UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

# Modeling and model predictive control of batteries to reduce a variability of load

*Author:*
Roman MILISHCHUK

*Supervisor:*
PhD. Tetiana BOGODOROVA

*A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science*

*in the*

Department of Computer Sciences
Faculty of Applied Sciences


Lviv 2020

# Declaration of Authorship

I, Roman MILISHCHUK, declare that this thesis titled, "Modeling and model predictive control of batteries to reduce a variability of load" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

**Modeling and model predictive control of batteries to reduce a variability of load**

by Roman MILISHCHUK

# *Abstract*

Model Predictive Control (MPC) has shown great success in different industrial applications, as it allows to have constraints on both, state and inputs. This ensures that processes will be run under tight performance specifications.

The main goal of this project is to develop an MPC controller which will determine the optimal operation of the battery, i.e. when and how the battery should be charged or discharged, by controlling input current for the battery, to reduce the amount of the money to pay for the required energy. For this task, the battery model with aging effects, which is able to discharge and charge, will be also developed.

Also, an MPC controller for efficient charging of the battery is developed, which allows calculating charging strategy, so without changing the charging time, the lifetime of the battery is prolonged.

Source code [1] is publicly available and can be used for future work in other studies.

[1] Github: github.com/Midren/MPC_for_battery_operation

# *Acknowledgements*

First of all, I express my gratitude to my supervisor Tetiana Bogodorova for her guidance throughout this thesis and for improving it in every aspect.

Also, I am grateful to the Ukrainian Catholic University, which drastically affected my future path by giving me the opportunity to study in the community of passionate people.

# Contents

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **Ah** | **A**mpere-**h**our |
| **SoC** | **S**tate **o**f **C**harge |
| **SoH** | **S**tate **o**f **H**ealth |
| **DoD** | **D**epth **o**f **D**ischarge |
| **CCCV** | **C**onstant **C**urrent **C**constant **V**oltage |
| **AC** | **A**lternating **C**urrent |
| **DC** | **D**irect **C**urrent |
| **RC** | **R**esistor-**C**apacitor |
| **MPC** | **M**odel **P**redictive **C**ontrol |
| **PID** | **P**roportional **I**ntegral **D**erivative |
| **BoL** | **B**eginning **O**f **L**ife |
| **EoL** | **E**nd **O**f **L**ife |
| **FMI** | **F**unctional **M**ock-up **I**nterface |
| **FMU** | **F**unctional **M**ock-up **U**nit |

# Chapter 1

# Introduction

## 1.1 Motivation

In order to reach global green-house emission reduction targets defined by the Paris Agreement and to keep global average temperature increase to well below 2ºC above pre-industrial levels, the future power energy systems will have to rely heavily on energy supplied by renewable energy sources such as wind and solar. Since renewable energy sources are inherently unpredictable and intermittent, it is necessary to design a balancing mechanism that will balance electricity supply and demand at all time. The most promising technology for balancing renewable energy sources with consumer demand is installation of electric batteries [1] [2], which will be charged when excessive or very affordable electric energy is available (typically in the periods of small demand) and discharged in periods when energy is scarce or expensive (typically in the periods of large demand). As battery has its limitation due to maximum capacity and maximum rate, at which it can be discharged, controller is required to know when battery should be charged or discharged. Moreover, expected lifetime of the battery depends on operation conditions, i.e. the charging strategy [3], so controller should not hasten the lifetime of the battery, but rather prolong it.

Model Predictive Control (MPC) has shown a great success in different industrial applications [4], and also was applied for power smoothing of wind generators [1]. Additionally, MPC controller combines advantages of both: feedback and non feedback control systems, and therefore is able to predict disturbances to the system, as well as react to them if they happened unexpectedly. Moreover, constraints are easily added, so controller won't be even choosing input that can violate them, what is crucial in a lot of cases. Therefore, chosen to be employed for the two control problems in this thesis. As MPC controller requires model for prediction and control, the model of a typical Li-ion battery has to be developed and validated in this work as well.

## 1.2 Goals

The main goal of this project is to develop a controller which will determine optimal operation of the battery, i.e. when and how the battery should be charged or discharged, by controlling input current for the battery, to reduce amount of the money to pay for the required energy.

For this task, a battery model will be developed, which is capable of simulating charging and discharging with different current input and constant temperature, as there controller will not be able to optimize it. Furthermore, model should be

capable of simulating ageing effects depending on the operation conditions of the battery.

Based on the battery model, Model predictive control (MPC) will be used to develop a controller which will be able to efficiently charge and discharge the battery, so speed of charging and lifetime of the battery will be maximized. Further, this controller to be extended enabling effectively reduce variability of load, i.e. daily difference between maximum and minimum power taken from the power grid.

## 1.3 Thesis structure

The rest of the thesis is organized as follows:

**Chapter 2. Battery modeling**
Analyzes scientific literature for the battery modeling including the datasheets of battery producers and proposes a model which can be used for model-based optimizations requiring battery runtime information and ageing effects.

**Chapter 3. Model Predictive Control for battery operation**
Describes how an MPC can be used for battery operation and proposes the developed implementations for two controllers:

- Controller for optimal charging of the battery
- Controller for reducing variability of load in that way minimizing the cost of energy for the consumer

**Chapter 4. Results and summary**
Summarizes the achieved results and their comparison and shares our ideas for future work.

# Chapter 2

# Battery modeling

## 2.1 Background Information

As level of digitalization rises in our world and smart homes are becoming reality, more items in our household have batteries. Also, amount of energy produced by renewable energy sources, such as photovoltaic or wind systems, usually fluctuates depending on the environment conditions, and battery storage are frequently used to have reliable amount of energy in any time. [5] [1] Other important usage of the batteries are electric vehicles, and hybrid electric vehicles. [6] All the batteries are working using the same principle: negative electrons are produced in the anode and transferred to the cathode, as they are created from different materials and so with different standard potentials. Also, electrolyte provides a medium through which charge-balancing ions can flow [7]. But depending on the materials used for electrode and electrolyte, battery has different properties which are applicable in diverse number of situations. The most important characteristics of batteries that are taken into consideration when modeling are:

- Nominal voltage - an output voltage of the battery. Even though with constant load battery voltage stabilizes near a nominal voltage value, but it has variations depending on the current state of charge of the battery.
- Nominal capacity - maximum amount of the energy stored in the battery. Due to ageing, batteries tend to decrease their capacity.
- Power density - amount of the energy stored in the battery per weight.
- Power rating - amount of the energy that battery can deliver at one time.

Different materials are used for electrodes (NiCad, NiMH, Lead-acid), but Li-ion batteries are the most common in consumer electronics. They are considered the state-of-the-art, as have very high power density and light weight. On the other hand, they are delicate and misusage can lead to explosion. It is unsafe to charge above maximum safe voltage or discharge below minimum, or charge with more current than battery can take. Usually, such information can be found in the datasheet. [8] So such batteries are usually accompanied with protection circuits which are preserve from such situations. [9]

During operation of a battery, a state of the battery is changing. This change can be described using several characteristics:

- State of Charge (SoC) - amount of energy currently stored in battery relative to usable capacity

- State of Health (SoH) - indicates the state of battery between Beginning of Life (BoL) and End of Life (EoL) in the scale from 0 to 1. Calculation of SoH depends on the application, as three main indicators are capacity, internal resistance and self-discharge. [10] For our problem capacity is the most important and will be used for SoH calculation.
- Voltage - measured voltage on the ends of the battery
- C-rate - rate of charge/discharge measured in **C** that is Amphere-hour capacity divided by 1 hour.
- Depth of discharge (DoD) - a fraction of the capacity which has been removed from battery during one discharge cyclee. Most manufacture specify a maximum DoD for a optimal performance, otherwise SoH of the battery will decrease at higher rates. [5]

## 2.2 Related works

In literature, many different models that simulate the behavior of Li-ion batteries are developed [11, 12, 13]. Electrochemical and analytical models usually represent systems with partial differential equations, the solution to which requires a lot of time and mathematical skills. Example of such are Shepherd and Generic battery models [14]. They also assume that internal resistances are constant, and so are incapable of modeling ageing effects. But it was shown, that ageing analytical model can be developed for lithium-ion cells. [15] Such models pay more attention to global system-level behavior, such as battery runtime or capacity. Also, such models are capable of modeling the aging of the batteries. Equivalent circuit-based models are the most common, as they can give current $I$ and voltage $V$ information, have more accurate results, and more intuitive for engineers [12]. Most of the electrical models of Li-ion batteries can be categorized into 3 groups: Thevenin-, impedance-, and runtime-based models.

### 2.2.1 Impedance-based model

The impedance-based model, which is shown in Figure 2.1, is based on electrochemical impedance spectroscopy to model an AC-equivalent impedance $Z_{AC}$ in the frequency domain [16]. The fitting process of $Z_{AC}$ to impedance spectra is difficult and nonintuitive. One of the main characteristics of the batteries is the State of Charge (SoC), which shows the fraction of energy that is currently stored in the battery. The impedance-based model works only for a fixed SoC and temperature setting [12], but for our problem, we need to be able to charge/discharge the battery and see how it affects the model, so this model is not applicable.

### 2.2.2 Thevenin-based model

The simplest form of Thevenin-based model is shown in Figure 2.2a, which consists of a voltage source $U_{OC}$, a series resistor $R_s$ and a resistor-capacitor (RC) circuit ($R_{t,s}$, $C_{t,s}$). The voltage $U_{OC}$ is assumed to be constant during all the simulations, which prevents capturing the run time information as well as DC response. So it cannot simulate the model, which is charging or discharging. The series resistor $R_s$ is used for getting an instantaneous voltage drop, while the RC circuit is necessary for modelling a transient behavior of the battery. Therefore, the simple form

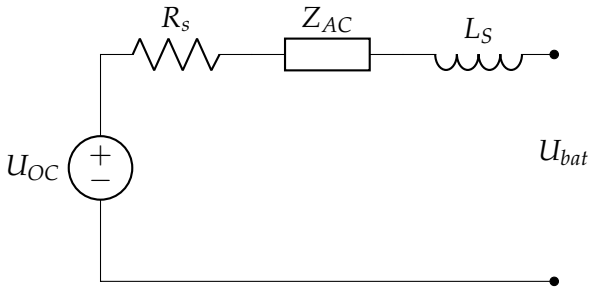
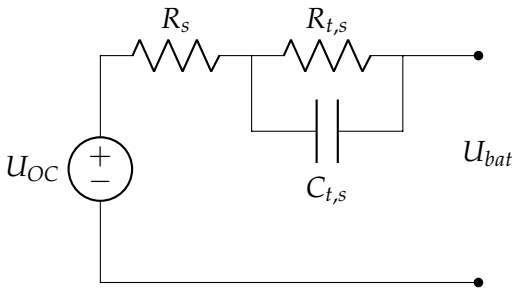
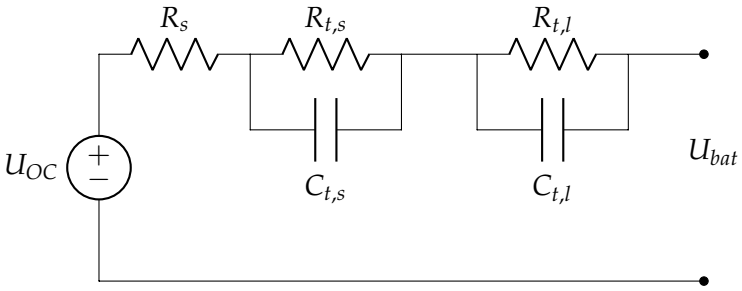



FIGURE 2.1: Impedance-based model




(A) 1 RC circuit




(B) 2 RC circuit

FIGURE 2.2: Thevenin-based models

of Thevenin-based model can model only short-term transient dynamics of the battery. But in multiple studies more RC circuits (e.g, 2 RC circuits in Figure 2.2b) can be added to improve accuracy and model long-term dynamics [12]. To take DC response or battery runtime into account, additional components can be added to the model, but not both if implementation is made in circuit simulators [11]. Also, it was found that circuit components are dependant on SoC with nearly constant values for 20 % - 80 % and change exponentially within 0 % - 20 % of SoC.[12].

