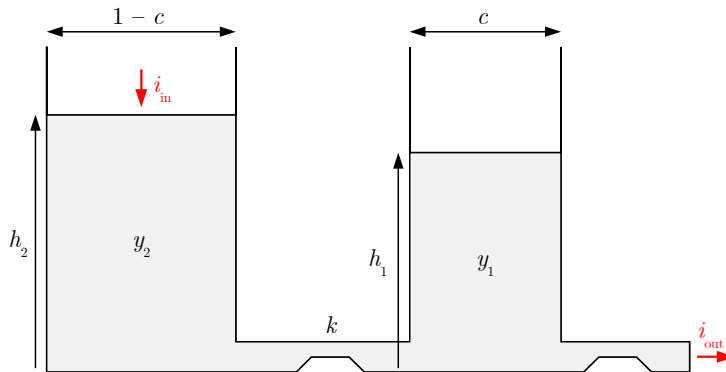


Figure 2.1: Kinetic Battery Model [JH09, WC10]

equations describe the charge in the wells

$$\dot{y}_1 = -i_{\text{out}}(t) + k (h_2 - h_1) \quad (2.8)$$

$$\dot{y}_2 = i_{\text{in}}(t) - k (h_2 - h_1) \quad (2.9)$$

with the initial conditions

$$y_1(0) = c Q_{\text{init}} \quad (2.10)$$

$$y_2(0) = (1 - c) Q_{\text{init}}. \quad (2.11)$$

The heights of the wells are given by $h_1 = \frac{y_1}{c}$ and $h_2 = \frac{y_2}{1-c}$. The model inputs $i_{\text{in}}(t)$ and $i_{\text{out}}(t)$ are defined as

$$i_{\text{in}}(t) = \begin{cases} -i(t) & i(t) < 0 \\ 0 & \text{else} \end{cases} \quad (2.12)$$

as well as

$$i_{\text{out}}(t) = \begin{cases} i(t) & i(t) > 0 \\ 0 & \text{else.} \end{cases} \quad (2.13)$$

The model has two parameters c and k , where c is between 0 and 1 and represents the partitioning of the charge in the two wells. It depends on the battery technology and is smaller in lead-acid batteries than in Li-ion batteries [KQ11]. This means that Li-ion batteries have a bigger available charge well, which corresponds with a smaller Rate Capacity Effect (b closer to 1 in the Peukert Model, Section 2.4.1). The parameter k describes how fast charge can flow between the two wells. The constant

$$k' := \frac{k}{c(1-c)} \quad (2.14)$$

is related to the diffusion rate and the inverse time constant of the Recovery Effect (cp. Equation 2.20 and 2.21).

According to [JH09] the unavailable charge is given by

$$Q_{\text{unavailable}} = (1 - c) (h_2 - h_1). \quad (2.15)$$

With Equation 2.15 the SoC can be calculated

$$\begin{aligned} SoC &= \frac{Q_{\text{available}}}{Q} = \frac{Q - Q_{\text{unavailable}}}{Q} = \\ &= \frac{1}{Q} (y_1 + y_2 - (1 - c) (h_2 - h_1)) = \\ &= \frac{1}{Q} \left(y_1 + y_2 - (1 - c) \frac{y_2}{(1 - c)} + (1 - c) \frac{y_1}{c} \right) = \\ &= \frac{1}{Q} \frac{1}{c} y_1. \end{aligned} \quad (2.16)$$

The coordinate transformation

$$\gamma = y_1 + y_2 \quad (2.17)$$

$$\delta = h_2 - h_1 \quad (2.18)$$

transforms Equation 2.8 and 2.9 into two separated differential equations

$$\dot{\gamma} = -i(t) \quad (2.19)$$

$$\dot{\delta} = \frac{1}{1-c} i_{\text{in}}(t) + \frac{1}{c} i_{\text{out}}(t) - k' \delta. \quad (2.20)$$

Now the *SoC* can be easily written as

$$\begin{aligned} \text{SoC} &= \frac{1}{Q} (\gamma - (1-c) \delta) = \\ &= \frac{1}{Q} \left(- \int_0^t i(\tau) d\tau + Q_{\text{init}} - (1-c) \delta \right) = \\ &= \text{SoC}_{\text{init}} - \frac{1}{Q} \int_0^t i(\tau) d\tau - \frac{1}{Q} (1-c) \delta \end{aligned} \quad (2.21)$$

which is the Coulomb Counting Equation 2.4 extended by the term $-\frac{1}{Q} (1-c) \delta$.

2.4.5 Stochastic Models

Another possibility to describe the characteristics of a battery are stochastic models [JH08, JH09, RVR03]. Stochastic models are abstract models which model discharging and effects like the Recovery Effect as stochastic processes. They use Markov Chains with about 10^6 states (see Fig. 2.2). When charging or discharging a battery, the state in the Markov Chain is changed

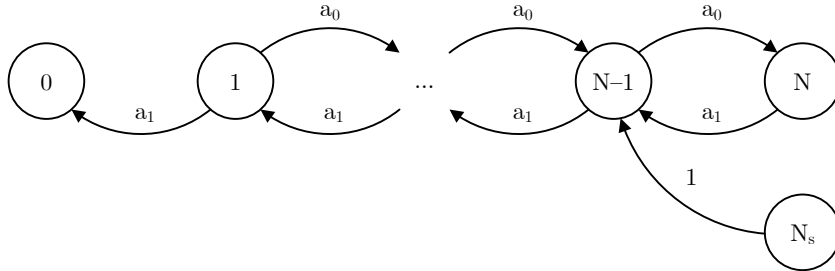


Figure 2.2: Markov Chain [JH09]

according to specific rules. State changes can also take place with a battery current equals to zero which gives a possibility to model the Recovery Effect. Stochastic models have been developed to estimate the life time for pulsed discharge of a battery where they have an average error of about 1% in comparison to electrochemical models. Though, it is unclear how well stochastic models perform for arbitrary currents. Also stochastic models need more computational performance than simple Coulomb Counting or the KiBaM. Thus they are not used in this work.

2.5 Generic Models

A big group of models are generic or mathematical models. Their equations are empirical and the parameters have, beside of the capacity Q , no physical relevance or equivalent. The models are built up of at least to two equations. The first describes the *SoC* as a function of the battery current i . This can be done by Coulomb Counting or a more complex model like the KiBaM. The other equation describes the voltage v as a function of the current i and state of charge *SoC*.

2.5.1 Unnewehr Universal Model

One of the simplest models is the Unnewehr Universal model [HB11]. Its model equation is:

$$v = E_0 + K SoC - R i \quad (2.22)$$

with the parameters E_0 , K and R . The term $E_0 + K SoC$ models a linear open circuit voltage (OCV), whereas R can be seen as internal resistor of the battery. Unnewehr also made R dependent on the SoC and suggested

$$R = R_0 + K_R SoC, \quad (2.23)$$

with the two empirical constants R_0 and K_R .

The Unnewehr Universal model is only valid in the middle SoC -region, where the open circuit voltage curve is flat and can be approximately described as a line. Hence, the Unnewehr Universal model is not suited as a model for this work.

2.5.2 Nernst Model

The Nernst model is mentioned several times in the literature (e.g. [HB11] or [ZQD09]) where it is referenced direct or indirect to the homepage of Thermoanalytics [10]. But at the homepage is no evidence of the model or its author. The model equation is

$$v = E_0 + c_1 \ln(SoC) + c_2 \ln(1 - SoC) - R i \quad (2.24)$$

with the parameters E_0 , c_1 , c_2 and R . The Nernst model has its name probably from the Nernst equation which relates the electrode potential to the reaction quotient of the chemical reaction in the electrode [12]:

$$E_{\text{cell}} = E_{\text{cell},0} - \frac{RT}{zF} \ln Q \quad (2.25)$$

where $E_{\text{cell},0}$ is the standard cell potential, R the universal gas constant, T the temperature in Kelvin, z the number of moles of electrons transferred in the cell, F the Faraday constant and Q the reaction quotient. The voltage of a cell is the difference of the two potentials of the electrodes:

$$v = E_{\text{cathode}} - E_{\text{anode}}. \quad (2.26)$$

The step from Equation 2.26 and 2.25 to 2.24 and replacing the two reaction quotients by SoC and $1 - SoC$ was not found in the literature. The Nernst model describes the open circuit voltage qualitative correct since the voltage drops when the battery is almost empty, is then quite flat when the SoC is between 0.3 and 0.7 and rises when the battery is almost full.

The Nernst model does not consider dynamic effects. Hence it is not a suitable model for this work. But it is a feasible open circuit voltage model for equivalent electrical circuit models (see Section 2.6).

2.5.3 Shepherd Model

The Shepherd Model [She65b, She65a] is a model which was developed for constant discharge of a battery. Its equation is

$$v = E_0 - R i - K \frac{Q}{Q - it} i + A e^{-B \frac{it}{Q}} \quad (2.27)$$

with the constant parameters E_0 , R , K , A , B . The variable it is defined as

$$it := \int_0^t i(\tau) d\tau. \quad (2.28)$$

Shepherd named the term $K \frac{Q}{Q-it} i$ polarization voltage v_p which he deduced from the unbalanced current density distribution at the electrodes. Together with R it models a state of charge dependent internal resistor which increases hyperbolic for decreasing SoC . The term $A e^{-B \frac{it}{Q}}$ models the gain of open circuit voltage when the SoC increases. Shepherd also suggested to add $C \cdot it$ to the equation if the open circuit voltage model is not accurate enough. Although the model was developed more than 50 years ago, Li and Ke [LK11] described it as “the best known voltage-current model for constant-current discharge”.

Shepherd developed his model for batteries which are fully charged. So Equation 2.28 should be adapted to

$$it := \int_0^t i(\tau) d\tau + it_{\text{init}} \quad (2.29)$$

when modeling batteries which are partially discharged. Comparing the Equations 2.4 and 2.29 gives

$$it = Q(1 - SoC) \quad (2.30a)$$

$$it_{\text{init}} = Q(1 - SoC_{\text{init}}). \quad (2.30b)$$

Now Equation 2.27 can be rewritten as

$$v = E_0 - Ri - K \frac{1}{SoC} i + A e^{-B(1-SoC)}. \quad (2.31)$$

Since the Shepherd Model does consider dynamic effects, the model is not accurate enough for this work. But it is the basis of the Modified Shepherd Model which will be described in Section 2.5.5.

2.5.4 Zheng Model

Zheng et al. [ZQD09] developed a battery model from the measurement of an OCV curve and the battery voltage response to a current pulse.

The authors divided the battery voltage in three parts.

1. An open circuit voltage v_{oc} , which is a function of the SoC . As OCV model they used a combination of Unnewehr Universal, Nernst und Shepherd Model:

$$v_{oc} = c_0 + c_1 \ln(SoC) + c_2 \ln(1 - SoC) + c_3 \frac{1}{SoC} + c_4 SoC \quad (2.32)$$

which is according to the authors a valid model between $0.02 \leq SoC \leq 0.98$. The OCV model has a maximal error of 2V which is less than 0.7%. The voltage drop ΔV_{SoC} in Fig. 2.3 is caused by a decrease of the SoC during the discharge pulse, resulting in a lower OCV.

2. An ohmic drop voltage v_{Ω} , caused by the internal ohmic resistance of the battery (ΔV_{Ω} in Fig. 2.3) and calculated by:

$$v_{\Omega} = R_{\Omega} \cdot i. \quad (2.33)$$

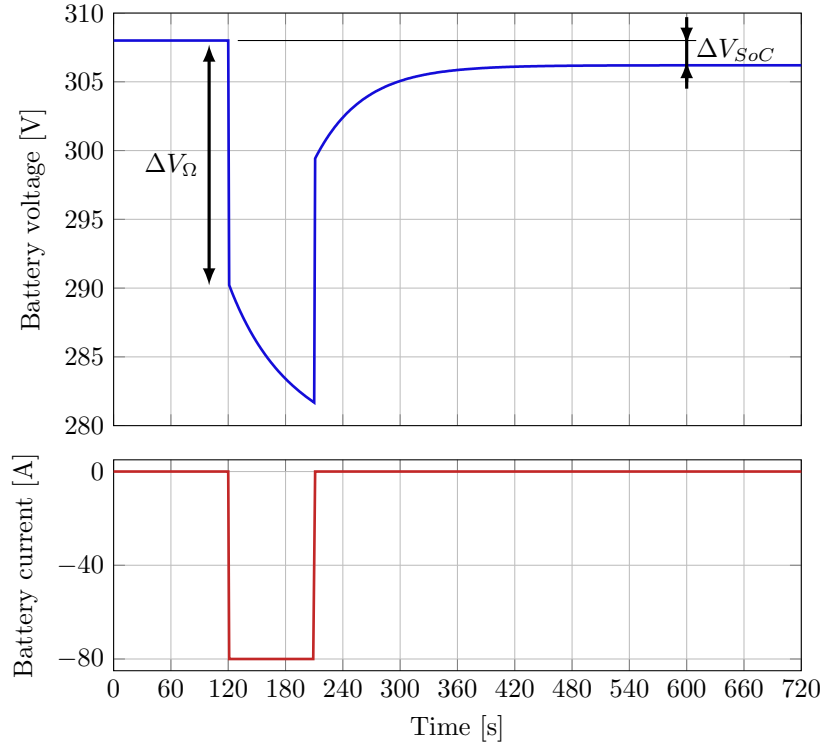


Figure 2.3: Current step response of the battery in [ZQD09]

3. A dynamic voltage called polarization voltage v_p . Zheng et al. model it with an exponential function. Following equation describes the polarization voltage for the discharge pulse of Fig. 2.3:

$$v_p(t) = v_p(t_0) e^{-\frac{t-t_0}{\tau_2}} + R_p i \left(1 - e^{-\frac{t-t_0}{\tau_1}} \right). \quad (2.34)$$

Their paper also consists of a general equation for the polarization voltage, but own MATLAB/SIMULINK simulations were not able to reproduce the measured voltage response with this equation. A valid general equation would be

$$\dot{v}_p = \frac{1}{\tau} (R_p i - v_p) \quad (2.35)$$

with

$$\tau = \begin{cases} \tau_1 & i < 0 \\ \tau_2 & \text{else} \end{cases}. \quad (2.36)$$

Although their general equation, or its description, is not correct, the polarization effect is a very important dynamic battery effect. It shows that the battery voltage is not only a function of the instantaneous current, but also from the past and it needs at least one ordinary differential equation to model the dynamic effects of a battery. The polarization effect is also considered in the Modified Shepherd Model and in equivalent electrical circuit models (see Section 2.6).

2.5.5 Modified Shepherd Model

Tremblay et al. [TDD07, OD09] enhanced the Shepherd Model for modeling not only constant discharge, but also charge and dynamic current. The equation of the so called Modified Shepherd Model (MSM) is

$$v = E_0 - R i - K \frac{Q}{Q - it} it - R_{\text{pol}} i^* + A e^{-\tilde{B}it} \quad (2.37)$$

with $\tilde{B} = \frac{B}{Q}$. They added the term $K \frac{Q}{Q - it} it$ to better represent the *SoC*-OCV characteristics and extended $K \frac{Q}{Q - it}$ to R_{pol} . The so called polarization resistance depends on the current direction and given by

$$R_{\text{pol}} = \begin{cases} K \frac{Q}{Q - it} & i^* > 0, \text{ discharge} \\ K \frac{Q}{it - \tilde{\lambda}Q} & i^* < 0, \text{ charge} \end{cases} \quad (2.38)$$

The polarization resistance models the rapidly voltage increase when the battery reaches the full charge. When the battery is charged, the polarization resistance increases until the battery is almost fully charged. This can be described by:

$$R_{\text{pol}} = K \frac{Q}{it - \tilde{\lambda}Q} \quad (2.39)$$

which is valid for lead-acid and Li-ion batteries. Experiments have shown, that $\tilde{\lambda} \approx 0.1$ [OD09].

The variable i^* stands for the low pass filtered current i . Tremblay et al. use without further comments a simple first order low pass filter with the differential equation

$$\tau \frac{di^*}{dt} + i^* = i. \quad (2.40)$$

Substituting it with *SoC*, Equation 2.37 can be rewritten as

$$v = E_0 - R i - K Q \frac{1}{\text{SoC}} - R_{\text{pol}} i^* + A e^{-B(1-\text{SoC})} \quad (2.41)$$

where the constant $E_0 + K Q$ is merged to one constant E_0 , and

$$R_{\text{pol}} = \begin{cases} K \frac{1}{\text{SoC}} & i^* > 0 \\ K \frac{1}{\lambda - \text{SoC}} & i^* < 0 \end{cases} \quad (2.42)$$

with $\lambda \approx 1.1$.

The authors explicitly mention some model assumptions respectively simplifications. They are valid also for the other generic models described above, but their authors didn't mention them explicitly:

- The internal resistance R is constant during charge and discharge and independent of the amplitude of the current. This is not true for real batteries, as for example shown in [LBK11], but still yield good results. Considering these issues would also increase the configuration effort a lot. Beside this the polarization resistance also can be counted to the internal resistance leading to a *SoC*- and current direction dependent internal resistance.

- The capacity is independent of the current rate. Thus, the Rate Capacity Effect is not considered. A possible solution would be replacing Coulomb Counting with the KiBaM (see Section 2.4.4), which covers the Rate Capacity and the Recovery Effect.
- The batteries behavior is not affected by temperature. This is a common simplification.
- The self discharge of the battery is not represented. As mentioned in Section 2.4.3 this could be considered by an additional current in the Coulomb Counting formula.
- The battery has no memory effect. This is a valid assumption for lead-acid and Li-ion batteries, but not for NiMH and NiCd batteries [OD09].

On the whole, the MSM fulfills the requirements of this work for a battery model: it seems not too computational expensive, with i^* the model considers dynamic effects, the model has only a few parameters whose parametrization effort seems sufficiently low. Therefore the MSM will be used and analyzed further on in this work.

2.5.6 Neural Networks

A complete different approach to model the behavior of batteries are artificial neural networks [CBT11, LWY09, TC08]. Neural networks use measurement data to teach the model the behavior of the battery. After this learning phase neural network behaves similar to the battery. The quality of the model depends on the size of the network, e.g. [CBT11] used 9600 knots, and the type of learning data.

Although there are some papers about this, modeling batteries with neural networks seems not very common. Neural networks for modeling batteries have a few disadvantages. For example, it is very difficult to predict the behavior of the model for a specific battery current, because the neural network is almost a black box. There is no intuitive understanding of the model. Also subsequent extensions like temperature dependency are complex to implement. Another disadvantage is the high memory requirement like almost 10000 knots for a constant discharge. Therefore neural networks will not be considered any more in this work.

2.5.7 Charge-Discharge Hysteresis

A battery voltage has a small charge-discharge hysteresis. For Li-ion batteries this can be neglected, whereas the voltage hysteresis can play a role for lead-acid, NiMH (Nickel metal hydride) and NiCd (Nickel Cadmium) batteries [OD09]. The authors use the equation

$$\dot{v}_{\text{hys}}(t) = c_0 |i| (-v_{\text{hys}}(t) + c_1 h) \quad (2.43)$$

with the parameters c_0 and c_1 and $h = 1$ for charge and $h = 0$ for discharge. The voltage v_{hys} replaces the term $A e^{-B(1-SoC)}$ at the MSM. Some other hysteresis models can be found in [HLP12]. As this work uses only measurement data from a Li-ion battery, hysteresis effects are not longer considered.

2.6 Equivalent Electrical Circuit Models

Since batteries are used in electrical circuits, it's an obvious approach to model a battery with electrical circuit components. These so called equivalent electrical circuit (EEC) models are widely used, discussed, analyzed and extended in the literature. EEC models have the advantage, that they offer an intuitive understanding of how the battery voltage will react for a given current. Also such a battery model can be easily implemented within the SIMULINK SimPowerSystems Toolbox, in SPICE (Simulation Program with Integrated Circuit Emphasis) models or other common simulation environments.

The simplest way would be the combination of an ideal voltage source and an internal resistor (see Fig 2.4a). This is only valid at a small area around an operating point near $SoC \approx 0.5$,

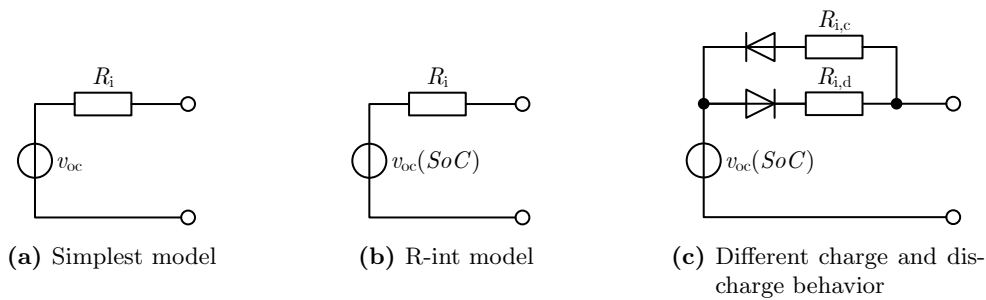


Figure 2.4: Various EEC models

where the SoC vs. OCV-curve is quite flat. To increase the model accuracy the constant voltage source has to be replaced by a state of charge dependent one (see Fig. 2.4b). The combination of a voltage source and an internal resistance is sometimes called “R-int”-model. The calculation of the SoC can either be done by a mathematical model like the Coulomb Counting or via an electrical circuit. Therefore a capacitor C_Q with the capacity Q is charged and discharged with the battery current i (see Fig. 2.5). Its voltage is

$$v_{C_Q} = v_{C_Q,init} - \frac{1}{C_Q} \int_0^t i(\tau) d\tau \quad (2.44)$$

which is equal to the SoC and equivalent to Coulomb Counting (cf. Equation 2.4). Constant self discharge can be modeled with a resistor R_{SD} parallel to C_Q .

The SoC -OCV relationship can result either from a look-up table with measurements or a mathematical model. Using a mathematical model, EEC models are equivalent to some of the generic models described above. For example the Nernst model and its corresponding Equation 2.24 can be directly transformed into an EEC model.

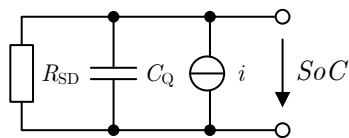


Figure 2.5: Modeling of SoC calculation in an EEC model

Batteries have slightly different behavior, depending on whether the battery is charged or discharged. Adding diodes for a charge and discharge branch can cope with this issue (see Fig. 2.4c), as for example shown in [HB11].

Batteries have dynamics effects like the polarization effect or diffusion. These effects are modeled with one or more RC-elements (see Fig. 2.6). The RC-elements are sometimes assigned to elec-

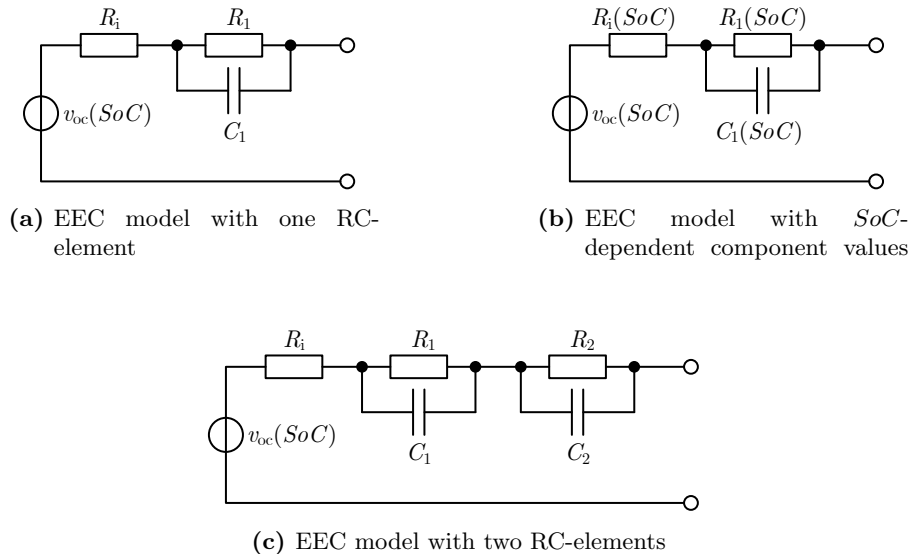


Figure 2.6: Various dynamic EEC models

trochemical effect of the battery: The first RC-element models the polarization effect, whereas the second represents diffusion [HLK10]. Of course, more RC-elements increase the accuracy (cf. Fig. 2.6c), but also increase the calibration effort and the computation time. [ECKF13] and [HLP12] come to the conclusion, that the additional effort to parametrize a second or a third RC-element is often not worth the comparatively small gain in accuracy. In this work, where the low execution time and easy parametrization are demanding requirements, EEC models with more than one RC-element are not longer evaluated. Of course, a charge/discharge dependency can also be implemented for models with RC-elements as it is for example done in [LBK11].

A higher impact on the accuracy of the model is achieved by making the resistor and capacitor value not constant but dependent on the *SoC* (see Fig. 2.6b). Strictly speaking, the models are no equivalent electrical circuits any more, because there are no resistors and capacitors whose values are a function of an online calculated value, but this is not an issue because the circuit is anyway only a simulation model. Similar to the OCV, the *SoC* dependency can be implemented by look-up table [ECKF13] or using mathematical functions [LBK11, CRM06, ZCSA10, TSK12, KK08]. Fig. 2.7 exemplarily shoes the component values of the *ePLB C020b* Li-ion battery mentioned in Section 1.4.3.

The next subsections will present three EEC models, beginning with a very detailed one, then an easier one and finally a hybrid battery model which uses the KiBaM instead of the usually used Coulomb Counting.

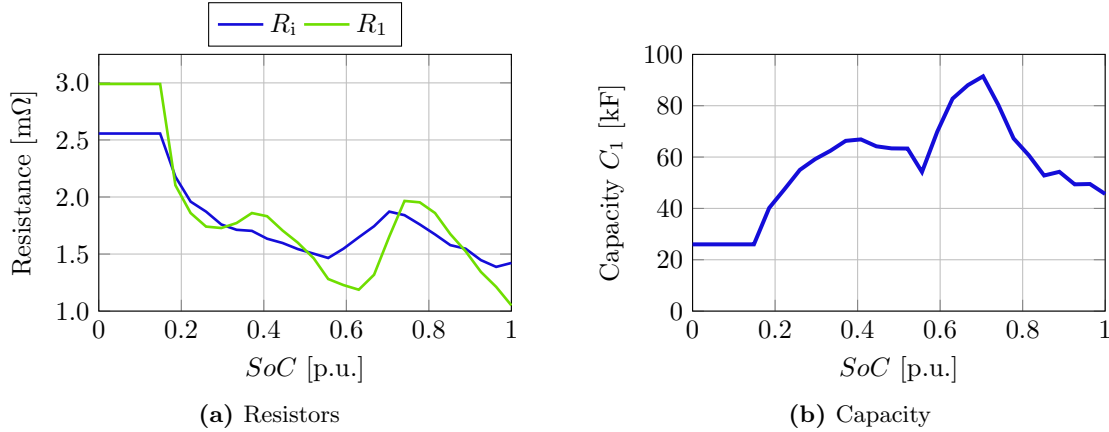


Figure 2.7: Component values as function of the SoC of a $ePLB C020$ Li-ion battery

2.6.1 Lam's Model

Lam et al. [LBK11] developed a circuit based battery model for modeling the $LiFePO_4$ battery of an electric vehicle (EV). They proposed an EEC model with two RC-elements (cp. Fig. 2.6c) whose values depend on the SoC . They neglected self discharge since this effect is very low for the investigated $LiFePO_4$ battery. Their OCV equation is

$$v_{oc} = a_1 e^{-a_2 SoC} + a_3 + a_4 SoC + a_5 e^{-\frac{a_6}{1-SoC}} \quad (2.45)$$

which was considered as temperature independent. Lem et al. modeled the resistors and capacitors to be dependent on SoC , current direction, C-rate and temperature. The internal resistor was modeled with

$$R_i(SoC) = b_1 SoC^4 + b_2 SoC^3 + b_3 SoC^2 + b_4 SoC + b_5 \quad (2.46)$$

with different parameters b_i for charge and discharge. For temperatures above $20^\circ C$, Equation 2.46 is extended to

$$R_{i,high}(SoC, T) = (b_1 SoC^4 + b_2 SoC^3 + b_3 SoC^2 + b_4 SoC + b_5) b_{t1} e^{\frac{b_{t2}}{T-b_{t3}}} \quad (2.47)$$

where T is the temperature in Kelvin and again with different parameters b_{ti} for charging and discharging. For temperatures below $20^\circ C$, Equation 2.46 is extended to

$$R_{i,low}(SoC, T) = (b_{t1} T^3 + b_{t2} T^2 + b_{t3} T + b_{t4}) SoC^4 + b_2 SoC^3 + b_3 SoC^2 + b_4 SoC + b_{t5} e^{\frac{b_{t6}}{T-b_{t7}}} \quad (2.48)$$

The C-rate dependency for high temperatures is modeled by

$$R_{i,high,C}(SoC, T, C_{rate}) = \begin{cases} R_{i,high}(SoC, T) (g_{i1d} C_{rate}^{g_{i2d}} + g_{i3d}) & \text{discharge} \\ R_{i,high}(SoC, T) (g_{i1c} (C_{rate}^{g_{i2c}} - 1) SoC + C_{rate}^{g_{i3c}}) & \text{charge} \end{cases} \quad (2.49)$$

with similar formulas for low temperature. This leads to 47 parameters for the SoC dependency, 117 additional parameters for the temperature dependency and six for the C-rate. Fig. 2.8 shows

the values of the component values for charge and discharge and exemplify the temperature dependency of the internal resistor. The equations in the paper and the according parameters, which are the basis of Fig. 2.8, do not exactly reproduce the figures from the papers, but the principle statements of the figures are the same.

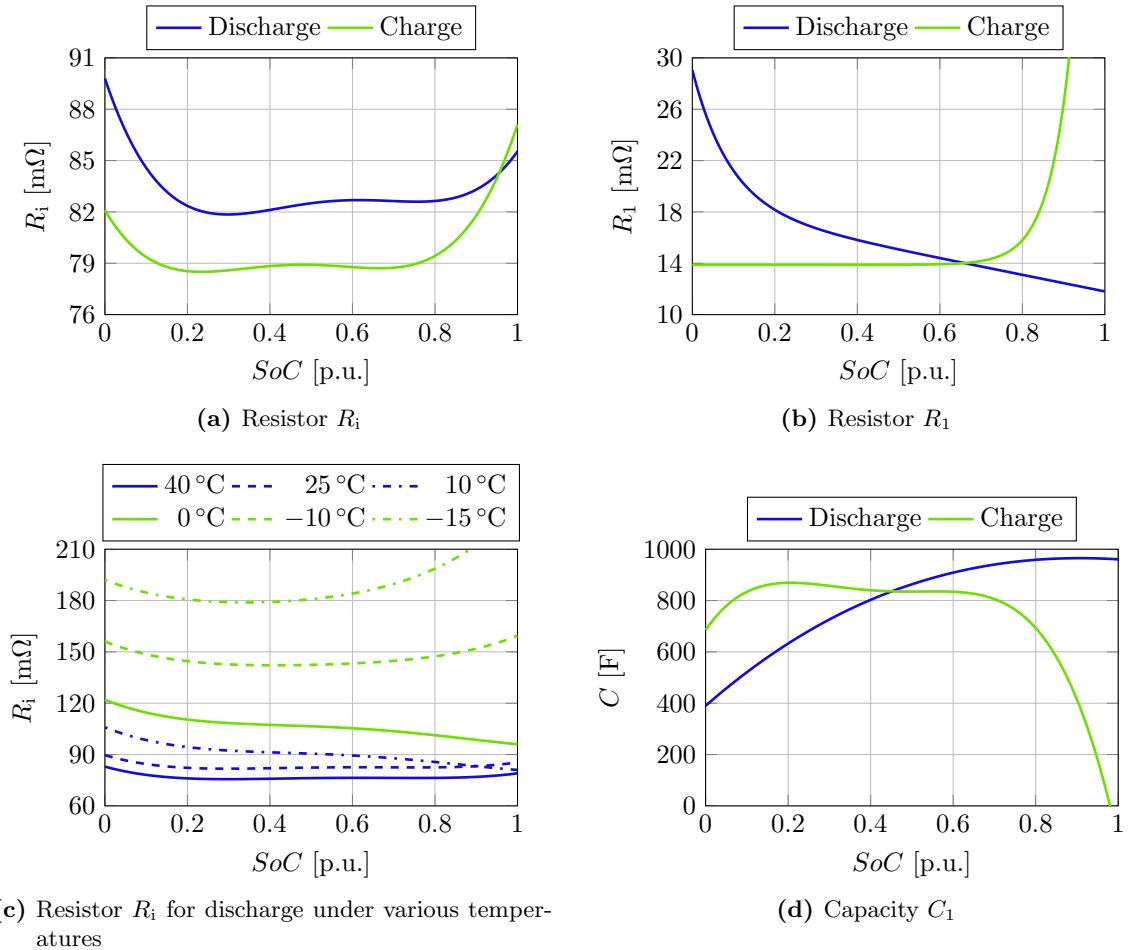


Figure 2.8: Resistor and Capacity values of the Lam model for various conditions [LBK11]

With this model, the battery voltage has a maximum error of 11 mV for a single cell (0.3% of the nominal voltage 3.3 V) at normal ambient temperature for both, charge and discharge. The error rises a little bit for low and high temperatures, but remains still quite small. It is obvious that even the 47 parameters for the SoC dependency are far too much for a “generic” battery model. But the model can be kept in mind as a reference. Also the OCV-model (Equation 2.45) is a useful model which will be used further on in this work.

2.6.2 Chen and Rincón-Mora's Model

Chen and Rincón-Mora [CRM06] developed a model for a *TCL PL-383562* polymer Li-ion battery. They also proposed an EEC model with two RC-elements, but no temperature dependency and less parameters than Lam [LBK11]. Their OCV-model is

$$v_{oc} = c_0 e^{-c_1 SoC} + c_2 + c_3 SoC + c_4 SoC^2 + c_5 SoC^3. \quad (2.50)$$

The parameters were extracted during discharge, but in contrast to [LBK11] also used for charge. All their resistors and capacities are modeled with an equation in the form of

$$x = a_0 e^{-a_1 SoC} + a_2. \quad (2.51)$$

Fig. 2.9 shows the according component values. It is worth mentioning that although the resistors

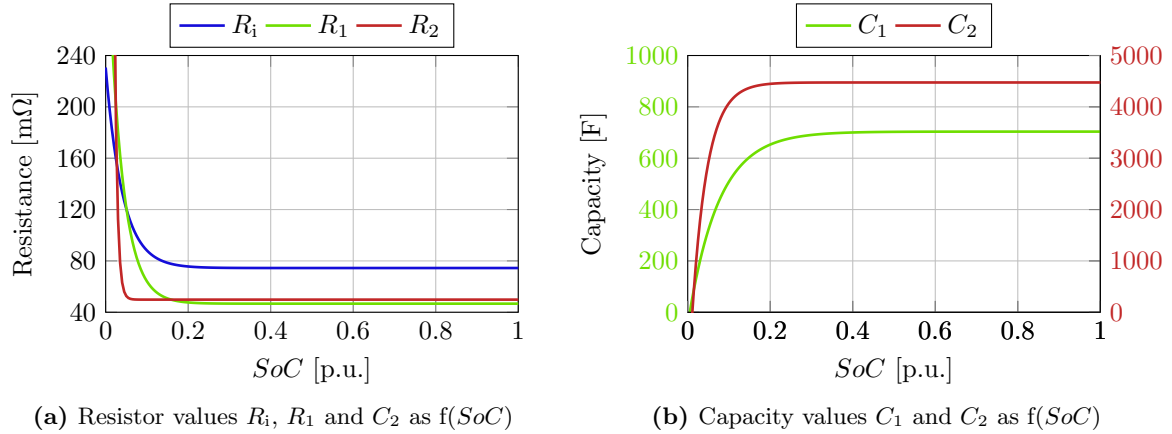


Figure 2.9: Resistor and Capacity values of Chen and Rincón-Mora's model [CRM06]

and capacitors are changing with different state of charges, the time constants $\tau_i = R_i C_i$ are in good approximation independent of the SoC. With SoC dependent resistors and capacitors according to Equation 2.51, their maximum error is 30 mV (0.8% of the nominal voltage 3.7 V), which is more than two times higher than Lam's battery model but still very accurate.

2.6.3 Kim and Qiao's Model

Kim and Qiao [KQ11] proposed a hybrid battery model. They used Rincón-Mora's model but used the KiBaM instead of Coulomb Counting and added a polynomial for $R_i(SoC)$. The hybrid model has 26 parameters; five more than the one from Chen and Rincón-Moras. The authors test their model with constant and pulsed discharge. The hybrid model shows its benefit at consecutive time where the unavailable charge is not negligible any more and small changes of the SoC have a higher impact on the battery voltage. As a result of the hybrid model predicts the runtime more accurate. A quantitative comparison of the errors is not done in the paper. They only mentioned, that the maximum error stayed below 1%.

2.7 Comparison of EEC and Generic Models

Generic mathematical models and EEC models have a different background and approach, but also show some similarities. Every EEC model has beside of the circuit its correspondent (differential) equations. This section will compare this equations with the generic models discussed in Section 2.5.

Some of the generic models, like the Unnewehr Universal, the Nernst and the Shepherd Model, can be directly transformed into an EEC model. They have the form

$$v = v_{oc}(SoC) - R i \quad (2.52)$$

which is also the equation of an EEC model with an internal resistor and no RC-element.

An EEC model with one RC-element has the differential equation

$$\tau \dot{v}_1 + v_1 = R_1 i \quad (2.53)$$

with $\tau = R_1 C_1$. This can be compared with the MSM and the Zheng model. The polarization voltage for discharge of the MSM is

$$\tau \dot{v}_p + v_p = K \frac{1}{SoC} i \quad (2.54)$$

and Zheng et al. used

$$\tau \dot{v}_p + v_p = R_p i \quad (2.55)$$

with different τ for charge and discharge. So the structures of the models are quite similar. The MSM can be transformed into an EEC model with *SoC*-dependent component values and Zheng's model into one with *SoC* and current direction dependent components. If the PT1-filter for i^* of the MSM is replaced by a second order filter, the model would behave like an EEC model with 2 RC-elements.

As already mentioned above, are EEC models whose components are functions of calculated variables strictly speaking no electrical circuits any more, but rather mathematical models with the background of an EEC model, although they are still called EEC models in the literature. So from a mathematical point of view EEC models are a subcategory of generic models.

Every mentioned model has a part which describes the OCV-*SoC* relationship. To compare the models their parameters are marked with $par_i > 0$ with an ascending i from 1 to 14. The values of these parameters are inconsiderable, since only the structure of the models are compared. Fig. 2.10 shows the OCV curve of the ePLB C020 Li-ion battery whose measurements will be used further on in this work to validate and evaluate the battery models. The figure shows, that a OCV curve typically has three parts.

