Daniela Dejaco

IDENTIFICATION, DIAGNOSIS AND CONTROL OF A THERMAL STORAGE SYSTEM

Graz University of Technology

Institute of Automation and Control

Supervisors:

Martin Horn, Univ.-Prof. Dipl.-Ing. Dr. techn.

Elio Usai, University of Cagliari

Graz, January 2016

Contents

Introduction

Energy storage will play a key role in developing an electricity system based on renewable energy. Generally, renewable energy systems are vitiated by the variable availability of the resources. The problem is that rises in energy generation do not necessarily correspond to rises in demand and vice versa. Therefore, energy storage is essential to balance supply and demand, as it can grant a back-up to intermittent renewable energy.
The efficient usage of energy storage systems in electricity distribution networks could significantly reduce costs and stabilize market prices. Furthermore, it could improve the security and efficiency of electricity transmission.

Currently, 99% of the worldwide storage capacity is represented by Pumped Hydro Storage Systems [14]. This form of energy storage was particularly attractive when the electricity network was composed of distributed grids with weak interconnection.
Nowadays other forms of storage, for instance batteries or chemical storage, are required to maximize the efficiency of a highly interconnected network. Unfortunately, their usage is still minimal or at an early stage of development.

The Thermal Storage System treated in this master thesis offers a relatively cheap way of storing energy as it allows to store the thermal energy itself. It is particularly interesting for solar thermal power plants. In this case the Thermal Storage System must provide energy when the solar radiation is not capable to fulfill the demand.
A huge disadvantage of this system is that it is not possible to exploit its entire capacity and the heat stored in it cannot be fully recovered. In this master thesis, we want to cope with this inconveniences by controlling the system.

As a start, in chapter **1** the concept of Thermal Storage System with packed bed is exposed. It is a thermally isolated cylinder containing a crushed material with a high thermal capacity. A fluid flowing through the cylinder exchanges thermal energy with the crushed material, where it is then stored to shift its delivery. A prototype of such a system was built in the course of the ESTATE lab project at the University of Cagliari. This laboratory plant will be the system to be controlled in this master thesis.
The main problem of this specific storage concept is that it's not possible to exploit the entire capacity during the charging phase. Furthermore, only a fraction of the energy stored in the cylinder can be later extracted. The principle aim of this master thesis is to optimize the charging efficiency by means of control.

In chapter **2** a mathematical model describing the charging dinamics of the Thermal Storage System is derived. The partial differential equations obtained cannot be solved analytically and must therefore be solved numerically. A discrete-time model is obtained as well as a continuous-time model. The models are then validated on the laboratory plant.

In order to understand the dynamics of the system, in chapter **3** some open-loop experiments are held on the numerical models. The results show that the charging efficiency can be improved choosing appropriate input trends.
Moreover, the desired behavior of the controlled system is outlined. Therefore, control variables and system outputs are specified.

Chapters **4**, **5** and **6** deal with control methods to improve the charging performance of the Thermal Storage System.
In chapter **4** the system is modeled as as a chain of first-order stable systems. The system is then controlled by a PI controller with variable-time Smith predictor, where the feedback variable is obtained by switching between the various outputs of the chain.
To control the system by means of Model Predictive Control, in chapter **5**, the system is identified as a nonlinear ARMAX-model. To do so, for various constant inputs, linear ARMAX models are identified. Afterwords, the local linear models are combined to a global nonlinear model. In appendix **A** a Monte-Carlo method was performed to describe the probability for the model to be stable using this approach.

In chapter **6** the system is controlled by a Fuzzy controller. As there is only one input that must control more than one output, the input is obtained as a weighted average over various Fuzzy controller outputs.

Finally, the achievements are summarized in the conclusive chapter.