The most frequent way[11], [12] for SoC calculation is by the use of the Ampere-Hour integral (Coulomb counting) method, which requires knowledge of inital SoC level. This method is based on the charge value that has been transferred into or out of the battery.

$$SoC = SoC_0 - \frac{1}{Q_{us}} \int I_{bat}(t)dt \tag{2.1}$$

where:

$SoC_0$ – Initial SOC
$Q_{us}$ – Usable charge capacity

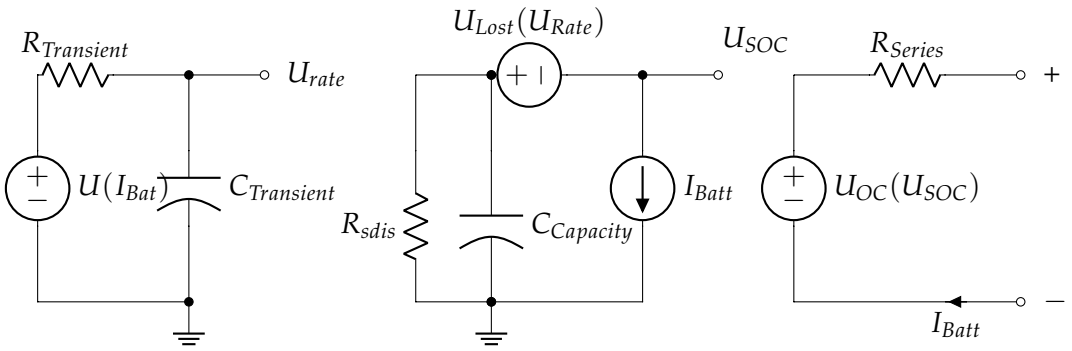



FIGURE 2.3: Runtime-based model

Further in [11] the researches show the dependence on temperature by Arrhenius relation 2.2 and high current values for long term transient resistance.

$$A = A_0 \cdot e^{-\frac{E_a}{RT}} \tag{2.2}$$

where $A$ is quantity of interest, $A_0$ the pre-exponential term, $E_a$ the activation energy, $R$ the gas constant and $T$ the temperature in Kelvin. However, in this work impact of using models at different temperatures is not important and can be neglected. To take into account an exponential behaviour of voltage response with low SoC values, exponential term can be used, as shown in 2.3 [12].

$$R(SoC) = R_0 + k_1 \cdot \exp^{k_2 \cdot SoC} \tag{2.3}$$

Another method of finding dependency between $SoC$ and circuit components is fitting high-order polynomial (i.e, of order 4-7), which gives more accurate results for series resistor value dependency on SoC, but is very battery-specific [11]. Thus, all dependencies of the Thevenin-based circuit parameters (Figure 2.2b) on SoC can be summarized as:

$$R_s(SoC) = R_{s_0} + k_{R_s,1} \cdot SoC + k_{R_s,2} \cdot SoC^2 + k_{R_s,3} \cdot SoC^3 + k_{R_s,4} \cdot SoC^4 \tag{2.4}$$

$$R_{t,s}(SoC) = R_{t,s_0} + \sum_{i=0}^{4} k_{R_{ts,i}} \cdot SoC^i \tag{2.5}$$

$$C_{t,s}(SoC) = C_{t,s_0} + \sum_{i=0}^{6} k_{C_{ts,i}} \cdot SoC^i \tag{2.6}$$

$$R_{t,l}(SoC) = R_{t,l_0} + k_{R_{tl,1}} \cdot \exp^{k_{R_{tl,2}} \cdot SoC} + k_{R_{tl,3}} \cdot SoC \tag{2.7}$$

$$C_{t,l}(SoC) = C_{t,l_0} + \sum_{i=0}^{6} k_{C_{tl,i}} \cdot SoC^i \tag{2.8}$$

where:

$k_{R_s,i}, k_{R_{ts,i}}, k_{C_{ts,i}}, k_{R_{tl,2}}, k_{C_{tl,i}}$ — parameters dependent on battery type

$R_{s_0}, R_{t,s_0}, C_{t,s_0}, R_{t,l_0}, C_{t,l_0}$ — initial parameter values of resistance, capacitance, which do not depend on SoC

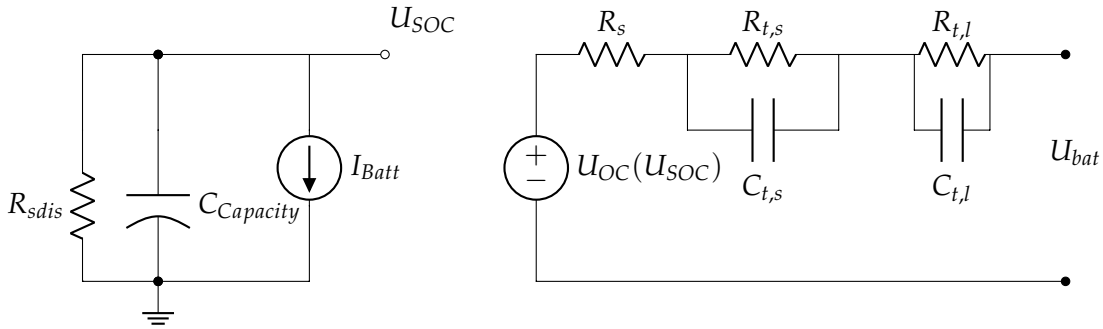



FIGURE 2.4: Combined electrical circuit-based model

### 2.2.3 Runtime-based model

Runtime-based models use a complex circuit network to simulate battery runtime and DC voltage response under constant discharge. However, it cannot accurately predict runtime or voltage response for varying load current [12]. An example of such model is shown in Figure 2.3.

### 2.2.4 Combined electrical circuit-based model

| Predicting Capability | Thevenin-based | Impedance-based | Runtime-based |
|---|---|---|---|
| DC | No | No | Yes |
| Transient | Yes | Limited | Limited |
| Battery Runtime | No | No | Yes |

TABLE 2.1: Comparison of various circuit models [12]

The Table 2.1 summarizes the main differences between models that were presented above. To deal with the downsides of each model, a combination of the Thevenin-based and runtime-based models can be used [12]. A new model, which is shown in Figure 2.4, is capable of simulating battery runtime, DC response and transient behaviour of the battery. It can be observed, that the left part with a capacitor $C_{Capacity}$ and variable current source $I_{Batt}$ models the capacity, SOC, and runtime of the battery. The output of the circuit (left part) is used to connect SoC $U_{SOC}$ to open-circuit voltage $U_{Bat}$ in a form of a variable voltage source $U_{OC}(U_{SOC})$. Thus, the right circuit is capable of showing transient behavior for different SoC. As $U_{SOC}$ is changing from 1 to 0 V during discharging and vice versa, it can be assumed to be equal to the SoC. It was found that $U_{OC}$ has only dependency on SoC level which is mainly nearly linear dependency, besides the SoC values are low (0-0.2) or high (0.8-1) where it has exponential behavior. [12] We can express this dependency as:

$$U_{OC}(SoC) = k_0 + k_1 \cdot SoC + k_2 \cdot \exp^{k_3 \cdot SoC} + + k_4 \cdot \exp^{\frac{k_5}{1-SoC}} \tag{2.9}$$

where $k_0, k_1, k_2, k_3, k_4, k_5$ - the coefficients which are calculated by curve fitting during parameter extraction

### 2.2.5 Models with aging effects

Another important aspect of a battery state is a state of health (SoH). In all previous models, the battery doesn't track previous charge/discharge cycles and runtime

conditions. Due to aging effects, usable capacity of a battery is decreasing, therefore, a capacity fading $\xi$ can be defined as:

$$\xi = Q_{nom} - Q_{use} \tag{2.10}$$

where

$Q_{nom}$ – nominal capacity of the battery
$Q_{use}$ – maximum usable capacity that battery currently can store
$\xi$ – capacity fading

The end of life condition depends on the application of a battery, but by convention is considered at 80 % of the nominal capacity. Capacity fading $\xi$ is considered as unique $SoH$ indicator, as self-discharge is omitted as it cannot affected by optimization, so $SoH$ can be calculated, as:

$$SoH = 1 - \frac{\xi}{0.2 \cdot Q_{nom}} \tag{2.11}$$

Capacity fading can be divided into two losses: calendar and cycling. The first one is mainly dependant on the battery's total working time and temperature, which affects electrode film growth [11]. While for the cycling loss the main factors are:

- Temperature
- Depth of Discharge
- Overcharge, when excess energy is delivered to recharge the battery
- C-rate

Two main approaches for models with capacity fading calculation are:

- Performance-based models, which are capable of simulating the change of perfomance of the battery, where different parameters (e.g, voltage, current, capacity) are monitored constantly, and when performance falls into some threshold, EoL of the battery is reached [17].
- Weighted-throughput models, where the weighted throughput of chosen parameter (e.g, charge processed, number of cycles, or time of operation) is linked to the EoL of the battery. The manufacturer usually gives information about lifetime under certain conditions [17].

As the runtime conditions are not always constant, we can define a set of stress factors variables, which are factors affecting cycling loss, which are mentioned above 2.2.5. In both models, the concept of the event is used. Event is time interval, during which the stress factors are not changing.

One of examples of the weighted-throughput model is the Palmgren-Miner rule model. The rule states that the life of a component under a sequence of variable loads is reduced each time by a finite value. This fraction corresponds to the ratio between the time spent under the given load condition and the number of cycles that the component would last if load condition were constant during whole life of the battery [17]. Thus, the EoL of the battery is reached, when sum of fraction for each event is a unit:

$$\sum_{i=1}^{E} \frac{\Delta t_i}{t_i^f(\sigma_i)} = 1 \tag{2.12}$$

where:

$E$ – number of events
$\Delta t_i$ – time during an event $i$
$\sigma_i$ – constant stress factor during an event $i$
$t_i^f$ – time to reach EoL with stress factor $\sigma_i$

Another approach is by the use of the damage accumulation model which is the performance-based model. The amount of capacity fading of battery is modeled as damage and we can express the rate at which damage develops as [11]:

$$\frac{d\xi(t)}{dt} = \phi(\xi, \sigma) \tag{2.13}$$

where $\phi$ is some function of capacity fading $\xi$ and stress factor $\sigma$ So capacity fading can be calculated as:

$$\xi = \sum_{i=1}^{E} \int_{t_{i-1}}^{t_i} d\xi_i(t)dt \tag{2.14}$$

where:

$t_i$ – end time of the event i

Palmgren-Miner rule-based and damage accumulation models are equivalent if damage rate can be factorised. A disadvantage of the first one is that whole history of capacity fading rate should be known which is hard to accomplish in real cases. Also, the damage accumulation model is easier for modifications [17].