1. The first part is usual operating area of a battery. There the OCV curve is quite flat, so the battery behaves more like an ideal voltage source. The area is approximately at a *SoC* between 0.3 and 0.7. To represent this flat part, most of the models (Unnewehr Universal, Shepherd, Zheng, MSM, Lam, Chen and Rinc3n-Mora) the term

$$par_1 + par_2 \cdot SoC. \quad (2.56)$$

Only the Nernst model has this flat part implicit in its combination $par_3 \cdot \ln(SoC) + par_4 \cdot \ln(1 - SoC)$. Hence it is not suitable for battery with a very flat middle OCV-*SoC* curve.

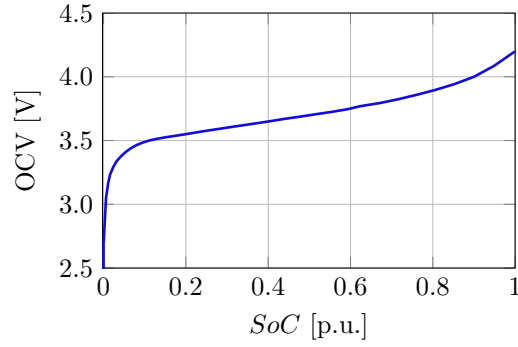


Figure 2.10: Open circuit voltage curve of a “ePLB C020” Li-ion battery

2. The OCV shows a significant break down when the SoC is going to zero. The models have different approaches for this break down. The Nernst model has

$$par_3 \cdot \ln(SoC); \quad (2.57)$$

the Shepherd and Modified Shepherd Model

$$- par_5 \cdot \frac{1}{SoC}; \quad (2.58)$$

Zheng the combination of them

$$par_6 \cdot \ln(SoC) - par_7 \cdot \frac{1}{SoC}; \quad (2.59)$$

while Lam and Chen/Rincón-Mora

$$- par_8 \cdot e^{-par_9 \cdot SoC}. \quad (2.60)$$

This shows a lower limit of the Nernst, Shepherd, modified Shepherd and Zheng model, because

$$\lim_{SoC \rightarrow 0} v_{oc}(SoC) = -\infty, \quad (2.61)$$

which is obviously not true for a real battery. So either every parametrization of the model has to be checked for an lower limit (Zheng et al. mention $SoC > 0.02$) or the equation has to be forced to zero by

$$v_{oc,corr}(SoC) = v_{oc}(SoC) \cdot (1 - e^{-c_0 \cdot SoC}) \quad (2.62)$$

with a sufficiently big constant $c_0 > 0$. This works not for every possible parametrization of the models, because the voltage can go below zero for $SoC > 0$, but it works for every parametrization of a real battery found in the literature.

The limit for Lam and Chen/Rincón-Mora is

$$\lim_{SoC \rightarrow 0} v_{oc}(SoC) = par_1 - par_8 \neq 0. \quad (2.63)$$

Theoretically this is not correct, since a totally empty battery would have zero voltage, but practically this is a valid assumption. The tested *ePLB C02* Li-ion battery, whose OCV curve is shown in Fig. 2.10, has $v_{oc}(SoC = 10^{-7}) = 2.7 \text{ V}$.

3. The third part is the increase of the OCV when the SoC is getting closer to one. Nernst and Zheng model this with the term

$$par_9 \cdot \ln(1 - SoC) \quad (2.64)$$

which is also going to infinity for $SoC \rightarrow 1$. This cannot be corrected by multiplying the equation with a term. So there is an upper limit of the model. Zheng et al. mention $SoC < 0.98$. The Shepherd and Modified Shepherd Model has

$$par_{10} e^{-par_{11}(1-SoC)}; \quad (2.65)$$

Lam has

$$par_{12} e^{-\frac{par_{13}}{1-SoC}}; \quad (2.66)$$

and Chen/Rincón-Mora use a polynom

$$-par_{13} \cdot SoC^2 + par_{14} \cdot SoC^3.$$

This three models have a definite reasonable value for $SoC = 1$. So there are no problems at the modeling.

So not all the models give reasonable values at the boundaries where the SoC is close to 0 and 1. Which of them is the most accurate model will show Chapter 3.

2.8 Summary of the Literature Review

Battery models have not been analyzed before in order to investigate their usability for PHIL simulations with easy to parametrize model. So this was done in this work. Electrochemical models are the most accurate, but too computation intense for current real time systems and the restraint of an execution time below 10 μ s. In contrast, some the discussed generic and EEC models are suitable for the given problem. These models consist of two parts. First the calculation of the SoC and second a part which has as output the battery voltage as a function of the current and SoC .

Coulomb Counting is a simple and effective way to calculate the SoC of a battery which basically integrates the current over time. A possible extension of it is the KiBaM which extends Coulomb Counting by a first order ordinary differential equation and then covers also the Rate Capacity as well as the Recovery Effect.

Out of the generic models, the modified Shepherd is the most promising one. It fulfills the requirements and will be used in this work. Also the EEC models with one RC-element seem promising. Resistor and Capacitor values as function of the SoC increase the accuracy, but also the parametrization effort. Further analysis will clarify if the additional effort is worth the gain in accuracy. In the next sections the Modified Shepherd Model and various EEC models will be parametrized and afterward compared with measurements of a battery.

3 Parametrization of Battery Models

Every model has its parameters and a model is only as good as its parametrization, because no matter how accurate a model could be, it is useless if the model is wrongly parametrized. Contrary, as mentioned in Section 1.4 and 2.1, the configuration effort of the model should be as low as possible since the focus of this work is a generic battery model. With the class of equivalent electrical circuit (EEC) models and the Modified Shepherd Model (MSM) two suitable models were presented in Chapter 2. This chapter will discuss their parametrization using two different ways.

There are several ways to parametrize a battery model. One is using only data sheet values and experience values. This is of course the least accurate method, but it needs no existing hardware, no experiments, and is hence a generic and cost-effective method. As mentioned at the requirements of the battery model, this would be the preferred method. An analysis will be done in this chapter to see how accurate it is and if this method is sufficiently accurate.

Another method is using experiments dedicated for the parametrization of a battery model. These could be for example the extraction of an open circuit voltage (OCV) curve, applying single current pulses until the battery is empty. This needs the existence of the battery under test, a programmable source and load, and measurement equipment, which is on the one hand very cost-intensive. But on the other side it is the most accurate way.

A third method is using (existing) measurements of complex waveforms, e.g. standardized driving cycles, and then applying optimization algorithms to extract the parameters out of these measurements. This takes a much higher configuration effort than using dedicated parametrization measurements. But if the measurements are already available, this method could be more accurate than using only data sheets. An additional advantage is, that no expensive measurement equipment is necessary. Parametrization using this method was not found during the literature research. So it is not clear if this approach is expedient and since parametrization is not the main focus of this work, this method is not further discussed in this work.

3.1 Parametrization Using Dedicated Measurements

Parametrization with dedicated experiments is the most accurate way but it has the drawback that the battery itself and expensive measurement equipment are necessary. An EEC model consists of a state of charge (*SoC*) dependent voltage source and resistors as well as capacitors.

The voltage source can be parametrized with an OCV-curve or a constant discharge curve of the battery, while the passive components can be parametrized with the response to a current step. If the resistors and capacitors are modeled with a *SoC*-dependency, the current step has to be conducted at various states of charge. The MSM can be parametrized in a similar way.

3.1.1 Parametrization of the OCV of an EEC Model

The *SoC* vs. OCV curve can be determined by discharging a fully charged battery with short current pulses. The amplitude of the current pulses should be low and the length short to avoid the Rate Capacity Effect, while the duration between two pulses should be long to take the Recovery Effect into account.

The literature review (c.f. Chapter 2) revealed a few different OCV-models, which are possible equations for the *SoC*-OCV dependency:

- Nernst model (Equation 2.24):

$$v_{oc} = E_0 + c_1 \ln(\text{SoC}) + c_2 \ln(1 - \text{SoC}) \quad (3.1)$$

- Zheng model (Equation 2.32):

$$v_{oc} = c_0 + c_1 \ln(\text{SoC}) + c_2 \ln(1 - \text{SoC}) + c_3 \frac{1}{\text{SoC}} + c_4 \text{SoC} \quad (3.2)$$

- OCV part of the Modified Shepherd Model (Equation 2.41):

$$v_{oc} = E_0 - K Q \frac{1}{\text{SoC}} + A e^{-B(1-\text{SoC})} \quad (3.3)$$

- Lam model (Equation 2.45):

$$v_{oc} = a_1 e^{-a_2 \text{SoC}} + a_3 + a_4 \text{SoC} + a_5 e^{-\frac{a_6}{1-\text{SoC}}} \quad (3.4)$$

- Chen and Rincón-Mora (Equation 2.50):

$$v_{oc} = c_0 e^{-c_1 \text{SoC}} + c_2 + c_3 \text{SoC} + c_4 \text{SoC}^2 + c_5 \text{SoC}^3. \quad (3.5)$$

The parameters to fit the model with the measurement can be determined by solving the optimization problem

$$\min_{\mathbf{x}} \sum_{k=1}^N (r_k(\mathbf{x}))^2 = \min_{\mathbf{x}} \|\mathbf{r}(\mathbf{x})\|^2 \quad (3.6)$$

with

$$r_k = v_{\text{meas},k}(\text{SoC}_k) - v_{\text{model},k}(\text{SoC}_k, \mathbf{x}). \quad (3.7)$$

where N is the number of measurements, \mathbf{x} is a vector containing the parameters of the model and r_k is the error between the measured value $v_{\text{meas},k}$ at point SoC_k and the value provided by the OCV model $v_{\text{model},k}$.

There are several algorithms to solve this non linear least squares problem and several programs which implement these algorithms. For this work MATLAB and the MATLAB *Curve Fitting Toolbox*

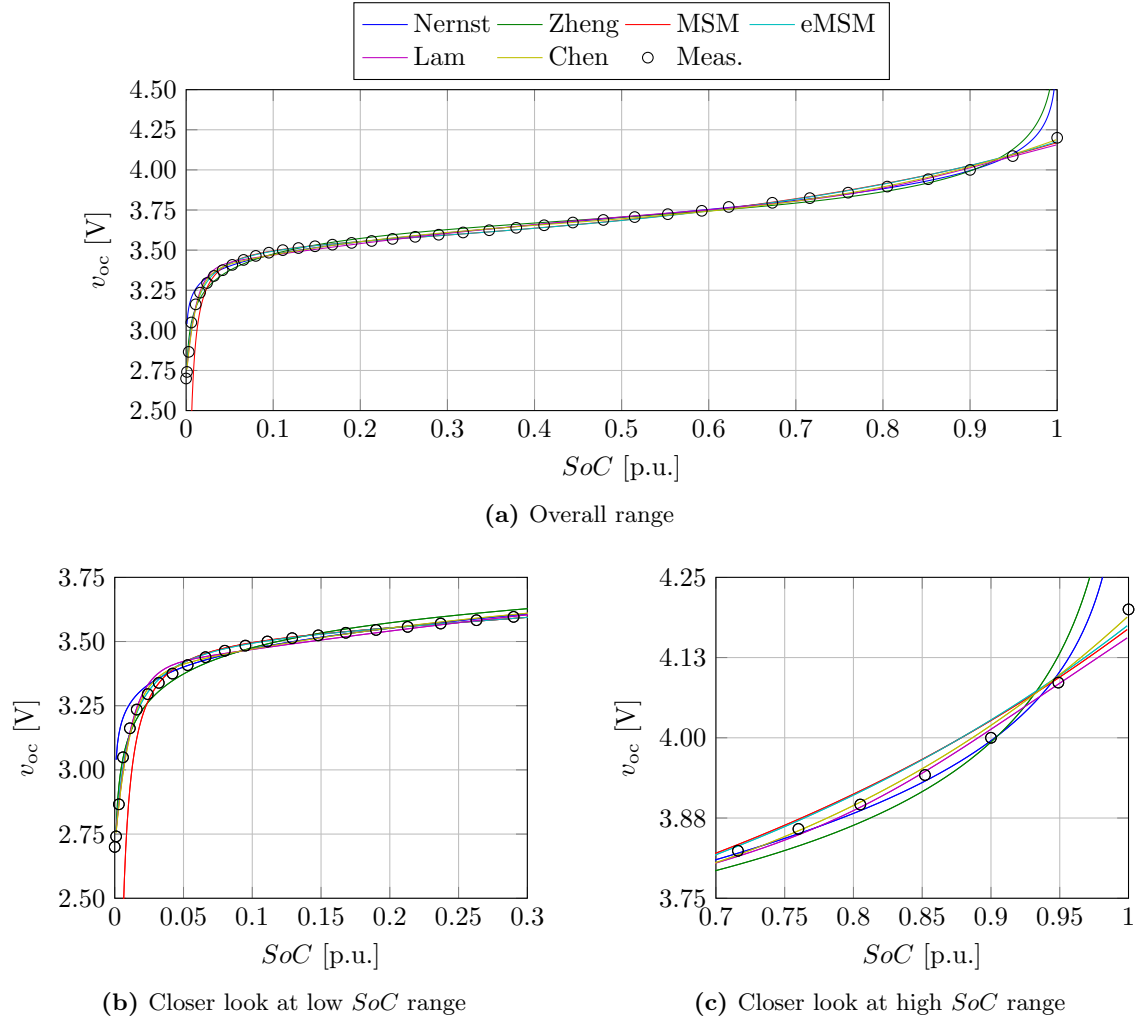


Figure 3.1: Fitting an OCV curve with different OCV models

with the algorithm *Trust-Region* and the setting *Robust on* was used. As already mentioned in Section 1.4.3, the input data are the measured OCV curve of an *ePLB C020* Li-ion battery. The result of the fitting process is shown in Fig. 3.1a.

The figure shows, that the various OCV models are in principle able to emulate the measurements and in the overall range the difference are hardly recognizable. Hence, Fig. 3.1b and Fig. 3.1c shows zooms of the edges of the SoC range. While all models are quite accurate at the middle flat part of the curve ($0.2 < SoC < 0.7$), they get inaccurate at the boundary areas. As discussed in Section 2.7, the Nernst and Zheng model tend to $\pm\infty$ when $SoC \rightarrow 0$ and $SoC \rightarrow 1$.

Also the OCV part of MSM tends to $-\infty$ when $SoC \rightarrow 0$. This issue can be coped by adding an additional parameter SoC_0 leading to

$$v_{oc} = E_0 - K Q \frac{1}{SoC + SoC_0} + A e^{-B(1-SoC)}. \quad (3.8)$$

This model can fit the OCV curve better and the parameter SoC_0 can also be inserted into the MSM leading to

$$v = E_0 - R i - K Q \frac{1}{SoC + SoC_0} - R_{\text{pol}} i^* + A e^{-B(1-SoC)} \quad (3.9)$$

with R_{pol} from Equation 2.42 and i^* from Equation 2.40. This model is from now on called **extended Modified Shepherd Model (eMSM)**. The corresponding open circuit voltage model is also shown in Fig. 3.1.

Beside an optical analysis which OCV is the best, it is possible to define a function whose value is an indicator of how well the model fits the measurements. A common number is the coefficient of determination, denoted R^2 . It is defined as

$$R^2 := 1 - \frac{\sum_{k=1}^N (y_k - f_k)^2}{\sum_{k=1}^N (y_k - \bar{y})^2}, \quad (3.10)$$

where N is the number of measured data values, y_k are the measured values, f_k the predicted and \bar{y} is the mean value of the measurements. If the model exactly matches the measurements R^2 would be 1, while fitting just with the mean value will lead to 0. Table 3.1 shows the R^2 values of the fitting. The OCV-part of the eMSM has the highest R^2 -value and is hence the most

Model	R^2	# of parameters
Nernst	0.9804	3
Zheng	0.9939	5
MSM	0.9723	4
eMSM	0.9986	5
Lam	0.997	6
Chen	0.9984	6

Table 3.1: R^2 -values of different OCV models

accurate. It is even better than the Lam and Chen-model who have one more coefficient. This means not, that the eMSM is more accurate than the other presented models for every battery and every cell technology, but for the tested Li-ion polymer battery. Hence, it will be used as OCV-model in this work further on. The resulting parameters of each model determined in the fitting can be found in the Appendix B.

3.1.2 Parametrization of the Passive Components on an EEC Model

The values of the resistors and capacitor(s) of an EEC model can be determined by analyzing the voltage response to either one discharge pulse, if only one parameter value is necessary, or to a pulsed discharge if a parametrization with SoC dependency is desired. Fig. 3.2 shows the pulsed discharge of the in this work used *ePLB C020* Li-ion battery (cf. Section 1.4.3). For an easier understanding the OCV is subtracted from the battery voltage to see only the voltage that is modeled with the internal resistance and the RC-element. Fig. 3.2 shows that the internal resistance is higher for lower SoC while the time constant $\tau = R_1 C_1$ is roughly constant. There are several ways of extracting the parameters out of these measurement. Two of them are described subsequently.

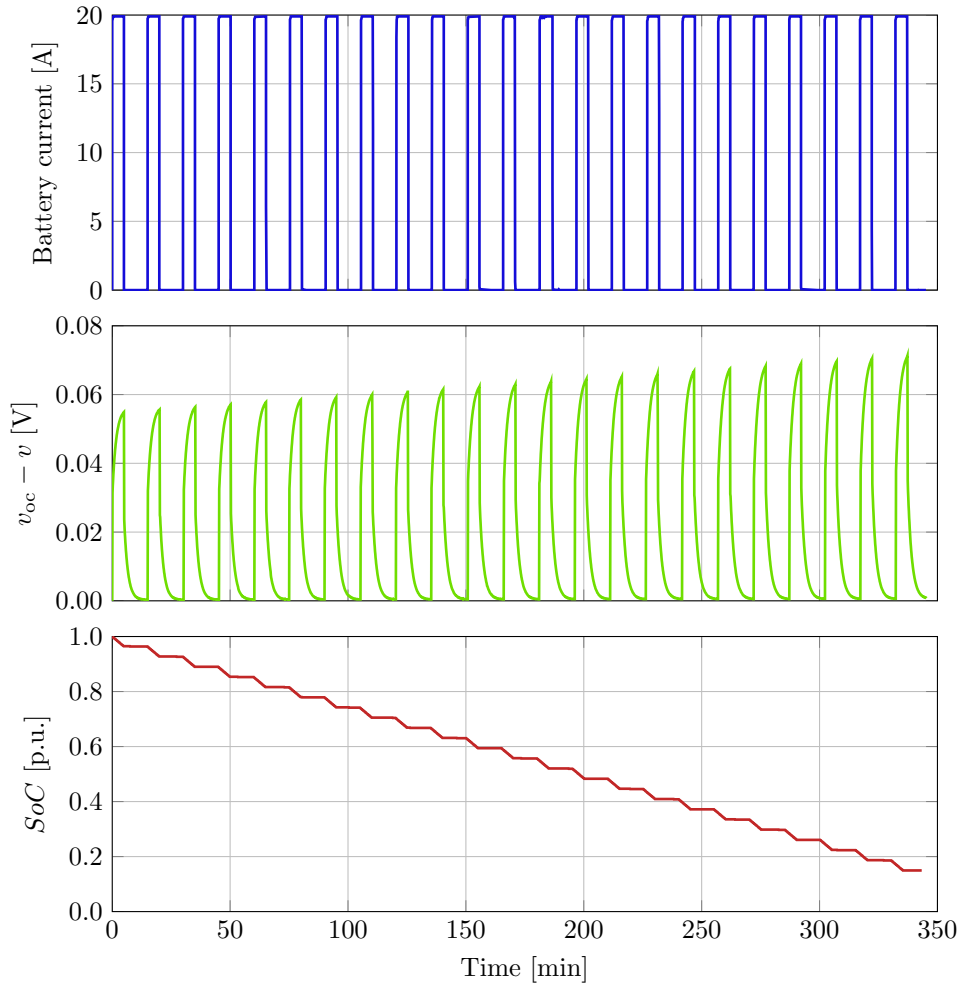


Figure 3.2: Pulsed discharge of an ePLB C020 Li-ion battery

3.1.2.1 Determining the Component Values using an Optimization Algorithm

The first way is to use the data as input data of a model and run an optimization algorithm to determine the parameters. This is the same type of problem as Equation 3.6, but with the current as an input instead of the state of charge. The optimization was done with the *Parameter Estimation* feature of MATLAB/SIMULINK. Fig. 3.3 contains the result of this parametrization process for a current step at an initial $SoC \approx 0.5$. It shows that the model closely matches the measurements. It confirms the statement from [ECKF13], that one RC-element is accurate enough to model the examined Li-ion polymer battery. The optimization process gives the following values for the parameters:

$$R_i = 1.723 \text{ m}\Omega$$

$$R_1 = 1.612 \text{ m}\Omega$$

$$C_1 = 65.268 \text{ kF}$$

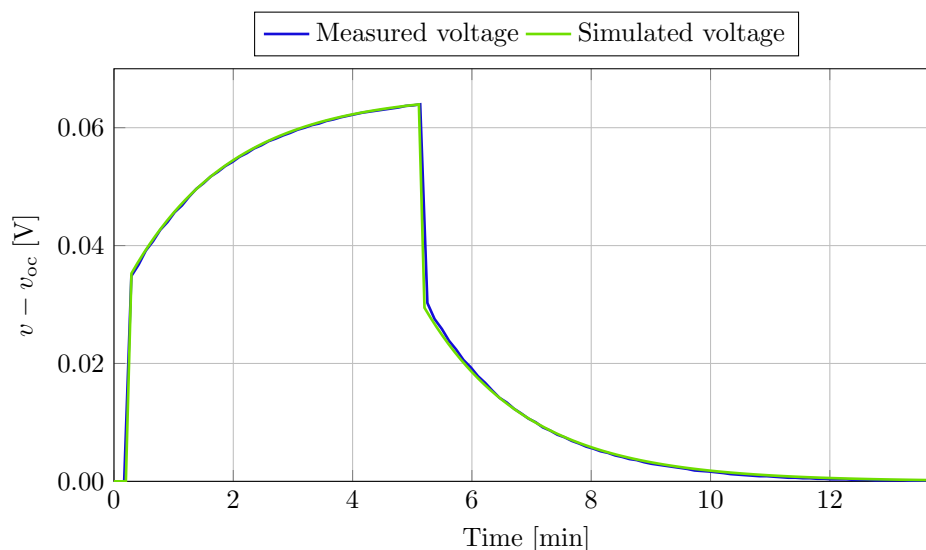


Figure 3.3: Comparison of $v_{oc} - v$ of a discharge current step of a Li-ion battery

This procedure can be done for each current step leading to a look-up table for SoC -dependent component values. Fig. 2.7 already showed this look-up table and Section 3.1.2.3 discusses mathematical models that fit the look-up table values.

This method to parametrize the model can also be used to parametrize an EEC model with two or more RC-elements, models with different values for charging and discharging, or models whose component values depend on the current rate or the temperature. This leads to multidimensional look-up tables and massively increases the configuration effort. Instead of look-up tables, the component values can also be determined by functions of SoC , current direction, current rate, temperature, etc like it is done by Lam et al. [LBK11] in Section 2.6.1.

3.1.2.2 Determining the Component Values with Pen and Paper

Another possibility to determine the values of the parameters is to measure voltages and time constants on a printed sheet with a ruler and then calculate the parameters. This method should only be done for EEC models with one RC-element because it is very difficult to precisely extract more than one time constant manually out of a graph like Fig. 3.3. In theory this method can be used for EEC models with SoC - and current amplitude dependent components, but in practice this makes no sense since the effort is just too high in comparison to the method described in Section 3.1.2.1.

Analyzing Fig. 3.3, the immediate voltage step for the 20 A current pulse is 0.0341 V, so

$$R_i = \frac{0.0341 \text{ V}}{20 \text{ A}} = 1.705 \text{ m}\Omega. \quad (3.11)$$

Metering the time constant $\tau_1 = R_1 C_1$ gives $\tau_1 = 112.5 \text{ s}$ (vs. 105.21 s with the SIMULINK parametrization). At the end of the step (300 s) the RC-element has a voltage of 0.03 V. The transient voltage is

$$v(t) = i R_1 \left(1 - e^{-\frac{t}{\tau}}\right), \quad (3.12)$$

so

$$R_1 = \frac{0.03 \text{ V}}{20 \text{ A} \left(1 - e^{-\frac{300 \text{ s}}{112.5 \text{ s}}}\right)} = 1.644 \text{ m}\Omega \quad (3.13)$$

and

$$C_1 = \frac{\tau_1}{R_1} = 68.422 \text{ kF}. \quad (3.14)$$

The error between the two methods is less than 5% of the parameter values, which is quite accurate for metering with a pen and a ruler.

3.1.2.3 Mathematical Models for the Passive Components

A parametrization with a pulsed discharge of the battery, like described in Section 3.1.2.1 leads to look-up tables for the values of the resistors and capacities at different states of charge. These look-up tables can be fitted with various mathematical functions.

The simplest fit is to calculate the mean value of the look-up table values resulting in an EEC model with constant component values. Calculating the mean values gives:

$$\begin{aligned} R_i &= 1.740 \text{ m}\Omega \\ R_1 &= 1.716 \text{ m}\Omega \\ C_1 &= 59.455 \text{ kF} \end{aligned}$$

which are only slightly different (1% for the internal resistance and 3% for the time constant τ) than the values extracted at a single discharge step at $SoC = 0.5$. So, parametrizing over the full SoC range and then calculating the average of the values is an unnecessarily high effort in comparison to performing just one discharge pulse in the middle SoC range. A possible extension would be a weighted average that is adapted to a particular application. For example, weighting values at lower state of charges less than other values, because a battery is typically rarely used in this operating point.

Beside a constant value, there are almost endless mathematical functions possible that try to fit the look-up table values. Some have been tested within this work. The result of the fitting with the MATLAB *Curve Fitting Toolbox* is shown in Table 3.2 and the according parameters of each fit are listed in Appendix B. The most common approach, fitting with a polynomial, leads to the best fit. So polynomials as mathematical model for the component values are used in Chapter 4 where the models are validated.

Model	R^2 value for		
	R_i	R_1	C_1
linear	0.52	0.53	0.13
polynomial, 4 th order	0.88	0.79	0.75
$c_0 + c_1 \cdot SoC^{c_2}$	0.63	0.62	0.40
$c_0 + c_1 \cdot e^{c_2 \cdot SoC}$	0.77	0.53	0.23
$c_0 \cdot e^{c_1 \cdot SoC} + c_2 \cdot e^{c_3 \cdot SoC}$	0.78	0.62	0.69

Table 3.2: R^2 value of various functions fitting the SoC look-up table

3.1.3 Parametrization of the Extended Modified Shepherd Model

The eMSM can be parametrized quite similar to an EEC model. For the sake of an easier understanding, the Equations 3.9, 2.42 and 2.40 are written again:

$$v = E_0 - R i - K Q \frac{1}{SoC + SoC_0} - R_{\text{pol}} i^* + A e^{-B(1-SoC)} \quad (3.15)$$

$$R_{\text{pol}} = \begin{cases} K \frac{1}{SoC} & i^* > 0 \\ K \frac{1}{\lambda - SoC} & i^* < 0 \end{cases} \quad (3.16)$$

$$\tau \frac{di^*}{dt} + i^* = i. \quad (3.17)$$

The parameters E_0 , K , SoC_0 , A and B of the eMSM can be determined in the same way as the OCV of a EEC model as described in Section 3.1.1. The parameter λ can be assumed to be 1.1 as mentioned in [OD09]. The parameters R and τ can be parametrized with a pulsed discharged as described in Section 3.1.2 so:

$$R = R_i = 1.723 \text{ m}\Omega \quad (3.18)$$

and

$$\tau = R_1 C_1 = 105.21 \text{ s}. \quad (3.19)$$

3.2 Parametrization Using Data Sheet Values

The parametrization of a battery model using only data sheet values and experience values has the advantage that there is no need for expensive batteries and measurement equipment. The drawback is the lower accuracy and that not all parameters can be determined. Data sheets always contain discharge profiles for constant current with various C-rates. Fig. 3.4 shows extracted data of discharge curves for constant discharge at various C-rates of the in this work used *ePLB C020* Li-ion polymer battery (data sheet in Fig A.4 on page 96).

For constant discharge the equation of an EEC model with one RC-element is reduced to

$$v_{\text{model}} = v_{oc}(SoC) - (R_i + R_1) i \quad (3.20)$$

and for the eMSM to

$$v_{\text{model}} = E_0 - \left(R - K \frac{1}{SoC} \right) i - K Q \frac{1}{SoC + SoC_0} + A e^{-B(1-SoC)}. \quad (3.21)$$

This means that it is not possible to determine the parameter C_1 of an EEC model as well as the spreading between R_i and R_1 with the data provided in the data sheet. A possible solution of this problem would be using experience values or a literature research about parametrizations of batteries with the same cell technology. Another solution would be combining the two presented parametrization methods and use the measurement of a current step response to determine R_i , R_1 and C_1 , while the OCV parameter are determined using data sheet information. This combines the advantages and disadvantages of both methods, since the time consuming measurement of an OCV-curve is not necessary.

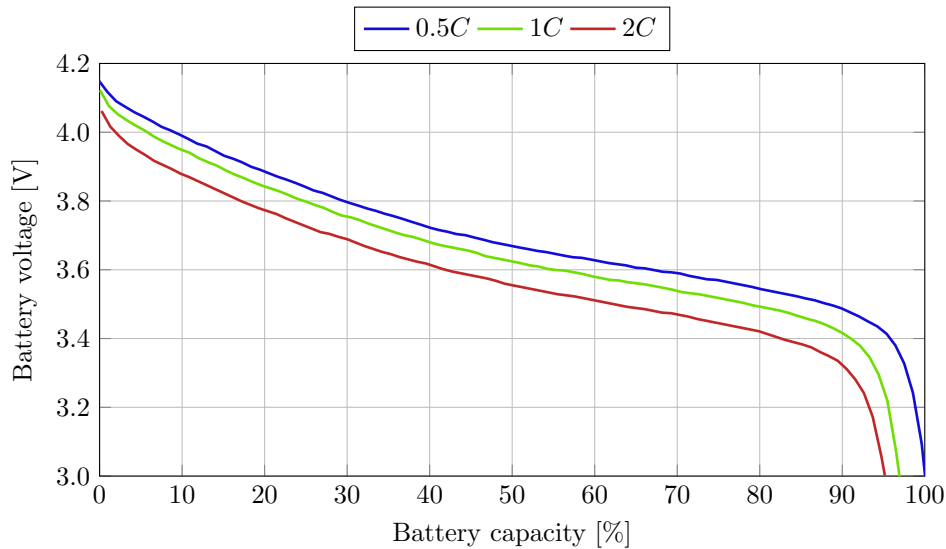


Figure 3.4: Battery voltage of a discharge of a Li-ion battery for various C-rates

The same applies for the parameter τ of the eMSM. A literature review showed, that R_1 and C_1 vary more for different batteries and cell technologies than the time constant τ . Also τ depends less on the SoC than R_1 and C_1 . So using the parameters from other batteries seems to be more appropriate for the eMSM than for the EEC. Hence, this parametrization method is not applicable respectively recommended for EEC models, but for the eMSM.

To parametrize the parameters of the eMSM, the numeric values have to be extracted out of the graph of the data sheet. There are several programs available to do this, e.g. *GetData Graph Digitizer*¹ or *DataThief*². The next step is to determine the parameters of a battery model such that the model fits the curve as accurately as possible. This problem is again an optimization problem in the form of Equation 3.6.

In Fig. 3.4 the abscissa shows the capacity at constant discharge, which is equivalent to the depth of discharge (DoD) in percent where

$$SoC = 1 - DoD.$$

For constant i and unknown R the parametrization is not unique since E_0 and Ri are both independent of SoC . A way to cope this issue is to use the *AC impedance at 1 kHz* $< 3\text{ m}\Omega$ mentioned at the data sheet. Fig. 3.5 shows the measured discharge profiles compared to the discharge modeled with the eMSM. The parameter vector $\mathbf{x} = [E_0, K, SoC_0, A, B]$ was determined using the MATLAB *Curve Fitting Toolbox* and a discharge profile with a C-rate of $0.5C$. The figures shows, that the model is capable of modeling the constant discharge of the battery at a C-rate of $0.5C$, but gets inaccurate for discharges with other C-rates.

Another way to deal with this under-determined optimization problem and is to use several discharge profiles, which led to an optimization problem in the form of

$$\min_{\mathbf{x}} \sum_{l=1}^M \sum_{k=1}^{N_l} (v_{\text{data sheet},k,l}(SoC_k, i_l) - v_{\text{model},k,l}(SoC_k, i_l, \mathbf{x}))^2 \quad (3.22)$$

¹GetData Graph Digitizer, <http://getdata-graph-digitizer.com/>

²DataThief, <http://datathief.org/>

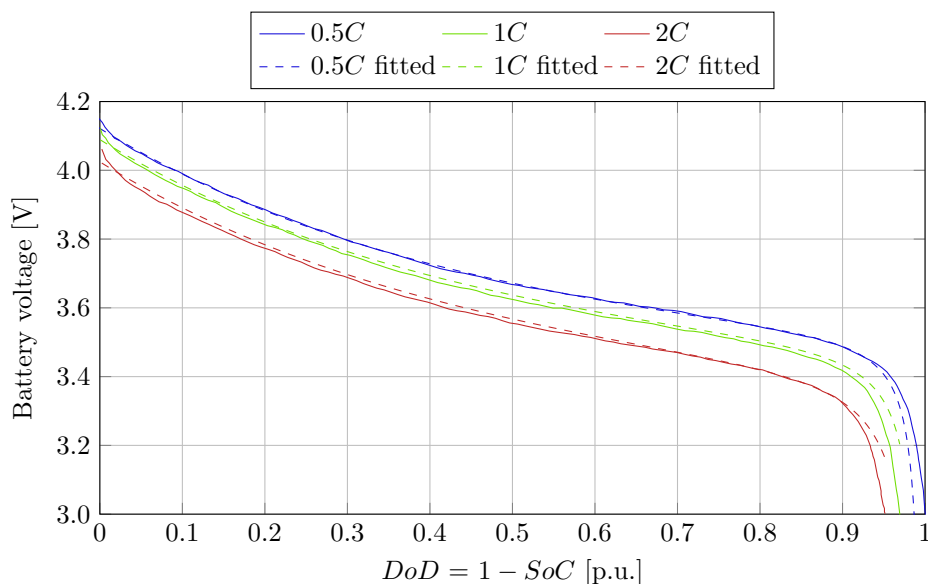


Figure 3.5: Measured discharge curves at various C-rates and eMSM parametrized with a single discharge curve at $0.5C$

where M are the numbers of discharge profiles and i_l is the current of discharge profile l which has N_l extracted points. The parameter vector \mathbf{x} additionally contains the resistor R . Fig. 3.6

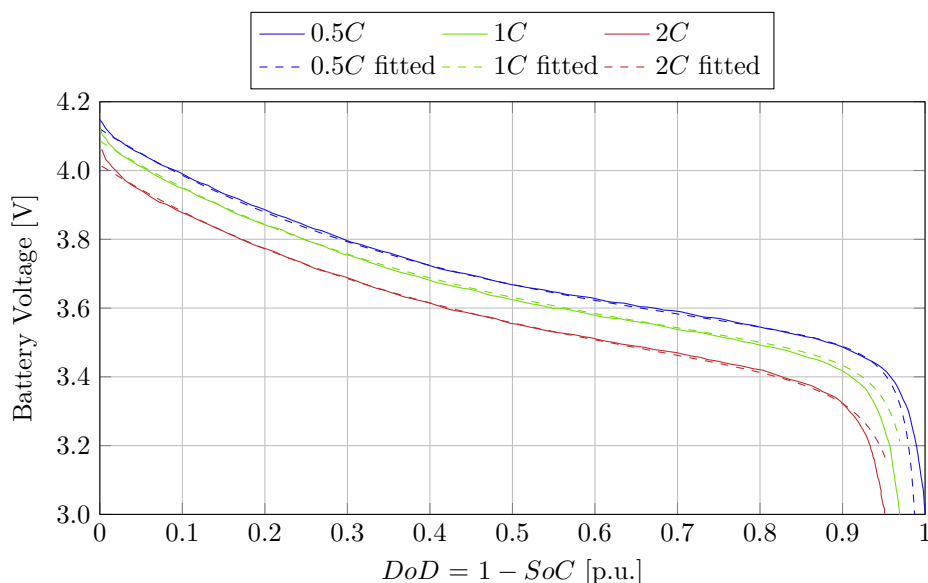


Figure 3.6: Measured discharge curves at various C-rates and eMSM parametrized with all discharge curves according to Equation 3.22

shows a parametrization obtained by solving the optimization problem of Equation 3.22. The figure shows the limits of the model where $SoC \rightarrow 0$ and $SoC \rightarrow 1$. At these areas the model and the measurements differ. Nevertheless, the parametrization has an overall R^2 of 0.999 which is quite accurate. Since other data and another fitting are used than in the fitting of the OCV curve in Section 3.1.1, the R^2 cannot be compared.

Table 3.3 shows a comparison of the parameters for the three presented ways of parametrization. The parameters E_0 , A and B are comparatively similar, while K , SoC_0 and R differ. Analyzing

Parameter	Parametrization using		
	measurements (Section 3.1)	one data sheet curve	multiple data sheet curves
E_0	3.490	3.503	3.508
R	$1.723 \cdot 10^{-3}$	$3 \cdot 10^{-3}$	$3.249 \cdot 10^{-3}$
K	$4.209 \cdot 10^{-4}$	$2.406 \cdot 10^{-4}$	$2.302 \cdot 10^{-4}$
SoC_0	0.010	0.001	0.001
A	0.692	0.656	0.653
B	2.395	2.244	2.296
τ	105.21	-	-

Table 3.3: Comparison eMSM's parameters for different ways of parametrization (parameters without units)

the model equation, the results of the parameter estimation (cf. Equation 3.21) show that this three parameters affect each other and the optimization probably stopped in a local maximum. Increasing R in the term

$$\left(R - K \frac{1}{SoC} \right) i$$

leads to a lower K when minimizing the residuum \mathbf{r} of the optimization and a decreased K in the term

$$K Q \frac{1}{SoC + SoC_0}$$

leads to a decreased SoC_0 . The time constant was estimated based on a short literature review to

$$\tau_{\text{est}} = 100 \text{ s.}$$

In summary, EEC models and the eMSM can be fully parametrized using simple measurement like an OCV-curve and the response to a current pulse. Therefore measurement equipment and power electronic for controlled discharge is necessary. A cheaper and faster, but less accurate method is to use only data which are available from data sheets. This method cannot determine all parameters, so experience values or data from literature review are necessary. The parametrization using data sheet values is only recommended for the eMSM where it has an error of a few percent in comparison to the parametrization by measurements.

3.3 Parametrization of the Kinetic Battery Model

The Kinetic Battery Model (KiBaM) has two parameters: the capacity ration c ($0 < c < 1$), which describes the distribution between the two charge wells and k , which is related to the time constant for balancing processes between the two wells. Kim et al. [KQ11] proposed to discharge a full battery with a very high current to get the parameter c . For $i \rightarrow \infty$ only the charge well y_1 will be emptied and the withdrawn charge is equal to $y_1(0)$ and $c = \frac{y_1(0)}{Q}$. Then k can be determined as the only parameter left by discharging the battery with various discharge currents.