In [13] and [11] it was shown that crack propagation and damage accumulation models can be combined for modeling of capacity degradation. To differentiate high SoC and DoD, $SoC_{avg}$ and $SoC_{dev}$ are introduced, as high $SoC_{avg}$ shows that battery is operating on high $SoC$ and high $SoC_{dev}$ shows that $DoD$ is high and $SoC$ during one cycle varies a lot. We can calculate these parameters for the event i as:

$$\begin{cases} SoC_{avg,i} = \frac{1}{t_{\Delta i}} \int_{t_{i-1}}^{t_i} SoC(t)dt \\ SoC_{dev,i} = 2\sqrt{3}\sqrt{\frac{1}{t_{\Delta i}} \int_{t_{i-1}}^{t_i} (SoC(t) - SoC_{avg,i})^2 dt} \end{cases} \tag{2.15}$$

Also, because battery cycles can vary in processed energy amount, effective throughput cycles are used instead of charge-discharge cycles:

$$N_{eff} = \int_{t_{i-1}}^{t_i} \frac{|I_{bat}(t)|}{2 \cdot Q_{nom}} dt \tag{2.16}$$

where $N_{eff}$ - number of effective cycles

Thus, the capacity fading can be calculated for charge-discharge cycle $i$ taking into account depth of discharge in the following way:

$$\xi_i = K_{co} \cdot N_{eff} \cdot e^{\frac{SoC_{dev,i}-1}{K_{ex}} \frac{T_{ref}}{T}} + 0.2 \frac{t_{\Delta i}}{t_{life}} \tag{2.17}$$

where:

$K_{co}, K_{ex}$ – battery specific constants, which are fitted to battery life data
$T_{ref}$ – reference temperature of 20°C in Kelvin
$t_{\Delta i}$ – duration of event $i$
$t_{life}$ – estimated calendar life at reference temperature

An adjustment for the average state of charge is performed, because the degradation rate proportional to the concentration of Lithium ions left in an active form.

$$\xi_i = (K_{co} \cdot N_{eff} \cdot e^{\frac{SoC_{dev,i}-1}{K_{ex}} \frac{T_{ref}}{T}} + 0.2\frac{t_{\Delta i}}{t_{life}}) \cdot e^{K_{soc} \frac{SoC_{avg,i}-0.5}{0.25}} \cdot (1 - L_i) \tag{2.18}$$

where:

$K_{soc}$ – battery specific constant, which are fitted to battery life data
$L_i$ – damage accumulation term

On each event iteration step $L_i$ takes a previous damage into account and can be expressed as:

$$L_i = \sum_{i=1}^{E} \xi_i \tag{2.19}$$

Also, $L_i$ can be calculated iteratively, but knowing accumulated damage from previous cycle:

$$L_i = L_{i-1} + \xi_i \tag{2.20}$$

## 2.3 Proposed implementation of battery model

For the considered task in this thesis, model should be able to simulate battery runtime with ageing. To achieve this, combined Thevenin-based and runtime-based model is used with additional logic for capacity fading calculation. In Figure 2.5 general architecture can be seen. A battery model was implemented using the Modelica language with a use of OpenModelica environment. Modelica is an object-oriented modeling language with a lot of dynamical models and its components in Modelica Standard Library that are used in engineering. Even though Modelica is a high-level language, where you can describe model in a form of algebraic and differential equations, or even connect components using graphical user interface, Modelica translates its models into C code, allowing to simulate models with a high performance. In addition to Modelica language advantages, OpenModelica provides free open-source simulation and modeling environment with additional tooling (i.e. linearization of models) and Python binding for OpenModelica compiler (omc) allowing to run models and visualize output using Python.

### 2.3.1 Thevenin-based model as a part of the proposed model

Thevenin-based model with 2 RC circuits is used as the base to the battery model. So it is able to show voltage-current information for a particular SoC and simulate short-term and long-term dynamics of the battery. To show different behaviour of the battery depending on SoC and charging/discharging cycle, equations from [11] are employed, but simplified for a condition of a constant temperature $T = 20°C$.

#### Dynamics inside Thevenin-based model

To apply model predictive control for optimal operation of the battery, we need to know the state of the system. Therefore, the model is described using a system of equations (2.21, 2.25, 2.24) that later can be used to find the state variables.

Using Kirchhoff's voltage rule, the voltage across the battery $U_{bat}$ (Figure 2.5) can be calculated in the following way:

$$U_{bat} = U_{OC} - U_{t,s} - U_{t,l} - U_s \tag{2.21}$$

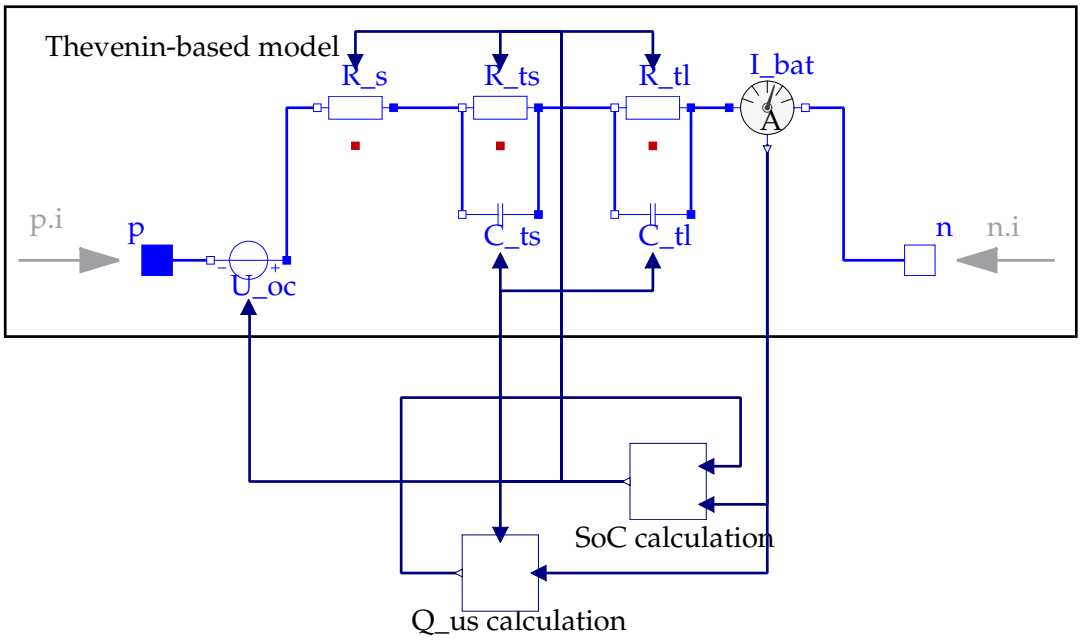



FIGURE 2.5: The model of battery implemented using Modelica

where:

$U_{bat}$ – battery voltage
$U_{OC}$ – open-circuit voltage
$U_{t,s}$ – voltage across the first RC circuit $(R_{t,s}, C_{th,s})$
$U_{t,l}$ – voltage across the second RC circuit $(R_{t,l}, C_{th,l})$
$U_s$ – voltage across the $R_s$ resistor

Using the Ohm's rule the voltage across $R_s$ resistor (Figure 2.5) depends on battery current and resistance $R_s$ in the following way:

$$U_s = \frac{I_{bat}}{R_s} \tag{2.22}$$

where:

$I_{bat}$ – battery current

Using Kirchhoff's current law and Ohm's rule, voltage across components in $RC$ circuit $(R_{t,s}, C_{t,s}$ (Figure 2.5) depends on current across each component in the following way:

$$\begin{cases} I_{bat} = i_1(t) + i_2(t) \\ U_{t,s} = \int \frac{i_1(t)}{C_{t,s}} dt \\ U_{t,s} = i_2(t) \cdot R_{t,s} \end{cases} \tag{2.23}$$

where:

$i_1$ – current across the capacitor $C_{t,ks}$
$i_2$ – current across the resistor $R_{t,s}$
$C_{t,s}$ – capacitance of the capacitor $C_{t,s}$
$R_{t,s}$ – resistance of the resistor $R_{t,s}$

We can combine equations from system 2.23 to formulate the voltage change across $U_{t,s}$:

$$\dot{U_{t,s}} = \frac{i_1(t)}{C_{t,s}} = \frac{I_{bat} - i_2(t)}{C_{t,s}} = \frac{I_{bat}}{C_{t,s}} - \frac{U_{t,s}}{R_{t,s} \cdot C_{t,s}} \tag{2.24}$$

By analogy, we can find a similar formulation for the voltage $U_{t,l}$ of the second RC circuit $(R_{t,l}, C_{th,l})$ (Figure 2.5):

$$\dot{U_{t,l}} = \frac{I_{bat}}{C_{t,l}} - \frac{U_{t,l}}{R_{t,l} \cdot C_{t,l}} \tag{2.25}$$

### 2.3.2 Runtime-based model as a part of the proposed model

The default Thevenin-based model is working only with constant SoC and voltage on open-circuit. To get a variation of SoC depending on the voltage of an open-circuit, the Thevenin-based model was combined with runtime-based model. As a result, in such an approach, $U_{oc}$ will vary voltage depending on the current SoC (Figure 2.5). The function which shows dependency between SoC and $U_{oc}$ is taken from [11] where the dependency was precomputed for all values of the SoC according to the characteristics of the real battery. Thus, the voltage source $U_{oc}$ in the model uses lookup table to change the voltage depending on the current level of SoC (Figure 2.5). To calculate the SoC of the battery, equation 2.1 for Coulomb counting method was used. The main disadvantage of this method is requirement to know the initial SoC, however, in our model, we always have this information.

### 2.3.3 Adding ageing parameter to the model

Due to ageing of the battery, such effects as capacity fading, self-discharge and internal resistance take place, but for our problem the capacity fading is of the main interest, as an affect of the total runtime of the battery. Therefore, for our application a combination of damage accumulation and crack propagation models was used based on [13], where equation for capacity fading is 2.18. As in real-life operation cycling losses are much higher than calendar losses [11] and impact from the latter cannot be decreased during optimization, therefore, the calendar losses are omitted. Thus, $0.2\frac{t_{\Delta i}}{t_{life}}$ term in equation 2.18 which responsible for the calendar losses is removed.

Even though the Arrhenius relation 2.2 was shown as the best way to predict capacity fading dependence on the temperature [11] [13], but for our task, the impact of using a model at different temperature is not considered and can be neglected. However, in previous work [11] was shown, that $E_a$ from Arrhenius relation 2.2 is dependent on $SoC$ to increase ageing speed during operation on high $SoC$. However, by the use of $SoC_{avg}$ 2.18 we are able to use this dependency without Arrhenius relation, as our model is neglecting temperature effects.

As a result, equation 2.18 that defines the capacity fading $\xi$ can be rewritten as:

$$\xi_i = K_{co} \cdot N_{eff} \cdot e^{\frac{SoC_{dev,i}-1}{K_{ex}}} \cdot e^{K_{soc}\frac{SoC_{avg,i}-0.5}{0.25}} \cdot (1 - L) \tag{2.26}$$

For model predictive control, the controller solves optimization problem at each step, and need to have feedback how the change of current input affects battery model, so $SoH$ of the battery will be maximized. Even though damage accumulation model operates over discharge-charge cycles, the damage accumulation model is

adapted to calculate capacity fading for current discharge-charge cycle iteratively at each step of model simulation.

Thus, during a current cycle $i$ the capacity fading $\xi_i$ is recalculated at each iteration using 2.26. The total capacity fading $\xi$ is calculated as a sum of the capacity fading during a current moment of time and a damage accumulation:

$$\xi = L_{i-1} + \xi_i \tag{2.27}$$

where $L_i$ - previously accumulated damage 2.20

Because of the new approach for calculation of the total capacity fading **??**, the capacity fading can not only increase, but also decrease during a battery operation cycle. However, when discharge or charge cycle is ended, changes to capacity fading $\xi$ are irreversible.