Discharging the battery with very high current close to a short circuit will damage or even destroy the battery [KHHM12]. Therefore this parametrization method is not recommended.

The model can be parametrized in another way. According to the Recovery Effect, the *SoC* and the open circuit voltage increases after a discharge pulse. The voltage increase behaves according to the KiBaM like a PT1-term with the time constant

$$\tau = \frac{1}{k'} = \frac{c(1-c)}{k}.$$

This time constant is in the range of 20 minutes for a Li-ion polymer battery and 5 minutes for a lead-acid battery [KQ11]. The time constant can be extracted from the voltage measurement of this discharge pulse. The voltage shows also the polarization effect (modeled by an RC-term in an EEC model), but since their time constants differ by more than a decade, they can be distinguished. After determining the parameter k' , the parameter c can be extracted from discharge profiles with different C-rates using a least squares optimization. Kim et al. [KQ11] parametrized the KiBaM for a 860 mAh Li-ion polymer cell and a 1.2 Ah lead-acid battery and got $c = 0.925$, and $c = 0.6$ respectively.

Unfortunately, for this work there were no measurements available which would give the possibility to parametrize a KiBaM for the used Li-ion battery. So it will be only parametrized by “trial and error” in Section 4.3.2.4 to show the influences of the Rate Capacity Effect, but is then not considered any more.

3.4 Scaling a Battery Model

A single battery cell has a voltage in the range of a few volts. Typical applications like home storage systems or electric vehicles operate at a voltage a couple of dozen higher. So several cells are connected in series to increase the battery voltage. Also several cells are often connected parallel to increase the capacity of the battery.

If the parametrization of the battery was not done for the whole battery pack, but only for a single cell, the battery model has to be scaled to a desired voltage and capacity. For testing and simulation issues it can also make sense to create a battery with a specific voltage and capacity which has no physical equivalent.

Let v_{des} be the desired battery voltage and Q_{des} the desired capacity. Then two factors can be defined:

$$scale_v := \frac{v_{\text{des}}}{v}, \tag{3.23}$$

$$scale_Q := \frac{Q_{\text{des}}}{Q}. \tag{3.24}$$

For real batteries the two factors can only be natural numbers, but for virtual batteries they can be any positive real number. With these definitions, the eMSM and EEC models can be scaled. For the *SoC* calculation with Coulomb Counting the capacity Q is just replaced by Q_{des} .

3.4.1 Scaling Equivalent Electrical Circuit Models

Connecting several equivalent electrical circuits of a battery in series and parallel lead to a network of identical voltage sources, resistors and capacitors. The resistors and capacitors of one cell model can be concentrated to one impedance and then Thévenin's theorem can be used, to replace this network by a single voltage source and an impedance.

From a circuit-wise point of view there is no difference, if the cells are first connected serial and then parallel (see Fig. 3.7):

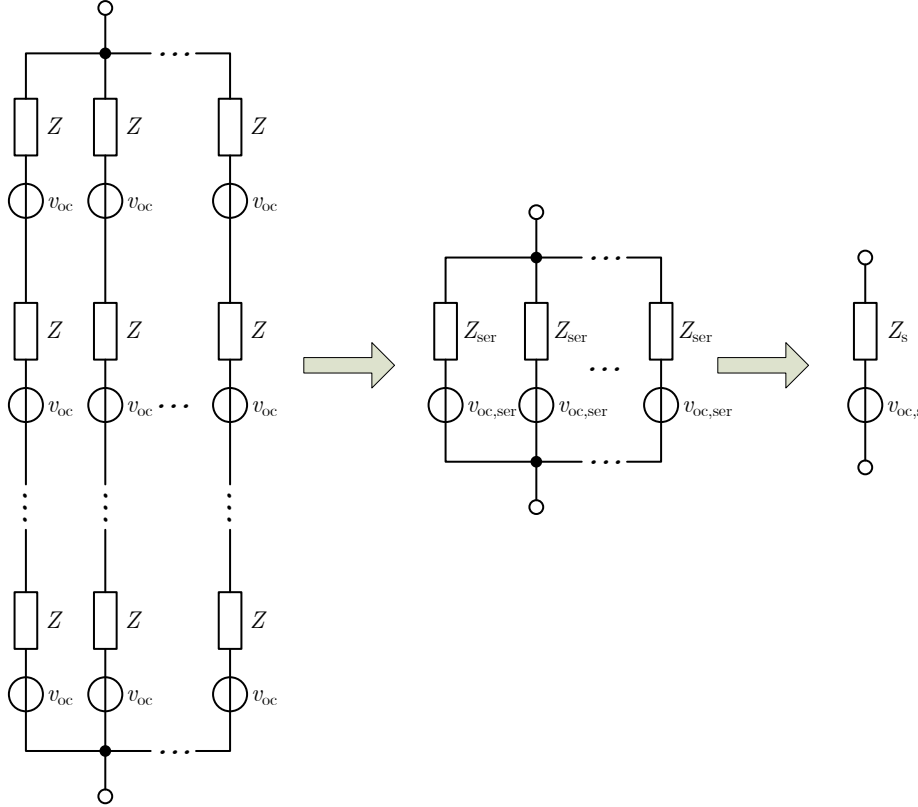


Figure 3.7: Applying Thévenin's theorem on a set of EECs of a battery pack

$$v_{oc,ser} = scale_v v_{oc} \quad \rightarrow \quad v_{oc,s} = v_{oc,ser} = scale_v v_{oc} \quad (3.25)$$

$$Z_{ser} = scale_v Z \quad \rightarrow \quad Z_s = \frac{1}{scale_Q} Z_{ser} = \frac{scale_v}{scale_Q} Z = k Z \quad (3.26)$$

or first parallel and then serial (see Fig. 3.8):

$$v_{oc,par} = v_{oc} \quad \rightarrow \quad v_{oc,s} = scale_v v_{oc,par} = scale_v v_{oc} \quad (3.27)$$

$$Z_{par} = \frac{1}{scale_Q} Z \quad \rightarrow \quad Z_s = scale_v Z_{par} = \frac{scale_v}{scale_Q} Z = k Z \quad (3.28)$$

Applying Thévenin's theorem to Fig. 3.7 or Fig. 3.8 leads to the Equations 3.25 and 3.26, which are used to scale the parameters of the model.

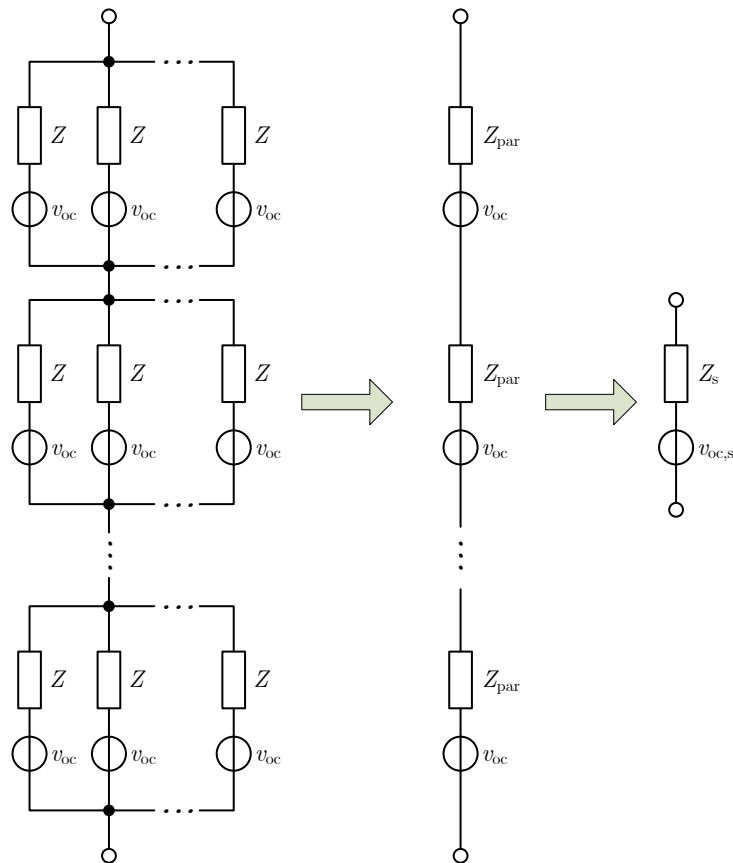


Figure 3.8: Applying Thévenin's theorem on a set of EECs of a battery pack

3.4.1.1 Scaling the Open Circuit Voltage Model

Following Equation 3.25 the OCV is multiplied with $scale_v$. This leads to a scaling of every entry in a look-up table model. Multiplying the model equation of a mathematical model does not always lead to a multiplication of a parameter, for example a_2 and a_6 of the Zheng model and c_1 in Chen/Rincón-Mora. The MSM and eMSM are a little bit different, since they contain the term $K Q \frac{1}{SoC}$ resp. $K Q \frac{1}{SoC + SoC_0}$. Since the capacity is also scaled, the parameter K has to be scaled with

$$K_s = \frac{scale_v}{scale_Q} K.$$

The parameters B and SoC_0 are unchanged.

3.4.1.2 Scaling the Component Values

With Equation 3.26 the values of the scaled resistors and capacities can be calculated:

$$\begin{aligned}
 Z_s &= k Z = k \left(R_i + \frac{R_1}{1 + j\omega R_1 C_1} + \cdots + \frac{R_n}{1 + j\omega R_n C_n} \right) = \\
 &= k R_i + \frac{k R_1}{1 + j\omega R_1 C_1} + \cdots + \frac{k R_n}{1 + j\omega R_n C_n} = \\
 &= k R_i + \frac{k R_1}{1 + j\omega (k R_1) \frac{C_1}{k}} + \cdots + \frac{k R_n}{1 + j\omega (k R_n) \frac{C_n}{k}} = \\
 &= R_{i,s} + \frac{R_{1,s}}{1 + j\omega R_{1,s} C_{1,s}} + \cdots + \frac{R_{n,s}}{1 + j\omega R_{n,s} C_{n,s}}.
 \end{aligned} \tag{3.29}$$

So the scaled resistors and capacities are

$$R_{i,s} = k R_i = \frac{scale_v}{scale_Q} R_i \tag{3.30}$$

$$R_{l,s} = k R_l = \frac{scale_v}{scale_Q} R_l \tag{3.31}$$

$$C_{l,s} = \frac{C_l}{k} = \frac{scale_Q}{scale_v} C_l \tag{3.32}$$

for an EEC model with n RC-elements and $l = 1 \dots n$.

At a real battery which consists of various cells, the resistors and time constants are not the same. But this makes only a little difference as long as the cells are from the same batch and have the same battery age, because the combination of multiple PT1-elements with similar time constants can be described in good approximation with a single PT1 term with the mean value of the time constants (see Fig. 3.9). Although the eleven time constants vary between 50 s and 150 s, there's hardly any difference between the average of them and a single PT1 with the average time constant 100 s.

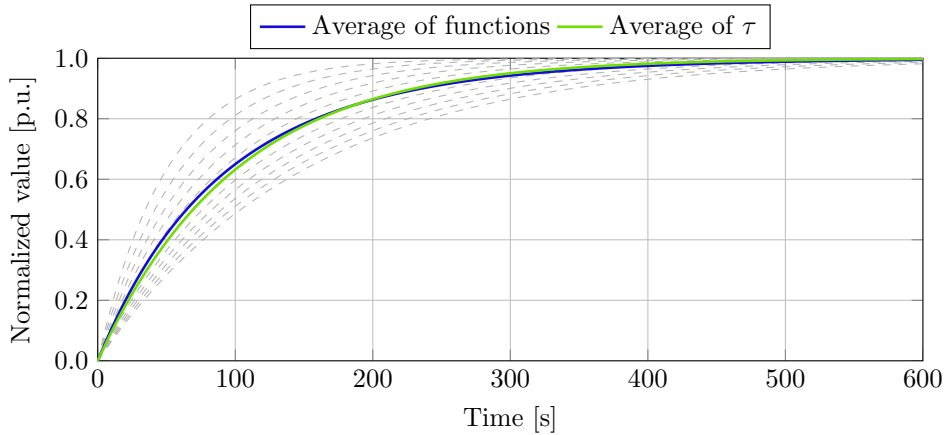


Figure 3.9: Comparison of the average of PT1-elements with different time constants and a PT1 with an average time constant

For more realistic smaller differences between the time constants, the two curves are congruent. So the simplification of combining $scale_v \cdot scale_Q$ identical impedances to one impedance using Equation 3.26 is acceptable.

3.4.2 Scaling the eMSM

As already mentioned in Section 3.4.1, the parameters of the OCV-part are scaled as followed:

$$E_{0,s} = scale_v E_0 \quad (3.33)$$

$$K_s = \frac{scale_v}{scale_Q} K \quad (3.34)$$

$$A_s = scale_v A. \quad (3.35)$$

The internal resistor is scaled to

$$R_s = \frac{scale_v}{scale_Q} R. \quad (3.36)$$

while parameters τ , λ , B and SoC_0 are unchanged.

To summarize, EEC models and the eMSM can easily be scaled from a certain parametrization to a battery with a desired voltage and capacity. Therefore only multiplications and divisions of the parameters with the scale factors to desired voltage and capacity are necessary.

4 Validation of the Battery Models

After a literature review on possible battery models and a discussion on how to parametrize them, the battery models are verified against measurements of a battery. This section starts with necessary error definitions, continues with a description of the software implementation of the models and finally validates the simulation results against measurements of a real battery.

4.1 Error Definition

The usual relative error definition

$$\epsilon = \frac{x_{\text{act}} - x_{\text{ref}}}{x_{\text{ref}}} \quad (4.1)$$

with x_{ref} as the reference value and x_{act} as the actual real value has the big disadvantage that the error tends to infinity if the reference value is close to or exact zero. Then the relative error isn't a meaningful quantity for an error anymore. This is especially important for the analysis of the current set value in Section 5.3. Hence, this work uses another error definition which is familiar with the measuring inaccuracy of measurement instruments. Instead of the current reference value, the absolute error is divided by maximum of the absolute values of the data profile. This leads the error definition

$$\epsilon_x(t_i) := \frac{x_{\text{act}}(t_i) - x_{\text{ref}}(t_i)}{\max_{t_i} |x_{\text{ref}}(t_i)|} \quad i = 1 \dots N \quad (4.2)$$

where N is the number of measurements of the data profile and t_i the time of each measuring point. On the basis of this weighted error, three additional errors can be defined. A maximum weighted error

$$\epsilon_{x,\text{max}} := \max_{t_i} |\epsilon_x(t_i)|, \quad (4.3)$$

a mean weighted error

$$\epsilon_{x,\text{mean}} := \frac{1}{N} \sum_{i=1}^N \epsilon_x(t_i) \quad (4.4)$$

and a weighted root mean squared error (RMS)

$$\epsilon_{x,\text{rms}} := \sqrt{\frac{1}{N} \sum_{i=1}^N (\epsilon_x(t_i))^2}. \quad (4.5)$$

These error definitions will be used in this chapter as well as in the following ones.

4.2 Implementation in Simulink

The models were implemented in MATLAB/SIMULINK because the real time system (RTS) that is used in this work offers the possibility to automatically parse SIMULINK models, convert them in C-code, and then compile and execute it. While the implementation of the extended Modified Shepherd Model (eMSM) is straight forward by implementing the model equations (Equations 3.15 to 3.17), there are various versions of equivalent electrical circuit (EEC) models. The open circuit voltage (OCV) as function of the state of charge (SoC) can be implemented as look-up table or as mathematical function and there are a few possible OCV-models (cf. Section 3.1.1). The passive components of an EEC model can either be constant or a function of the SoC . For EEC models with a SoC -dependence, this dependency can be implemented in two ways. First, by using a look-up table and second by mathematical functions. Look-up tables with a linear interpolation between the values would be the intuitive way, because the parametrization with pulsed discharge directly produces a look-up table, but the literature review showed, that mathematical functions are far more common, e.g. Lam’s Model in Section 2.6.1 or Chen and Rincón-Mora’s Model in Section 2.6.2. The reason for this circumstance is not clear and also no explanation was found during the literature review. A reason could be that the typical linear fitting between two look-up table values results in a non continuously differentiable behavior which can cause troubles for some solvers.

Since this work will be used furthermore at the Energy Department of the Austrian Institute of Technology (AIT), all reasonable EEC model variants and the eMSM were implemented to build a battery model library. These are

- the eMSM and an EEC model with constant component values because they are easily parametrizable battery models,
- EEC models with mathematical functions for the OCV and the component values, because these are the common models in the literature as well as
- EEC models with look-up tables because the Mobility Department of AIT analyzes batteries and parametrizes their models this way.

Also combinations of mathematical functions and look-up tables are possible, which leads to Table 4.1 where all the implemented EEC models are listed. Although this work does not use EEC models with two RC-elements (compare Section 2.6), every model in Table 4.1 was also implemented with two RC-elements due to probable future work.

OCV	Passive components		
	constant	$f(SoC)$	look-up table
$f(SoC)$	✓	✓	✗
look-up table	✓	✓	✓

Table 4.1: Implemented EEC battery models

Both, the eMSM and the EEC models use Coulomb Counting to calculate the SoC . Coulomb Counting was implemented with an resettable integrator, so that the SoC can be reset to certain values during run-time (see Fig. 4.1). If SoC set value is changed, the current value of SoC_{in} is unequal (“ \sim ” in MATLAB notation) to the last one and the integrator will be reset and starts with the initial condition SoC_{in} . The gain-block with 3600 is used to scale between Ah, the

common unit of Q , and As which is the SI-unit. The value of the integrator is limited between 0 and 1, so that the SoC always has valid values.

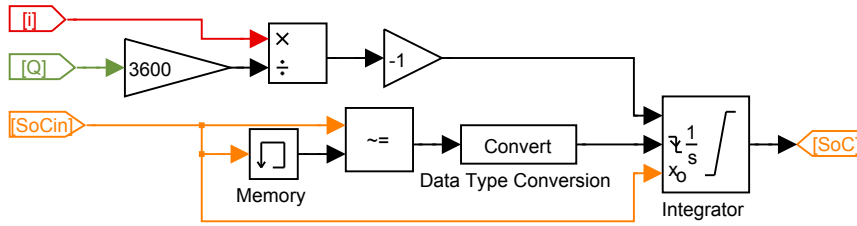


Figure 4.1: Implementation of Coulomb Counting in MATLAB/SIMULINK

The eMSM model was mainly implemented in an *Embedded Matlab Function*, because this was faster and clearer than an implementation with standard SIMULINK blocks. A screenshot is shown in Fig. 4.2. The elements of the “para”-structure are the parameters of the as described

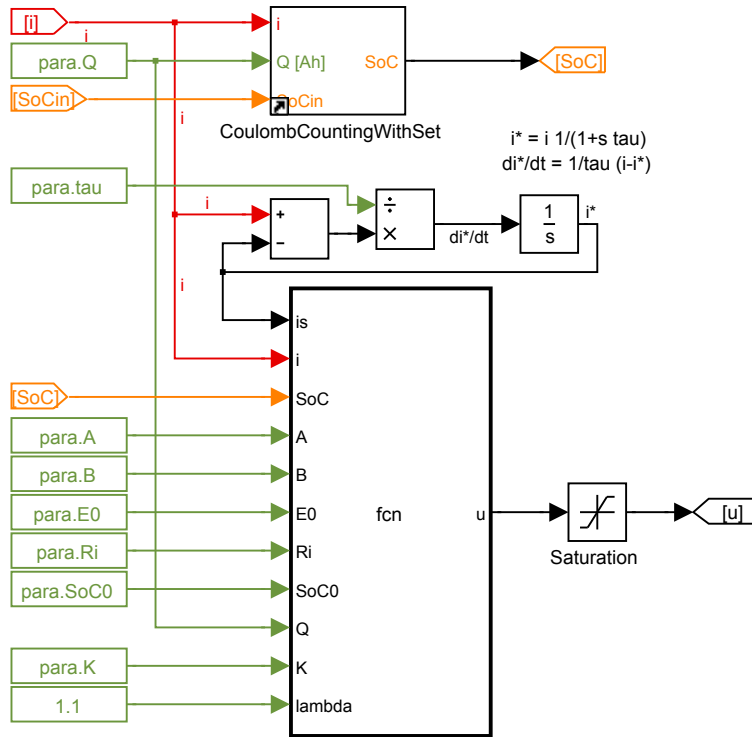


Figure 4.2: Implementation of the eMSM in MATLAB/SIMULINK

in Section 3.1.3. The saturation block after the calculation of the model output ensures that the voltage is limited to zero when the mathematical model produces values below zero.

All EEC models have a parallel RC-element in common. Its differential equation

$$i = \frac{v}{R} + C \frac{dv}{dt} \quad (4.6)$$

was implemented with standard SIMULINK blocks with i as input and v as output value (see Fig. 4.3). Two *Saturation*-blocks were used to ensure that the parameter values are greater than eps , where eps is the distance from 1 to the next larger double precision number ($2^{-52} \approx 2.2 \times 10^{-16}$).

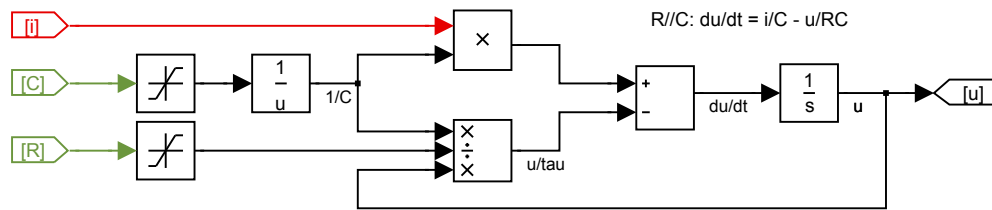


Figure 4.3: Implementation of an RC-element in MATLAB/SIMULINK

The OCV-models analyzed in Section 3.1.1 were implemented in one *Embedded Matlab Function* where the OCV model can be changed via a parameter, so that the model can be changed easily. This way, the OCV-model can be changed without changing the SIMULINK model, but only in the initialization script which is very practical when simulating different batteries or different parametrizations.

The functions for the *SoC*-dependency of the passive components were implemented the same way than the OCV for the same reasons. According to Section 3.1.2.3, following functions were implemented

- a polynomial up to 4th order,
- the function $x = c_0 + c_1 e^{-c_2 SoC}$, and
- the function $x = c_0 + c_1 SoC^{c_2}$.

As an example, Fig. 4.4 shows a screenshot of the implementation of an EEC model, where the OCV is calculated via a look-up table and the component values with functions of *SoC*. To calculate the *SoC* the block described in Fig. 4.1 is used. The OCV is calculated with the standard SIMULINK block *Lookup Table Dynamic* with linear interpolation between the look-up table values. The custom Simulink block *Parameterfitting Model* implements the *Embedded Matlab Function* described above. All necessary parameters of the model are stored in the structures *paraOCV* and *paraComp*.

The direct implementation of an electrical circuit without any additional blocks produces a negative battery voltage when an empty battery is discharged with a high current due to the term $R_i \cdot i$. This case can only happen in the model and not for a real battery, but the battery model has to produce reasonable values at any point. There are several ways to cope with this issue. One could be to set the input current to zero when $SoC = 0$ and the $i > 0$ (discharge). There still would be a change in the output voltage due to the capacity C_1 . Another possible solution would be to keep the output voltage at the same value when $SoC = 0$ and the battery is still discharged. Also a combination of the two is possible. In the end, a more simple solution was implemented as a low execution time is an important requirement for the battery model. It is a combination of limiting the battery voltage to a value greater or equal to zero and setting the voltage to zero when the state of charge is zero. This way the battery voltage can never be negative and has a defined behavior if the state of charge equals to zero.

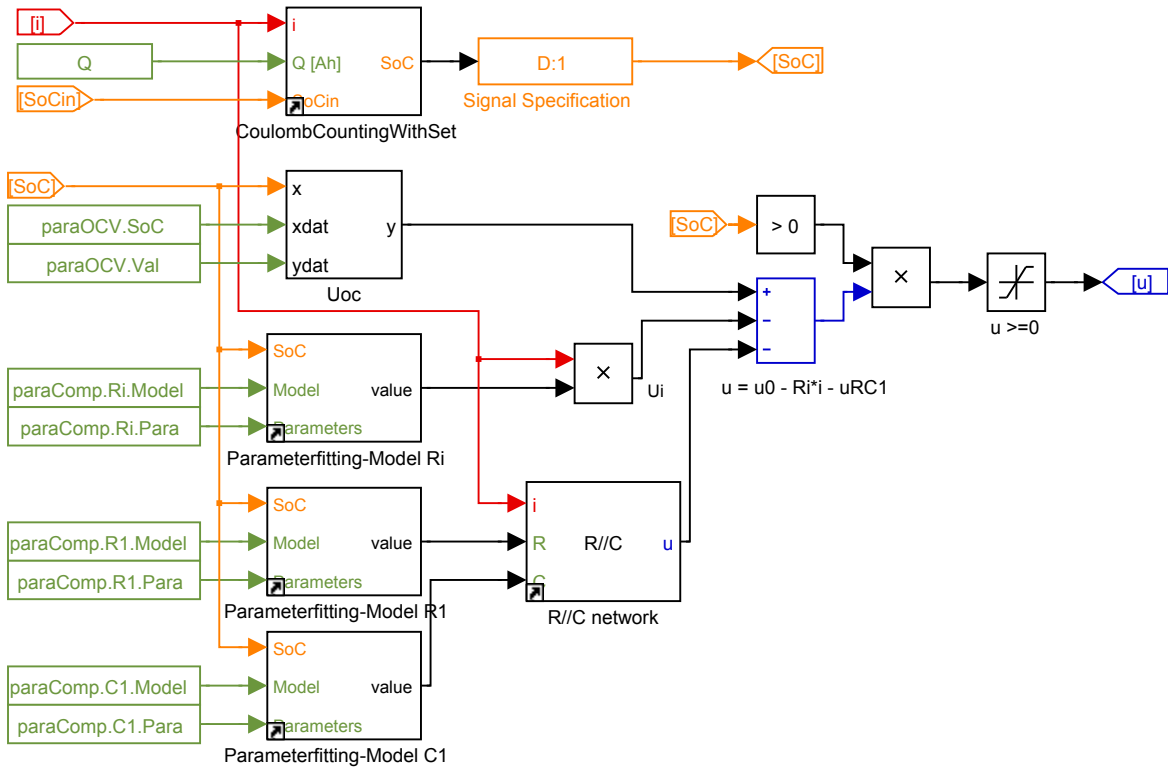


Figure 4.4: Implementation of an EEC model with one RC-element in MATLAB/SIMULINK

4.3 Validation

To validate the battery models, measurements of a *ePLB C020* Li-ion battery were used. As previously mentioned in Section 1.4.3, these are the measurements for the work in [ECKF13] where a current profile on the basis of the FTP-72 driving cycle (Federal Test Procedure, [4]) was repeatedly applied to a full charged battery until the battery was empty. The current profile of one cycle is shown in Fig. 4.5.

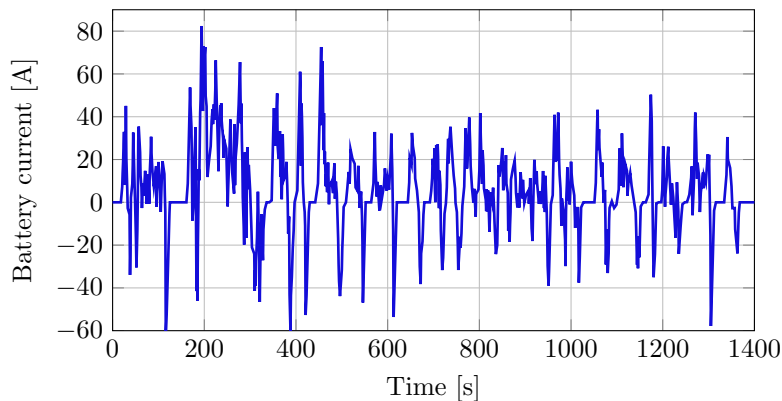


Figure 4.5: Test current profile on the basis of the FTP-72 driving cycle [4]

Because of reasons of clarity, not all the implemented battery models are included in this section. Table 4.2 shows the models which are used. Other combinations of modeling the passive com-

#	Model type	Passive comp.	OCV	Parametrization	
				meas.	data sheet
1	EEC, 1 RC-element	look-up table	look-up table	✓	✗
2	EEC, 1 RC-element	$f(SoC)$	$f(SoC)$	✓	✗
3	EEC, 1 RC-element	$c_0 + c_1 \cdot SoC$	$f(SoC)$	✓	✗
4	EEC, 1 RC-element	constant	$f(SoC)$	✓	✗
5	eMSM	-	-	✓	✗
6	eMSM	-	-	✗	✓

Table 4.2: Battery models which will be compared with the measurements

ponents and the OCV are also possible, but they are not listed in the table and in the analysis below, since there is hardly any new information in comparison to the equivalent model with the other OCV calculation.

In the used data base (cf. Section 1.4.3 and [ECKF13]) look-up table values for the OCV and the passive components were available. Out of them, other parametrizations (cf. Table 4.2) were extracted. This was already done in Section 3.1.1 for the OCV and in Section 3.1.2.3 for the resistors and the capacity.

The EEC models were only simulated with parametrization data from measurements, while for the eMSM, both parametrization with measurement data and data sheet parametrization was used. So overall six different models and parametrizations are compared.

4.3.1 Estimating the Capacity and Choosing the Right Solver

A closer look at the measurement data revealed that the real capacity of the battery is a few percent higher than the nominal capacity of 20 Ah of the data sheet. So the real capacity had to be estimated. The measurement data starts with a phase of constant voltage charging. After this phase, the current is zero for some time. At the end of this rest period ($t = t_{\text{start FTP-72}}$) the battery voltage is 4.175 V. The OCV-*SoC* look-up table can be interpolated at this point, which gives $SoC = 98.89\%$. After this rest period the FTP-72 is repeated several times until the battery voltage falls below 2.7 V the first time, followed with a rest period. At the end of last rest period ($t = t_{\text{end FTP-72}}$), the voltage is 2.768 V, which equals according to the OCV-*SoC* look-up table $SoC = 0.14\%$. Now the Equation 2.4 can be used to estimate the capacity

$$SoC(t_{\text{end}}) = SoC(t_{\text{start}}) - \frac{1}{Q} \int_{t_{\text{start}}}^{t_{\text{end}}} i(\tau) d\tau \quad (4.7)$$

$$\approx SoC(t_{\text{start}}) - \frac{1}{Q_{\text{est}}} \sum_{k=1}^N i(t_{\text{start}} + k \cdot t_s) \cdot t_s \quad (4.8)$$

where t_s is the sample time of the measurements ($t_s = 10$ ms) and $N = \frac{1}{t_s}(t_{\text{end}} - t_{\text{start}})$ the number of measurements between the start time of the FTP-72 cycles and the end of them. The capacity is

$$Q_{\text{est}} = \frac{1}{SoC(t_{\text{start}}) - SoC(t_{\text{end}})} \sum_{k=1}^N i(t_{\text{start}} + k \cdot t_s) \cdot t_s$$

which leads with $t_{\text{start}} = 602.69 \text{ s}$ and $t_{\text{end}} = 27\,335.33 \text{ s}$ to

$$Q_{\text{est}} = 20.86 \text{ Ah.} \quad (4.9)$$

This value was used as capacity for all the simulations that use parametrization data from measurements (Models #1 to #5 in Table 4.2).

First simulations of the eMSM implemented according to Section 4.2 showed a not negligible error at the *SoC* calculation because the automatical step size feature of SIMULINK chooses too big maximum step sizes. Various combinations of solver and step sizes were tried following the guideline *Choose a Solver* from the MATHWORKS homepage [6]. Finally the solver *ode45* with *max step size* = 0.1s was chosen as a tradeoff between accuracy and simulation duration. It produces an error in the state of charge calculation of 0.04% at the end of the measurement which was about nine hours.

4.3.2 Simulating EEC Models and the eMSM

After parametrizing the battery models, implementing them in SIMULINK and defining an error, the simulations of the models were conducted. The current measurements of a repeatedly applied FTP-72 cycle (cf. Fig. 4.5) were used as input for the models.

As a representative result, Fig. 4.6 shows the simulated battery voltage of an EEC model (Model #1 according to Table 4.2). The battery voltage is underestimated a little bit when the state of charge is close to zero and quite accurate in the middle *SoC*-area. The error between the model and the measurements increases significantly when the *SoC* falls below 15%. This effect is the same for all the 6 simulated models.

Model	ϵ_{max}	ϵ_{mean}	ϵ_{rms}
1	8.86%	0.47%	1.51%
2	8.40%	0.49%	1.49%
3	8.58%	0.50%	1.51%
4	8.81%	0.51%	1.53%
5	7.76%	0.46%	1.31%
6	77.24%	-4.99%	17.49%

(a) Repeated FTP72-cycle from $SoC_{\text{init}} = 98.89\%$ till the battery is empty

Model	ϵ_{max}	ϵ_{mean}	ϵ_{rms}
1	0.72%	0.37%	0.38%
2	0.38%	-0.01%	0.08%
3	0.45%	-0.05%	0.11%
4	0.42%	-0.04%	0.10%
5	0.49%	0.03%	0.12%
6	2.32%	0.49%	0.66%

(c) One FTP-72 cycle with $SoC_{\text{init}} = 52.15\%$

Model	ϵ_{max}	ϵ_{mean}	ϵ_{rms}
1	1.05%	-0.67%	0.73%
2	0.73%	-0.50%	0.51%
3	0.77%	-0.50%	0.51%
4	0.85%	-0.55%	0.55%
5	0.93%	-0.50%	0.51%
6	2.58%	-0.94%	1.01%

(b) One FTP-72 cycle with $SoC_{\text{init}} = 98.89\%$

Model	ϵ_{max}	ϵ_{mean}	ϵ_{rms}
1	6.86%	4.18%	4.39%
2	6.96%	4.59%	4.78%
3	6.91%	4.70%	4.88%
4	6.94%	4.76%	4.93%
5	6.77%	4.29%	4.47%
6	9.29%	6.37%	6.54%

(d) One FTP-72 cycle with $SoC_{\text{init}} = 9.06\%$

Table 4.3: Errors between the measured and simulated voltage; model numbers according to Table 4.2

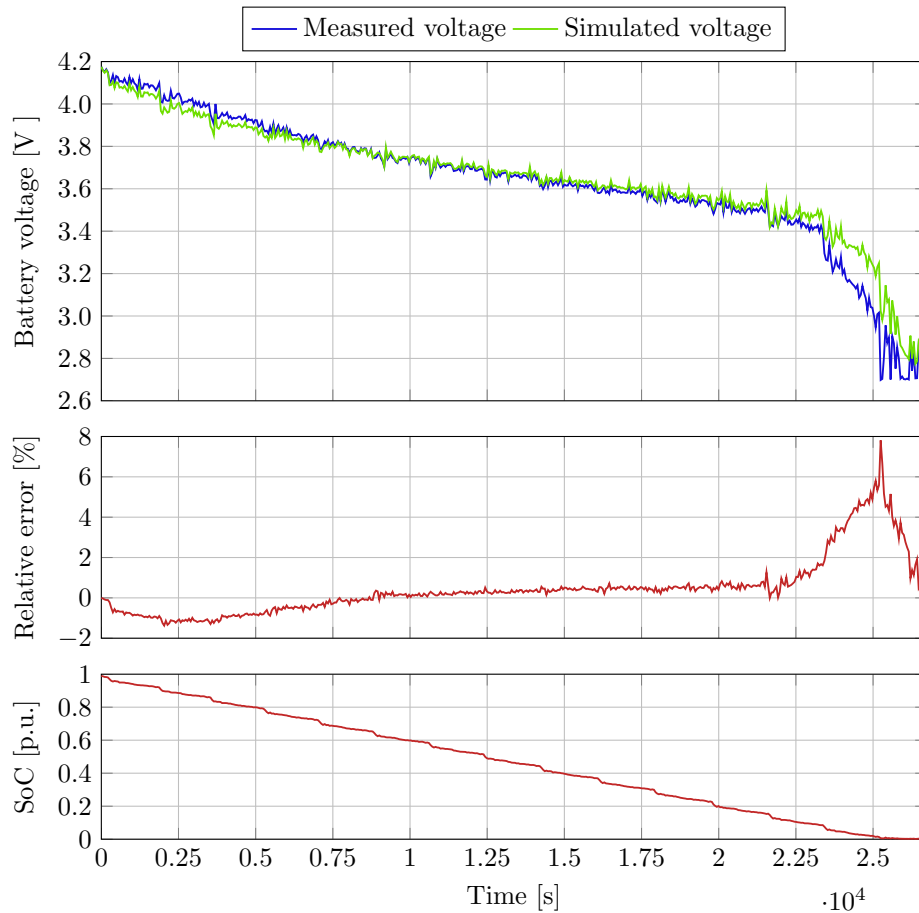


Figure 4.6: Comparison of the measured and simulated battery voltage for a repeatedly applied FTP-72 cycle. The model is an EEC model with one RC-element, OCV and components are modeled with look-up tables (model #1 in Table 4.2).

Table 4.3 shows the three defined errors for every simulated model. The table is divided in four sub-tables, one for the overall range and three for one FTP-72 cycle at the upper SoC_{init} range ($SoC_{init} = 98.89\%$), the middle ($SoC_{init} = 52.15\%$) and the lower ($SoC_{init} = 9.06\%$), respectively. Models 1 to 5, so all the models which were parametrized with measurement data, are quite similar. Beside the lower SoC -area, the maximum error is below 1.5% and the RMS error below 1%. There is no significant difference between the errors of the five models. Especially, it is very surprising that there is hardly any difference between the EEC model with constant passive components and those with look-up tables or polynomials. This circumstance disagrees with the literature analyzed in the literature review where a dependency of the state of charge is proposed. Due to the fact that the models are so similar, it makes hardly any sense to plot them together in one figure, because only little information can be withdrawn out of the graphs beside of the fact, that they are similar.

Only the graphs of simulation data which are necessary to make the important conclusions are shown in this chapter. The results of all the 24 simulations which are the basis of the data of Table 4.3 are listed in Appendix C.

4.3.2.1 Upper SoC-Range

Fig. 4.7 shows the eMSM in comparison with the measurement over one FTP-72 cycle with $SoC_{init} = 98.89\%$. The figures show an offset error of approximately 0.02 V, but beside of this, the model can reproduce the measurement qualitatively correct. Apparently, the internal resistance is determined correctly. The effect of the quite long time constant $\tau \approx 100$ s cannot be seen in a signal with comparatively fast changes.