### 2.3.4 Parameters identification

For Thevenin-based models, it is required to find values of coefficients, which was used to describe the model (2.9, 2.4, 2.5, 2.6, 2.7, 2.8). The main approach for identifying those parameters is to analyse voltage during pulse charging and discharging cycles. During pulse charging, battery is charge with constant current for some amount of time and then stopped charging, so transient behavior of voltage of response can be seen for particular value of $SoC$. Current can be constant during all pulses [18] or alternating [19]. The advantage of the latter one is that less testing should be done, but it is impossible to observe how charging with higher current affects aging of the battery [11].

rest times (when battery is not charging nor discharging), charging and discharging.

Commonly [18] [19], $U_{oc}$ is assumed to be only dependent on $SoC$ while parameters for others circuit components ($R_s$, $R_{t,s}$, $C_{t,s}$, $R_{t,l}$, $C_{t,l}$) constant. However, in [11] was shown dependence of $SoC$ on resistance and capacitance. So, function of those (2.6, 2.7, 2.8) were fitted by high-order polinomials

As, the model's parameters identification requires a knowledge of the hardware characteristics (e.g. battery specification from the producer) and the laboratory testing of the real equipment that is time-consuming. In this work the parameters for the circuit components are used from [11]. These parameters correspond to the A123 APR18650M1 battery dynamics (parameters can be seen in Appendix A) However, some parameters as temperature dependence are neglected. The parameters for capacity fading are taken from [13]. Even though the parameters in [11] and [13] are provided for different batteries (A123 ANR26650M1A and A123 ANR18650M1), it can be seen in Fig. 2.6, that those two batteries have similar capacity fading cycles. Data was extracted from corresponding datasheets [8] [20]. The battery operation was tested in the same conditions, where batteries were charged and discharged multiple times with 1$C$ rate, and during each cycle battery was fully charged and discharged, so $DoD$ is 100 %.

## 2.4 Simulation results

To validate the resulted model of the Li-ion battery, the essential characteristics of the battery were tested and validated against those presented in the scientific literature. First of all, the resulted parameters of the circuit components that depend

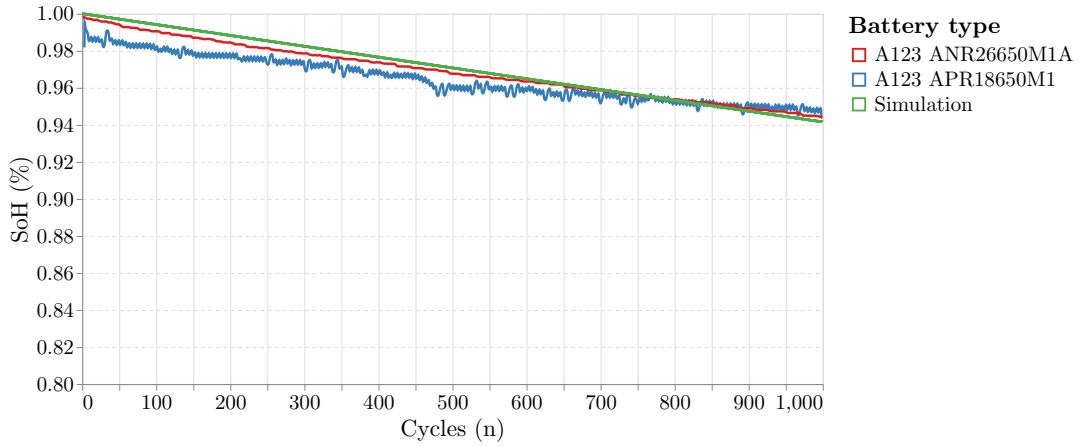



FIGURE 2.6: Capacity fading for two types of batteries and the simulated model (Figure 2.5)

on SoC, that are calculated by running discharge cycle with 100 % DoD, were validated against the corresponding characteristic in [11]. In Figure 2.8 and Figure 2.7 the obtained dependencies of the open circuit voltage $U_{oc}$, capacitances $C_{t,l}$, $C_{t,s}$, and resistances $R_s$, $R_{t,l}$, $R_{t,s}$ (Figure 2.5) are plotted for charging and discharging cycles of the battery operation. The results in Figure 2.8 and Figure 2.7 show the similarity of the parameters with respect to the reference characteristics in the literature [11]. Thus, the plotted characteristics are mostly the same for charging and for discharging cycle (with small difference) of the battery.

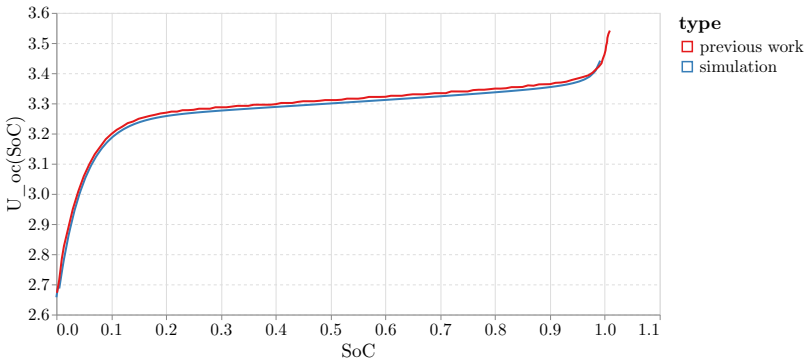



FIGURE 2.7: Comparison of the open circuit voltage $U_{oc}$ dependency on SoC of the proposed model and the reference model [11]

Next, voltage response for charging and discharging was tested. As voltage response of battery operation in [11] were taken during rapid test-procedure [19], resulted model was also simulated with same profile (Figure 2.9) for comparison. The main advantage of this test is that results from one charge and discharge cycle can be analyzed to extract model parameters. So researchers in [11] has used to extract parameters which are described in Section 2.3.4. During the test the obtained current $I_{req}$ characteristic is presented in Figure 2.9, where positive current $I_req$ corresponds to discharging and negative for charging. While the comparison of voltage response $U_{bat}$ of the simulated battery with respect to the reference [11] can be seen in Figure 2.10. The close agreement between simulation results and measurement data indicates that the developed model allows to predict runtime and transient voltage response accurately.