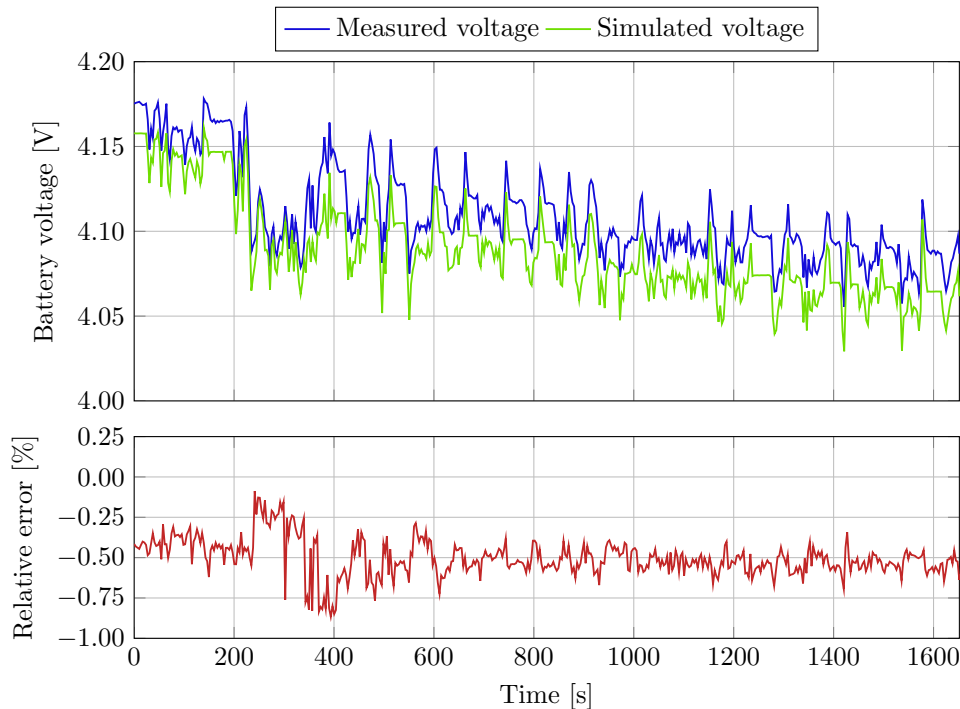


Figure 4.7: Comparison of the measured and simulated battery voltage for one FTP-72 cycle with $SoC_{init} = 98.89\%$ of the eMSM (Model #5 in Table 4.2).

4.3.2.2 Middle SoC-Range

In the middle SoC area, the simulation almost matches the measurements, which can be seen in Fig. 4.8. The RMS error is below 0.15% and the maximum error below 1% for the Models #2 to #5. Only the error of the model with look-up tables is higher. There is no evident reason why this model performs worse than the other models. Fig. 4.9 shows the same input as Fig. 4.8, but for the Model #1. In contrast to Model #2, there is a slight offset error which leads to 0.3% higher mean and RMS error.

Simulations with other combinations of modeling the OCV and the passive components showed, that using a look-up table for the resistor's and capacity's value increases the accuracy, while using a look-up table for the OCV decreases the accuracy. It seems that OCV model which tries to fit the look-up table values accidentally leads to a model which is more accurate than the original look-up table. Nevertheless, the error of both models is in the dimension of one percent which is more than sufficiently accurate.

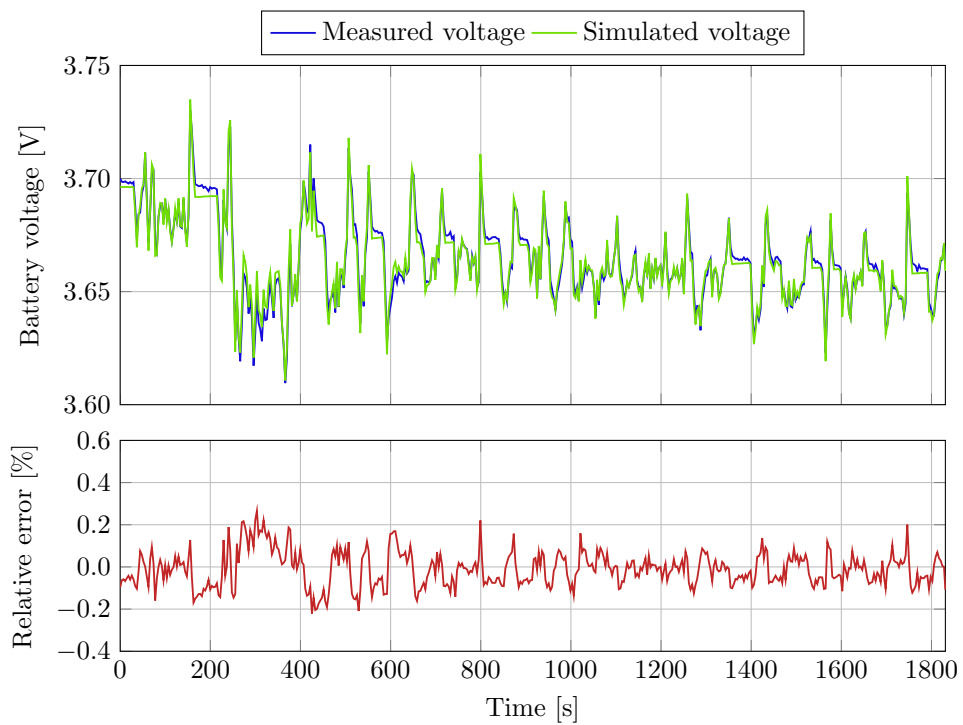


Figure 4.8: Comparison of the measured and simulated battery voltage for one FTP-72 cycle with $SoC_{init} = 52.15\%$ of an EEC model with one RC-element, OCV and comp. values as $f(SoC)$ (Model #2 in Table 4.2)

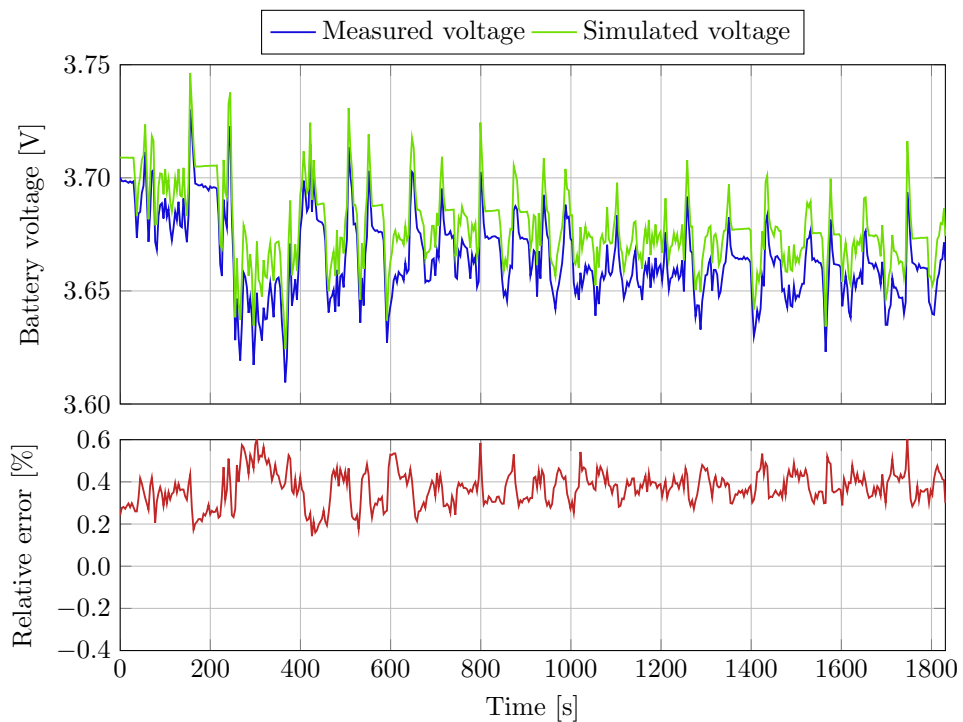


Figure 4.9: Comparison of the measured and simulated battery voltage for one FTP-72 cycle with $SoC_{init} = 52.15\%$ of an EEC model with one RC-element, OCV and comp. values as look-up table (Model #1 Table 4.2)

4.3.2.3 Lower SoC-Range

As shown in Fig. 4.6 and Table 4.3d, the error of all models increases significantly when the battery is almost discharged. Fig. 4.10 shows a comparison of the measurement with the look-up table model for one FTP-72 cycle with $SoC_{init} = 9.06\%$. Here, the model isn't capable anymore to reproduce the measurements, although the OCV in the model is close to the measured one, and the transient behavior of the two voltages matches quite good, which would indicate that the parametrization of the resistors suits well.

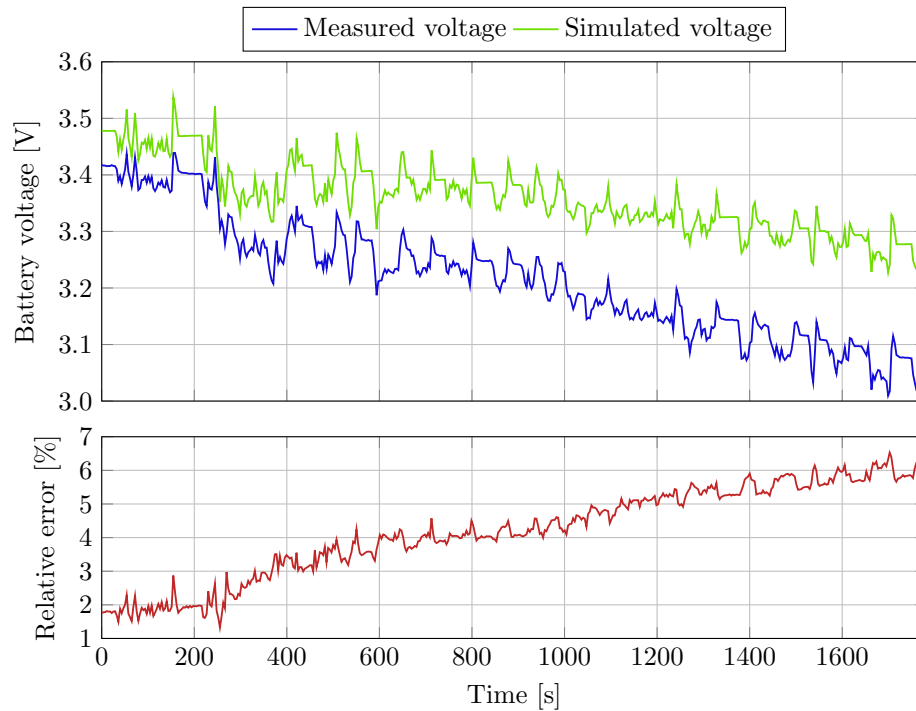


Figure 4.10: Comparison of the measured and simulated battery voltage for one FTP-72 cycle with $SoC_{init} = 9.06\%$ of an EEC model with one RC-element, OCV and comp. values as look-up table (Model #1 Table 4.2).

4.3.2.4 Using the KiBaM to Increase the Accuracy

The overestimation of the voltage for ongoing simulation time suggests the assumption that the *SoC* determination calculates too high states of charge. This is because of the fact that the Rate Capacity Effect is not taken into account at Coulomb Counting, which is used to calculate the *SoC*. A model that includes this effect is the Kinetic Battery Model (KiBaM, see Section 2.4.4). As mentioned in Section 3.3, there was no measurement data available to explicitly parametrize the KiBaM. Hence, its parameters were determined by “trial and error” leading to

$$\begin{aligned}c &= 0.75 \\k &= 0.0005.\end{aligned}$$

The result of the simulation of Model #1 with KiBaM instead of Coulomb Counting is shown in Fig. 4.11. The figure shows that the KiBaM is able to model the earlier dropping of the battery voltage due to the Rate Capacity Effect, but also increases the error at the high *SoC* sector. Thus, additional effort is needed to parametrize the EEC model in combination with the KiBaM. This is too far afield from the focus of this master thesis and hence done in future work.

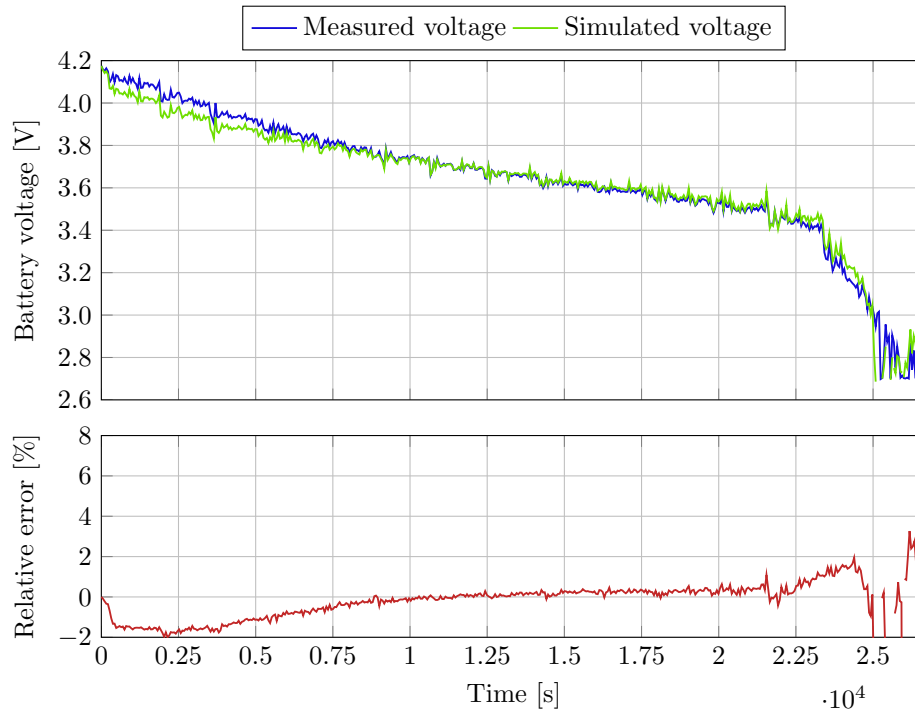


Figure 4.11: Comparison of the measured and simulated battery voltage for a repeatedly applied FTP-72 cycle using an EEC model with KiBaM instead of Coulomb Counting to estimate the *SoC*.

4.3.2.5 Extended Modified Shepherd Model with Data Sheet Parametrization

The last simulated model was an eMSM which was parametrized with data sheet curves (model #6 in Table 4.2). Fig. 4.12 shows this model in comparison to the measurement for a continued

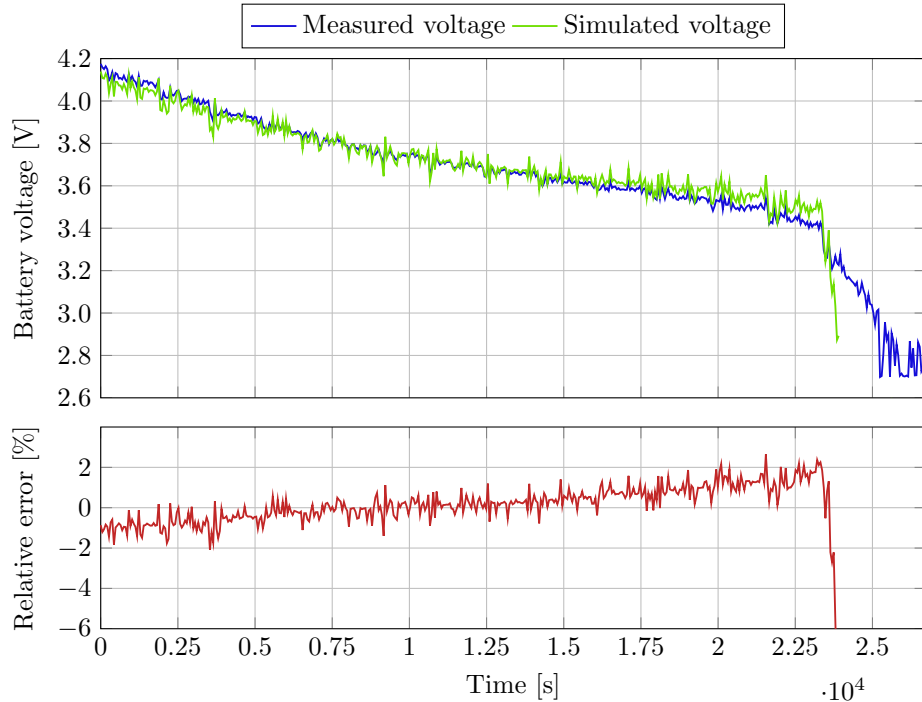


Figure 4.12: Comparison of the measured and simulated battery voltage for a repeatedly applied FTP-72 cycle using an eMSM which was parametrized with data sheet curves.

applied FPT-72 cycle. The model can reproduce the measurement quite well until the time is about $2.35 \cdot 10^4$ s. At this point, the difference between the estimated capacity of 20 Ah and the real capacity of 20.86 Ah becomes noticeable. While the real battery still has a $SoC \approx 4\%$, the modeled battery is empty, wherefore the voltage breaks down.

Although the model is only parametrized on the basis of the discharge curves in the data sheet, the mean and RMS error until $t = 2.2 \cdot 10^4$ s is below 2%. Fig. 4.13 shows a detailed view of 4.12 one FPT-72 cycle with an initial state of charge of 52.15%. It can be clearly seen in the figure that the resistance R is too high. The changes of voltage due to the internal resistance are roughly twice as large for the model as for the measurements which goes in hand with Chapter 3, where the data sheet parametrization led to $R = 3.249$ m Ω in comparison to $R = 1.723$ m Ω (see Table 3.3). This shows that the static resistance of a battery is higher than the transient one. The value of the resistance was determined by continuous discharge profiles and hence overestimates the dynamic resistance. This is consistent with the EEC models where the AC resistant is smaller than the DC resistant.

Neglecting the inaccurate capacity which leads to a completely wrong voltage after $2.2 \cdot 10^4$ s, the maximum error is below five percent and the RMS error below two percent. This error is higher than for the other models which were parametrized with data sheet data, but still unexpectedly low. Thus, the eMSM parametrized with data sheet values and with a good choice of the time constant τ , which can be guessed due to experience values or a literature research, can be used

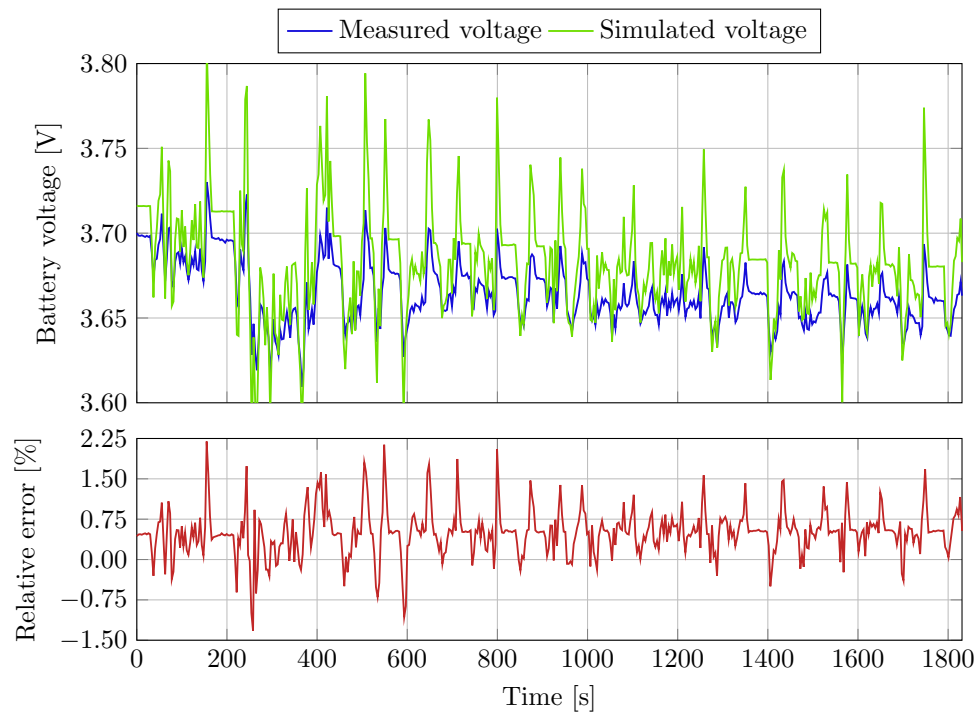


Figure 4.13: Comparison of the measured and simulated battery voltage for one FTP-72 cycle with $SoC_{init} = 52.15\%$ using an eMSM which was parametrized with data sheet curves.

to model the principle behavior of the battery quite accurate. But it cannot predict the battery lifetime because the nominal capacity is not equal to the real capacity (cf. Fig. 4.12).

Taking into account the requirements of the battery model (Section 1.4 and 2.1), the errors of Table 4.3, the figures in this chapter and in Appendix C, the extended Modified Shepherd Model (eMSM) turned out to be the best model in terms of parametrization effort, accuracy and the easy possibility to parametrize the model without measurements. Therefore this model will be used for the PHIL experiments in this work further on.

5 PHIL Implementation

After a literature research about battery models, their parametrization and validation, the Power Hardware-in-the-Loop (PHIL) simulations can finally be implemented and conducted. This chapter will deal with the involved components and the PHIL-simulation setup, while Chapter 6 discusses the results of the simulation.

The basic setup of a PHIL simulation using the ideal transformer model was shown in Fig. 1.3. In a normal PHIL use case, the amplifier would be connected to the hardware under test and the “virtual rest of system” the battery model with a battery management system, and for example an energy management system with a domestic house model, a market model, etc. In this work, only a setup to create a desired battery current is needed in order to validate the emulated battery against measurements of a real battery.

To reproduce the measurements of the battery within the PHIL simulation, a variable battery current in both energy directions is needed. Due to the lack of a programmable bidirectional current source and load, a combination of a manually switchable constant load and a programmable source is used. As power amplifier a bidirectional power source *TopCon TC.GSS.32.600.400* from Regratron was used, because this device was available at the test bed. This leads to the implemented PHIL setup shown in Fig. 5.1.

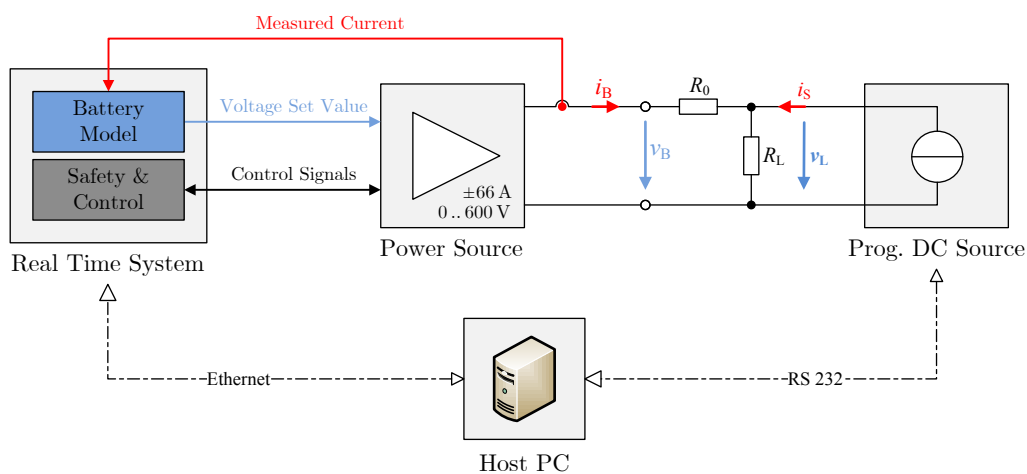


Figure 5.1: Implemented PHIL setup with constant load and programmable DC source

The power source can be controlled with an analog interface via the real time system (RTS) and is capable of voltages up to 600 V and currents up to ± 66 A. More information about its internals and how to control the power source will be mentioned in Section 5.2. The resistor R_0 was used to limit the current between the power amplifier and the programmable DC source if their terminal voltage varies due to their internal control loops. As resistor R_L a manually switchable power resistor and as programmable DC source a *Regatron TopConQuadro TC.P.16.600.400.S* were used. More information about how to use them to create a desired battery current will be discussed in Section. 5.3. This section is followed by a section which discusses how to set the right resistance for R_L . The PHIL setup is completed by a host computer which is used to control the real time system (RTS) over Ethernet and the programmable DC source via an RS 232 interface.

5.1 OPAL-RT Real Time System

An OPAL-RT real time system is used for this work. It consists of a commercial off-the-shelf server motherboard with two quad core Intel i7 CPUs operating a real time Red Hat Linux. An additional FPGA (field-programmable gate array) provides timing features and the possibility to connect I/O modules with digital and analog in- and outputs. Together with the OPAL-RT RTS comes RT-LAB, an open real time simulation software environment. This gives the possibility to execute MATLAB/SIMULINK models on the RTS in real time processing electrical signals sampled via the analog input modules and controlling external hardware via the output modules.

RT-LAB is composed of several parts. One of them is a SIMULINK library with blocks to read and write SIMULINK signals from and to the I/O modules and to monitor the model. Another is the functionality of transforming a SIMULINK model into C code which is then compiled to create an executable program without MATLAB runtime.

Execution Time of the Models on the Real Time System

The low execution time of the battery model is a crucial requirement. In a typical future application the battery model will be part of a bigger PHIL simulation containing the simulation of an AC microgrid. At this kind of simulations the RTS also generates the voltage curves for the AC phases which are amplified with fast power amplifiers. For this kind of simulations a step time of 10 μ s is desirable, which leads to an execution time, including processing the measurements and model output, of under 10 μ s of the battery model.

For a comparison of the execution time the extended Modified Shepherd Model (eMSM, Section 3.1.3) and three different equivalent electrical circuit (EEC) models with one RC-element were implemented. These EEC models are:

- a look-up table for the open circuit voltage (OCV) and a look-up table for the state of charge (*SoC*) dependency of the resistors and capacity, linear interpolation between the look-up table entries,
- OCV as a function of the *SoC* and constant resistors and capacities,
- OCV and R/C values as a function of the *SoC*.

The models were executed simultaneously, each running at one core of the RTS. Their execution time is shown in Table 5.1. The execution time can be measured via the *OpMonitor* block of

Model	Execution time
eMSM	2.65 μ s
EEC, one RC-element, Rs and C as look-up table	2.35 μ s
EEC, one RC-element, Rs and C are constant	2.90 μ s
EEC, one RC-element, Rs and C as function of <i>SoC</i>	4.35 μ s

Table 5.1: Execution time of different battery models

RT-LAB. The table shows that the execution time of all four implemented model is below the limit of 10 μ s. There is also enough time left for the communication with other subsystems of the simulation, for filtering and for offset correction of the measurements. So all the four implemented models are suitable for the given task.

Nevertheless, Table 5.1 shows some differences between the models. As expected, the look-up table model is the fastest, since only linear approximations on a relatively small table are necessary, while the other models contain more computation intense mathematical function like division or exponential function. The eMSM is roughly 10% faster than an EEC model with one RC-element and constant parameters although their structure and complexity is more or less the same. This can result from different execution times of standard Simulink blocks and embedded MATLAB functions. The execution time of the model with parameters as a function of the *SoC* sticks out. It is about 50% higher than the others while its accuracy is not significantly higher than the others (see Chapter 4). Using the KiBaM instead of Coulomb Counting results in an approx. 1.5 μ s higher execution time.

5.2 Power Amplifier

An essential component of a PHIL simulation is the power amplifier, which operates in this case in the first and fourth quadrant, meaning that it can provide energy (“battery is being discharged”) and absorb energy (“battery is being charged”). In this work the programmable bidirectional DC supply *Regatron TopCon TC.GSS.32.600.400* has been used, because it was available at the research test bed of AITs Energy department and fulfilled the requirements regarding the terminal voltage and power capability. Its topology is shown in Fig. 5.2. This power source contains three rectifiers/inverters (number 3, 5 and 7 in Fig. 5.2), a DC intermediate circuit between two of them (#4) and a transformer between the second and the third rectifier. This provides up to 600 V DC voltage with a galvanic isolation between AC and DC side. To reduce noise on the

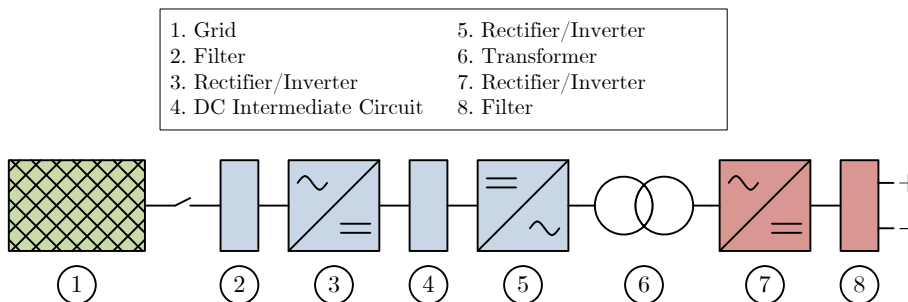


Figure 5.2: Topology of the Regatron TC.GSS bidirectional DC source

lines and external disturbance, the AC and DC side has an output filter (#2 and #8). With this combination of rectifiers, inverters and an intermediate circuit, DC power can be fed back to the AC grid instead of heating power resistors.

The TC.GSS has a maximum DC current of 66 A in both directions. The overall power is limited to 32 kW so there is a constant power mode between 53.3 A@600 V and 66 A@485 V. Fig. 5.3 shows this circumstance. The figure also shows, that due to the internal structure, the sink mode

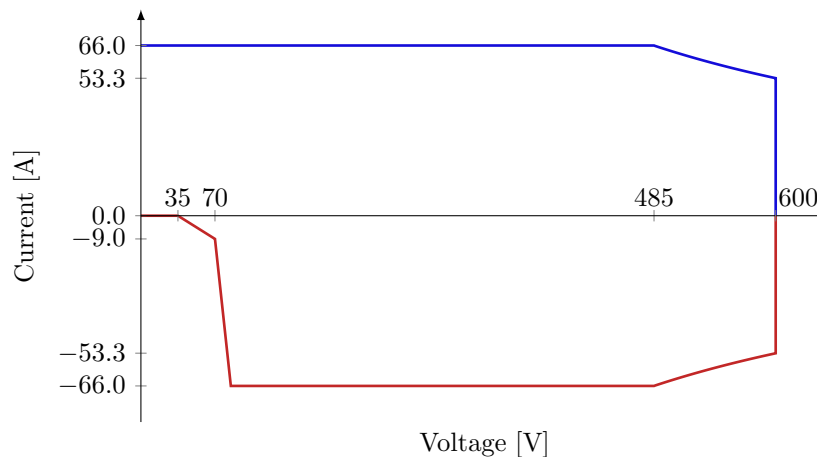


Figure 5.3: System operational area of the Regatron TC.GSS

is only possible above a specific minimal voltage. At 70 V the current is limited to -9 A and the nominal current is available from 85 V on. This is an important restriction since it implies a minimal battery voltage for charging.

The *TC.GSS* has an internal digital controller and can operate in constant voltage (CV), constant current (CC) and constant power (CP) mode. The term “constant voltage” is misleading, since it is a voltage regulation and the output provides a set value change within a few milliseconds. The TC.GSS has three inputs (voltage, current and power) which are treated as maximum values. Its internal controller increases the output power until the first of these three limits is reached and alternatively decreases it until all the three values fall below the limits.

For this work the voltage regulation mode (CV mode) is used, because the output of the battery model is its voltage which is then the set value for the DC supply. The other two input values are used for safety and security reason, e.g. to not exceed the maximum permitted current in the cables.

The bidirectional DC supply can be controlled in a few different ways. At first, there is a human machine interface (HMI) directly on the device. It can be used, to set constant values for voltage, current and power, where the limiting one is active. The HMI can also be used to set the cut-off frequency for the input filter of the reference value and for the output filter of the actual values. The HMI also provides a button to quickly disable the output and zero-potential it, which is very important for safety issues.

Regatron also provides optional RS 232 and USB interfaces together with a LabVIEW and C/C++ API (application programming interface). So the DC supply can be controlled and supervised via LabVIEW or a self written program.

Another available interface is the analog interface. The reference values for voltage, current and power can be directly set through a voltage between 0 V and 10 V which corresponds to 0% until 100% of the nominal values. The analog interface also provides internal measurements of the DC current and voltage. This is well suitable for an easy and fast control via the RTS with its analog in- and outputs. Since the analog interface also provides digital inputs to activate and deactivate the *TC.GSS*, it is used in this work.

5.2.1 Control of the Power Amplifier with the RTS via its Analog Interface

The analog output modules of the RTS have a range of -16 V to 16 V where the SIMULINK signal value directly corresponds to the analog output voltage. As mentioned above, the power amplifier has a range of 0 V to 10 V . For the voltage regulation 10 V conform to the maximum output voltage of 600 V and 0 V to 0 V . The same applies for the current, merely that the range is -10 V to 10 V for $-i_{\text{nom}}$ (-66 A) to i_{nom} ($+66\text{ A}$). So the analog output value is

$$v_{v,\text{out},\text{analog}} = 10\text{ V} \cdot \frac{v_{\text{set}}}{v_{\text{nom}}} \quad (5.1)$$

and

$$v_{i,\text{out},\text{analog}} = 10\text{ V} \cdot \frac{i_{\text{set}}}{i_{\text{nom}}}. \quad (5.2)$$

The power input is reversed so 0 V to 10 V correlate with $\pm 32\text{ kW}$ to 0 kW . This is done in order that the input does not have to be connected if a power regulation or maximum power limit is not used. This lead tho the equation

$$v_{p,\text{out},\text{analog}} = 10\text{ V} \cdot \left(1 - \frac{p_{\text{set}}}{p_{\text{nom}}}\right) \quad (5.3)$$

to adapt the SIMULINK signal value for the analog output module.

The analog input has to be activated with a rising edge on one specific pin of the analog interface. Also the output has to be enabled with another pin. This output has to be activated manually for safety issues. And finally it is practical to connect the clear-error-pin with the RTS, to reset the DC supply after an error occurred, so it can be used again. These three pins are controlled with the RTS too.

The *TC.GSS* also provides internal measurements of the DC current and voltage via the analog interface. The particular pins are connected to the analog input modules of the RTS. They work with the same principle than the set values, but the other way around. So the measured value is

$$\text{measured output value} = \text{nominal value} \cdot \frac{\text{analog interface value}}{10\text{ V}}. \quad (5.4)$$

To summarize, the used pins are shown in Table 5.2.

The *TC.GSS* uses a D-SUB 25-pin connector, while the RTS has BNC (Bayonet Neill–Concelman) connectors. Hence, a small adapter board was built. This board gives also the possibility to enable the analog interface with a button and to insert an emergency stop button in the interlock circuit.

Pin	Name	Description	Type
1	AGND	analog ground for inputs and outputs	–
2	VREF	voltage set-point	analog in
3	IREF	current set-point	analog in
4	IACT	measurement of the DC current	analog out
8	CLEAR_ERROR	clear an error to reuse it after an error	digital in
9	VOLTAGE_ON	enable and disable the output	digital in
14	PREF	power set-point	analog in
16	VACT	measurement of the DC voltage	analog out
17	COM	digital ground	–
20	ANALOG_REFERENCE_SELECT	activate the analog interface	digital in

Table 5.2: Used pins of the power amplifiers analog interface

5.2.2 Correcting Offset- and Linearity-Error

First measurements showed an offset error between set values via the analog interface and the actual value. So the input/output relation was further analyzed. Fig. 5.4 shows the experimental setup. The RTS and the DC supply were connected via the analog interface. A host PC with a

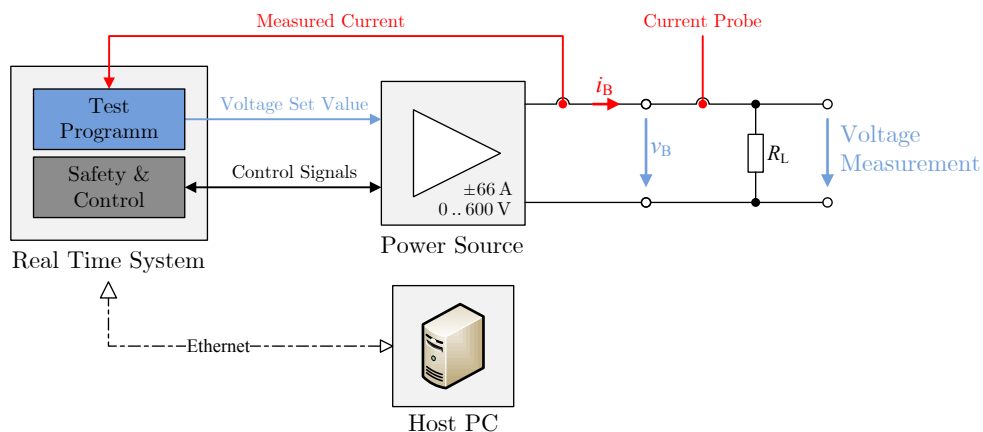
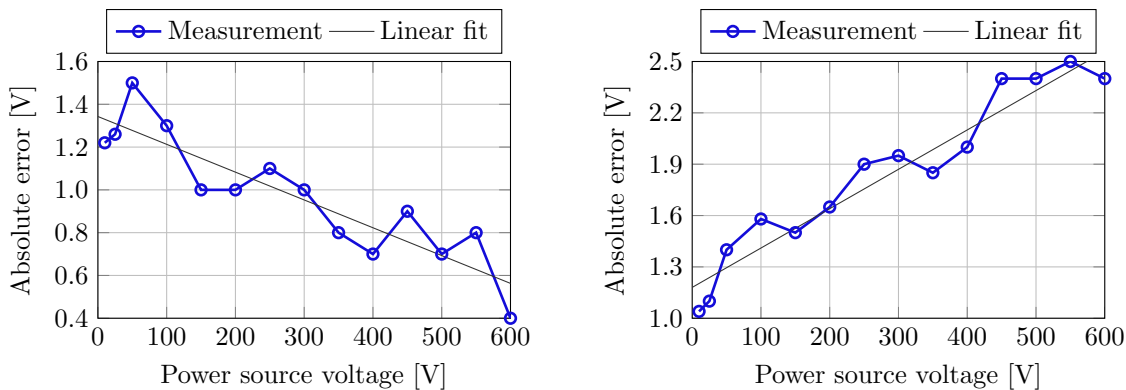


Figure 5.4: Experimental setup for measuring the linearity error

GUI (graphical user interface) was used to set the reference values and read the measurements of the actual values. The DC supply was connected with a constant load of $\approx 40\Omega$ to ensure a current below 16 A which is the maximum current of the used test cables. The measurements showed two aspects. On the one hand there is an error between the set value (pin VREF and IREF on the analog interface of the *TC.GSS*) and the actual value. On the other hand, there is

an error at the measurements of the actual values which are available over the output pins of the analog interface (pin VACT and IACT). These four errors are identified below.

1. **Error between voltage set value and actual voltage.** This error is caused by the DACs (digital-to-analog converter) of the RTS, the ADC (analog-to-digital converter) of the *TC.GSS* and the internal control loop of the *TC.GSS*. To determine the error, different voltage set values were set up via the GUI at the host PC and then measured with a *Fluke 189* multimeter. Fig. 5.5a shows the absolute error between the voltage set values and the



(a) Absolute error between voltage set values and measured voltages (b) Absolute error between voltage measured with a multimeter and voltage displayed at the RTS

Figure 5.5: Absolute errors between set values, measured values by a multimeter and values displayed at the RTS

measured actual voltages. The according data can be found in the Appendix in Table D.1. The figure shows that the offset error is not constant, but decreasing for higher voltages. So there is a linearity error. The error can be approximated by the equation

$$\text{offset error} = \text{set value} - \text{output value} = k \cdot \text{output value} + d \quad (5.5)$$

with $k = -0.0013$ and $d = 1.3429$ V. The average error is 1 V which is only $\approx 0.157\%$ of the nominal value. Reversed, the linear approximation can be used to generate more accurate output values, by adjusting the set value before converting it to an analog voltage. This adjustment is

$$\text{output value} = \frac{\text{set value} - d}{1 + k}. \quad (5.6)$$

A simpler adjustment would be just to remove the average error of 1 V.

2. **Error between actual voltage and measured voltage at the real time system.** This error is caused by voltage measurement of the *TC.GSS*, its DAC and the ADC of the RTS. For determining the error of the voltage measurement, the input value was adjusted manually until the multimeter showed the desired value. Then the measured value was noticed. Fig. 5.5b shows the offset dependent on the output voltage. The according data can be found in Table D.2 in the Appendix. A linear approximation with Equation 5.5 gives $k = 0.0023$ and $d = 1.18$ V with an average error of 1.8 V. The accuracy of the voltage measurement is increased by adapting the input value with Equation 5.6.

3. **Error between current set value and actual current.** This error is caused by the DACs of the RTS, the ADC of the *TC.GSS* and the internal control loop of the *TC.GSS*. The error was determined in the same way than for the voltage. The set value showed a quite constant offset of 0.1 A which is in the same range than the one for the voltage (also about $\approx 0.15\%$ of the nominal value). So it is not corrected with Equation 5.6, but just with subtracting the offset.
4. **Error between actual current and measured current at the real time system.** This error is caused by current measurement of the *TC.GSS*, its DAC and the ADC of the RTS, and was determined in the same way than for the voltage. The analysis of the current measurement led to $k = -0.005$ and $d = 0.0547$ A with an average error of 0.03 A and the measurement was adapted again by applying Equation 5.6. The according data is given in Table D.3 and D.4.

5.3 Variable Load and Source

The PHIL-setup together with the battery model are validated against measurements of a real battery. To reproduce the measured battery current, a variable load and source is needed. For this work, only a programmable DC source was available, but with an additional constant load, it was possible to generate variable battery emulator current in both the directions. Therefore a manually switchable power resistor R_L , which was available in the lab, was used. Fig. 5.1 shows the according setup. The programmable DC source is a *Regatron TopConQuadro TC.P.16.600.400.S* and the resistor a manually adjustable three phase 27 kW load. Their specifications can be found in Appendix A on page 93.

The power amplifier as well as the programmable DC source have internal controllers to tune their set values. Small differences δv of the two terminal voltages, based on response times of their controllers, would result in high compensating current $i_{\text{comp}} = \frac{\delta v}{R_{\text{cable}}}$. Hence a resistor R_0 is connected between them to limit this compensation current. In the lab a 0.5Ω power resistor was available, which provided good results and therefore used.

By applying Kirchhoff's laws, a desired battery emulator current i_B can be generated by setting the output current i_S of the programmable DC-source to

$$i_S = -i_B + \frac{v_L}{R_L}. \quad (5.7)$$

To achieve this, an additional controller is needed, which sets the output current for a desired i_B and a measured voltage v_L . This was done via a LabVIEW program which ran on the host PC of the RTS. Section 5.4 will discuss the choice of the right value for R_L . The written LabVIEW program is able to read a battery current profile from a csv-file with variable time stamps and generate the desired battery current. Its pseudo code is shown in Algorithm 5.1.

First the program opens the csv-file and starts a timer. Then a loop is entered. Every loop pass a set value for the programmable DC source is calculated using the desired battery current i_B of the present csv-file row, the measurement of v_L and Equation 5.7. If the timer value is bigger than the time entry in the next file row, the row index is increased by one and a new value for i_B is set. This loop runs until the last entry in the file is reached.

Regatron, the vendor of the programmable DC source, provides a LabVIEW VI to control the *TopConQuadro* via the serial RS232 interface. This VI was adapted to reduce the execution time

Algorithm 5.1 Basic idea of the LabVIEW Program to generate a variable battery current

```

load csv. file
init communication with DC source
start timer

get first ref value for i_B
get first time_next_change (in the following row)
while EOF not reached
    if current_time > time_next_change
        set row pointer to next row
        get new ref value for i_B
        get new time_next_change
    end
    read current DC source voltage v_L
    calculate i_S = -i_B + v_L/R_L
    set i_S as set value for the DC source
end

```

significantly in order to make the control more accurate. Neither the operating system on the PC, nor LabVIEW is real time capable. So the time of one loop pass is around ≈ 50 ms with a variation of a few milliseconds. However, the LabVIEW program can generate a desired i_B under changing v_B . Fig. 5.6a shows the current set values and the measured current. Fig. 5.6b shows a

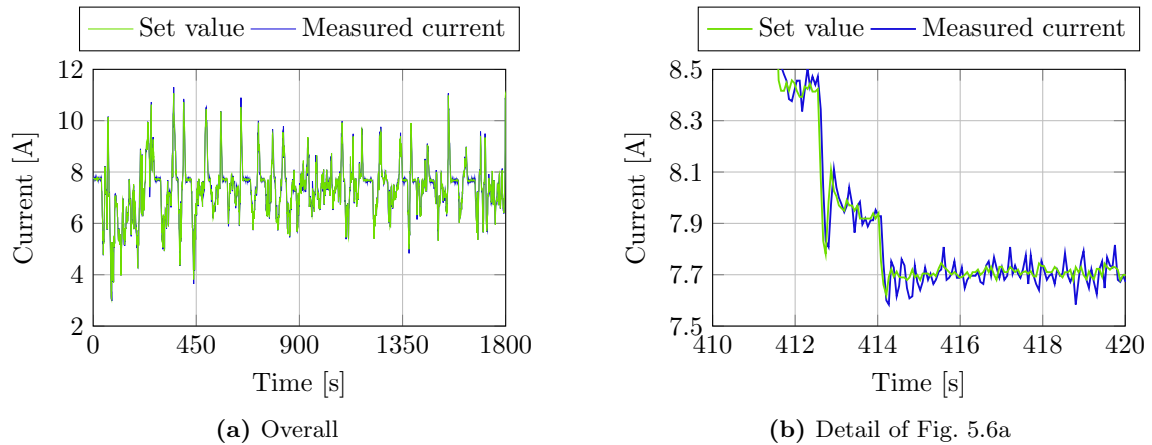


Figure 5.6: Set value and actual value of the current i_S controlled with the LabVIEW program

zoom of the curve. The figure shows that the DC source can generate the desired current profile, but with a small lag. The figure also shows a deviation which is a combination of measurement noise and the control error due to the simple controller of the LabVIEW program. Following the error definition of Equations 4.4 and 4.5 on page 46, the mean error $\epsilon_{x,\text{mean}}$ is smaller than 0.01% and the RMS (root mean squared) error $\epsilon_{x,\text{rms}}$ is smaller than 1.5%. This was considered to be small enough to run the PHIL test.

5.4 Preparation of Measurements and Scaling the Battery Model

Before running the PHIL test, the measurements of the Li-ion battery have to be filtered, because the colleagues who performed the measurement used a controllable resistor load to discharge the battery. While switching the resistor load, the load is disconnected for a short time (cf. Fig. 5.7). These fast spikes would cause troubles for the programmable DC-Source, so the signal was filtered

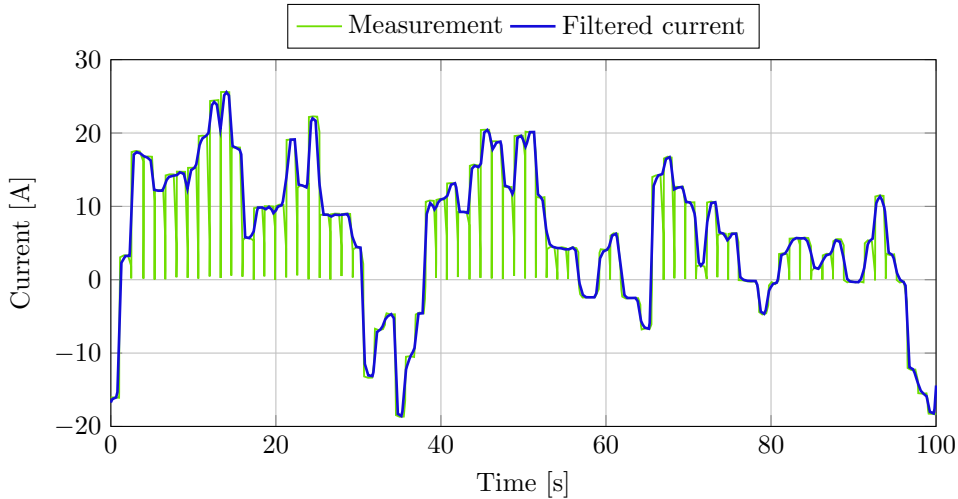


Figure 5.7: Battery test current and filtered current as input for the programmable DC source

with a PT3 low pass filter with the transfer function

$$G(s) = \frac{1}{(1 + s/\omega_c)^3} \quad (5.8)$$

and $\omega_c = 15 \text{ s}^{-1}$.

The measurement data is from one single cell with a maximum current $i_{\text{meas,max}} = 86.163 \text{ A}$. The voltage of a Li-ion cell is about 4 V , but as shown in Fig. 5.3, the battery simulator cannot absorb power under a terminal voltage of 70 V . Also the cables in the lab environment were not capable of currents above $i_{\text{max}} = 16 \text{ A}$. As a result, the battery model has to be scaled for the PHIL test. This scaling is similar to the one in Section 3.4 with the difference, that the battery modeled is scaled to be able to compare measurements with a PHIL simulation that cannot have the same magnitudes of voltage and current due to hardware and safety limits.

The minimal battery simulator voltage for working as a sink is $v_{\text{min}} = 70 \text{ V}$ (compare with Section 5.2). Thus the voltage must not fall below 70 V when $i \leq 0$. For currents greater than zero, there is no voltage limit, since the *TC.GSS* can provide energy at any voltage (see Fig. 5.3). Due to the internal resistance of the battery, the battery voltage increases when the battery is charged. Hence, the minimal terminal voltage occurs when the charge current is almost zero. This leads to a scale factor for the battery voltage:

$$\text{scale}_v = \left[\frac{v_{\text{min}}}{\lim_{i \rightarrow 0} v(i, \text{SoC}_{\text{min}})} \right] = \left[\frac{v_{\text{min}}}{v_{\text{oc}}(\text{SoC}_{\text{min}})} \right] \quad (5.9)$$

with SoC_{min} as the minimal state of charge where the battery is considered as completely empty. A SoC_{min} of 0% would not work, because $v_{\text{oc}}(0\%) = 0 \text{ V}$. A percentage of 0.5% turned out to be

a good trade-off between the SoC_{\min} close to 0% and to have a low maximum voltage for safety reasons. As a result the maximal PHIL-battery voltage can be calculated as

$$v_{\max} = v_{oc} (SoC_{\max}) \cdot scale_v \quad (5.10)$$

with SoC_{\max} as the maximal SoC where the battery is considered as fully charged. A SoC_{\max} of 100% can be used if the OCV model has no point of discontinuity at this point. For example that's the case for the eMSM, but not for the Nernst model.

The eMSM with the parametrization as done in Section 3.1.3 leads to

$$\begin{aligned} scale_v &= 24, \\ v_{\max} &= 100.15 \text{ V}. \end{aligned} \quad (5.11)$$

The second scale factor $scale_Q$ is basically freely selectable with the only condition that no current in any cable is higher than i_{\max} . The scale factor is the ratio between the PHIL battery current i_B and the current of the original measurement

$$scale_Q = \frac{i_B}{i_{\text{meas}}}. \quad (5.12)$$

It makes sense to choose $scale_Q$ is as close as possible to 1 to minimize the difference between the original measured current and the PHIL current. Since $i_{\text{meas},\max}$ is greater than i_{\max} , i_B should be maximized without violating the boundary conditions

$$\begin{aligned} |i_B| &< i_{\max}, \\ i_S &< i_{\max}, \\ i_L = i_B + i_S &< i_{\max}. \end{aligned} \quad (5.13)$$

The value of the resistance R_L has to be determined to solve this optimization problem.

Applying Kirchhoff's and Ohm's laws in Fig. 5.1 lead to

$$\begin{aligned} i_S &= -i_B + \frac{v_L}{R_L} = \\ &= -i_B + \frac{v_B - i_B R_0}{R_L} = \\ &= -i_B \left(1 + \frac{R_0}{R_L}\right) + \frac{v_B}{R_L}. \end{aligned} \quad (5.14)$$

The load current i_L is

$$\begin{aligned} i_L &= \frac{v_B - i_B R_0}{R_L} \\ &= -i_B \frac{R_0}{R_L} + \frac{v_B}{R_L} \end{aligned} \quad (5.15)$$

and the battery current i_B is

$$i_B = \left(\frac{v_B}{R_L} - i_S \right) \frac{1}{1 + \frac{R_0}{R_L}}. \quad (5.16)$$

The current i_B in Equation. 5.16 is monotonically decreasing for higher R_L , whereas the absolute value of the current is monotonically increasing when the battery is charged. The battery can be

charged if $i_S > \frac{v_B}{R_L}$. Hence a higher i_S or a lower v_B increases $|i_B|$ for constant R_L . Vice versa, the battery is discharged if $i_S < \frac{v_B}{R_L}$. To achieve a maximum $|i_B|$, i_S should be as small and v_B as high as possible.

Since all these dependencies on i_S and v_B are monotonically decreasing or increasing, it is sufficiently to consider only the minimum and maximum values to find the R_L that maximizes $|i_B|$. Fig. 5.8 shows the current i_B as function of R_L at these minimum and maximum values discarding the areas where $i_L > i_{\max}$ (occurs when i_S is close to i_{\max} and R_L is small). The range of R_L is limited

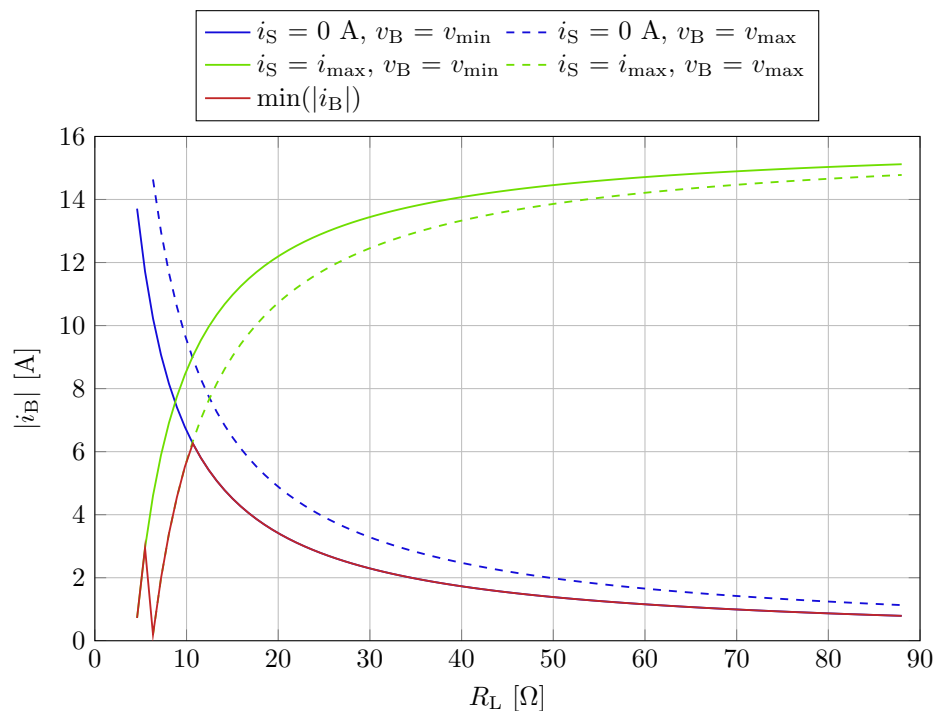


Figure 5.8: Battery current i_B for the minimum and maximum values of i_S and v_B

to the range the switchable load is capable of. The red line in Fig. 5.8 shows the minimum of the other four curves. The maximum of this curve gives the maximum $i_{B,\max} = |i_B|$ and the corresponding R_L . For every other value of R_L exists a combination of a i_S and v_B where either the boundary conditions of Equation 5.13 violated or i_B is smaller than $i_{B,\max}$. For example a R_L of $40\ \Omega$ would make a higher charge current possible, but the discharge current will be smaller than $i_{B,\max}$. The value of R_L can be calculated by solving the equation

$$\begin{aligned}
 |i_B(i_S = 0\ \text{A}, v_B = v_{\min})| &= |i_B(i_S = i_{\max}, v_B = v_{\max})| \\
 \left(\frac{v_{\min}}{R_L}\right) \frac{1}{1 + \frac{R_0}{R_L}} &= -\left(\frac{v_{\max}}{R_L} - i_{\max}\right) \frac{1}{1 + \frac{R_0}{R_L}}.
 \end{aligned} \tag{5.17}$$

Equation 5.17 results in

$$R_L = \frac{v_{\min} + v_{\max}}{i_{\max}} = \frac{70\ \text{V} + 100.15\ \text{V}}{16\ \text{A}} = 10.63\ \Omega \tag{5.18}$$

and

$$i_{B,\max} = \frac{v_{\min}}{R_0 + R_L} = \frac{70\ \text{V}}{0.5\ \Omega + 10.63\ \Omega} = 6.29\ \text{A}. \tag{5.19}$$

Algorithm 5.2 Brute force algorithm to find out R_L and $i_{B,\max}$

```

i_Bmax = 0
R_Lmax = 0

for every i_B in [0, i_max]
    for every R_L in [min(realizable R_L), max(realizable R_L)]
        for every v_B in [v_min, v_max]
            calculate i_S and i_L when charging with i_B
            calculate i_S and i_L when discharging with i_B
        endfor

        if max(i_S) < i_max && max(i_L) < i_max && i_B > i_Bmax
            i_Bmax = i_B
            R_Lmax = R_L
        endif
    endfor
endfor

```

$i_{B,\max}$ is now the maximal possible i_B which does not violate the boundary conditions and $R_{L,\max}$ is the corresponding R_L

To test if the above line of argument is correct, a script was written that finds out R_L and $i_{B,\max}$ using a brute force algorithm (see Algorithm 5.2). It tries all possible $i_B \in [-i_{\max}, i_{\max}]$ for every realizable resistor R_L in the possible battery voltage range $v_B \in [v_{\min}, v_{\max}]$ in a fine discretization and saves the maximum i_B that satisfies the boundary conditions of Equation. 5.13. This script leads to the same $i_{B,\max}$ than Equation 5.19.

The value of the switchable load should be chosen as close as possible to the theoretical optimum of Equation 5.18. It does not matter if the set resistor at the switchable load is smaller or bigger. The closed possible realizable resistor by the switchable load is $R_L = 11.0 \Omega$ which is fortunately very close to the calculated value. This results in

$$i_{B,\max} = \frac{v_{\min}}{R_0 + R_L} = \frac{70 \text{ V}}{0.5 \Omega + 11.0 \Omega} = 6.09 \text{ A}$$

and

$$scale_Q = \frac{i_{B,\max}}{i_{\text{meas},\max}} = \frac{6.09 \text{ A}}{86.163 \text{ A}} = 0.07. \quad (5.20)$$

Now the measurements and the battery model can be scaled with Equations 5.9 and 5.12. The battery model is scaled according to Section 3.4. Also the current measurement is scaled with $scale_Q$. The scaled current is then the set value for the LabVIEW program according to Section 5.3 and Equation 5.7.

6 PHIL Simulations

In the previous chapters various battery models, their suitability for Power Hardware-in-the-Loop (PHIL) simulations and their parametrization have been discussed, followed by the validation of the models. Then the PHIL setup has been described, including offset correction of real time PHIL measurements and the scaling of the existing battery measurements to be able to compare them with the PHIL results. Thus, all the preparation work is done to be able to run the final PHIL simulations. This chapter will first briefly discuss the post processing of the measurements and then analyze the performed PHIL simulations.

6.1 Recording and Post Processing of Measurements

To analyze the PHIL simulations, several measurements and calculated values were stored both at the real time system (RTS) and the programmable DC source. The data values of the real time system were stored in a file with a sampling frequency of 1 kHz. This was done by using the SIMULINK block *OpWrite* from RT-LAB. These values were

- the time t ,
- the measured battery current i_B which is the input of the battery model,
- the state of charge (*SoC*) calculated from the current i_B ,
- the emulated battery voltage $v_{B,set}$ as output of the battery model (set value),
- and measured voltage v_B (actual value).

Also the measurements of the programmable DC source, which were available within the written LabVIEW VI, were stored. These values were

- the time t_{LV} ,
- the current set value $i_{S,set}$,
- the measured current i_S ,
- the measured load voltage v_L ,
- and the estimated battery current $i_{B,LV} = \frac{v_L}{R_L} - i_S$.

In contrary to the measurements of the real time system, which were stored synchronous every 1 ms, the LabVIEW VI was executed on a standard PC operating Windows XP. Hence, the measurements were stored asynchronous approximately every 50 ms with a variety of a few

milliseconds. To be able to compare the measurements, they have been interpolated onto a equally spaced 1 ms time base.

Also the real time system and the LabVIEW program were started manually, so their measurements were not synchronized. To compare both measurement sources, the currents i_B and $i_{B,LV}$ were cross correlated in order to find out the time offset between the measurements and correct the offset. After these two steps, the measurements were on the same time base and synchronized and ready for analysis.

6.2 Discharging of a Battery with a Constant Resistor

Before the final PHIL simulation with an FTP-72 cycle (federal test procedure) as input, a trial run with a constant resistive load was performed to test the closed loop of real time system and amplifier. The setup is the same than used for correcting the offset and linearity error described in Section 5.2.2 and shown in Fig. 5.4.

As battery model the extended Modified Shepherd Model (eMSM, see Section 3.1.3) was used with the parametrization done in Chapter 3. The battery was scaled to a nominal voltage of 76.65 V with a capacity Q of 0.25 Ah in order to achieve a for the amplifier suitable voltage and a small discharge time. The short discharge time was chosen for practical reasons to see significant changes of the voltage in a few seconds instead of minutes.

The discharge of this scaled battery with an initial state of charge $SoC_{init} = 0.15$ and a discharge resistance of $R_L = 17.8 \Omega$ is shown in Fig. 6.1. The absolute error between the offline simulation (pure software simulation) and the PHIL simulation stays under ± 2 V and the weighted relative

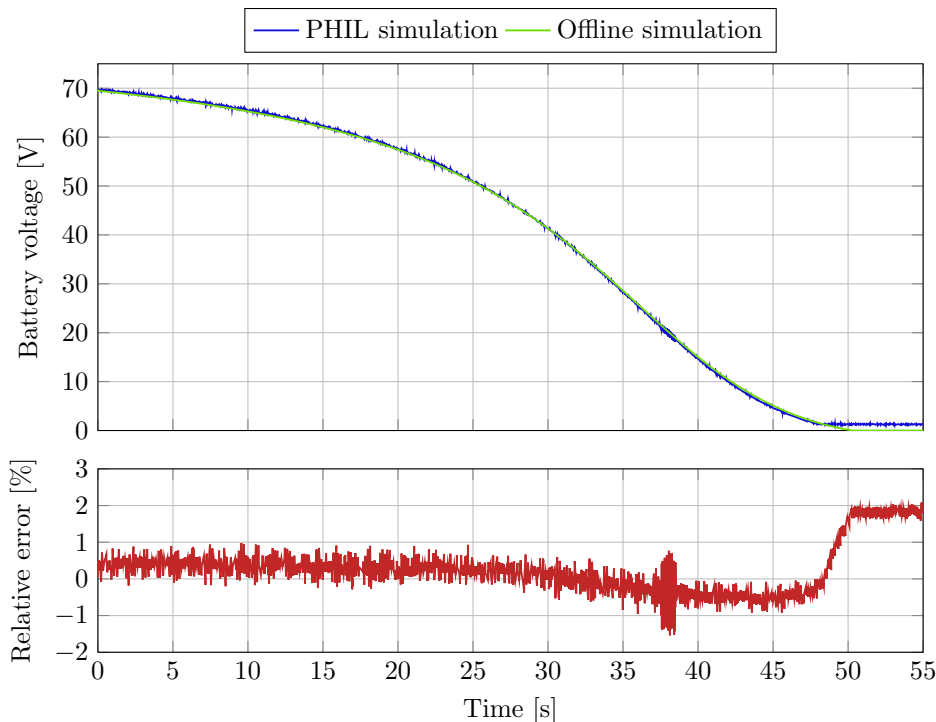


Figure 6.1: PHIL simulation of the discharge of an emulated battery with a constant resistor

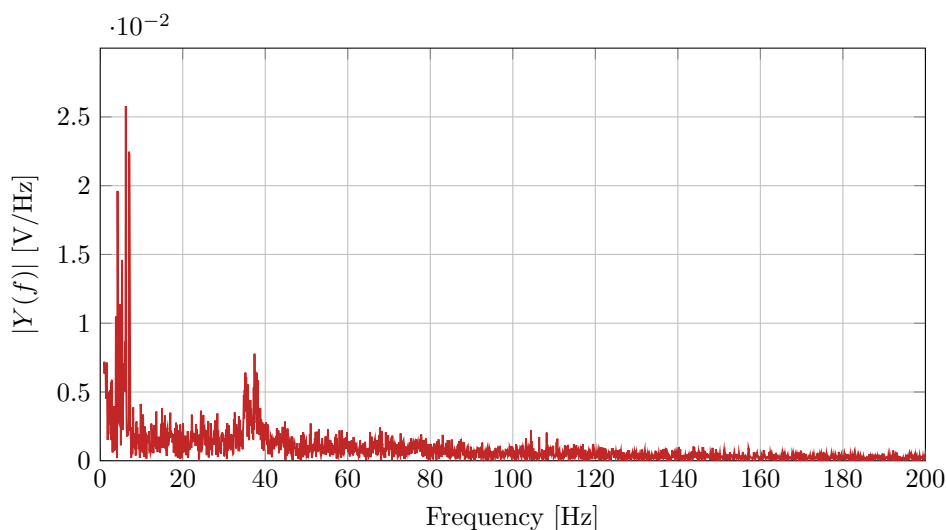


Figure 6.2: Spectrum of the absolute error between offline simulated and PHIL simulated battery voltage (without direct component)

error under $\approx 2\%$ (according to the error definition of Equation 4.2 on page 46). The absolute error is roughly independent of the set value and consists of two parts. The figure also shows that the power source is not capable of setting the output voltage to zero, but remains at $\approx 1.5\text{ V}$. This is not a big issue for practical experiments, since this is only 0.25% of the nominal voltage of 600 V and usual operating voltages won't be around 0 V . The output voltage can be set to zero by disabling the `VOLTAGE_ON` digital input of the power source (cf. Table 5.2).

Fig. 6.2 shows the amplitude spectrum of the absolute error between the offline simulated voltage and the PHIL simulation voltage. It was calculated via a Fourier transformation of the measurements. The direct component is not shown because its amplitude is about ten times higher than the second highest component. The figure illustrates, that the error consists not only of noise, but also components with 6 Hz , 7 Hz , 35 Hz and 38 Hz . This is probably due to the internal control loop of the power source that sets the output voltage of the source according to the set value. A correlation of the periodical errors and the RTS with a sampling frequency of $f_s = \frac{1}{10\ \mu\text{s}} = 100\text{ kHz}$ could not be found.

This simple PHIL simulation shows that the combination of the present RTS and the power source is capable of reproducing the results of the offline simulation with a maximum error of 2.16% and a RMS error of 0.69% . Hence it can be continued with the final PHIL simulation.

6.3 Simulating the FTP-72 Cycle

Now all the preparatory work is done to reproduce the battery measurements with an emulated battery in a PHIL simulation. Chapter 2 discussed potential battery model, in Chapter 3 they have been parametrized and afterwards validated against the measurements. Chapter 5 described the PHIL implementation, which RTS and power amplifier were used and how they interact. Their interaction was then verified in Section 6.2.

As last PHIL simulation in this work a current on the basis of the FTP-72 cycle (cf. Section 1.4.3 and Fig. 4.5 on page 50) is used. Here, the current is bidirectional and this experiment is a practical use case, because the cycle is a standardized driving cycle for cars. As battery model the eMSM was used with the parametrization values listed in Table 3.3 and scaled according to Equation 5.11 and 5.20. Following the analysis in Section 4.3, three FTP-72 cycles were executed: one with an almost fully charged battery, one with an initial SoC of about 50% and one cycle with an almost empty battery. The comparison between the offline simulations and the PHIL simulations showed that their difference is independent from the voltage of the simulated battery. Hence, only analysis for the cycle with $SoC_{init} \approx 0.5$ is shown further on in this chapter.

6.3.1 Comparing the Currents

At the original measurement, the discharge current was generated by a switchable resistor. While switching, the conductance was zero for short moments. This created troubles for the programmable DC source. Thus it was filtered (see Fig. 5.7 and Equation 5.8). After filtering, the current was scaled according to Equation 5.20 to ensure that the current did not exceed the maximum current of the used cables. The current values are then used as input for the programmable DC source which sets a current i_S in order to create the desired battery current i_B (see Equation 5.7 and Fig. 5.1 for the experimental setup). Fig. 6.3 shows a comparison between the set value i_B and its actual measurement. The figure basically shows the same results than Fig. 5.6. First, there is a noise with an RMS (root mean squared) error of $\approx 1\%$, which is the combination of a noise and its impact on the current set value. But then the figure shows also overshoots when the gradient of the set value is too high. This overshoots are generated by the

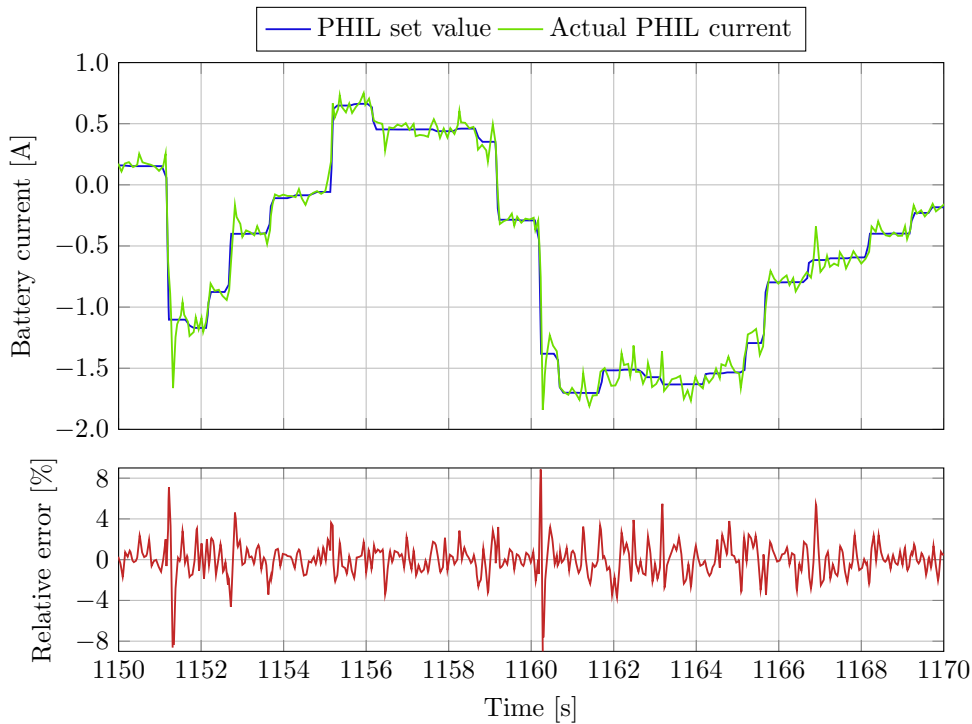


Figure 6.3: Comparison between the set value of the battery current and the actual generated PHIL current

internal control algorithm of the programmable DC source and result in a short error up to 20%. But still, an input current for the battery emulator with an overall RMS error of $\epsilon_{\text{RMS}} = 1.23\%$ and a mean error $\epsilon_{\text{mean}} < 0.01\%$ can be created, which was considered to be accurate enough. Future work has to show if these overshoots can be reduced by changing internal parameters or filtering the set values. It is important to mention that the error of the current influences the output through the calculation of the battery voltage (cf. Equation 3.15). The output voltage results in a current which is then measured again, affects the voltage and so on.

Also the type of modeled battery makes a difference, because a lead-acid battery has a higher internal resistance R_i than a Li-ion battery as well as increasing the capacity normally decreases the internal resistance. An error ϵ_i of the current directly affects the output voltage by $R_i \cdot \epsilon_i$.

6.3.2 Comparing the States of Charge

The error of the generated PHIL current produces not only a directly error at the output value. It also influences the state of charge (*SoC*) estimation. Fig. 6.4 shows a comparison between the estimated *SoC* from the original battery measurements and the *SoC* calculated during the PHIL simulation in the battery model. To estimate the state of charge of the measurements, Coulomb Counting (Equation 2.4) was used with the capacity $Q = 20.86 \text{ Ah}$ and $\text{SoC}_{\text{init}} = 98.89\%$ (cf. Section 4.3 and Equation 4.9).

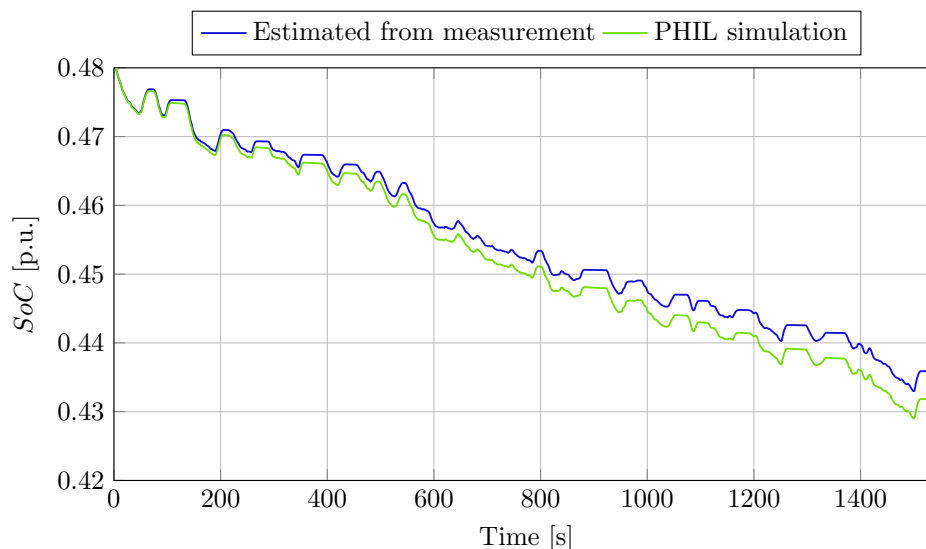


Figure 6.4: Comparison between the estimated *SoC* from battery measurements and the PHIL simulation

The figure shows, that the noise of the PHIL current has no noticeable influence on the *SoC* calculation. But despite the offset correction done in Section 5.2.2, there is still an offset error which leads to a drift of the *SoC*. After one cycle, the error is about 0.4%.

6.3.3 Comparing the Battery Voltages

Finally a comparison of the battery voltage of the offline simulation from Chapter 4 and the PHIL simulation is done. The offline simulation results are scaled according to Equation 5.11 to

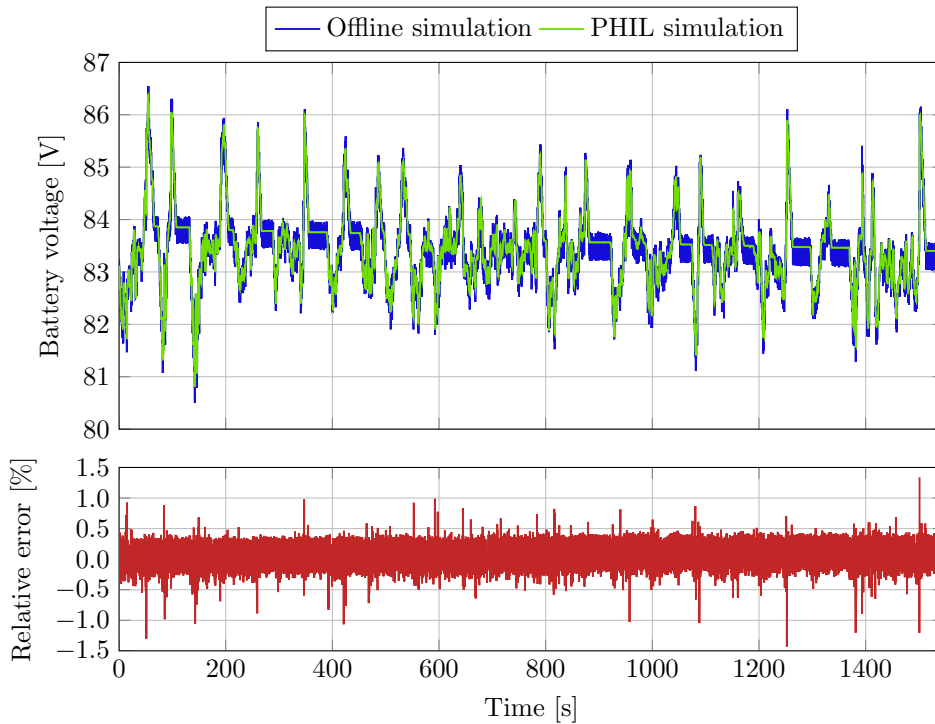


Figure 6.5: Comparison of the battery voltages between the offline simulation and the PHIL simulation

be able to compare them. Fig. 6.5 shows the two voltages. The figure shows that there is an error varying between $\pm 0.5\%$ or ± 0.3 V. When there are fast changes of the battery current, the error of the resulting change of the battery voltage increases up to $\pm 1.5\%$ or ± 0.9 V. The average error is slightly increasing which correlates with the error of the *SoC* calculation. The mean error is below 0.01%, the RMS error 0.15% or 0.13 V and the maximum error 1.72% or 1.48 V. It is important to mention also the absolute errors in Volt and not only the weighted relative errors, because the absolute errors are almost independent of the average of the battery voltage. A PHIL simulation near to the nominal voltage $v_{\text{nom}} = 600$ V of the power source, would reduce the relative error by a factor of 6. Hence, it makes no sense to scale the voltage of the PHIL simulation down to a single cell and compare it with the original measurements of the battery.