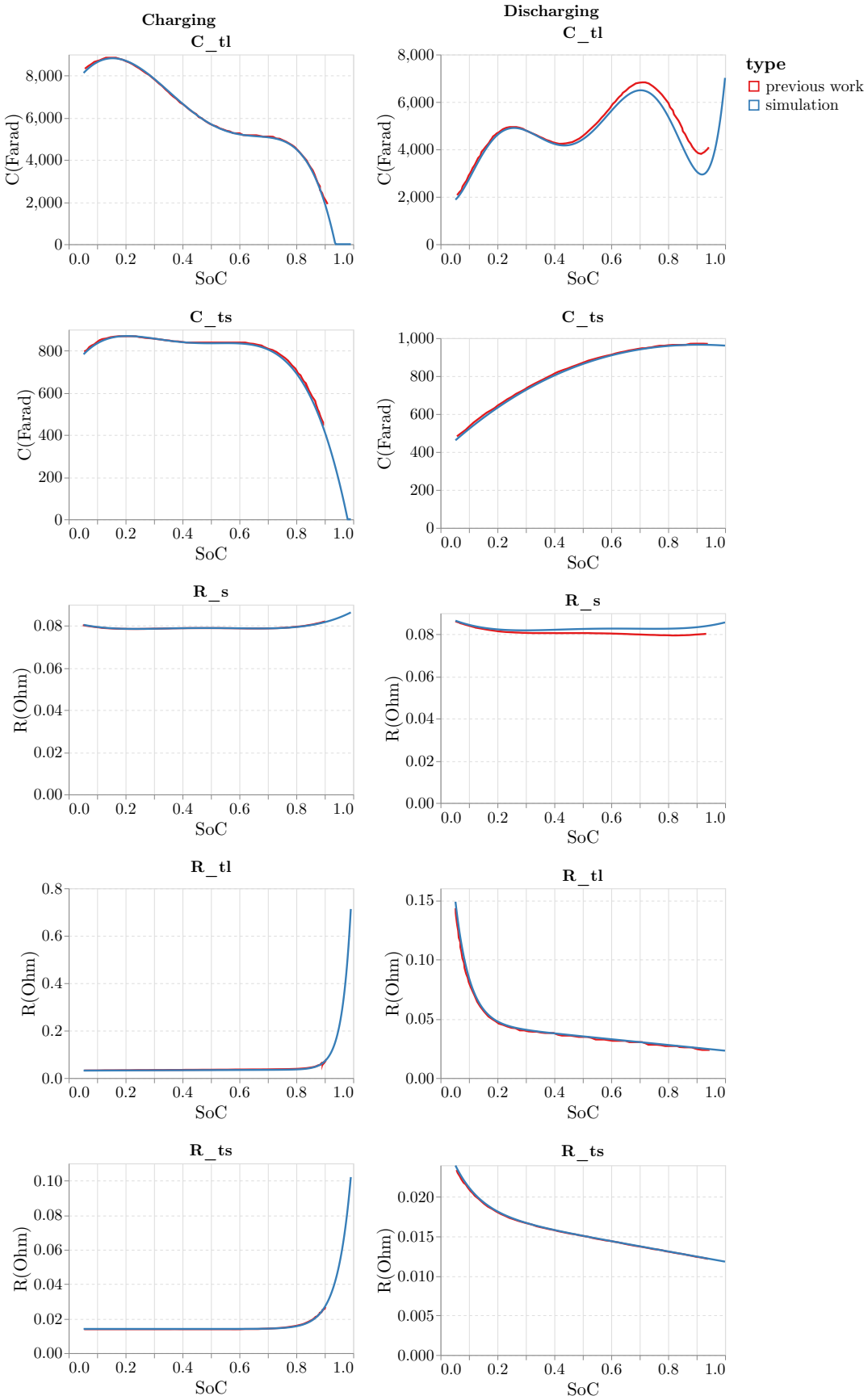



FIGURE 2.8: Parameters dependence from SoC

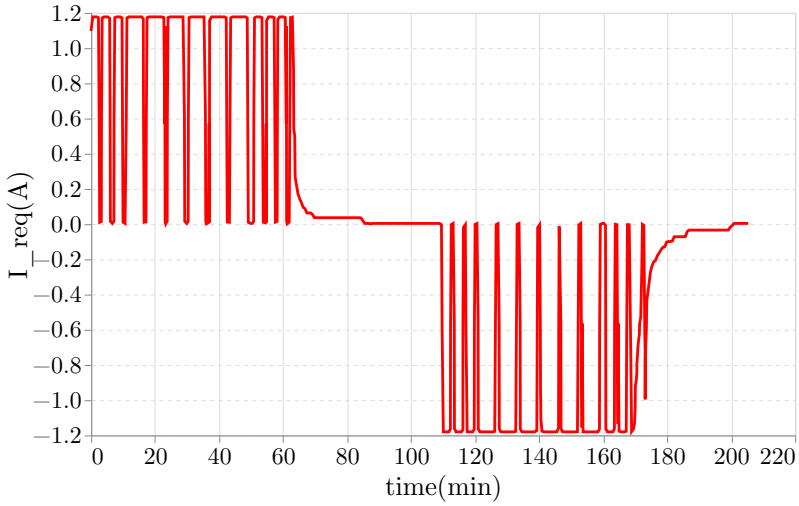



FIGURE 2.9: Test cycle for determining the model parameters

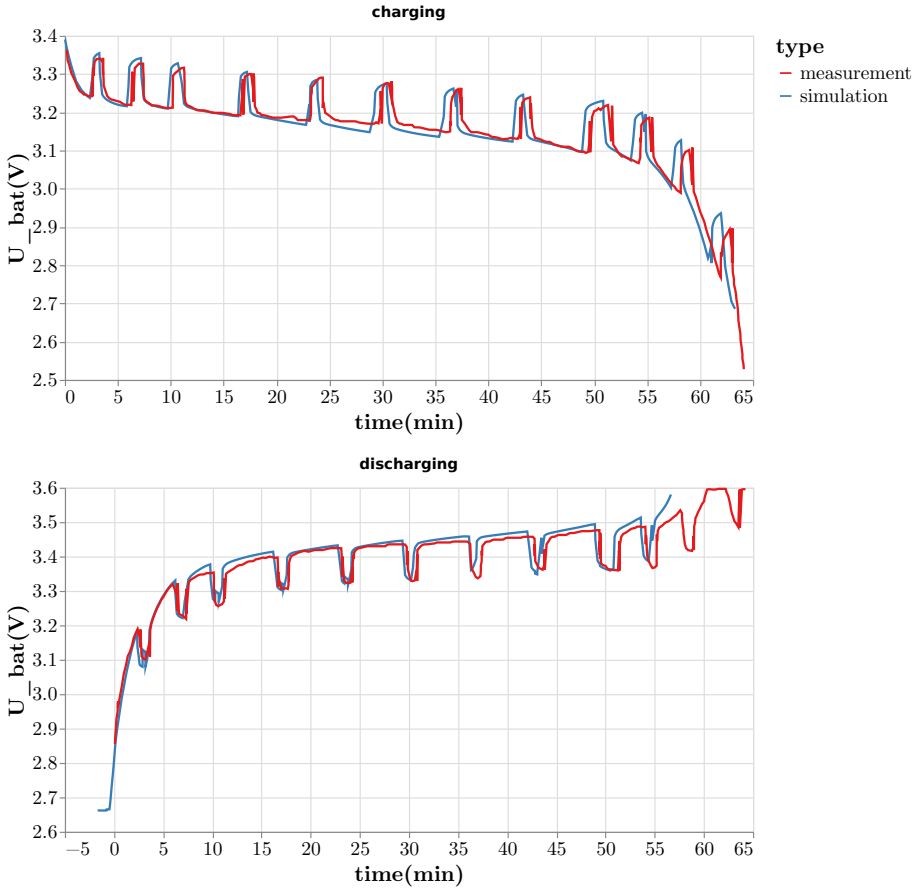



FIGURE 2.10: Battery voltage response during test cycle

Finally, to validate the model, it was used to simulate 1000 full cycles of charging and discharging, so $DoD$ was 100 %, with 1C rate for charging and discharging (so one cycle continued for 1 hour) and with 1.3C rate for charging and 2.1C rate for discharging (Figure 2.11). The resulted characteristics of the model were compared to the characteristics in the datasheets [8] Thus, from the resulted plots of Figure 2.11 one can conclude that the simulated battery is capable of modelling capacity fading depending on different battery usage cycles. However, simulated capacity fading shows linear trend, while measurements from real model have a slight exponential behavior at the beginning of life (BoL) . To improve the simulated behaviour, the weight of the damage accumulation term $L$ (2.26) could be increased to have more rapid capacity fading in the BoL. In addition, the capacity fading that presented as SoH of the battery with 1.3C/2.1C rates was tested by manufacturer for different temperature (45C comparing to 25C for 1C/1C test) [8]. Despite omitted heat effects in the proposed model, rates of capacity fading are similar.

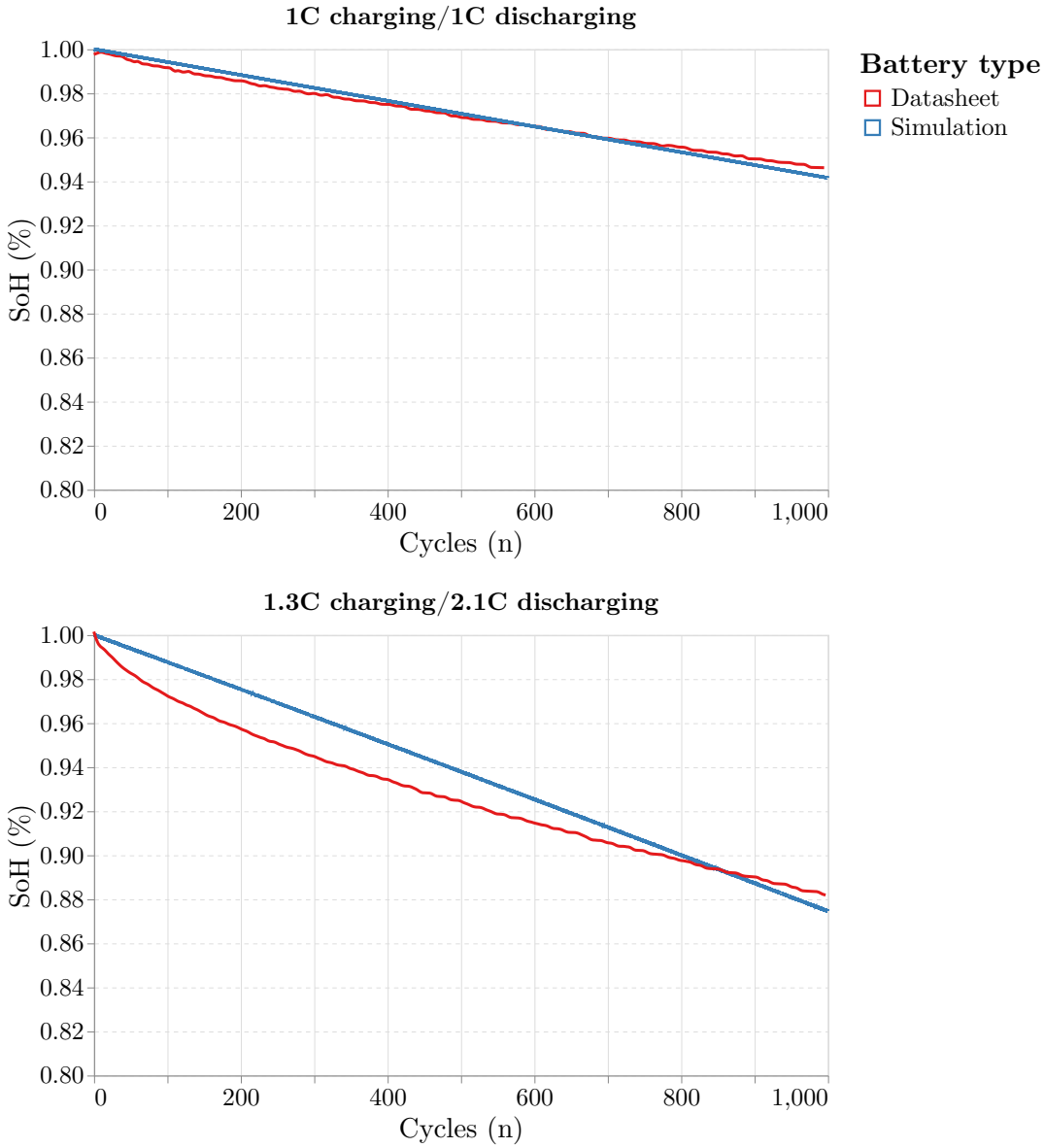



FIGURE 2.11: Capacity fading for different load

# Chapter 3

# Model Predictive Control for battery operation

## 3.1 Background information

Control systems are used in various aspects of our life and allow to achieve a desired response of the system by controlling its output. There are two types of control systems: open-loop and closed-loop, or with no feedback and with feedback. The main difference is that the last one has not only the reference input, which should be achieved, but also the output of the system going through the feedback loop into the input of the system. This makes the controller more stable, as it can adapt future input by reacting to the changes in the output and passing the change via the feedback block to the input. Open-loop on other hand can just predict which input should be given to the system and are less accurate and less stable [21].

One of the most important and widely used controllers is the PID controller. It is the closed-loop controller which applies a correction to the plant output based on proportional, integral, and derivative terms. It is common to use one or two components of PID controller by setting control gain to zero. Using proportional control alone for changing reference value will always result in some error, as it requires error to compensate it by adapting input. To overcome this drawback of the propotional control, the integral component is used to eliminate residual offset by integrating the error over time [21]. A derivative component, on other side, tries to predict a future trend of the error and sometimes is omitted to make the system more robust for noisy data.

The battery can be considered a controlled object, as it commonly requires controlling inputs during its operation. For example, tracking and control of battery characteristics, such as SoC or voltage, is required to prevent misuse of Li-ion batteries in protection circuits [9]. Another control application for the battery is to track and tune voltage and current of the battery depending on the charging strategy. The most common charging strategy is constant-current/constant-voltage, where battery is charged with constant voltage until the nominal voltage is achieved, after that charging and supporting nominal voltage until the battery is fully charged. Typical curves for current and voltage can be seen in Figure 3.1. As we know the desired trajectories of current and voltage, the PI controller can be used to charging [22].

Model Predictive Control (MPC) made an enormous impact on industrial control engineering and inspired many academics for further research. The main advantages of MPC over an ordinary PID controller are:

- It can handle problems with multiple inputs and outputs
- It can easily put constraints on input, as well as on state variables of the controlled object

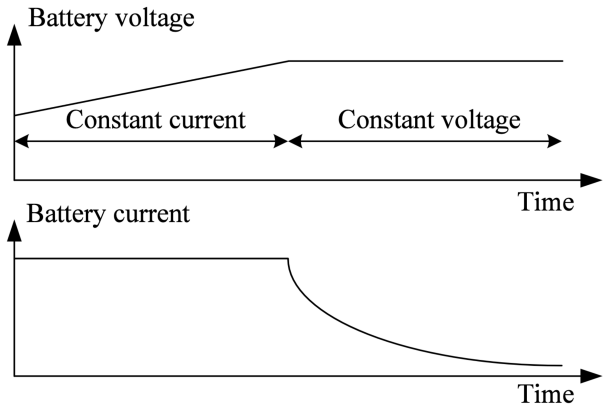



FIGURE 3.1: Constant-current/constant-voltage strategy[22]

- It combines advantages from both: open-loop and closed-loop controllers.

As it can be guessed from the name, MPC is model-based control in the sense, that it uses an internal model of process to predict behavior of the object. The general scheme of MPC optimization operation is presented in Figure 3.2. The controller at each state predicts behavior for $p$ time points ahead up to prediction horizon and changes an input of the object aiming to minimize the difference between reference trajectory and the predicted output. Most commonly, a least-squared optimization problem is solved to find an input that would optimize a controlling aim - a cost function. Also, it is widespread to use different prediction and control horizons, to reduce computation complexity [4].