A detailed look at the error reveals that it is not noise, but deterministic. Fig. 6.6 shows a closer look at this error. It is the superposition of various sines with various amplitudes. This becomes clearer with a look at the spectrum of the error which is shown in Fig. 6.7. There are nine frequency components starting at 7.5 Hz with an increment of ≈ 1.9 Hz. The spectrum is different to the one of the constant discharge in Fig. 6.2. Hence, it seems plausible that this error comes from the combination of the power source and the programmable DC source. They both have an internal control loop: one that controls the voltage v_B , the other the current i_S . The two control algorithms are interacting with each other which produce this periodic error. This is a classical phenomenon that appears at PHIL simulations where the power amplifier adds a new error that would not be here if the experiment would be conducted with a real battery instead of the emulated battery.

An extensive analysis of the interaction between the two controlled source and their internal parameters is necessary to find out if and how the error can be reduced as well as if input filters

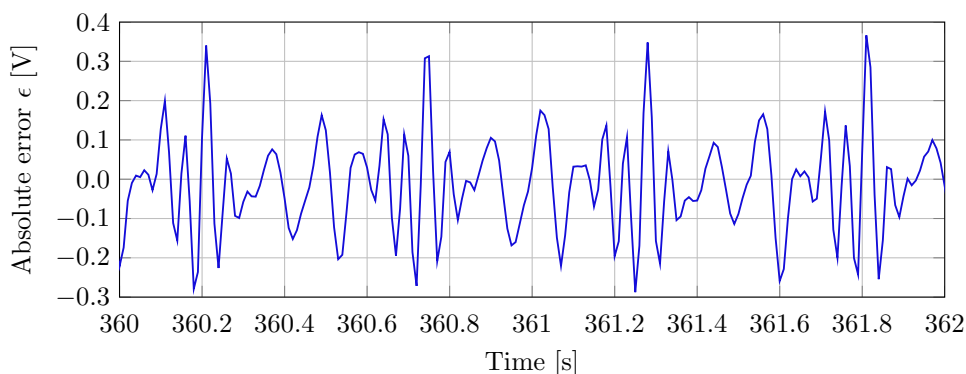


Figure 6.6: Closer look at the error between offline and PHIL simulation

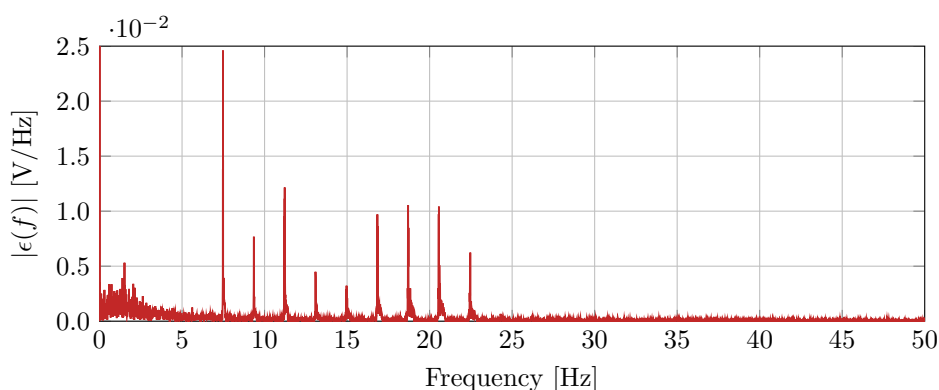


Figure 6.7: Spectrum of the absolute error between offline and PHIL simulation

at the RTS can significantly oppress this error. This goes beyond the scope of this work and hence was not done.

To sum up the PHIL simulations, the setup of the OPAL-RT RTS together with the Regatron bidirectional power source is capable of reproducing the offline simulations. The interaction between the two controlled source, the bidirectional power source and the programmable DC source, creates a deterministic error which can be described with the superposition of 9 sine signals between 7.5 Hz and 22.7 Hz . This error is independent of the mean value of the actual emulated battery voltage and about ± 0.4 V. It makes no sense to compare the measurement of one battery cell with the results of a scaled PHIL simulation, since the measurements were filtered to remove spike that the programmable DC cannot handle well. Also with the “constant” error and variable scaled PHIL battery voltages, the rescaled error can be arbitrary increased or decreased at will. But it makes sense to compare the PHIL simulation with an offset simulation, where the used setup leads to an RMS error of 0.15% and 1.72% maximum error for a battery voltage of about 84 V. These errors are sufficiently low, so the battery emulator can be considered as well-performing. A conservative estimation of the total error between the measurements of a cell and the results of the PHIL simulation would be adding the maximum relative errors. This leads in a maximum error smaller than 5% for an state of charge area between 10% and 100% and an error smaller than 10% for the whole state of charge range.

7 Discussion and Conclusion

In this final chapter of the thesis, the work is reviewed and discussed. Also some ideas for future work are mentioned, followed by personal experiences. Finally a summary and conclusion about this work is presented.

7.1 Review and Discussion

The battery emulator developed in this work will extend an existing test bed which already emulates an AC microgrid to be able to test battery related power electronics. According to Section 1.4 on page 5 the requirements for the battery model of such a battery emulator are a generic model which models the battery voltage as a function of current and state of charge, an execution time below 10 μ s for the existing OPAL RT real time system (RTS), and an easy parametrization. The output of the model will be the set value for a bidirectional power source and the measured current of the source the input of the model. As a result of a state of the art analysis of battery modeling, discussing the parametrization of battery models and validating of battery models (Chapters 2 to 4) the extended Modified Shepherd Model (eMSM) turned out to be the most suitable model in terms of the requirements. After choosing the right battery model, a Power Hardware-in-the-Loop (PHIL) setup was implemented and verified. The necessary steps to accomplish this result are reviewed and discussed subsequently.

7.1.1 Requirements and the eMSM as Most Suitable Model

The boundary condition of an execution time below 10 μ s of the battery model was not such a hard condition than expected. It was obvious very soon in the state of the art analysis that electrochemical models are computationally too expensive to meet this condition for reasonable expensive state of the art real time systems. And the remaining possible models (equivalent electrical circuit models and extended Modified Shepherd Model) have absolutely no problem to fulfill the execution time boundary condition. So there was no need for optimizations in the implementation to reduce the execution time.

The finally chosen extended Modified Shepherd Model fulfills all the requirements for the battery model complexity defined in the problem statement in Section 1.4 on page 5. One reason for this is that the requirements of the model were not definitely given at the beginning of the work. They have been a result of a process during the literature research, because at the beginning there

was hardly any knowledge about battery modeling and the complexity as well as the quality of battery models. At the beginning there was the plan to create a battery model for lead-acid as well as 2nd life Li-ion batteries. Also the temperature should be a parameter of the model so that there can be distinguished between a cold start of an electric vehicle in the winter and in the summer. But there was only measurement data for a new Li-ion battery recorded at 23 °C available (cf. Section 1.4.3). So these two tasks were postponed to future work (see Section 7.2).

Another reason why the eMSM fulfills all the requirements for the model is that there have been no quantitative requirements for the accuracy or a maximum error. The model should just be as accurate as possible under the other boundary conditions. After the offline simulations to validate the battery models and the PHIL simulations to verify the PHIL setup, their accuracies have been discussed with experienced colleagues and considered to be high enough.

It was surprising that hardly anybody in the literature mentioned the similarities of equivalent electrical circuit (EEC) models and generic mathematical models. Although they have a different approach, their equations are very similar, like it is worked out in Section 2.7. It seems that the trend in battery modeling goes to EEC models. Probably because of the intuitive understanding of the behavior, the scalability by adding or removing RC-elements and the simple “drag and drop”-implementation in programs like SIMULINK. Its toolboxes *Simscape* and *SimPowerSystems* are both able to simulate electrical circuits by drawing them. Their parametrization is also very straight forward by measuring the open circuit voltage and the voltage response to discharge pulses.

Contrariwise, a textual implementation would be less intuitive and more complex, but faster and more flexible, especially when the component values depend on the state of charge (*SoC*), the current direction and rate, or the temperature. Strictly speaking, the model isn’t an EEC model anymore, but a generic mathematical model. A good example of the benefits of generic model is the term for the polarization resistance

$$R_{\text{pol}} = \begin{cases} K \frac{1}{\text{SoC}}, & i^* > 0 \\ K \frac{1}{\lambda - \text{SoC}}, & i^* < 0. \end{cases}$$

of the extended Modified Shepherd Model (Equation 3.16 on page 37). With this simple term, the eMSM is able to model the charge and discharge efficiency subject to *SoC* qualitatively correct. The polarization resistance increases when an almost empty battery is discharged or if an almost full battery is charged. A higher resistance results in higher losses which correlate with the charge and discharge efficiency. None of the other simulated models in this work is capable of modeling this behavior.

7.1.2 Parametrization of the Models

In contrary to fulfilling the short execution time boundary condition, the parametrization of the models was far more complicated than thought at the beginning. In theory, it is easy to determine the parameters of an open circuit voltage model (OCV model) from measurements by using an optimization algorithm. But in practice, a good choice of the initial values is crucial. Otherwise the algorithms get stucked in a local optimum far away from the global optimum. This applies all the more accurate OCV model tested (Zheng, eMSM, Lamand Chen, cf. Table 3.1 on page 33), but not for the Nernst model. So some “trial and error” is needed to find out good initial values. Unfortunately this is far away from loading the measurements, pressing a button and getting the correct result.

The automatic determination of the resistor and capacitor values of an EEC model from a pulsed discharge took some time, but is now ready to use. Here, the selection of the initial values is not so crucial than for the OCV curve.

The extraction of the data out of the discharge curves from a data sheet was very straight forward. There are several programs that claim to automatically do this after defining the axis manually. All of the tested programs have been able to perform this, which is probably because these curves are monotonically decreasing. Out of interest, the voltage response to a pulsed discharge (cf. Fig. 3.2 on page 34) was also analyzed, but there most of the programs fail to extract the numeric values of the curve out of a raster graphics. They got stuck around at an “edge” when a pulse started or finished. So in the worst case, the data have to be extracted by manually clicking on points at the curve. This is not difficult, but time consuming. After extracting the data out of the figure, the parametrization process is the same than for a parametrization with measurement data.

As a matter of interest, a second battery was parametrized with data sheet figures. This was a *6FM150-X* lead-acid battery from *VISION*, because a home storage system, that contained four of these batteries, was placed for a few weeks in the research test bed to perform some tests. Unfortunately the quality of the figures in the data sheet of the *6FM150-X* lead-acid battery was very bad (cf. Fig. A.3 on page 95). The discharge curves are not monotonically decreasing, which should be the case for a lead-acid battery. The curves were not smooth at all and an integral over the current results in a withdrawn charge of more than 25% over the nominal capacity. In a nutshell, the discharge profiles are useless for an exact parametrization of a battery model. This shows that a parametrization on the basis of the data sheet is not always possible.

It is difficult to decide whether the requirement of an easy to parametrize model is met or not. The parametrization takes some manual steps and is not completely automated. However, the parametrization of an electrochemical model would be far more complicated. The parametrization of the Nernst model would be more robust, but it is less accurate than the other examined OCV models, especially when $SoC > 95\%$ and the OCV tends to infinity. So some manual steps cannot be avoided.

The parametrization process revealed that the accuracy of the OCV part of the Modified Shepherd Model can be increased significantly by slightly changing the model equation and adding a new parameter SoC_0 . At this so called extended Modified Shepherd Model, the point of break down of the battery voltage for low state of charges can be moved to a certain SoC with this new parameter. With this improvement the eMSM has the best accuracy under all analyzed OCV models with one parameter less the Lam model and Zheng model, which were second and third ranked, concerning accuracy. Hence, the OCV part of the eMSM was used as OCV model for the EEC models.

7.1.3 Validation of the Models

A very important first step of the validation was choosing the right solver and a small enough maximum step sizes for the simulation. The *auto* option of SIMULINK for the maximum step size produces too high step sizes resulting in a wrong calculation of the SoC via Coulomb Counting (see Equation 2.4 on page 13). The drop of the SoC is overestimated resulting in states where the calculated SoC is zero whereas the original experiment is still going on. Too small step sizes or fixed step size solver unnecessary massively increase the simulation time. Following the *Choose a*

Solver guideline from the MATHWORKS homepage [6] and performing some simulations resulted in the solver *ode45* with a maximum step size of 0.1 s. This example shows that the automatic settings of SIMULINK cannot always be trusted and that it is important to check the settings and results manually.

As expected, the validation of the measurements showed that the withdrawn charge of the batteries varies a few percent compared to the nominal capacity. This makes it impossible to make exact life time predictions of a battery. Due to the Rate Capacity Effect (cf. Section 2.4.1) the usable capacity depends on the discharge current and is under normal working conditions always smaller than the real capacity. Therefore, the nominal capacity of a battery is smaller than the real capacity to achieve usable capacities in the range of the nominal capacity. The problem statement mentioned the typical duration of a PHIL simulation is in the range of seconds to an hour. In this time range, the Rate Capacity Effect has hardly any impacts, especially for Li-ion battery. But the figures in Chapter 4 with the repeatedly applied test cycle showed that the effect plays a role for longer experiments. If there are PHIL simulation with longer durations, or if the battery model is used for offline simulations, the Rate Capacity Effect and with it the Recovery Effect must be taken into account.

A solution would be replacing the Coulomb Counting with the Kinetic Battery Model (KiBaM). Fig. 4.11 on page 57 shows that using the KiBaM, the *SoC* prediction is more accurate and the error is less. For this figure, the parameters of the KiBaM were determined by trial and error. The resulted parametrization not only decreased the error at a low *SoC* but also increased the error at the beginning of the simulation. This shows that the parametrization is not optimal and that there should be a combined parametrization of the OCV curve and the KiBaM. Such a combined parametrization will be done in future work.

Beside the inaccurate state of charge prediction for long durations, the accuracy is very high. At the high and medium *SoC* area, the maximum error of the models parametrized with data sheet is below 1%. The RMS (root mean squared) error for one FTP-72 cycle (federal test procedure) is below 0.15% (see Table 4.3 on page 52). The accuracy is so high, that the EEC model with constant resistor and capacitor values is accidentally more accurate than the one with look-up tables for the values. In theory, this should not happen, because the constant value-model is a simplification of the look-up table-model. But the difference between them is so small and the accuracy so high, that the in theory less accurate model accidentally leads to a more accurate simulation result.

With the difference between the mean error and the RMS error, it is possible to estimate whether the error is mainly a constant error, which can be ascribed to the OCV model, or if inaccuracy is caused by model weakness. A look at Table 4.3 on page 52 reveals that the difference between these two errors is quite small for a cycle with initial state of charge of 98.89% and 52.15%, but higher for the cycle with $SoC_{init} = 9.06\%$. This suggests the assumption that the EEC models as well as the eMSM are capable of modeling the behavior for high and middle states of charge, but show signs of weakness when the *SoC* is low. However, the main cause of the higher error is still the wrong state of charge estimation.

The simulation of the eMSM with data sheet parametrization has an error smaller than $\pm 2\%$ in comparison to the measurements of the real battery. The only problem is that the estimated capacity (nominal capacity) is a few percent lower than the real capacity. So the voltage breaks down too early because according to the model the battery is empty, while at the measured battery, the *SoC* was still a few percent. So life time predictions are not possible, but the

qualitatively behavior can be reproduced respectively predicted within an accuracy of $\pm 2\%$. This is a very satisfying result, considering that only data sheet information and an estimation of the time constant τ was used.

7.1.4 PHIL Implementation and Simulation

Originally, the PHIL implementation and simulation only should be two smaller parts of a few. But as always, the devil's in the detail and in the end the PHIL implementation and simulation took the most effort. Beginning with how to use the real time system, how to design a SIMULINK model that can be compiled for the real time system, how to compile a model and execute it at the RTS. Followed by learning how to use the analog to digital converters and process analog input signals, as well as how to use the digital to analog converters to use SIMULINK signals as analog output signals. After studying how to use and control the *Regatron TC.GSS* power source, a small connector board was built in order to control the power source via the RTS. A closer look at the manual of the *TC.GSS* revealed that there is a minimum terminal voltage of $\approx 70\text{ V}$ for working as a sink and feeding energy into the grid. This limits the operating range of the battery emulator and makes it impossible to emulate a typical 48 V lead-acid storage system.

Then different kinds of measurement tools were tested. It turned out, that the *TC.GSS* creates a lot of electromagnetic pollution and current clamps cannot be used. So the internal measurement of the power source, which was accessible via its analog interface had to be used. Future work has to show, if there is another possibility to deal with this issue. After the decision of using the internal measurement of the power source, a correction of offset and linearity errors of the measuring chain was performed.

It took some time to find a method respectively a setup that is able to reproduce a scaled version of the current from the original battery measurement because there was no second bidirectional power source available. The final solution is a combination of a constant load and a programmable current source *Regatron TopConQuadro TC.P* (for the setup see Fig. 5.1 on page 60). The *TopConQuadro* can be controlled via a RS 232 interface and a LabVIEW program. Because the current source measures also its terminal voltage and the value is available within LabVIEW, it is possible to create a desired battery emulator current independent of the battery emulators voltage. To achieve this, a LabVIEW program had to be written that reads a csv-file with battery current set values, reads the voltage measurement and calculates the output current of the programmable current source (cf. Section 5.3 on page 67). The library for the RS 232 communication provided by *Regatron* had to be tuned in order to reduce the latency of setting new set values and reading measurements. This reduced execution time resulted in an increased accuracy.

For the PHIL simulations the measurements as well as the battery model have to be scaled because of limitations of the power source and of the used cables. When a battery is charged, it absorbs electrical energy and stores it as chemical energy. When the built battery emulator is "charged", the energy is not lost by heating power resistors, but fed back into an AC grid. Due to the internal architecture of the amplifier (cf. Fig. 5.2 on page 62), the minimal voltage for feeding is limited to 70 V . Hence, the battery voltage had to be scaled from a nominal voltage of 3.65 V to $\approx 88\text{ V}$. Also the cables of the lab setup are only capable of maximum 16 A , while the original measurements had a maximum current of about 86 A . Therefore, the capacity of the emulated battery and the input current had to be scaled. This shows that the battery emulator cannot emulate every battery voltage up to the nominal voltage of the power source. Also that

the test bed needs thicker cables reproduce typical current for some battery applications. But this is an issue than can be easily solved.

The PHIL simulation showed a deterministic error of the battery emulator voltage which is the superposition of nine sine waves from 7.5 Hz to 22.7 Hz with a resulting amplitude of about 0.3 V and a maximum error of 1.48 V. This error clearly exceeds the noise as the spectrum of the error in Fig. 6.7 on page 79 shows and is in good approximation independent on the amplitude of the battery voltage. The deterministic error probably caused by the interaction between the internal control loops of the two controlled sources. This is a classical PHIL issue, where an additional error appears, that would not exist when using a real battery instead of the emulated battery. Future work has to show if this problem also occurs for other hardware under test and how the error can be reduced.

With filtering the current as input of the model, scaling the battery voltage by the factor 24 and an error that is almost independent of the voltage it makes no sense to compare the PHIL results to the original measurements. First, the battery and the battery model have a different input. Secondly, the PHIL voltage cannot be scaled back to a cell voltage, because the error scales not this way. A very conservative estimation is adding the maximum relative errors between the measurement and the simulation of a single cell, and between the offline simulation and the PHIL simulation. This results in a maximum error smaller than 5% for $0.1 < SoC \leq 1$ and an error smaller than 10% for the whole state of charge range. This error was considered to be small enough for replacing a physical battery with a battery emulator.

7.2 Future Work

Possible future work can be divided in extending the battery model for other battery types, improving the accuracy of the model and improving the signal handling between the RTS and the power source. Some of the improvements can be implemented in a few hours and just need input data for parametrization, while others require extensive work.

7.2.1 Extending the Model for other Battery Types

One wide field of future work is the extension to other battery types. In principle, the battery models used in this work are capable of modeling not only Li-ion batteries, but also lead-acid, NiMH (Nickel metal hydride) and NiCd (Nickel Cadmium) batteries. They only need to be parametrized. So a logical future work is to use measurements from other batteries and parametrize the presented battery models. Also, several parametrizations of batteries can be found in the literature, but their quality and accuracy is not guaranteed. The literature review showed that the results of the papers are often sugarcoated.

The battery voltage of lead-acid, NiMH and NiCd batteries have a hysteresis effect (see Section 2.5.7 and [OD09]). When parametrizing batteries with this type, the hysteresis effect should be considered to increase the accuracy. The additional implementation effort is small, because only one differential equation has to be implemented (Equation 2.43 on page 21) and even more simple hysteresis models can be found in [HLP12]. But it is not clear how complex the additional configuration effort is.

In the future, there will be a huge availability of Li-ion batteries at the end of their life cycle. Their capacity decreased due to capacity fading. So their primary usage is not economic any more. This are for example heavily used batteries from an electric vehicle. But there are fields of applications where the capacity per kg or the capacity per m³ is not so important, but the costs per Wh – for example home storage systems or storage systems for primary regulation and power quality. This is a huge potential field for battery energy storage systems, battery related power electronics and battery emulators.

7.2.2 Improving the Model Accuracy

The validation in Section 4.3 showed that the Kinetic Battery Model (KiBaM) improves the state of charge (*SoC*) estimation for long simulations, because it models the Rate Capacity and the Recovery Effect. Further measurements and work is necessary to systematically parametrize the KiBaM in combination with an equivalent electrical circuit (EEC) model or the extended Modified Shepherd Model. The method described in [KQ11] is in my opinion not the right way because on the one hand it includes discharge with a “very large load” which can destroy the battery and on the other hand, the KiBaM and the EEC model are parametrized separately. Fig. 4.11 on page 57 shows that a separate parametrization leads to higher error under some conditions than with Coulomb Counting. This should not happen, since the Coulomb Counting is a special case of the KiBaM.

The behavior of batteries depends on the temperature of the battery. This is a circumstance that is neglected in this work. Some authors already included temperature dependency like the in this work several times cited work of Lam et al. [LBK11]. The open circuit voltage vs. *SoC* curve is usually assumed to be independent of the temperature, while the capacity is modeled $Q \propto \exp -\frac{1}{T}$. But of course, more complex models are possible. While equation for the capacity is based on the Arrhenius equation [11], which is an equation used to model the temperature dependence of reaction rates in chemistry, the temperature dependency of the values of the resistors and capacitors is modeled empirically. Extracted values from measurements are either stored in look-up tables with interpolation between the table values, or mathematical functions which fit these values. But in general, the resistance is lower for higher temperatures, while the capacitances are higher.

The temperature of a battery is not always only given by the ambience temperature. Every battery has its losses, which heat the battery and for some use cases the thermal losses cannot be dissipated completely and the battery warms up. A possible extension of the battery model is the extension with a thermal model that models the heat generation as a result of the losses in the battery, the convective heat as well as the conductive heat transfer to the ambience and the temperature rise of the battery.

Another possibility to extend a battery model and improve its accuracy is to make the parameters current rate dependent. The implementation is in principle the same than for the temperature dependency as for example done in [LBK11] or [TSK12].

7.2.3 Improving the Interaction of RTS, Power Source and HuT

The analysis in Section 6.3.3 showed that there is a deterministic error between the set value and the actual value of the power sources voltage. This error is probable due to the interaction between the controlled voltage of the power source and the controlled current of the programmable DC source. This phenomenon will likely occur also in future applications where the battery emulator is connected with battery inverters, electric vehicle charge stations and so on. Maybe the instant accuracy is not high enough, so an analysis and justification of the internal control parameters of the power source will be necessary. This can also include additional band-rejection filter, implemented in a SIMULINK model at the RTS.

By now, the input signals of the RTS are filtered with a simple PT1 low pass filter in order to reduce the noise. Since the low execution time of the models allows additional computation effort without violating the boundary of 10 μ s, more complex filters are possible to better reduce the noise.

The Regatron TC.GSS power source creates a lot of EMC pollution (electromagnetic compatibility). This makes it very difficult to use current clamps with Hall effect sensor. For example, a *Lem PR30* current probe measured up to 2 A when put around a 20 cm long wire which was placed near to the power source. Hence, the internal current measurement of the power source was used as input signal for the RTS. Additional work has to show, if the Regatron device creates unwanted electromagnetic interference with other devices and what is need to be done, to be able to use current clamps.

7.3 Personal Experiences and Perceptions

In the beginning it was difficult for me to recognize the usability of a paper and to decide before reading most of the paper whether the quality of the paper is good or not. I also tried to reproduce the results of the papers by implementing the models and simulating them and was not always successful. This led me to the impression that the results of some papers are sugarcoated. In the end, for me a good indicator to assess the quality of a paper turned out to be its figures. If they are carefully created and have a good quality, the rest of the paper often also has a good quality and vice versa. In future work, I will also focus mainly on journal papers and less on conference papers.

I decided to write this thesis in English for two reasons. First, if this work should not be only a means to an end, but read and used by other people, it should be written in English, because the potential number of readers is a few hundred times higher. Second, I saw this as a kind of investment: spend additional time and effort to write the thesis in English, but then have better skills in future professional life. But writing the thesis in English was much harder than expected, especially the switching between the daily life in German, and the composing of this document in English. Time will tell if the investment will pay off or if it was jut unnecessary additional effort.

I really love to experiment or simulate and see what happens if this or that is changed. But in retrospect, I should have focused more on reproducibility of the simulations and results, as well as keep in mind a camera-ready documentation and not only notes and documentation for myself. So for my future work, every simulation that produces a notable result will be enclosed with a script that runs the simulation, but also includes all the necessary pre- and post-processing. So

if a simulation will be conducted again with slightly different input parameters, it will just need a few clicks to do this. This will take off some of the fun of the work, but reduce the overall effort. Also the development of SIMULINK models, e.g. a battery model, and simulations which use them will be separated on a very early state of development. This is obvious and very easy to realize for text-oriented programming languages like C or Java, but not for SIMULINK.

The more I have read about batteries and battery modeling, the more I got the impression, that there is no deep understanding about the internal processes inside a battery, what happens exactly and why. This was very surprising for me, because batteries have been explored for more than 50 years. For example, the original Shepherd Model [She65a, She65b] is from the 1960ies. There are accurate models which describe the battery voltage. But it is difficult to find papers about aging mechanisms, or a theoretical description of how to build a battery that lasts more discharge cycles or whose capacity depends less on the temperature. Either this is not completely known yet or it is a well-kept secret of the battery manufacturers.

7.4 Summary and Conclusion

This work presents a working battery emulator for Power Hardware-in-the-Loop simulations. The battery emulator extends the research test bed from the Energy department of the Austrian Institute of Technology. So far, this test bed can emulate an AC microgrid with varying frequency, voltage and net impedance, as well as photovoltaic (PV) panels and programmable RLC loads (cf. Fig. 1.4 on page 6). This gives the possibility to test PV inverters, energy and demand side management strategies, and so forth. With the developed battery emulator the test bed also can be used to analyze battery inverters, home storage systems, charging stations for electric vehicles, demand shifting strategies and so on.

This fields of application need battery voltages up to a few hundred Volts and in particular cases more than 20 kW battery power. The simultaneous emulation of an AC grid requires time steps of 10 μ s. Since the battery model is calculated in another core of the same RTS, the maximum calculation period is also limited to 10 μ s. A literature review revealed neither that battery models been analyzed in terms of their applicability for PHIL simulation, nor work about implementations of battery emulators with this fields of applications. Hence, this work is of high relevance to this field.

Beside the low execution time, the requirements of such a battery model are

- modeling the battery voltage (output of the model) as a function of the battery current (input of the model) and the state of charge,
- calculating the state of charge,
- taking polarization effects of the battery into account, and
- the model should be generic to be able to model different types of batteries.

Additionally, this work defined another requirement: an easy parametrization is favored. At best, only data sheet information and experience values should be necessary to parametrize the battery model, being aware that this will decrease the accuracy. This work focused on Li-ion batteries because only measurements from a Li-ion battery had been available. A future extension to other battery technologies was always kept in mind.

A state of the art analysis (cf. Chapter 2) about battery modeling yielded several types of battery models. Electrochemical models model the chemical processes of the battery. They consist of several differential equations, some of them are partial differential equations. Electrochemical models are the most accurate models, but their computational effort is too high for state of the art real time simulation systems and the condition an execution time below 10 μ s. Hence, they do not comply with the requirements for the desired battery model of this work.

Another type of battery models are generic models, also often named mathematical models. They consist of an empirical equation that describes the battery voltage as a function of the current and state of charge, and at least of a second equation that calculates the state of charge. Various generic models from the literature have been discussed and the Modified Shepherd Model (MSM, cf. Section 2.5.5) turned out to be the most suitable model among them.

Equivalent electrical circuit models (EEC models) pursue another approach. They consist of a voltage source (open circuit voltage, OCV), whose voltage is a function of the state of charge, an internal resistance and none, one or more RC-elements, depending on the desired accuracy of the model. Under the aspect of a low execution time and configuration effort, it is clear, that one RC-element is the preferred grade of accuracy. EEC models with one RC-element meet all the requirements. The literature review showed several mathematical models for the *SoC*-dependency of the OCV. Also the resistor and capacitor values can be either constant or a function of the *SoC*.

Six different OCV-models, three EEC models and the MSM have been analyzed and compared in detail. The parametrization process in Chapter 3 showed that the accuracy of the MSM can be increased significantly by adding an additional parameter. This applies both for the model itself, as well as its OCV-part used as OCV model for an EEC model. This model was named extended Modified Shepherd Model (eMSM, equations in Section 3.1.3) and was used further on in this work instead of the MSM. The eMSM turned out to be parameterizable by data sheet values and an estimation of a time constant, while a parametrization of EEC models only with data sheet data is more difficult.

Both, generic models and EEC models have the state of charge as state variable. The *SoC* is typically determined by Coulomb Counting that basically calculates an integral of the current. Coulomb Counting needs less computation time, but it does not take the Rate Capacity and Recovery Effect into account. This increases the error of the model for long simulation durations. A solution of this problem is using the Kinetic Battery Model (KiBaM, see Section 2.4.4 and 3.3) instead. The KiBaM consists of two charge wells, an inflow, a drain and a pipe between them. Two differential equations describe the changes of the filling heights of the wells. With this simple abstract model, the KiBaM is able to model these two effects that are caused by diffusion effects inside the battery. Unfortunately, no measurements have been available to systematically parametrize the KiBaM. Hence, Coulomb Counting was used for the validation.

After the parametrization, six models have been validated against measurements of a Li-ion battery from a standardized FTP-72 driving cycle (cf. Section 1.4.3 and Fig. 4.5 on page 50). These six models are four different EEC models, all parametrized based on measurements, an eMSM also parametrized with measurements and an eMSM parametrized with data sheet values (more details in Table 4.2 on page 51). The five models, which have been parametrized with measurements, have roughly the same accuracy. In the typical operating area of a battery between 10% and 100% state of charge, the maximum error is lower than 3% and the RMS error about 1.5%. The eMSM is the most accurate model, but the difference is not significant. At lower

state of charges, the error increases up to more than eight per cent. This has two reasons. First, the models fail to accurately predict the behavior in *SoC*-area of the battery. Second, the non modeled Rate Capacity Effect has an appreciable impact. The models overestimate the usable capacity and hence, overestimate the battery voltage. In comparison, a model with “trial and error”-parametrized KiBaM did not have this effect and the error was smaller than 3% until the battery was almost empty. But this way of parametrization is not applicable for real PHIL use-cases.

As expected, the eMSM that was parametrized with data sheet values has a higher error, but it still remains under 10%. For a driving cycle with an initial state of charge of about 50%, the maximum error is 2.32%, which is notably low. Overall, the eMSM was chosen to be the most suitable model in terms of accuracy, parametrization effort and the boundary condition of a low execution time. Therefore it was used in the PHIL simulations.

After the models had been validated, they were adapted to be able to run at the OPAL-RT real time system and executed afterwards. An analysis of the execution time showed, that all models easily meet the requirement of a time below 10 μ s. As expected the fasted model is the EEC model that uses look-up tables with an execution time of 2.35 μ s (cf. Table 5.1 on page 62). The eMSM has 2.65 μ s with the second lowest execution time. Thus, enough time remains to filter and process the input data.

The next step was building the actual battery emulator (see Chapter 5). This includes the implementation of some control and security mechanisms, the setting of the simulated battery voltage, the measurement of the actual PHIL current and its filtering within the RTS. The PHIL setup was verified with two simulations. First, the discharge of an emulator battery with a constant resistor. This was done to test the principle behavior of the battery emulator. After the experiment succeeded, the emulator was tested with the measured current of the standardized driving cycle. This profile contains dynamic currents in both directions, as well as rest periods. In order to reproduce this variable current, a combination of a constant load and a programmable current source was used and a LabVIEW program that generates a desired program was written.

The used power source has a limitation of minimal 70 V terminal voltage for feeding energy from the DC side back into the AC grid. Also the measurements from the FTP-72 cycle had a maximum of 86 A whereas the available cables in the lab were only capable of 16 A. Hence, the parametrized battery model and also the current measurements as set values had to be scaled. The original current set values contained short breakdowns to zero because they had been created with a digital potentiometer that disconnects for a short time while switching. These short break downs would have caused problems when using this signal as set value for the programmable DC source. Hence, the set values have been filtered with a low pass filter (see Fig. 5.7 on page 69).

After these three steps, the PHIL simulations had been performed with the eMSM as battery model. The experiment showed a maximum error of 1.48 V and an RMS error of 0.3 V between the offline and the PHIL simulated battery voltage. An error of 1.48 V is about 1.72% of the battery emulator voltage of 86 V. It is important to mention the absolute error in Volt and not a relative error, because the error is quite independent of the terminal voltage. A Fourier analysis of the error showed that it is not a noise but a deterministic superposition of nine sines starting at 7.5 Hz with an increment of ≈ 1.9 Hz (cf. Fig. 6.7 on page 79). The error is most likely caused due to the interaction two controlled sourced. This a classical PHIL phenomenon that would not occur with the use of a real battery. Up to which extent this phenomenon will also occur for another power hardware under test in a future PHIL experiment, depends on the hardware and

the use case. So further analysis and how to suppress this error has to be done for the specific device.

Since the battery model is scaled, the original current is scaled and filtered, and the error between offline and PHIL simulation is roughly independent of the terminal voltage, it makes no sense to scale the PHIL simulation down to a cell voltage of 3.65 V and compare the measurements of the real battery with the results of the PHIL simulation. A conservative estimation of the overall error between the real and the emulated battery is adding the maximum relative errors. This results in a maximum error smaller than 5% in a typical operating range between 10% and 100% state of charge and in an error smaller than 10% for the whole state of charge area.

The developed battery emulator was already used in the Vehilce2Grid-Inverter project [1]. In this project the battery emulator replaced the battery on an electric vehicle in order to test a novel charging station. This work is also the basis of two conference papers [SKLL13, SKLL14], presented at the 39th “Annual Conference of the IEEE Industrial Electronics Society” (IECON 2013) and the 23rd “IEEE International Symposium on Industrial Electronics” (ISIE 2014).

A side result of this work is a SIMULINK library for battery modeling. This library emerged out the model comparisons and simulations in this work and consists not only of the models that have been validated in Chapter 4, but of EEC models with one and two RC-elements and the eMSM. The OCV of any EEC model can be chosen as look-up table or as one of the six in Section 3.1.1 compared OCV models. The resistor and capacity values can be either from *SoC* dependent look-up table values, constant, or any of the five in Section 3.1.2.3 presented mathematical models. The *SoC* can be determined by Coulomb Counting or the Kinetic Battery Model. With this variety of different models, a lot of already parametrized and validated battery models from the literature can be used. The library is rounded by a battery management system to prevent a battery from exhaustive discharge or overcharge, a charge controller for constant current/constant voltage charging and some controlled loads. This library has already been used for offline simulations and pre-simulation of laboratory tests in two projects of the AIT and for simulations in a diploma thesis of a colleague.

To conclude, in this work a battery emulator of Li-ion batteries between 70 V and 600 V battery voltage has been implemented. The battery is simulated by the extended Modified Shepherd Model, which is a self-made extension of an existing model from the literature. This model is more accurate than equivalent electrical circuit models with one RC-element and also suitable to be parametrized only with data sheet information. The model was validated against measurements from a standardized driving cycle and has, parametrized with measurements, in the typical operating area ($0.1 < SoC \leq 1$) a maximum error of 3% and an RMS error of 1.31%. The whole battery emulator has a maximum error of about 5%, which is for a lot of use cases low enough to replace a real battery with a battery emulator. With such a setup, the battery voltage, capacity and state of charge can be set immediately and arbitrary. Thus, a battery does not have to be charged between two tests, which could take hours. Also there is no need to buy several batteries with different types, voltages or capacity, which saves time and money. So this work will help to make tests of battery related power electronics faster, cheaper and more flexible.

A Used Hardware

In this work several hardware has been used. These have been

- a real time system (RTS) that calculates the battery model and controls the
- bidirectional power source, which is used as power amplifier for the battery emulator,
- and a programmable DC source as well as
- a switchable load to create a bidirectional battery emulator current.

The setup of the hardware is shown in Fig. 5.1 on page 60. The RTS is described in Section 5.1, while the specifications of the other hardware are listed below. Excerpts of the data sheets of the used batteries are listed afterwards.

Power Source

As power amplifier a bidirectional power source *TC.GSS.32.600.400.S* from *Regatron* was used. Its specifications are listed in Table A.1. More information can be found at [9].

Parameter	Value
AC line voltage	$3 \times 360 - 440$ V
AC line frequency	48 – 62 Hz
DC power Range	0 – ± 32 kW
DC voltage Range	0 – 600 V
DC current range	0 – ± 66 A
static accuracy load regulation	$< \pm 0.1\%$ FS
static accuracy line regulation	$< \pm 0.1\%$ FS
response time load regulation	< 2 ms

Table A.1: Specifications of the *TC.GSS.32.600.400.S* bidirectional power source

Programmable DC source

To create a variable battery emulator current a *Regatron TopConQuadro TC.P.16.600.400.S* programmable source was used. Its specifications are listed in Table A.2. More information can be found at [8].

Parameter	Value
AC line voltage	$3 \times 360 - 440$ V
AC line frequency	48 – 62 Hz
DC power Range	0 – 16 kW
DC voltage Range	0 – 600 V
DC current range	0 – 32 A
static accuracy load regulation	$< \pm 0.1\%$ FS
static accuracy line regulation	$< \pm 0.1\%$ FS
response time load regulation	< 2 ms

Table A.2: Specifications of the *TC.GSS.32.600.400.S* bidirectional power source

Manually Switchable Load

Since the programmable DC source can only provide energy, but not absorb, an additional load was necessary. This load was a manually switchable load. A picture of it is shown in Fig. A.1.



Figure A.1: Picture of the manually switchable AC load

Originally, the load is a three phase AC load with maximum 9 kW switchable in 600 W steps. The according internal circuit of the load is shown in Fig. A.2. Via smart wiring and switch settings there are a lot of possible overall resistor values. Section 5.3 deals with how the combination of the programmable DC source and the load can be used to create a variable battery emulator current in both directions.