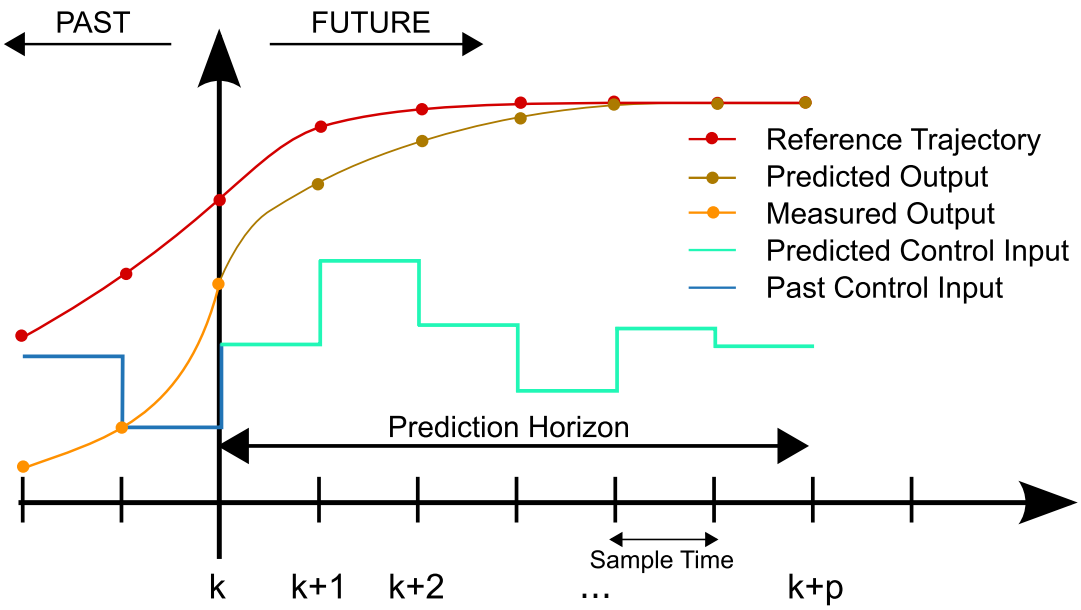



FIGURE 3.2: Model predictive control operation scheme [23]

To formulate an MPC problem, we need to specify the cost function $J(x,u,y)$, which to be minimized to find an optimal control, where $x$, $u$, $y$, $z$ vectors to define state, controlled inputs, measured outputs, controlled outputs respectively. But a common practice is to have y and z the same.

If model is linear controlled object's model can be described as state-space model:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_{k+1} = Cx_k \end{cases} \tag{3.1}$$

where $A$, $B$, $C$ - are state, input, output matrices respectively.

However, if model is not linear it can be described as:

$$\begin{cases} x_{k+1} = f(x_k, u_k) \\ y_{k+1} = g(x_k, u_k) \\ z_{k+1} = h(x_k, u_k) \end{cases} \tag{3.2}$$

where $f$, $g$, $h$ - nonlinear functions

## 3.2 Related works

The lower price of energy during the periods of low demand, encourage consumers to accumulate the stored energy in the batteries to use it later. Therefore, the consumer who accumulated an energy during the low demand tries to release electricity stored in the battery to cover required consumption or even sell to the grid to gain money for that.

Several successful attempts to reduce the variability of the load of the network by the use of batteries earning some cost for that has been presented in [1], [2], [24]. This helped the power system operators to perform peak shaving supporting energy demand and in this way preserving a balance of the power grid.

Minimization problems were formulated to reduce load variance in [1], [24], but $SoH$ of the battery was not taken into account, which resulted in charging nearly to full capacity during low load and vice versa. On the other hand, in [2] the cost was used as the objective function to show how profitable is to use a battery storage for peak shaving task. As the wrong usage of the battery, where capacity fading has high rates, increases costs, $SoH$ was taken into consideration. $SoH$ consisted of the calendar and cyclic aging, however the impact from $DoD$ was neglected. As in [11] is shown, $DoD$ can have a great impact on the capacity fading, therefore further investigation is required. Moreover, it was shown that peak shaving using battery storage is profitable and can be used for providing energy to large industrial loads.

MPC (shown in [1]) is able reduce variability of energy production output for wind generators. Furthermore, stable generic controller was developed. Usage for such controllers is crucial for generators based on renewable resources as their output is highly fluctuating. Adding additional variable for optimization ($SoH$), makes the minimization problem competing between maximizing $SoH$ and minimizing peaks' height. One of the approaches to deal with it is by use of equivalence factor [6], which will penalize very large $SoH$ changes or very small peak shaving.

To reduce the variability of load, the battery will be charged and discharged in the particular moments, when the price is the most pleasant, so we can define efficient battery charging as a subproblem, which will be solved during runtime. MPC was already applied to optimal charging [3], and can be extended to use different energy prices depending on the network load. The main idea is by the use of slack variables for reference trajectory of $SoC$ and $SoH$, which can be precalculated or chosen by the user, and then solve least-squares quadratic problem of tracking those variables.

## 3.3 Implementation

The model which is able to predict the future behavior of controlled object is crucial for MPC. Generally, there are two types of models that can be used: first-principle nonlinear model and linear model obtained by system identification techniques [4]. The first one is built upon equations that describe underlying physical, chemical, and thermodynamic processes in the controlled object, and is not always feasible. The second one is built using statistical methods and is more like a black box. In this work the battery model with ageing which was developed in Chapter 2 is used.

Modelica is an equation-based modeling language that is convenient for engineers, but it doesn't have support for optimization problems by default. Despite the fact, that Optimica, a toolbox that provides an extension to Modelica model allowing to include optimization models, tries to fix this problem, there are no third-party libraries providing MPC framework, and optimization problems should be implemented from scratch. Python, on the other hand, provided multiple third-party solutions, that can simplify development [25], [26]. Thus, the decision to combine optimization tools in Python with the Python interface to access the battery model written in Modelica is the most efficient and applied in this work.

### 3.3.1 Jmodelica.org

Jmodelica is an open-souce platform, which is based on Modelica models together with optimization models. It also provides a Python package that allows interaction with models. However, since latest releases of Jmodelica stopped being open-source, this library supports only Python2, despite the End of Life announcement for it. In addition, JModelica.org stopped being open-source in 2019. An alternative open-source solution is MPCPy library [27], but it mainly focuses on problems for buildings system and has problems with extensibility for custom cost function. It works by translating Python code in JModelica, however adding new functionality in the library is not simple.

### 3.3.2 Functional mock-up interface

A functional mock-up interface (FMI) is a free standard, which can be used to integrate different third-party dynamic models and exchange them between different software tools. It is done by translating the developed model into functional mock-up unit (FMU), which can be used in different environments. In FMU all information, required to simulate the model, is stored (such as variables, equations, inputs). For example, such software products as Simulink, OpenModelica, Dymola, and JModelica.org are capable to export (or import) FMU. Moreover, the PyFMI library [28] was developed to allow interaction with FMU from Python.

### 3.3.3 Own implementation

Even though Python-based solution for MPC using FMU was developed previously [29], it is not available online. Thus, the MPC algorithm is implemented from scratch. SciPy library was used for optimization per step. During optimization model state should resets multiple times. As OpenModelica implementation does not support canGetAndSetFMUstate flag for exported FMU, which requires set specific state to the model, model is rerunned with the same inputs from the start.

## 3.4 MPC for efficient charging of the battery

The main objective of the controller is by preserving the required time for the charging, prolong the life time of the battery.

### 3.4.1 MPC problem formulation for battery charging

Based on the battery model, that was described in previous Chapter 2 next state, input and output vectors are used:

$$x = [Q_{cur}, U_{t,l}, U_{t,s}, SoC_{avg}, SoC_{dev}, N_{eff}, L]$$
$$u = [I_{bat}]$$
$$y = [SoC, SoH]$$

where $Q_{cur}$ is the current charge stored in the battery. Other variables are presented in Chapter 2.

Reference tracking problem was previously used for charging [30]. The main idea is that from given reference signal $y_r$ some state-input ($x_r$, $u_r$) is generated, and controller tries to reach those values for model. So for the horizon length of $N$, cost function at step $k$ is:

$$J = \sum_{i=0}^{N-1} \left( \|x(i) - x_r(k+i)\|_Q^2 + \|u(i) - u_r(k+i)\|_R^2 \right) + \|x(N) - x_r(k+N)\|_P^2 \quad (3.3)$$

where $x(i)$ is the state predicted by our model, and we are setting initial value $x(0)$ to $x_k$ – the state of the system at the step $k$, $\|\cdot\|_A^2$ - is squared weighted 2-norm, i.e $\|x\|_A^2 = x^T Ax$ .

Also, $P$, $Q$, $R$ are weighted matrices, that serve to archieve desired dynamic behavior by balancing tracking error against the control effort required to achieve it. The last term of the equation 3.3 is the terminal constraint, which allow to ensure stability of the battery, so each controller at each optimization step considers what happens beyond the prediction horizon [4].

However, if change rates of battery outputs satisfy the conditions:

$$0 \leq (\frac{\delta SoC_r}{\delta t})_{min} \leq (\frac{\delta SoC_r}{\delta t}) \leq (\frac{\delta SoC_r}{\delta t})_{max} \quad (3.4)$$
$$0 \leq (\frac{\delta SoH_r}{\delta t})_{min} \leq (\frac{\delta SoH_r}{\delta t}) \leq (\frac{\delta SoH_r}{\delta t})_{max} \quad (3.5)$$

and therefore reference signals have reachability (state of the battery which have such outputs is reachable), cost function can be rewritten as an output tracking problem [3]. Moreover, because of the limitation on $SoC_r$ and as $SoC$ is just the integral of input, input weight matrix $R$ can be assigned to 0, and it will not impact the stability of the system. So cost function 3.3 can be rewritten as:

$$J = \sum_{i=0}^{N-1} \left( \|y(i) - y_r(k+i)\|_Q^2 \right) + \|y(N) - y_r(k+N)\|_P^2 \quad (3.6)$$

Output weight matrix $Q$ allows us to control pursuing two distinctive objectives: maximizing $SoC$ for faster charging and maximizing $SoH$ to reduce damage to the battery. So, we can define matrix $Q$ as $\begin{bmatrix} 1 & 0 \\ 0 & w \end{bmatrix}$ , where $w$ is a penalizing factor for

$SoH$ degradation and should be chosen tuned during testing, or obtained by use of optimization algorithms. If the model is linear, and can be expressed by $A$, $B$, $C$ matrices like in 3.1, matrix $P$ can be calculated using the discrete Lyapunov equation [3]:

$$(A + BK)^T P(A + BK) - P = -(Q + K^T RK) \tag{3.7}$$

where K is stabilizing control gain such that (A+BK) is Schur. For this approach, model was linearized during each optimization step, but the time required for linearization is too big and accuracy dropped greatly (see Figure 3.3). Therefore, for the nonlinear case, it was decided to have no constraints for the system to ensure stability, and cost function has is defined as:

$$J = \sum_{i=0}^{N-1} \left( \|y(i) - y_r(k+i)\|_Q^2 \right) \tag{3.8}$$

So the only one constraint for the MPC is the input constraint, such that current $I_{req}$ should be inside bound $[-10C; 10C]$ where $C$ - C-rate.

### 3.4.2 Simulation Results

In [3] researchers proposed running optimization with $w = 0$ to get reference signal $SoC_r$, and choose $SoH_r$ considerin application. So to get $SoH_r$, the optimization was run with the limit on $I_{req} \in [-2C; 2C]$. Thus, during the optimization, the MPC controller will be balancing between faster charging and minimization battery damage.

During simulations, three different scenarios were possible depending on the value of $w$. The first one is if $w$ is too small, controller is just ignoring changes in $SoH$ and tries to charge as fast as possible. The second one is when $w$ is too big, and the simplest solution is not to charge the battery at all, as a change of $SoC$ level has a high cost for changes in $SoH$. The last one is actually where changes in $SoC$ and $SoH$ are competing and depending on $w$ faster charging of lower damage can be achieved. From the last scenario, suboptimal $w$ equal to 3000 was taken, as optimization of full charge usually takes from 15 to 30 minutes.

MPC with linearization, where at each optimization step battery model was linearized and optimization was performed using a linear model, was implemented. However, linearization takes a lot of computational time, therefore, the optimization with linearization worked twice slower than MPC with nonlinear model. In addition, small changes in $SoH$ were often not visible in the linear model. Thus, we can see on 3.3 that the charging of the battery was performed with a high C-rate in the start, where capacity fading per step was small, leading to the drastic drop of $SoH$.

In contrary, we can see on 3.3, that the MPC controller without linearization 3.8 is capable of efficient battery charging, even with the same charging time as with a constant 2.5$C$ rate to perform an optimal charging to have less damage to the battery.