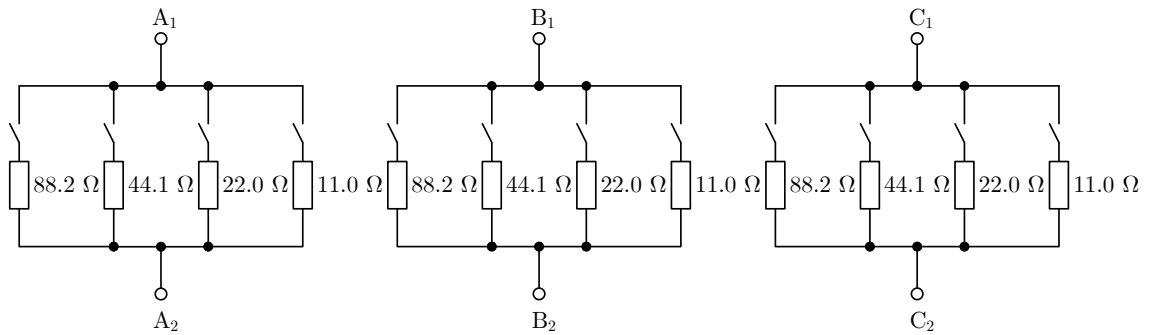


Figure A.2: Internal circuit of the switchable load

Data Sheets of the used Batteries

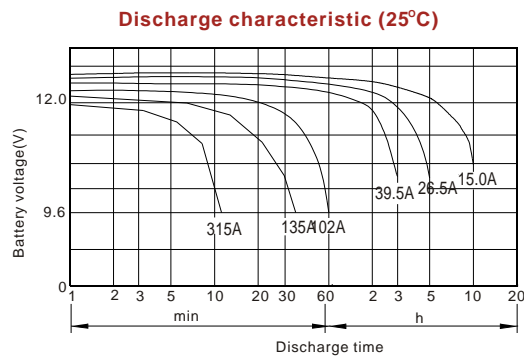


Figure A.3: Excerpt of the data Sheet of a *VISION 6FM150X* Lead-acid battery

ePLB C High Energy Product



Technology

Lithium Ion Polymer Battery
 Li[NiCoMn]O₂-based Cathode
 Graphite-based Anode
 High Energy Density
 Optimized for PHEV, EV

Product General Specification

Mechanical Characteristics

Model	C020
Length	217.0 ± 1 mm (excluding terminal)
Width	129.0 ± 1 mm
Thickness	7.2 ± 0.2 mm
Weight	approx. 428 g

Electrical Characteristics

Nominal Voltage	3.65 V
Nominal Capacity	20 Ah
AC Impedance (1 KHz)	< 3 mΩ
Specific Energy	174 Wh/Kg
Energy Density	370 Wh/L
Specific Power(DOD50%, 10sec)	2300 W/Kg
Power Density(DOD50%, 10sec)	4600 W/L

Operating Conditions

Charge Conditions :	
Recommended Charge Method	CC/CV
Maximum Charge Voltage	4.15 V
Recommended Charge Current	0.5 C Current

Discharge Conditions :	
Recommended Voltage Limit for Discharge	3.0 V
Lower Voltage Limit for Discharge	2.5 V
Maximum Discharge Current (Continuous)	up to 5 C Current
Maximum Discharge Current (Peak < 10sec)	10 C Current

Operating Temperature :	-30 °C / + 55 °C
Recommended Charge Temperature	0 °C / + 40 °C
Storage Temperature	-30 °C / + 55 °C

Cycle Life at 25 °C : (1 C Charge / 1 C Discharge, DOD100%)	1000 Cycles to 80% Nominal Capacity
---	-------------------------------------

ePLB C020 Performance

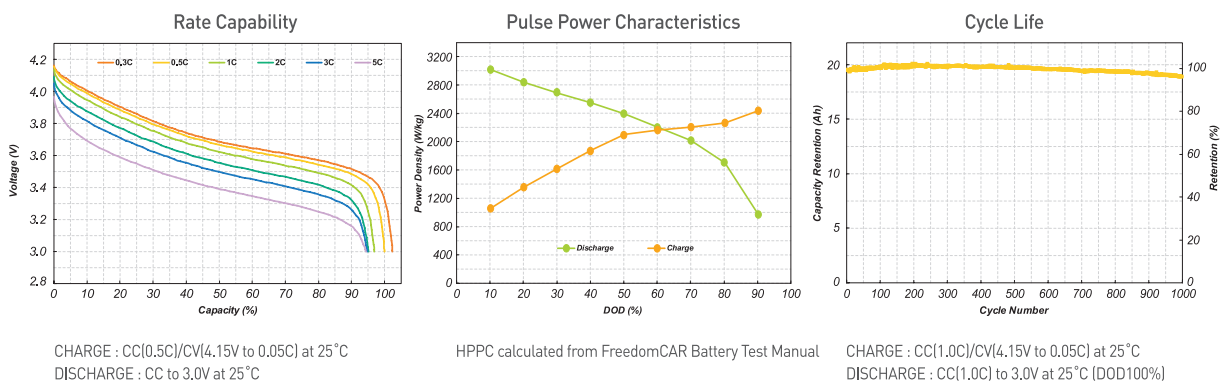


Figure A.4: Data Sheet of a EiG ePLB C020 Li-ion battery

B Computed Parameters

Several EEC models have been parametrized in Chapter 3 on the basis of measurements from an *ePLB C020* Li-ion battery (cf. Section 1.4.3). The particular parameters of these parametrizations are listed below.

Parameters of OCV Model Parametrization

This section contains the parameters of the open circuit voltage (OCV) models parametrized in Section 3.1.1. The comparison of the accuracy of the model in Table 3.1 is based on the values of Table B.1 to B.6. The model name and its equation is given in the captions of the tables.

Parameter	Value
E_0	3.664
c_1	0.09043
c_2	-0.1481

Table B.1: Nernst: $v_{oc} = E_0 + c_1 \ln(SoC) + c_2 \ln(1 - SoC)$

Parameter	Value
c_0	3.854
c_1	0.1578
c_2	-0.2145
c_3	1.39e-7
c_4	-0.3755

Table B.2: Zheng: $v_{oc} = c_0 + c_1 \ln(SoC) + c_2 \ln(1 - SoC) + c_3 \frac{1}{SoC} + c_4 SoC$

RC Models

Various mathematical models to model the *SoC* dependency of the passive components have been analyzed. The results of the parameter fitting are shown in Table B.7 to B.12. These parametrizations are the basis of the R² in Table 3.2.

Parameter	Value
E_0	3.471
A	0.706
B	2.254
$K \cdot Q$	0.006815

Table B.3: OCV part of the Modified Shepherd Model: $v_{oc} = E_0 - K \cdot Q \frac{1}{SoC} + A e^{-B(1-SoC)}$

Parameter	Value
E_0	3.49
A	0.6918
B	2.395
$K \cdot Q$	0.008779
SoC_0	0.01009

Table B.4: OCV part of the extended Modified Shepherd Model: $v_{oc} = E_0 - K \cdot Q \frac{1}{SoC + SoC_0} + A e^{-B(1-SoC)}$

Parameter	Value
R_i	1.7401 m Ω
R_1	1.7159 m Ω
C_1	59.455 kF

Table B.7: EEC model with one RC-element and constant component values

Parameter	Value
a_1	-0.685
a_2	90.03
a_3	2.718
a_4	1.439
a_5	1.677
a_6	0.914

Table B.5: Lam: $v_{oc} = a_1 e^{-a_2 SoC} + a_3 + a_4 SoC + a_5 e^{-\frac{a_6}{1-SoC}}$

Parameter	Value
c_0	-0.660
c_1	92.74
c_2	3.356
c_3	1.347
c_4	-2.16
c_5	1.647

Table B.6: Chen and Rincón-Mora: $v_{oc} = c_0 e^{-c_1 SoC} + c_2 + c_3 SoC + c_4 SoC^2 + c_5 SoC^3$

Parameter	Value		
	R_i	R_1	C_1
c_0	2.175 m Ω	2.065 m Ω	48.06 kF
c_1	-0.7891 m Ω	-0.7592 m Ω	20.66 kF

Table B.8: EEC model with one RC-element and component values modeled as straight line: $c_0 + c_1 \cdot SoC$

Parameter	Value		
	R_i	R_1	C_1
c_0	2.635 m Ω	3.020 m Ω	24.30 kF
c_1	-1.360 m Ω	-1.737 m Ω	49.88 kF
c_2	-10.89 m Ω	-17.48 m Ω	297.4 kF
c_3	26.41 m Ω	41.40 m Ω	-503.8 kF
c_4	-15.5 m Ω	-24.23 m Ω	173.4 kF

Table B.9: EEC model with one RC-element and component values modeled as polynomial 4th order: $c_0 + c_1 \cdot SoC + c_2 \cdot SoC^2 + c_3 \cdot SoC^3 + c_4 \cdot SoC^4$

Parameter	Value		
	R_i	R_1	C_1
a	0.688 m Ω	8.106 $\mu\Omega$	-184.8 F
b	-0.3856	-2.702	-2.847
c	0.7919 m Ω	1.498 m Ω	65.52 kF

Table B.10: EEC model with one RC-element and component values modeled with $a \cdot SoC^b + c$

Parameter	Value		
	R_i	R_1	C_1
a	5.242 m Ω	1.358 m Ω	-35.45 kF
b	-11.39	-4.583	-322.3
c	1.574 m Ω	1.447 m Ω	61.45 kF

Table B.11: EEC model with one RC-element and component values modeled with $a \cdot e^{b \cdot SoC} + c$

Parameter	Value		
	R_i	R_1	C_1
a	8.468 m Ω	1.606 m Ω	$1.028 \cdot 10^8$ F
b	-15.29	-4.764	2.632
c	1.717 m Ω	1.447 m Ω	$-1.028 \cdot 10^8$ F
d	-0.1053	-0.01215	2.632

Table B.12: EEC model with one RC-element and component values modeled with $a \cdot e^{b \cdot SoC} + c \cdot e^{d \cdot SoC}$

C Simulation results

Chapter 4 discusses the validation of the battery models on the basis of measurements of a standardized FTP-72 driving cycle (federal test procedure). Overall, six models have been tested, four different equivalent electrical circuit (EEC) models with one RC-element, and two extended Modified Shepherd Models (eMSM), one parametrized with measurements and one with data sheet values (cf. Table 4.2 on page 51). Four different ranges have been analyzed

- a repeatedly applied FTP-72 driving cycle to a full battery until the battery was completely discharged
- a single FTP-72 driving cycle at an initial state of charge (*SoC*) of 98.89%,
- a single FTP-72 driving cycle at an initial *SoC* of 51.12%, and
- a single FTP-72 driving cycle at an initial *SoC* of 9.06%.

Out of these 24 simulations several weighted errors have been calculated (see Table 4.3 on page 52). The according simulated battery voltages are shown subsequently in Figure C.1 to C.24.

Repeatedly Applied FTP-72 Cycle to a Full Battery

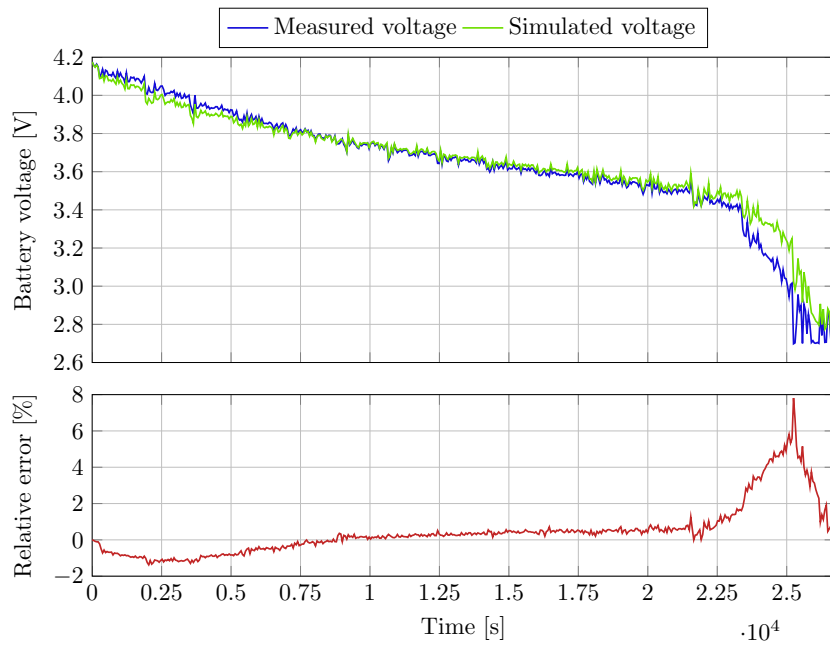


Figure C.1: EEC model with one RC-element, OCV and components are modeled with look-up tables (model #1 in Table 4.2)

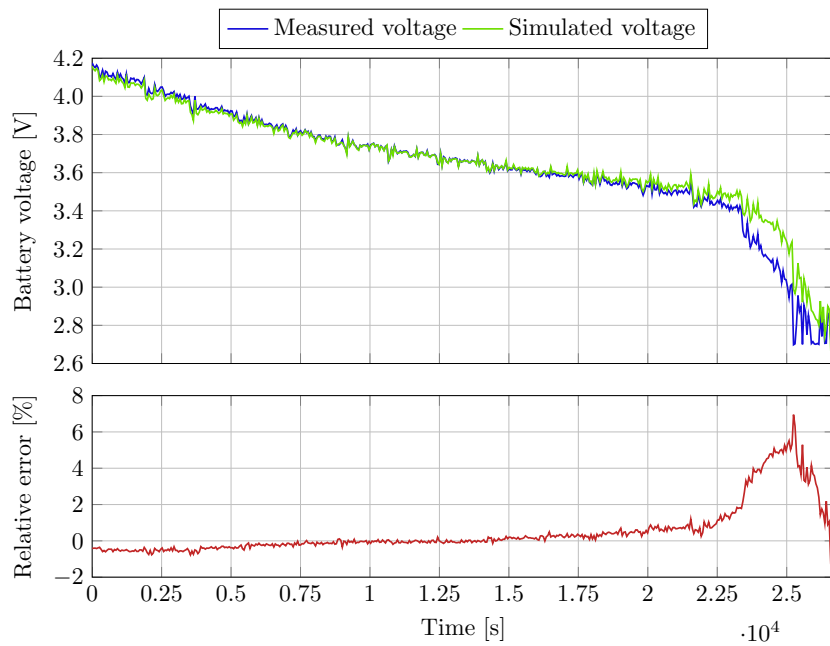


Figure C.2: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* with a polynomial of the 4th order (model #2 in Table 4.2)

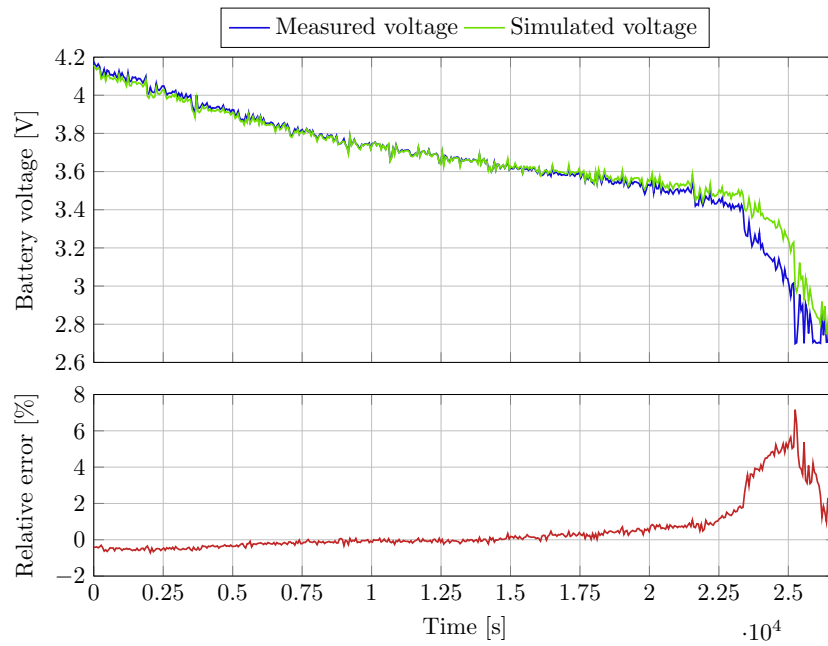


Figure C.3: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* as straight line (model #3 in Table 4.2)

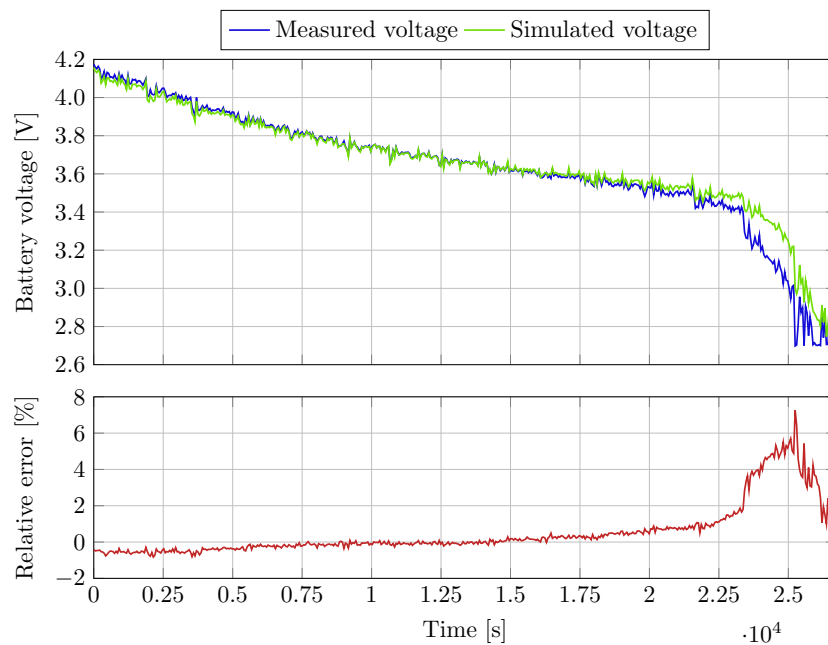


Figure C.4: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are constant (model #4 in Table 4.2)

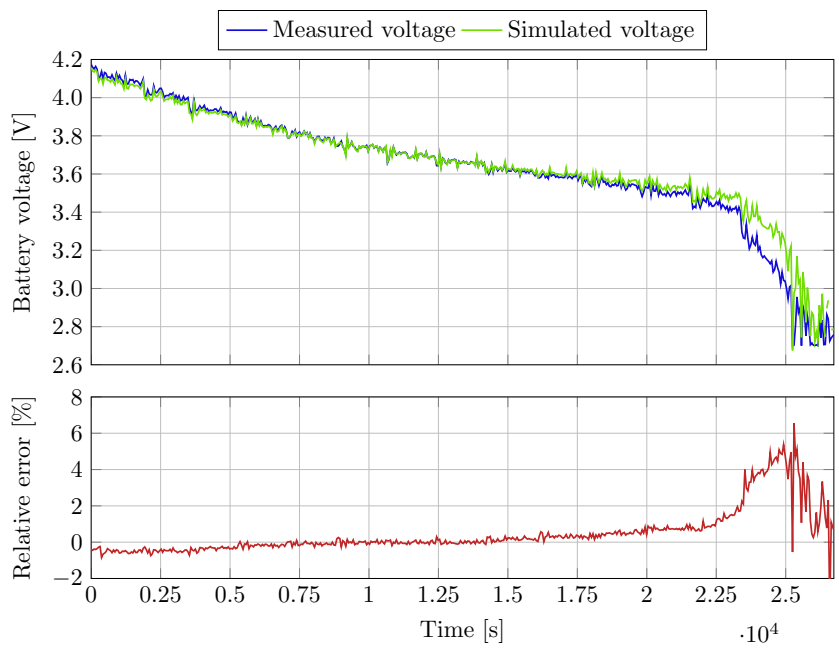


Figure C.5: eMSM parametrized with measurement data (model #5 in Table 4.2)

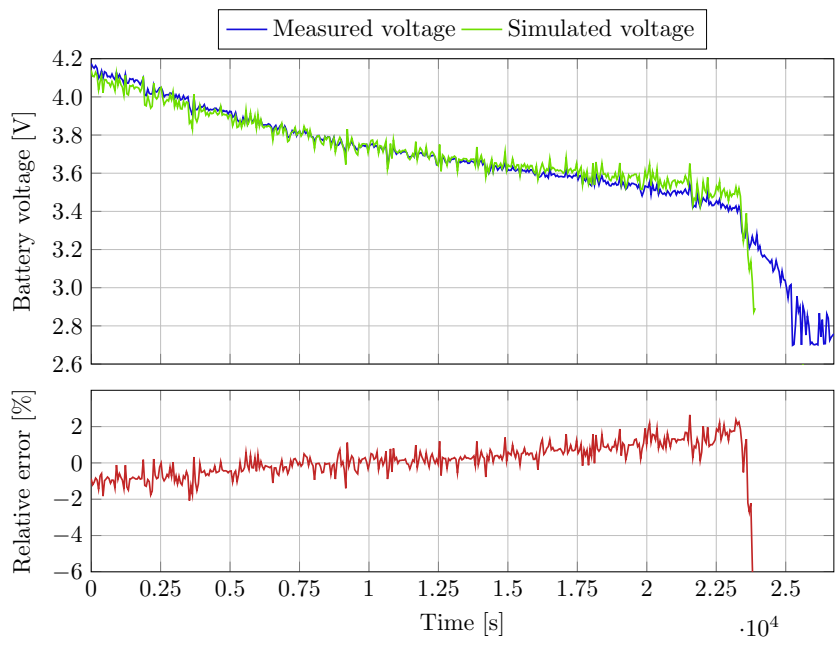


Figure C.6: eMSM parametrized with data sheet curves (model #6 in Table 4.2)

One FTP-72 Cycle with an Initial SoC of 98.89%

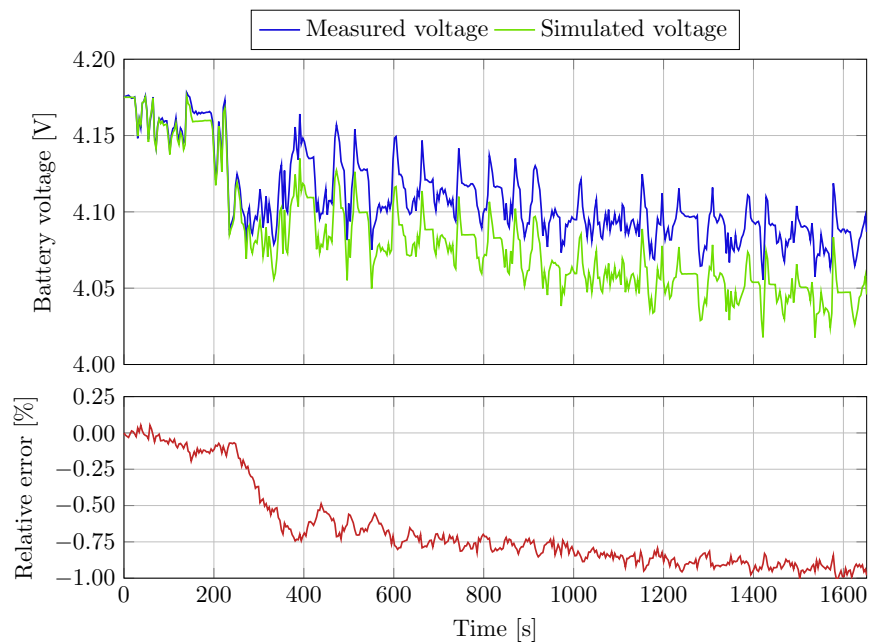


Figure C.7: EEC model with one RC-element, OCV and components are modeled with look-up tables (model #1 in Table 4.2)

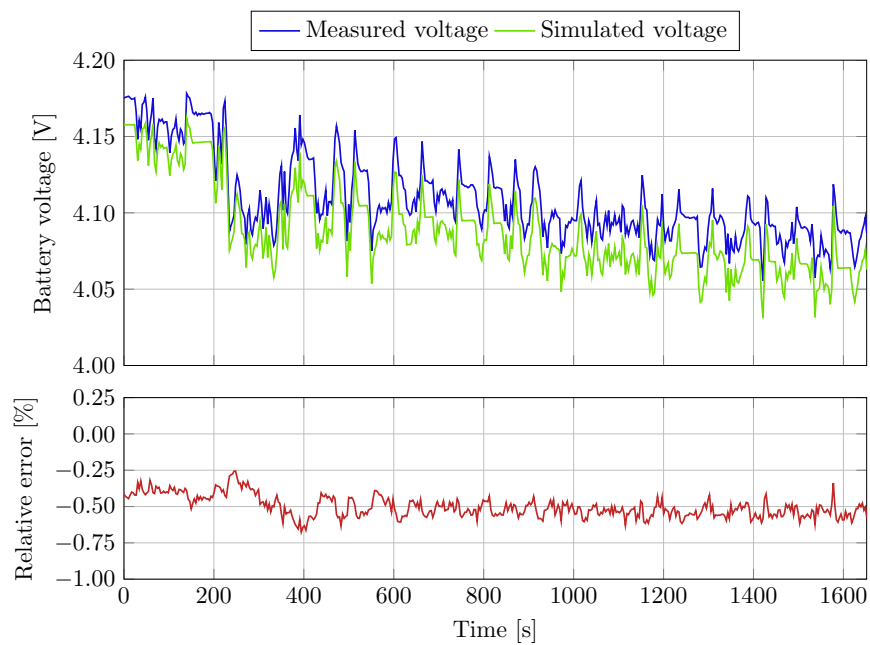


Figure C.8: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* with a polynomial of the 4th order (model #2 in Table 4.2)

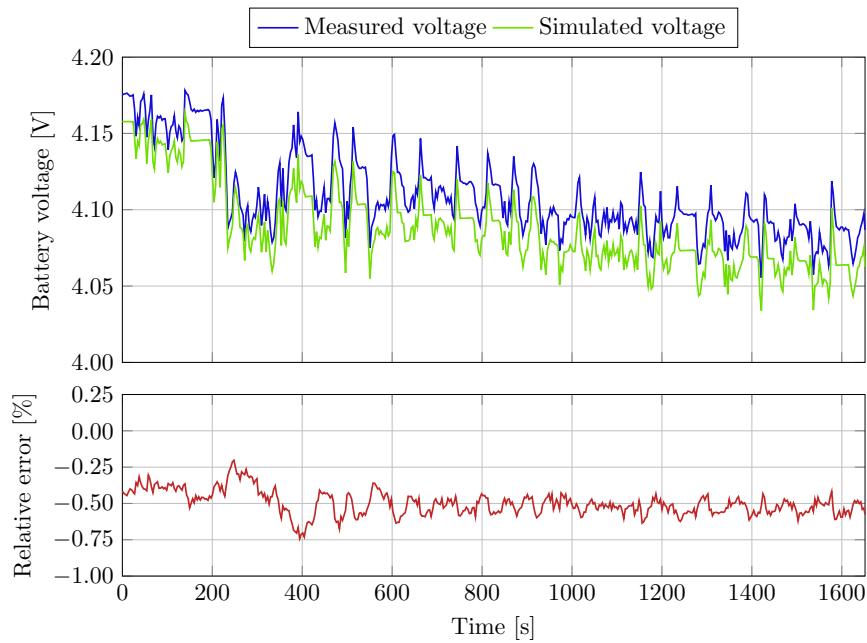


Figure C.9: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* as straight line (model #3 in Table 4.2)

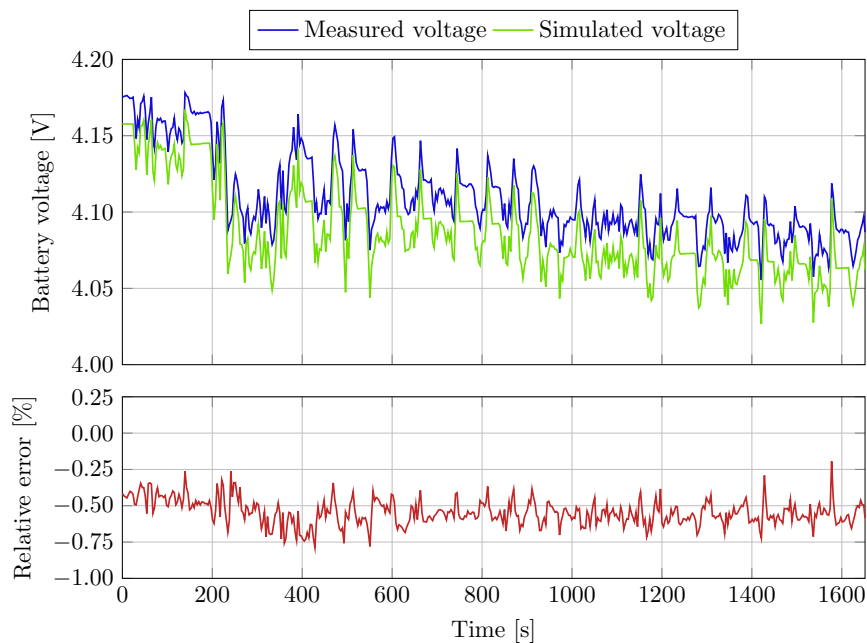


Figure C.10: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are constant (model #4 in Table 4.2)

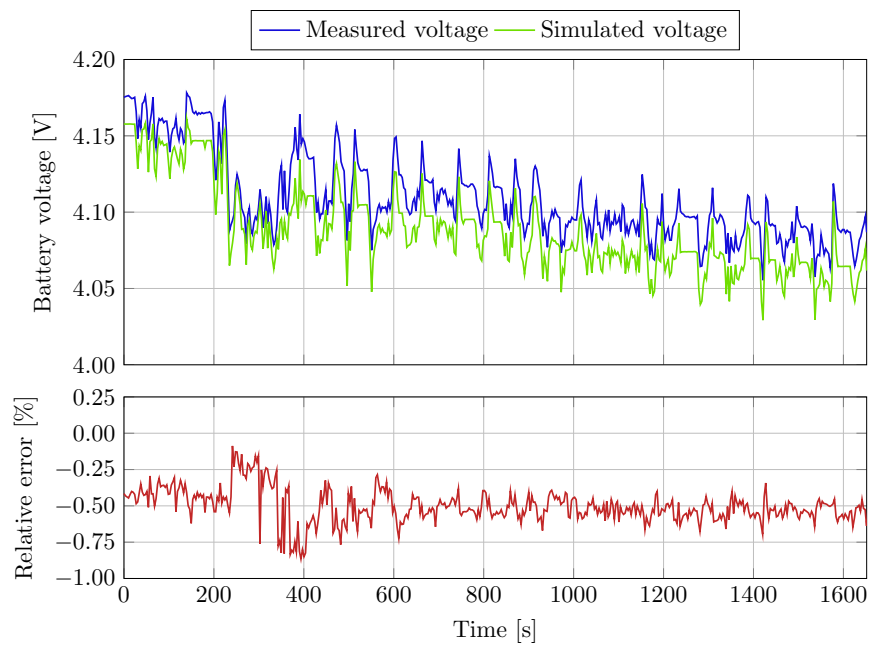


Figure C.11: eMSM parametrized with measurement data (model #5 in Table 4.2)

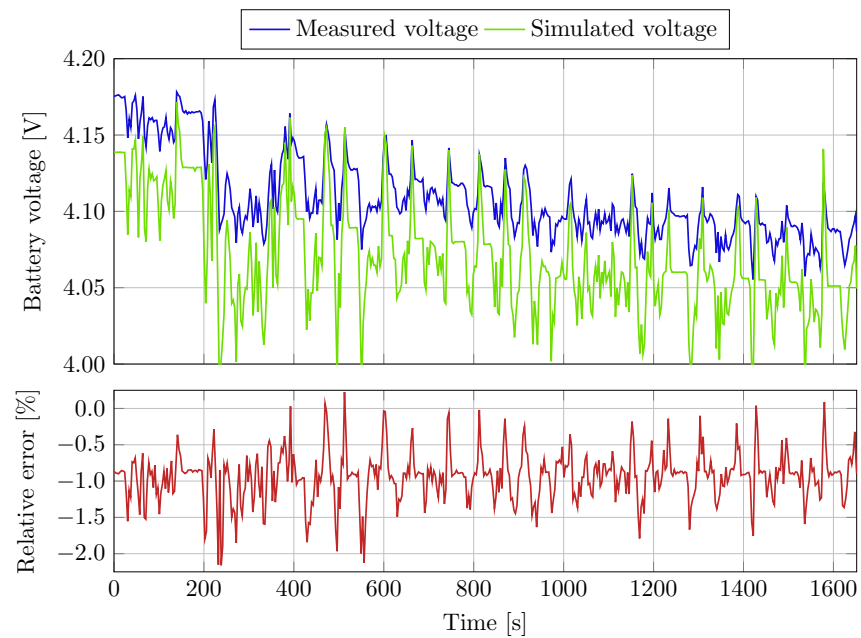


Figure C.12: eMSM parametrized with data sheet curves (model #6 in Table 4.2)

One FTP-72 Cycle with an Initial SoC of 51.12%

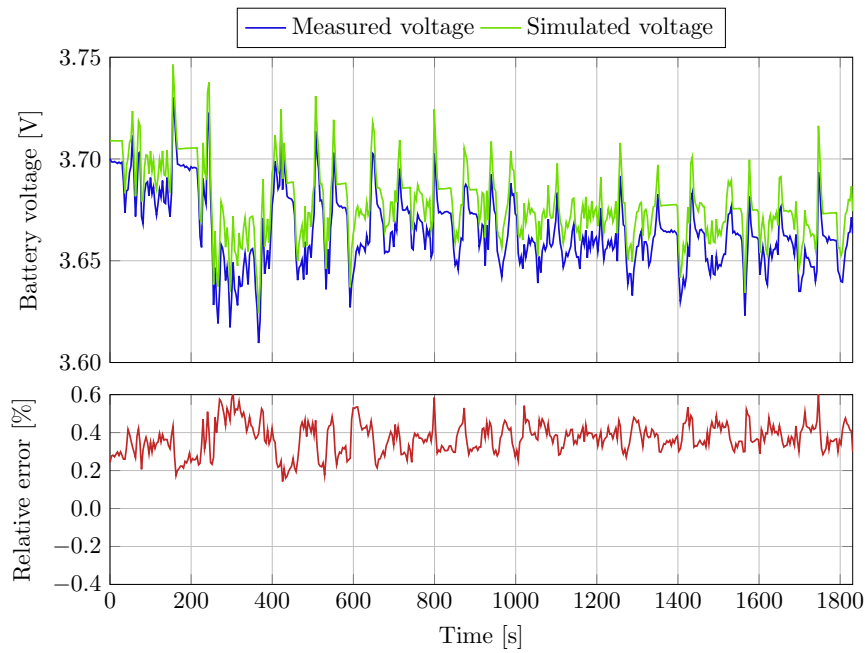


Figure C.13: EEC model with one RC-element, OCV and components are modeled with look-up tables (model #1 in Table 4.2)

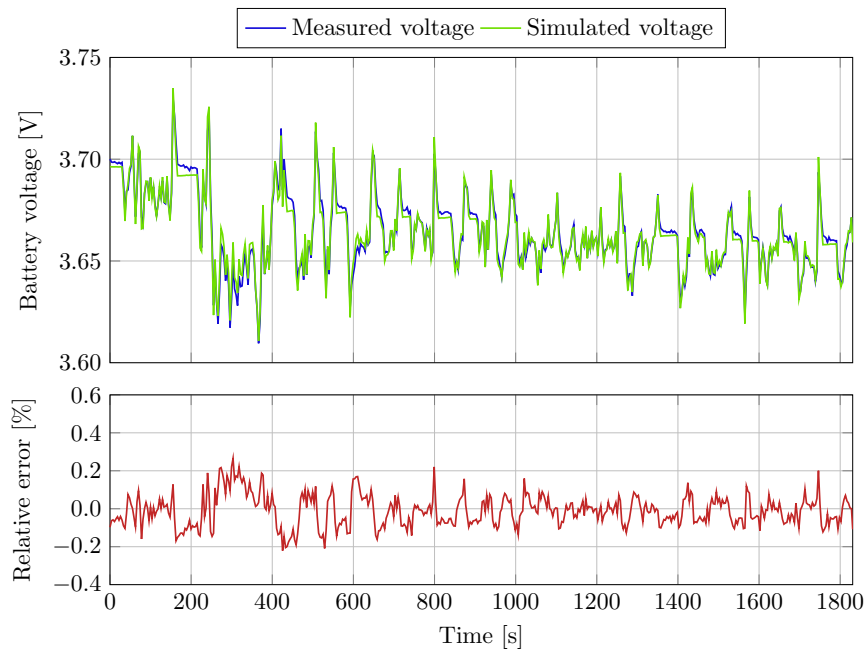


Figure C.14: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* with a polynomial of the 4th order (model #2 in Table 4.2)

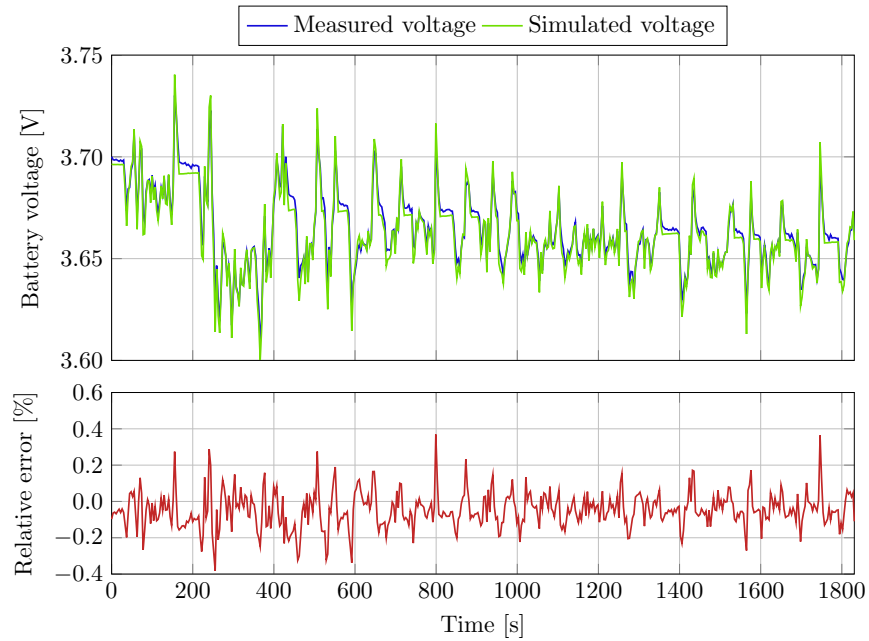


Figure C.15: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* as straight line (model #3 in Table 4.2)