## 3.5 MPC to reduce variability of load

The main objective of the controller is to minimize total amount of money required to spend on the energy. This objective is justified by the fact that consuming

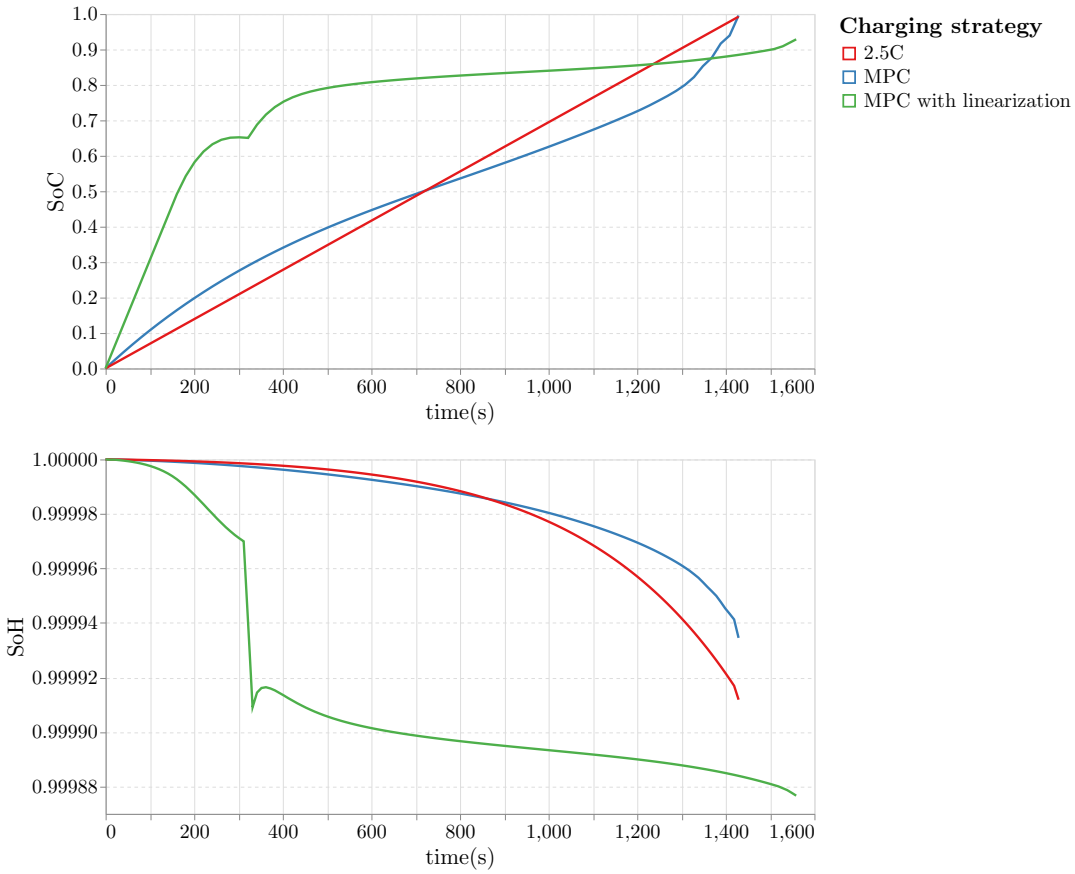



FIGURE 3.3: Comparison of different charging strategies

the portion of load during peak demand is typically much more expensive than consuming energy during minimum load demand. For this task it is assumed that the electricity price per GWh is given by the following equation:

$$Cost[USD/GWh] = 5 + 0.5 * P_{load}[GW] + 0.05 * P_{load}^2[GW] \tag{3.9}$$

where $P_{load}$ is the current load on the network. The load profile of an exemplary country (Sweden) is available during 13 years (2005 - 2017) with hourly resolution expressed in gigawatts [GW], and therefore will be used as $P_{load}$. One week from December 2017 will be used, and the load profile can be seen on the Figure 3.4.

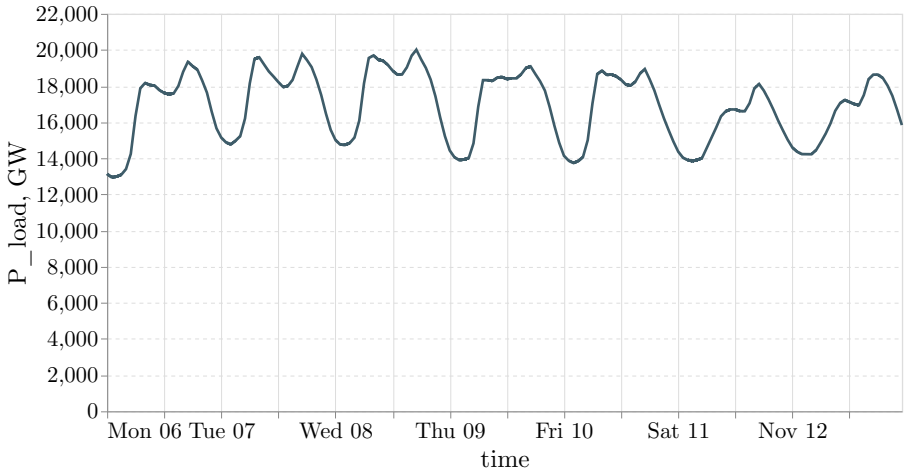



FIGURE 3.4: The load profile of Swedish grid in December 2017

The battery model from Chapter 2 was used for prediction of the battery outputs, such as $SoC$, $SoH$ and $U_{bat}$. But as the battery parameters were used for A123 APR18650M1, which has capacity only 1.1 $Ah$ [20], it will be assumed, that multiple of batteries are used together synchronously, so output from the battery will be scaled:

1. Total capacity of the batteries is 800GWh
2. Maximum charging/discharging power of batteries is 200GW

### 3.5.1 MPC problem formulation for reducing variability of load

Similarly to controller for efficient charging presented in 3.4, MPC controller for reducing variability of load is also formulated, as output tracking problem, but for amount of money for required amount of the energy. Therefore, controller has a reference the amount of money $M_r$ which should be paid for the each optimization step. Thus, cost function $J$ for step $k$ :

$$J = \sum_{i=0}^{H_c} \left\| Cost \left( \int_{t_{k+i}}^{t_{k+i+1}} (P_{load} - P_{bat,i}) \Delta t \right) - M_r(k+i) \right\|^2 \tag{3.10}$$

where

$H_c$ – controlling horizon
$k$ – a number of step during MPC optimization
$M_r$ – reference amount of money is USD
$Q_{us}$ – sum of usable capacities of all batteries
$P_{load}$ – power of the load network
$P_{bat,i}$ – power of the battery
$\Delta t$ – time inverval of optimization step

However, as it can be seen on Figure 3.4, that periods of low and high demands has big gap between each other in a few hours. Previously researchers in [31] were using 10 minutes as length of an optimization step for MPC. Therefore, this approach would be needed to have controlling horizon $H_c$ equal to 36, to be able to take into account next 6 hours. As optimization for such long period will take a lot of computational power and time, it was decided to have not equal prediction and controlling horizons. For steps greater than prediction horizon $H_p$, it is impossible to know power of the battery, as current with 0 $A$ is given as input to the model for non controllable region. Therefore, for the interval $[H_c; H_p]$, parameter $k_i$ is introduced to specify how much energy we expect to use from the battery in the time interval $[H_c; H_p]$. The energy of the battery can be used in constant amounts for each iteration, for even distribution of energy per step, or depending on the amount of load per step.

Since the goal is that the batteries should be charged efficiently, for this problem we can use the same approach as in Section 3.4, by providing reference signal $SoH_r$ and penalizing factor $w$ for $SoH$ degradation.

Thus, cost function $J$ can be rewrittten as:

$$\begin{aligned} J = &\sum_{i=0}^{H_c} \left\| Cost \left( \int_{t_{k+i}}^{t_{k+i+1}} (P_{load} - P_{bat,i}) \Delta t \right) - M_r(k+i) \right\|^2 + \\ &w \cdot \sum_{i=0}^{H_c} \|SoH - SoH_r\|^2 + \\ &\sum_{i=H_c}^{H_p} \left\| Cost \left( \int_{t_{k+i}}^{t_{k+i+1}} (P_{load} \Delta t) - k_i \cdot SoC_i \cdot Q_{us} \right) - M_r(k+i) \right\|^2 \end{aligned} \tag{3.11}$$

MPC was constrained on the input, as batteries has the maximum charging rate. Additionaly, MPC was constrained, so $SoC$ should be in the bounds $[0.2, 0.8]$ to reduce damage to the model.

### 3.5.2 Simulation results

The real measurements of the consumed power in Swedish grid in 6 December of 2017 was taken for tests. To get a reference signal $M_r$ for controller input, the load profile should be analyzed. The percentiles of 10 and 60 % of load were taken to localize periods of low and high demand that can be observed in 3.5. Reference signal $M_r$ was calculated from considering the load profile with assumption that amount of power from network should be always less then 65 % percentile and greater then 10 % percentile.

On the Figures 3.6 and 3.8 resulted power consumption from the network and the battery can be seen. Blue area corresponds to the energy that was bought from the network. Red area with power more then 0 corresponds to the time intervals when

power from batteries were taken, while negative power means that the battery was charging.

On 3.6 we can see suboptimal solution. Despite the fact that battery was charged in the period of low demand, and charged during high, most of the energy were taken not in the highest peak. The reason for it is that reference signal required the controller to decrease power consumption from network for too large period of time, and batteries were not capable of providing so much energy. To overcome this issue, different approaches can be done:

- Add weight term $w_p$ to $\sum_{i=H_c}^{H_p} \left\| Cost \left( \int_{t_{k+i}}^{t_{k+i+1}} (P_{load}\Delta t) - k_i \cdot SoC_i \cdot Q_{us} \right) - M_r(k+i) \right\|^2$ in 3.11, so difference between reference and predicted price in the interval $[H_c; H_p]$ will have more impact
- Provide more feasible reference signal into the controller.

Optimization run for reducing variability of 24 hours required nearly 10 hours of computations, so tuning parameter requires a lot of time, so it more feasible reference signal was taken. So upper bound for amount of load taken from network was increased from 65 % percentile to 75%, and result of such change on optimization can be seen in Figure 3.8.

Also, we can see on 3.6 that near 3 am, there was a slight variation, where battery after charging consistent charging, discharged with a small current. By further examination of the battery state, it was discovered that $SoC$ constraint was violated, so the battery has $SoC >= 0.8$, and therefore started discharging.

Overall, the controller is capable of reducing variability of load (see Figure 3.8). Even though such controller can be used for real-time operations, there is a big amount of required computations, therefore the computational improvement can be reached when the model of the battery is simplified.

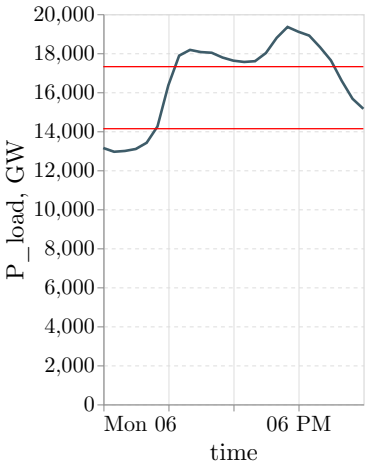



FIGURE 3.5: 10 and 60 percentiles

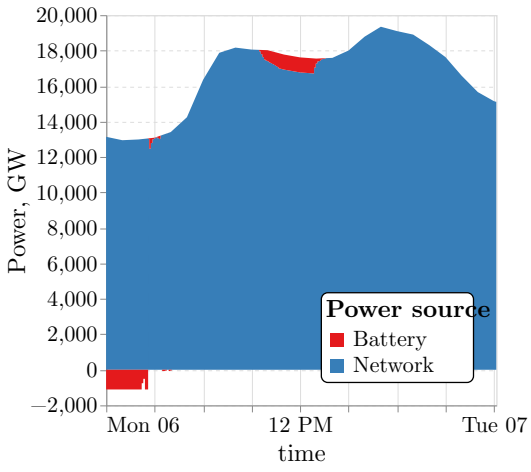



FIGURE 3.6: Distribution between power sources

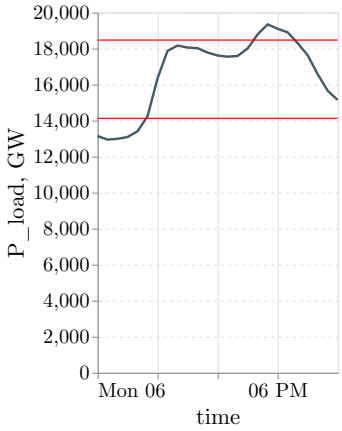



FIGURE 3.7: 10 and 75 quantiles

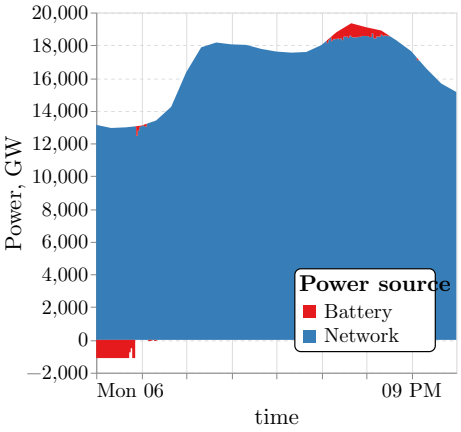



FIGURE 3.8: Distribution between power sources

# Chapter 4

# Results and summary

## 4.1 Results

The results of this work show a lot of perspective of usage of Model Predictive Control for battery operation. The battery model with ageing effects was developed, and later used in several MPC controllers. The first MPC controller was able to perform efficient charging of the battery, and the second one reducing variability of load by performing peak shaving.

Thus, the research contributions:

- The battery model with ageing effects implemented in Modelica, which can be used in any FMI-compliant software, and validated against the references in the scientific literature including the datasheets that are provided by the battery manufacturer
- The Python framework for MPC using FMU models is developed.
- MPC controller for efficient charging of the battery considering state of battery health with maximizing the speed of charge or discharge of the battery is developed and validated
- MPC controller for reducing variability of load aiming for minimum expenses for battery operation in terms of price of electricity is developed and validated using the real data of the load in Swedish grid

## 4.2 Future work

Based on the presented results in this work, the following improvements can be made:

- Use other than SciPy library for solving optimization problem, to speed up optimization process
- Add support of FMU canGetAndSetFMUstate for MPC Python framework flag to be able reset model without running from scratch, if model was exported using Dymola environment
- Add estimation of the state of the model for MPC Python framework, the battery model used in MPC is not identical to real battery. So if developed controllers will be used in real time systems, state of the real battery should be estimated to reduce errors.
- Add forecasting of the load, so the controller for reducing variability of load can be run on unseen data

# Appendix A

# Battery model parameters

TABLE A.1: Battery model parameters

| Parameter | Charging | Discharging |
|---|---|---|
| $R_s$ | | |
| $R_{s0}$ | 8.98e-2, | 8.210e-2 |
| $k_{R_s,1}$ | -7.216e-2 | -4.1006e-2 |
| $k_{R_s,2}$ | 2.273e-1 | 1.609e-1 |
| $k_{R_s,3}$ | -2.892e-1 | -2.518e-1 |
| $k_{R_s,4}$ | 1.298e-1 | 1.369e-1 |
| $R_{t,s}$ | | |
| $R_{t,s0}$ | 1.827e-2 | 1.4e-2 |
| $k_{R_{t,s1}}$ | 1.080e-2 | 7.13e-11 |
| $k_{R_{t,s2}}$ | 11.03 | -21.11 |
| $k_{R_{t,s3}}$ | -6.463e-3 | 03 |
| $R_{t,l}$ | | |
| $R_{t,l0}$ | 4.722e-2 | 3.1e-2 |
| $k_{R_{t,l1}}$ | 2.95e-1 | 8.913e-15 |
| $k_{R_{t,l2}}$ | 20.00 | -32.23 |
| $k_{R_{t,l3}}$ | -2.420e-2 | 4.473e-3 |
| $C_{t,s}$ | | |
| $k_{C_{t,s0}}$ | 389.7 | 6.849e2 |
| $k_{C_{t,s1}}$ | 1408 | 2.340e3 |
| $k_{C_{t,s2}}$ | -1007 | -1.013e4 |
| $k_{C_{t,s3}}$ | 169.7 | 1.723e4 |
| $k_{C_{t,s4}}$ | 0 | -1.026e4 |
| $k_{C_{t,s5}}$ | 0 | 0 |
| $k_{C_{t,s6}}$ | 0 | 0 |
| $C_{t,l}$ | | |
| $k_{C_{t,l0}}$ | 2.232e3 | 7.144e3 |
| $k_{C_{t,l1}}$ | -3.102e4 | 2.283e4 |
| $k_{C_{t,l2}}$ | 5.998e5 | -8.124e4 |
| $k_{C_{t,l3}}$ | -2.958e6 | -4.009e3 |
| $k_{C_{t,l4}}$ | 6.271e6 | 2.042e5 |
| $k_{C_{t,l5}}$ | -6.007e6 | -1.541e5 |
| $k_{C_{t,l6}}$ | 2.130e6 | 0 |

TABLE A.2: circuit components

| | |
|---|---|
| $K_{co}$ | 3.66e-5 |
| $K_{ex}$ | 0.717 |
| $K_{soc}$ | 0.916 |
| | |

TABLE A.3: damage accumulation model

# Bibliography


[1] M. Khalid and A.v. Savkin. "A model predictive control approach to the problem of wind power smoothing with controlled battery storage". In: *Renewable Energy* 35.7 (2010), 1520–1526. DOI: 10.1016/j.renene.2009.11.030.

[2] Rodrigo Martins et al. "Optimal Component Sizing for Peak Shaving in Battery Energy Storage System for Industrial Applications". In: *Energies* 11.8 (2018), p. 2048. DOI: 10.3390/en11082048.

[3] Changfu Zou, Chris Manzie, and Dragan Nesic. "Model Predictive Control for Lithium-Ion Battery Optimal Charging". In: *IEEE/ASME Transactions on Mechatronics* 23.2 (2018), 947–957. DOI: 10.1109/tmech.2018.2798930.

[4] Jan Marian Maciejowski. *Predictive control: with constraints*. Prentice Hall, 2008.

[5] *How to choose the best battery for a solar energy system*. URL: https://www.energysage.com/solar/solar-energy-storage/what-are-the-best-batteries-for-solar-panels/.

[6] A. Sciarretta, M. Back, and L. Guzzella. "Optimal Control of Parallel Hybrid Electric Vehicles". In: *IEEE Transactions on Control Systems Technology* 12.3 (2004), 352–363. DOI: 10.1109/tcst.2004.824312.

[7] Mischa. *How a battery works*. 2018. URL: https://www.science.org.au/curious/technology-future/batteries.

[8] *ANR26650M1 Datasheet*. URL: https://altitec.no/media/pdf/ANR26650M1.pdf.

[9] Lady Ada. *Li-Ion LiPoly Batteries*. URL: https://learn.adafruit.com/li-ion-and-lipoly-batteries.

[10] *Learn About Batteries*. URL: https://batteryuniversity.com/learn/.

[11] Languang Lu et al. "A review on the key issues for lithium-ion battery management in electric vehicles". In: *Journal of Power Sources* 226 (2013), 272–288. DOI: 10.1016/j.jpowsour.2012.10.060.

[12] M. Chen and G.A. Rincon-Mora. "Accurate Electrical Battery Model Capable of Predicting Runtime and I–V Performance". In: *IEEE Transactions on Energy Conversion* 21.2 (2006), 504–511. DOI: 10.1109/tec.2006.874229.

[13] Alan Millner. "Modeling Lithium Ion battery degradation in electric vehicles". In: *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply* (2010). DOI: 10.1109/citres.2010.5619782.

[14] Jun-Hyeok Kim et al. "Modeling of Battery for EV using EMTP/ATPDraw". In: *Journal of Electrical Engineering and Technology* 9.1 (2014), 98–105. DOI: 10.5370/jeet.2014.9.1.098.

[15] Richard L. Hartmann. "An aging model for lithium-ion cells". PhD thesis. 2008.

[16] Hartmut Hinz. "Comparison of Lithium-Ion Battery Models for Simulating Storage Systems in Distributed Power Generation". In: *Inventions* 4.3 (2019), p. 41. DOI: 10.3390/inventions4030041.

[17] V. Marano et al. "Lithium-ion batteries life estimation for plug-in hybrid electric vehicles". In: *2009 IEEE Vehicle Power and Propulsion Conference* (2009). DOI: 10.1109/vppc.2009.5289803.

[18] Ari Hentunen, Teemu Lehmuspelto, and Jussi Suomela. "Time-Domain Parameter Extraction Method for Thévenin-Equivalent Circuit Battery Models". In: *IEEE Transactions on Energy Conversion* 29.3 (2014), 558–566. DOI: 10.1109/tec.2014.2318205.

[19] Suleiman Abu-Sharkh and Dennis Doerffel. "Rapid test and non-linear model characterisation of solid-state lithium-ion batteries". In: *Journal of Power Sources* 130.1-2 (2004), 266–274. DOI: 10.1016/j.jpowsour.2003.12.001.

[20] *APR18650M1 Datasheet*. URL: https://www.batteryspace.com/prod-specs/6612.pdf.

[21] M. F. Golnaraghi and Kuo Benjamin C. *Automatic control systems*. McGraw-Hill Education, 2017.

[22] Danijel Pavkovic et al. "Battery current and voltage control system design with charging application". In: *2014 IEEE Conference on Control Applications (CCA)* (2014). DOI: 10.1109/cca.2014.6981481.

[23] Martin Behrendt. *A discrete MPC scheme*. 2009. URL: https://en.wikipedia.org/wiki/Model_predictive_control#/media/File:MPC_scheme_basic.svg.

[24] Yichen Zhang et al. "Battery energy storage scheduling for optimal load variance minimization". In: *2018 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)* (2018). DOI: 10.1109/isgt.2018.8403324.

[25] Python-control. *Python Control Systems Library*. URL: https://github.com/python-control/python-control.

[26] Sdu-Cfei. *Python library: ModestPy*. URL: https://github.com/sdu-cfei/modest-py.

[27] Lisa Rivalin. "MPCPy: An Open-Source Software Platform for Model Predictive Control in Buildings". In: (May 2017).

[28] Modelon AB. *PyFMI*. URL: https://github.com/modelon-community/PyFMI.

[29] Alfred Spiessens Lieve Helsen Javier Arroyo Bram van der Heijde. "A Python-Based Toolbox for Model Predictive Control Applied toBuildings". In: (July 2018).

[30] D. Limon et al. "MPC for tracking piecewise constant references for constrained linear systems". In: *Automatica* 44.9 (2008), 2382–2387. DOI: 10.1016/j.automatica.2008.01.023.

[31] Deepranjan Dongol, Thomas Feldmann, and Elmar Bollin. "A Model Predictive Control based Peak Shaving Application for a Grid Connected Household with Photovoltaic and Battery Storage". In: *Proceedings of the 7th International Conference on Smart Cities and Green ICT Systems* (2018). DOI: 10.5220/0006685300540063.