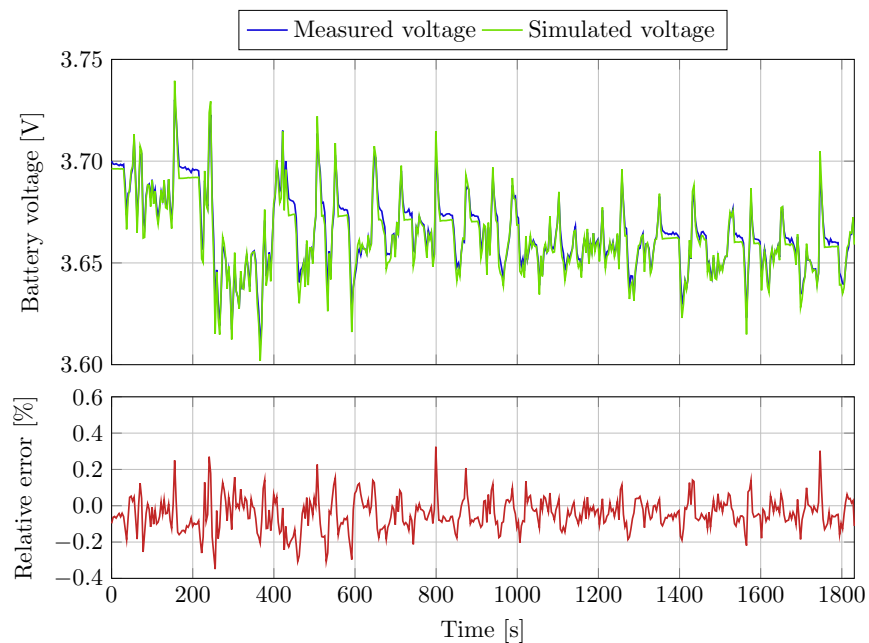


Figure C.16: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are constant (model #4 in Table 4.2)

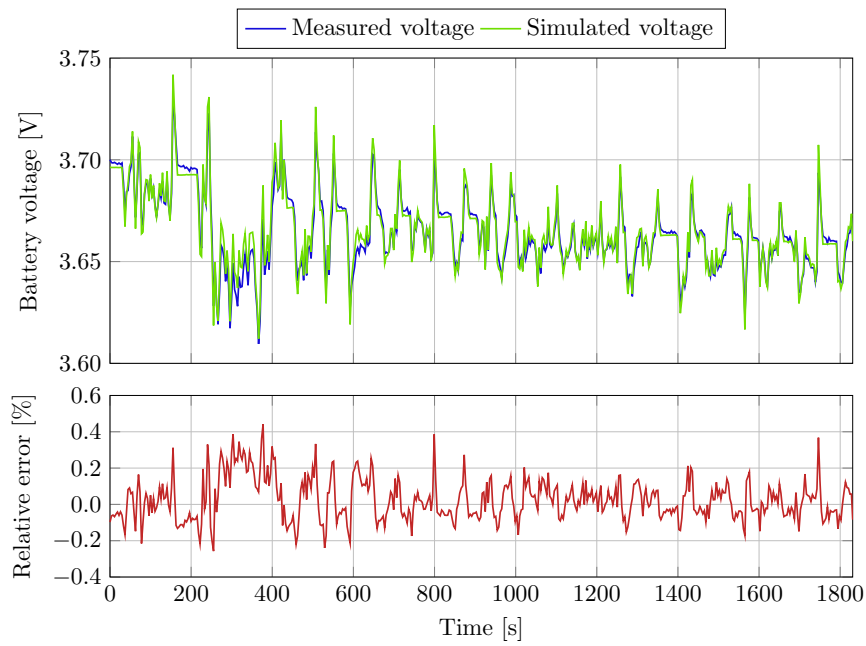


Figure C.17: eMSM parametrized with measurement data (model #5 in Table 4.2)

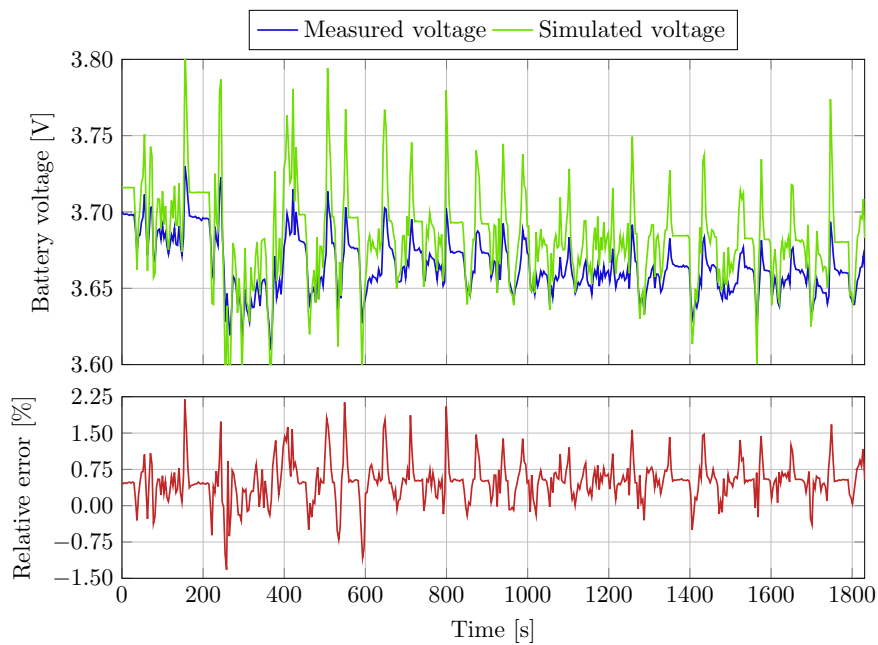


Figure C.18: eMSM parametrized with data sheet curves (model #6 in Table 4.2)

One FTP-72 Cycle with an Initial SoC of 9.06%

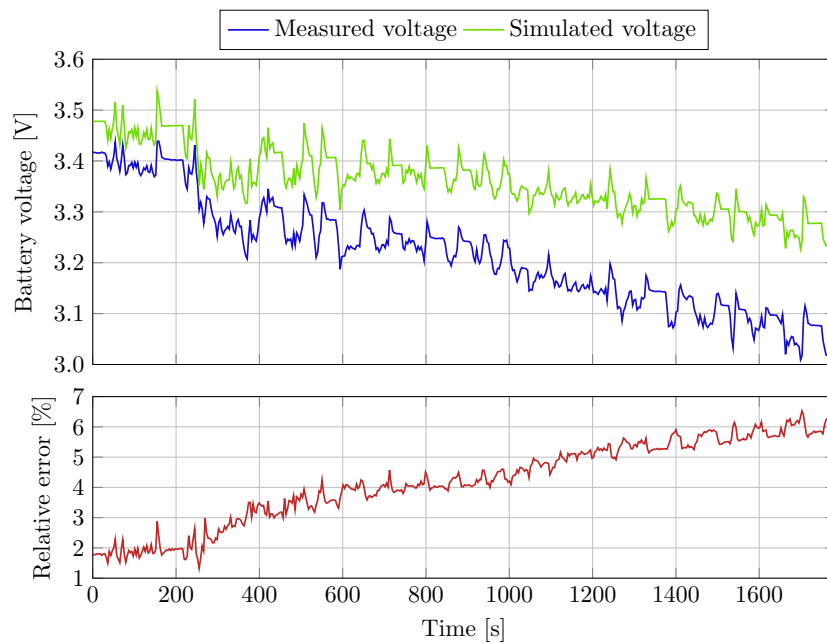


Figure C.19: EEC model with one RC-element, OCV and components are modeled with look-up tables (model #1 in Table 4.2)

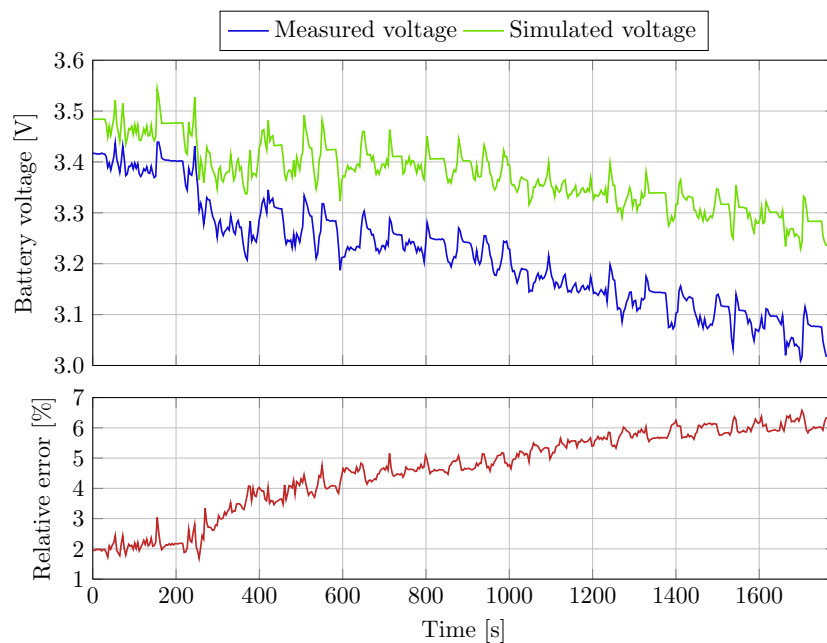


Figure C.20: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* with a polynomial of the 4th order (model #2 in Table 4.2)

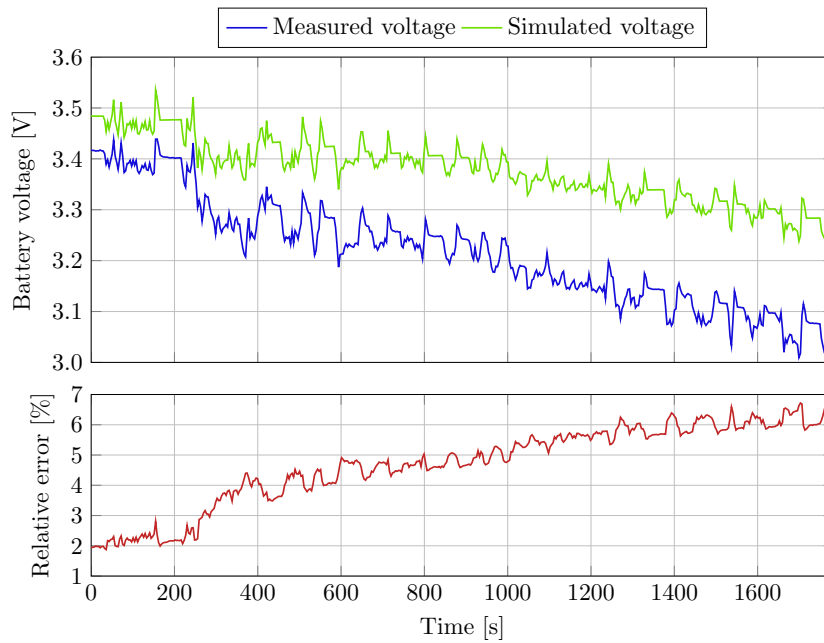


Figure C.21: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are modeled dependent of the *SoC* as straight line (model #3 in Table 4.2)

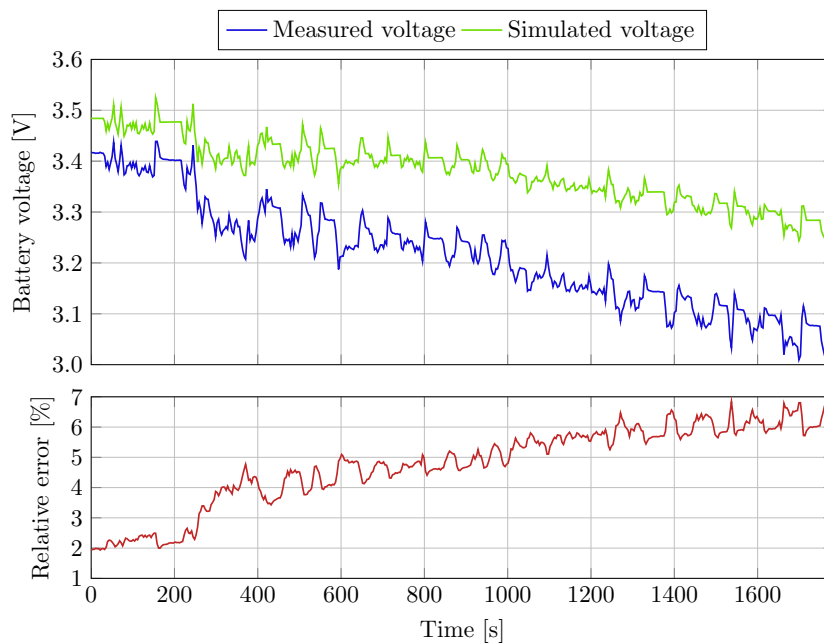


Figure C.22: EEC model with one RC-element, OCV is modeled with the OCV part of the eMSM, component values are constant (model #4 in Table 4.2)

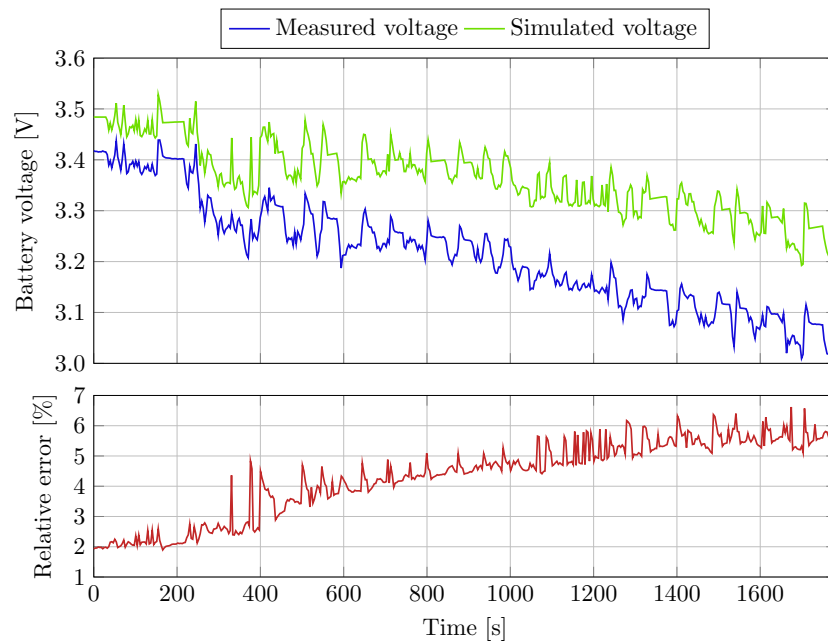


Figure C.23: eMSM parametrized with measurement data (model #5 in Table 4.2)

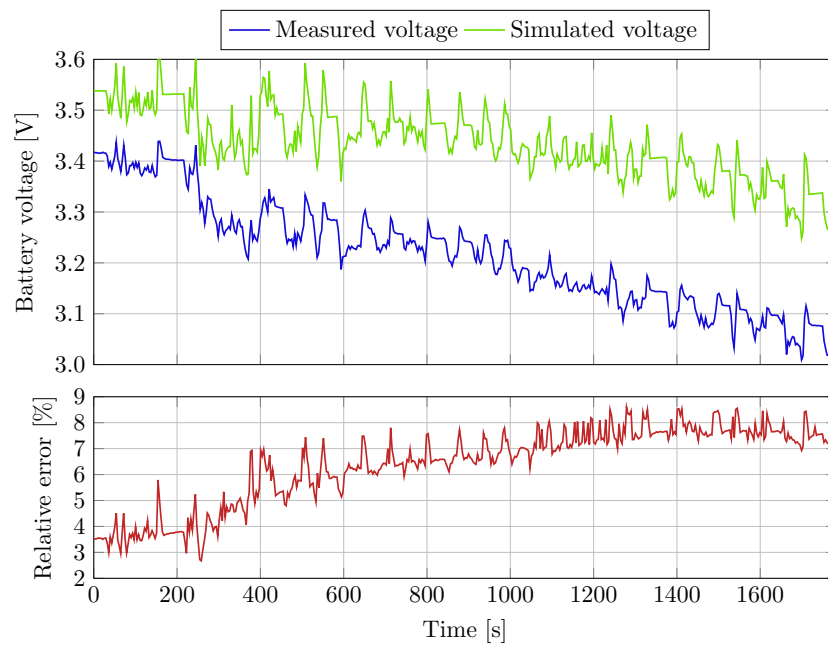


Figure C.24: eMSM parametrized with data sheet curves (model #6 in Table 4.2)

D Measurement Data of Offset Correction

Measurements showed that there are offset errors between set values at the real time system (RTS) and the actual value of the power source (cf. Fig. 5.4 for the measurement setup). Furthermore there is an error between the actual value and the value shown at the RTS. This goes both for the voltage as well as the current. In Section 5.2.2 these errors have been reduced by using a linear correction function. The underlying measurements are shown in Table D.1 to D.4.

Set value [V]	Measured value [V]	Absolute error [V]
10	11.2	1.2
25	26.3	1.3
50	51.5	1.5
100	101.3	1.3
150	151.0	1.0
200	201.0	1.0
250	251.1	1.1
300	301.0	1.0
350	350.8	0.8
400	400.7	0.7
450	450.9	0.9
500	500.7	0.7
550	550.8	0.8
600	600.4	0.4

Table D.1: Offset between voltage RTS set values and actual voltages

Actual value [V]	Displayed value [V]	Absolute error [V]
10	11.0	1.0
25	26.1	1.1
50	51.4	1.4
100	101.6	1.6
150	151.5	1.5
200	201.7	1.7
250	251.9	1.9
300	302.0	2.0
350	351.9	1.9
400	402.0	2.0
450	452.4	2.4
500	502.4	2.4
550	552.5	2.5
600	602.4	2.4

Table D.2: Error between actual voltage value (measured by a multimeter) and displayed value at the RTS

Set value [A]	Measured value [A]	Absolute error [A]
1.00	1.096	0.10
2.00	2.090	0.09
3.00	3.090	0.09
4.00	4.100	0.10
5.00	5.100	0.10
6.00	6.100	0.10
7.00	7.100	0.10
8.00	8.100	0.10
9.00	9.100	0.10
9.50	9.600	0.10

Table D.3: Absolute error between RTS current set values and actual current

Set value [A]	Measured value [A]	Absolute error [A]
1.01	1.045	0.05
1.98	2.050	0.05
2.99	3.038	0.04
4.00	4.040	0.04
5.00	5.030	0.03
6.00	6.030	0.03
6.98	7.000	0.00
7.98	8.020	0.02
8.99	9.010	0.01
9.49	9.510	0.01

Table D.4: Absolute error between actual current value (measured by a multimeter) and displayed value at the RTS

Literature

- [CBT11] CAPIZZI, G. ; BONANNO, F. ; TINA, G. M.: Recurrent Neural Network-Based Modeling and Simulation of Lead-Acid Batteries Charge-Discharge. In: *Energy Conversion, IEEE Transactions on* 26 (2011), Nr. 2, S. 435–443. <http://dx.doi.org/10.1109/TEC.2010.2095015>. – DOI 10.1109/TEC.2010.2095015. – ISSN 0885–8969
- [CKC⁺10] CHATURVEDI, N.A. ; KLEIN, R. ; CHRISTENSEN, J. ; AHMED, J. ; KOJIC, A.: Modeling, estimation, and control challenges for lithium-ion batteries. In: *American Control Conference (ACC), 2010*, 2010, S. 1997–2002
- [CRM06] CHEN, Min ; RINCON-MORA, G.A.: Accurate electrical battery model capable of predicting runtime and I-V performance. In: *IEEE Transactions on Energy Conversion* 21 (2006), Juni, Nr. 2, S. 504–511. <http://dx.doi.org/10.1109/TEC.2006.874229>. – DOI 10.1109/TEC.2006.874229. – ISSN 0885–8969
- [DDFS08] DI DOMENICO, D. ; FIENGO, G. ; STEFANOPOULOU, A.: Lithium-ion battery state of charge estimation with a Kalman Filter based on a electrochemical model. In: *IEEE International Conference on Control Applications, 2008. CCA 2008*, 2008, S. 702–707
- [Dep13] DEPARTMENT OF ENERGY AND CLIMATE CHANGE: *Electricity Generation Costs*. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/269888/131217_Electricity_Generation_costs_report_December_2013_Final.pdf. Version: Dezember 2013
- [DS06] DOERFFEL, Dennis ; SHARKH, Suleiman A.: A critical review of using the Peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries. In: *Journal of Power Sources* 155 (2006), April, Nr. 2, 395–400. <http://dx.doi.org/10.1016/j.jpowsour.2005.04.030>. – DOI 10.1016/j.jpowsour.2005.04.030. – ISSN 0378–7753
- [DVM12] DAO, Thanh-Son ; VYASARAYANI, Chandrika P. ; MCPHEE, John: Simplification and order reduction of lithium-ion battery model based on porous-electrode theory. In: *Journal of Power Sources* 198 (2012), Januar, 329–337. <http://dx.doi.org/10.1016/j.jpowsour.2011.09.034>. – DOI 10.1016/j.jpowsour.2011.09.034. – ISSN 0378–7753
- [ECKF13] EINHORN, M. ; CONTE, F.V. ; KRAL, C. ; FLEIG, J.: Comparison, Selection, and Parameterization of Electrical Battery Models for Automotive Applications. In: *IEEE Transactions on Power Electronics* 28 (2013), März, Nr. 3, S.

- 1429–1437. <http://dx.doi.org/10.1109/TPEL.2012.2210564>. – DOI 10.1109/TPEL.2012.2210564. – ISSN 0885–8993
- [Eur10] EUROPEAN COMMISSION: *The Role of Battery Electric Vehicles, Plug-in Hybrids and Fuel Cell Electric Vehicles*. http://ec.europa.eu/research/fch/pdf/a_portfolio_of_power_trains_for_europe_a_fact_based_analysis.pdf. Version: November 2010
- [Eur13] EUROPEAN CLIMATE FOUNDATION: *Climate Change: Action, Trends and Implications for Business - The IPCC's Fifth Assessment Report, Working Group 1*. <http://www.europeanclimate.org/documents/IPCCWebGuide.pdf>. Version: September 2013
- [EVH⁺10] EDRINGTON, C. S. ; VODYAKHO, O. ; HACKER, B. ; AZONGHA, S. ; KHALIGH, A. ; ONAR, O.: Virtual battery charging station utilizing power-hardware-in-the-loop: Application to V2G impact analysis, 2010, S. 1–6
- [Fra13] FRAUNHOFER-INSTITUT FUER SOLARE ENERGIESYSTEME ISE: *Stromgestehungskosten Erneuerbare Energien*. <http://www.ise.fraunhofer.de/de/veroeffentlichungen/veroeffentlichungen-pdf-dateien/studien-und-konzeptpapiere/studie-stromgestehungskosten-erneuerbare-energien.pdf>. Version: November 2013
- [GLD02] GAO, Lijun ; LIU, Shengyi ; DOUGAL, R.A.: Dynamic lithium-ion battery model for system simulation. In: *IEEE Transactions on Components and Packaging Technologies* 25 (2002), September, Nr. 3, S. 495 – 505. <http://dx.doi.org/10.1109/TCAPT.2002.803653>. – DOI 10.1109/TCAPT.2002.803653. – ISSN 1521–3331
- [GWDW02] GOMADAM, Parthasarathy M. ; WEIDNER, John W. ; DOUGAL, Roger A. ; WHITE, Ralph E.: Mathematical modeling of lithium-ion and nickel battery systems. In: *Journal of Power Sources* 110 (2002), August, Nr. 2, 267–284. [http://dx.doi.org/10.1016/S0378-7753\(02\)00190-8](http://dx.doi.org/10.1016/S0378-7753(02)00190-8). – DOI 10.1016/S0378-7753(02)00190-8. – ISSN 0378–7753
- [HB11] HUSSEIN, Ala Al-Haj ; BATARSEH, Issa: An Overview of Generic Battery Models. Detroit, Michigan, Juli 2011
- [HLK10] HE, Yongsheng ; LIU, Wei ; KOCH, Brain J.: Battery algorithm verification and development using hardware-in-the-loop testing. In: *Journal of Power Sources* 195 (2010), Mai, Nr. 9, 2969–2974. <http://dx.doi.org/10.1016/j.jpowsour.2009.11.036>. – DOI 10.1016/j.jpowsour.2009.11.036. – ISSN 0378–7753
- [HLP12] HU, Xiaosong ; LI, Shengbo ; PENG, Huei: A comparative study of equivalent circuit models for Li-ion batteries. In: *Journal of Power Sources* 198 (2012), Januar, 359–367. <http://dx.doi.org/10.1016/j.jpowsour.2011.10.013>. – DOI 10.1016/j.jpowsour.2011.10.013. – ISSN 0378–7753
- [IEA13] IEA INTERNATIONAL ENERGY AGENCY: *Global EV Outlook - Understanding the Electric Vehicle Landscape to 2020*. http://www.iea.org/publications/freepublications/publication/GlobalEVOutlook_2013.pdf. Version: April 2013
- [IEA14] IEA INTERNATIONAL ENERGY AGENCY: *Technology Roadmap Energystorage*. 2014
- [IEC11] IEC INTERNATIONAL ELECTROTECHNICAL COMMISSION: *Electrical Energy Storage Whitepaper*. 2011

- [JGV⁺11] JONG, Erik d. ; GRAAFF, Roald d. ; VAESSEN, Peter ; CROLLA, Paul ; ROSCOE, Andrew ; LEHFUSS, Felix ; LAUSS, Georg ; KOTSAMPOPOULOS, Panos ; GAFARO, Francisco: *European White Book on Real-Time Powerhardware-in-the-Loop testing*. 2011. – ISBN 978-3-943517-01-9
- [JH08] JONGERDEN, M. R. ; HAVERKORT, B. R. H. M.: Battery Modeling / Centre for Telematics and Information Technology University of Twente. Version: Januar 2008. <http://eprints.eemcs.utwente.nl/11645/>. Enschede, Januar 2008 (TR-CTIT-08-01). – Technical Report
- [JH09] JONGERDEN, M.R. ; HAVERKORT, B.R.: Which battery model to use? In: *IET Software* 3 (2009), Dezember, Nr. 6, S. 445–457. <http://dx.doi.org/10.1049/iet-sen.2009.0001>. – DOI 10.1049/iet-sen.2009.0001. – ISSN 1751-8806
- [KHHM12] KALLFAB, C. ; HOCH, C. ; HILGER, A ; MANKE, Ingo: Short-circuit and overcharge behaviour of some lithium ion batteries. In: *2012 9th International Multi-Conference on Systems, Signals and Devices (SSD)*, 2012, S. 1–5
- [KK08] KROEZE, R.C. ; KREIN, P.T.: Electrical battery model for use in dynamic electric vehicle simulations. In: *IEEE Power Electronics Specialists Conference, 2008. PESC 2008*, 2008, S. 1336–1342
- [KQ11] KIM, Taesic ; QIAO, Wei: A Hybrid Battery Model Capable of Capturing Dynamic Circuit Characteristics and Nonlinear Capacity Effects. In: *IEEE Transactions on Energy Conversion* 26 (2011), Dezember, Nr. 4, S. 1172–1180. <http://dx.doi.org/10.1109/TEC.2011.2167014>. – DOI 10.1109/TEC.2011.2167014. – ISSN 0885-8969
- [LBK11] LAM, Long ; BAUER, P. ; KELDER, E.: A practical circuit-based model for Li-ion battery cells in electric vehicle applications. In: *Telecommunications Energy Conference (INTELEC), 2011 IEEE 33rd International*, 2011, S. 1–9
- [LCP12] LEE, James L. ; CHEMISTRUCK, Andrew ; PLETT, Gregory L.: One-dimensional physics-based reduced-order model of lithium-ion dynamics. In: *Journal of Power Sources* 220 (2012), Dezember, 430–448. <http://dx.doi.org/10.1016/j.jpowsour.2012.07.075>. – DOI 10.1016/j.jpowsour.2012.07.075. – ISSN 0378-7753
- [LK11] LI, Shuhui ; KE, Bao: Study of battery modeling using mathematical and circuit oriented approaches. In: *2011 IEEE Power and Energy Society General Meeting*, 2011, S. 1–8
- [LWY09] LI, Gechen ; WANG, Haiying ; YU, Zhilong: New Method for Estimation Modeling of SOC of Battery. In: *WRI World Congress on Software Engineering, 2009. WCSE '09* Bd. 2, 2009, S. 387–390
- [Mar11] MARTINEZ, J.A: Modeling and characterization of energy storage devices. In: *2011 IEEE Power and Energy Society General Meeting*, 2011, S. 1–6
- [MM93] MANWELL, James F. ; MCGOWAN, Jon G.: Lead acid battery storage model for hybrid energy systems. In: *Solar Energy* 50 (1993), Mai, Nr. 5, 399–405. [http://dx.doi.org/10.1016/0038-092X\(93\)90060-2](http://dx.doi.org/10.1016/0038-092X(93)90060-2). – DOI 10.1016/0038-092X(93)90060-2. – ISSN 0038-092X
- [OD09] O. TREMBLAY ; DESSIAINT: Experimental Validation of a Battery Dynamic Model for EV Applications. In: *World Electric Vehicle Journal* (2009)
- [Peb13] PEBRIYANTI, Grace: A lithium-ion battery modeling for a HIL-battery simulator. In: *2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 2013, S. 185–190

- [Pol08] POLICY DEPARTMENT ECONOMIC AND SCIENTIFIC POLICY: *Outlook of Energy Storage Technologies*. Februar 2008
- [RBAI11] ROMERO-BECERRIL, Aldo ; ALVAREZ-ICAZA, Luis: Comparison of discretization methods applied to the single-particle model of lithium-ion batteries. In: *Journal of Power Sources* 196 (2011), Dezember, Nr. 23, 10267–10279. <http://dx.doi.org/10.1016/j.jpowsour.2011.06.091>. – DOI 10.1016/j.jpowsour.2011.06.091. – ISSN 0378–7753
- [RSB08] REN, Wei ; STEURER, Michael ; BALDWIN, Thomas L.: Improve the Stability and the Accuracy of Power Hardware-in-the-Loop Simulation by Selecting Appropriate Interface Algorithms. In: *IEEE Transactions on Industry Applications* 44 (2008), Nr. 4, 1286–1294. <http://dx.doi.org/10.1109/TIA.2008.926240>. – DOI 10.1109/TIA.2008.926240. – ISSN 0093–9994
- [RV01] RAKHMATOV, D.N. ; VRUDHULA, S.B.K.: An analytical high-level battery model for use in energy management of portable electronic systems. In: *IEEE/ACM International Conference on Computer Aided Design, 2001. ICCAD 2001*, 2001, S. 488–493
- [RVR03] RAO, R. ; VRUDHULA, S. ; RAKHMATOV, D.N.: Battery modeling for energy aware system design. In: *Computer* 36 (2003), Dezember, Nr. 12, S. 77 – 87. <http://dx.doi.org/10.1109/MC.2003.1250886>. – DOI 10.1109/MC.2003.1250886. – ISSN 0018–9162
- [She65a] SHEPHERD, C. M.: Design of Primary and Secondary Cells II . An Equation Describing Battery Discharge. In: *Journal of The Electrochemical Society* 112 (1965), Januar, Nr. 7, 657–664. <http://dx.doi.org/10.1149/1.2423659>. – DOI 10.1149/1.2423659. – ISSN 0013–4651, 1945–7111
- [She65b] SHEPHERD, Clarence M.: Design of Primary and Secondary Cells I . Effect of Polarization and Resistance on Cell Characteristics. In: *Journal of The Electrochemical Society* 112 (1965), Januar, Nr. 3, 252–257. <http://dx.doi.org/10.1149/1.2423518>. – DOI 10.1149/1.2423518. – ISSN 0013–4651, 1945–7111
- [SKJ⁺13] SONG, Hyun-Sik ; KIM, Tae-Hoon ; JEONG, Jin-Beom ; KIM, Byoung-Hoon ; SHIN, Dong-Hyun ; LEE, Baek-Haeng ; HEO, Hoon: Verification of battery system model for environmentally friendly vehicles using a battery hardware-in-the-loop simulation. In: *IET Power Electronics* 6 (2013), Februar, Nr. 2, S. 417–424. <http://dx.doi.org/10.1049/iet-pel.2012.0293>. – DOI 10.1049/iet-pel.2012.0293. – ISSN 1755–4535
- [SKLL13] SEITL, C. ; KATHAN, J. ; LAUSS, G. ; LEHFUSS, F.: Selection and implementation of a generic battery model for PHIL applications. In: *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, S. 5412–5417
- [SKLL14] SEITL, C. ; KATHAN, J. ; LAUSS, G. ; LEHFUSS, F.: Power hardware-in-the-loop implementation and verification of a real time capable battery model. In: *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, 2014, S. 2285–2290
- [SM13] SHIAU, Jaw-Kuen ; MA, Chien-Wei: Li-Ion Battery Charging with a Buck-Boost Power Converter for a Solar Powered Battery Management System. In: *Energies* 6 (2013), März, Nr. 3, 1669–1699. <http://dx.doi.org/10.3390/en6031669>. – DOI 10.3390/en6031669. – ISSN 1996–1073

- [SRW07] SMITH, Kandler A. ; RAHN, Christopher D. ; WANG, Chao-Yang: Control oriented 1D electrochemical model of lithium ion battery. In: *Energy Conversion and Management* 48 (2007), September, Nr. 9, 2565–2578. <http://dx.doi.org/10.1016/j.enconman.2007.03.015>. – DOI 10.1016/j.enconman.2007.03.015. – ISSN 0196–8904
- [SW06] SANTHANAGOPALAN, Shriram ; WHITE, Ralph E.: Online estimation of the state of charge of a lithium ion cell. In: *Journal of Power Sources* 161 (2006), Oktober, Nr. 2, 1346–1355. <http://dx.doi.org/10.1016/j.jpowsour.2006.04.146>. – DOI 10.1016/j.jpowsour.2006.04.146. – ISSN 0378–7753
- [TC08] TINA, G. ; CAPIZZI, G.: Improved lead -acid battery modelling for photovoltaic application by Recurrent Neural Networks. In: *International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2008. SPEEDAM 2008*, 2008, S. 1170 –1174
- [TDD07] TREMBLAY, O. ; DESSAINT, L.-A. ; DEKKICHE, A.-I.: A Generic Battery Model for the Dynamic Simulation of Hybrid Electric Vehicles. In: *IEEE Vehicle Power and Propulsion Conference, 2007. VPPC 2007*, 2007, S. 284 –289
- [TSK12] THIRUGNANAM, K. ; SAINI, H. ; KUMAR, P.: Mathematical modeling of Li-ion battery for charge/discharge rate and capacity fading characteristics using genetic algorithm approach. In: *2012 IEEE Transportation Electrification Conference and Expo (ITEC)*, 2012, S. 1 –6
- [VDE13] VDE - FORUM NETZTECHNIK/NETZBETRIEB: *Anschluss und Betrieb von Speichern am Niederspannungsnetz*. Juni 2013
- [VLF11] VIEHWEIDER, Alexander ; LAUSS, Georg ; FELIX, Lehfuss: Stabilization of Power Hardware-in-the-Loop simulations of electric energy systems. In: *Simulation Modelling Practice and Theory* 19 (2011), August, Nr. 7, 1699–1708. <http://dx.doi.org/10.1016/j.simpat.2011.04.001>. – DOI 10.1016/j.simpat.2011.04.001. – ISSN 1569–190X
- [WC10] WANG, Tao ; CASSANDRAS, C.G.: Optimal discharge and recharge control of battery-powered energy-aware systems. In: *2010 49th IEEE Conference on Decision and Control (CDC)*, 2010, S. 7513–7518
- [YCC⁺12] YOO, Cheol-Hee ; CHOI, Won-Jun ; CHUNG, Il-Yop ; WON, Dong-Jun ; HONG, Sung-Soo ; JANG, Byung-Jun: Hardware-in-the-loop simulation of DC microgrid with Multi-Agent System for emergency demand response. In: *2012 IEEE Power and Energy Society General Meeting*, 2012, S. 1 –6
- [ZCSA10] ZHANG, Jiucui ; CI, Song ; SHARIF, H. ; ALAHMAD, M.: Modeling Discharge Behavior of Multicell Battery. In: *IEEE Transactions on Energy Conversion* 25 (2010), Dezember, Nr. 4, S. 1133 –1141. <http://dx.doi.org/10.1109/TEC.2010.2048904>. – DOI 10.1109/TEC.2010.2048904. – ISSN 0885–8969
- [Zha13] ZHANG, Tan: *The Economic Benefits of Battery Energy Storage System in Electric Distribution System*. <http://www.wpi.edu/Pubs/ETD/Available/etd-042513-131620/unrestricted/tzhang.pdf>. Version: April 2013
- [ZQD09] ZHENG, Minxin ; QI, Bojin ; DU, Xiaowei: Dynamic model for characteristics of Li-ion battery on electric vehicle. In: *4th IEEE Conference on Industrial Electronics and Applications, 2009. ICIEA 2009*, 2009, S. 2867 –2871

Internet References

- [1] AIT Austrian Institute of Technology. V2G-Inverter - Multi-Purpose Inverter als Basis fuer Smart Grids: Analyse der Wirkungskette Verteilnetz bis Batterie mit bidirektionaler Ladung, 2013. URL: http://energyit.ict.tuwien.ac.at/wp-content/uploads/2010/11/8_Strasser.pdf.
- [2] AIT Austrian Institute of Technology. SmartEST Laboratory, October 2014. URL: <http://www.ait.ac.at/departments/energy/research-areas/energy-infrastructure/smart-grids/smartest-laboratory/?L=1>.
- [3] BD Batteries. Battery types and battery efficiency - VRLA vs GEL vs AGM - peukert's law, August 2014. URL: <http://www.bdbatteries.com/peukert.php>.
- [4] US Environmental Protection Agency (EPA). Emission test cycles: FTP72, January 2014. URL: <http://www.dieselnet.com/standards/cycles/ftp72.php>.
- [5] European Commission. Europe 2020 - EU-wide headline targets for economic growth, October 2014. URL: <http://ec.europa.eu/europe2020/targets/eu-targets/>.
- [6] MathWorks. Choose a solver - MATLAB & simulink - MathWorks, 2014. URL: <http://www.mathworks.de/de/help/simulink/ug/choosing-a-solver.html>.
- [7] John Newman. Dualfoil, August 2014. URL: <http://www.cchem.berkeley.edu/jsngrp/fortran.html>.
- [8] Regatron. TopCon quadro, October 2014. URL: <http://www.regatron.com/en/products-topcon/topcon-quadro-tc-p/topcon-product-overview>.
- [9] Regatron. TopCon TC.GSS, October 2014. URL: <http://www.regatron.com/en/products-topcon/bidirectional-power-supply-gss>.
- [10] thermoanalytics.com. Battery modeling, August 2014. URL: <http://thermoanalytics.com/docs/batteries.html>.
- [11] UC Davis Chem Wiki. Arrhenius equation, October 2014. URL: http://chemwiki.ucdavis.edu/Physical_Chemistry/Kinetics/Modeling_Reaction_Kinetics/Temperature_Dependence_of_Reaction_Rates/The_Arrhenius_Law/Arrhenius_Equation.
- [12] UC Davis Chem Wiki. Nernst equation, August 2014. URL: http://chemwiki.ucdavis.edu/Analytical_Chemistry/Electrochemistry/Nernst_Equation.
- [13] Wikipedia. Fukushima daiichi nuclear disaster, October 2014. Page Version ID: 627607608. URL: https://en.wikipedia.org/w/index.php?title=Fukushima_Daiichi_nuclear_disaster.