Chapter 1

System description

The scientific name of the storage system to be controlled in this thesis is Packed Bed Energy Storage System. It is principally meant for storing thermal energy of solar thermal power plants. As in this case there is no need to convert the energy, the losses incurred by energy transformation can be prevented. It is a relatively cheap method of storing the energy, while for example storing electricity is much more expensive. [5]
This storage system is mostly applied for short-therm energy storage, for instance to smooth out the plant output during intermittently cloudy weather conditions. Hence, times of mismatch between energy supply by the sun and energy demand can be reduced.

1.1 General operation mode

The mode of operation of this specific storage system is very simple and is shown in figure **1.1**. The Thermal Storage System consist of a vertically placed thermally isolated cylinder. The entire volume of the cylinder is filled with small balls with a high thermal capacity. As there is a significant void fraction between the balls, they form a porous packed bed. If there is more thermal energy available than needed, a hot liquid is introduced into the top of the cylinder and it gradually conveys the heat to the balls. While the upper part of the cylinder is heated, the lower part still has the initial temperature of the balls. This results in a temperature gradient through the coil, which is called thermocline. Therefore, the fluid that exits from the

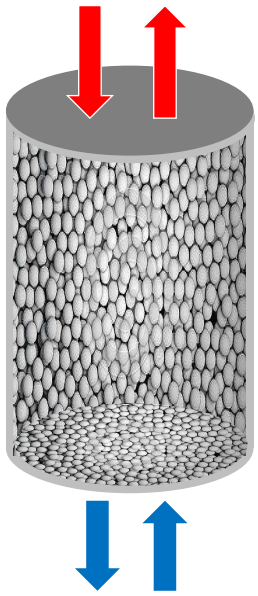

Figure 1.1: Scheme of the energy exchange in the Thermal Storage System

bottom of the cylinder is cold.
As soon as energy is demanded, for example because the sun radiation is limited or non-existing, a cold liquid is sent through the bottom of the cylinder. While flowing through the coil, the liquid absorbs the thermal energy stored in the balls and the heated liquid exits from the top of the cylinder.

1.2 Design criteria

The selection of the appropriate system is a compromise between cost and benefit. The cost of such a system mainly depends on the storage material itself, the heat exchanger for charging and discharging the system and the cost for the space and enclosure for the Thermal Storage System.
The key issue of the design of such a thermal storage system is the amount of energy that it can store and provide, which mainly depends on the choice of the material of the packed bed. It must have a very high thermal capacity, but must be resistant to high temperatures. Other crucial requirements on

the system behavior are: [5]

- Good heat transfer from fluid to the storage medium
- The charging and discharging procedure should be completely reversible for a high number of cycles
- The thermal losses should be negligible
- The system should be easy to control

1.3 Laboratory plant

In the course of the ESTATE-LAB project, a prototype of a Packed Bed Energy Storage System was built according to the design standards in [2]. The project is founded by the MIUR (the Italian Ministry of Education, Universities and Research) and its overall aim is to develop technologies for solar thermal power plants. It is particularly interesting for Sardinia, where the average yearly sun radiation is $1800\,\mathbf{kWh/m^2}$. [2]
The ESTATE-LAB project involves the following institutions: [8]

- Project coordinator: CRS4
- Sardegna Ricerche
- University of Cagliari: Institute of Mechanical Engineering and Institute of Electrical and Electronical Engineering
- R.T.M S.p.A.
- Sapio S.r.l.

The laboratory plant, which is shown in figure **1.2**, is located at the Campus of Monserrato in Cagliari.
The space in the laboratory occupied by the system is approximately $15\,\mathbf{m^2}$, while it's height is almost $4\,\mathbf{m}$. The cylinder has a diameter of approximately $60\,\mathbf{cm}$ and is $1.8\,\mathbf{m}$ tall. To minimize the thermal losses, the cylinder is enclosed by a $10\,\mathbf{cm}$ thick insulation wall, which is composed of stone wool with a protecting galvanized steel laminate.
The upper and lower end of the cylinder are delimited by very dense grids,

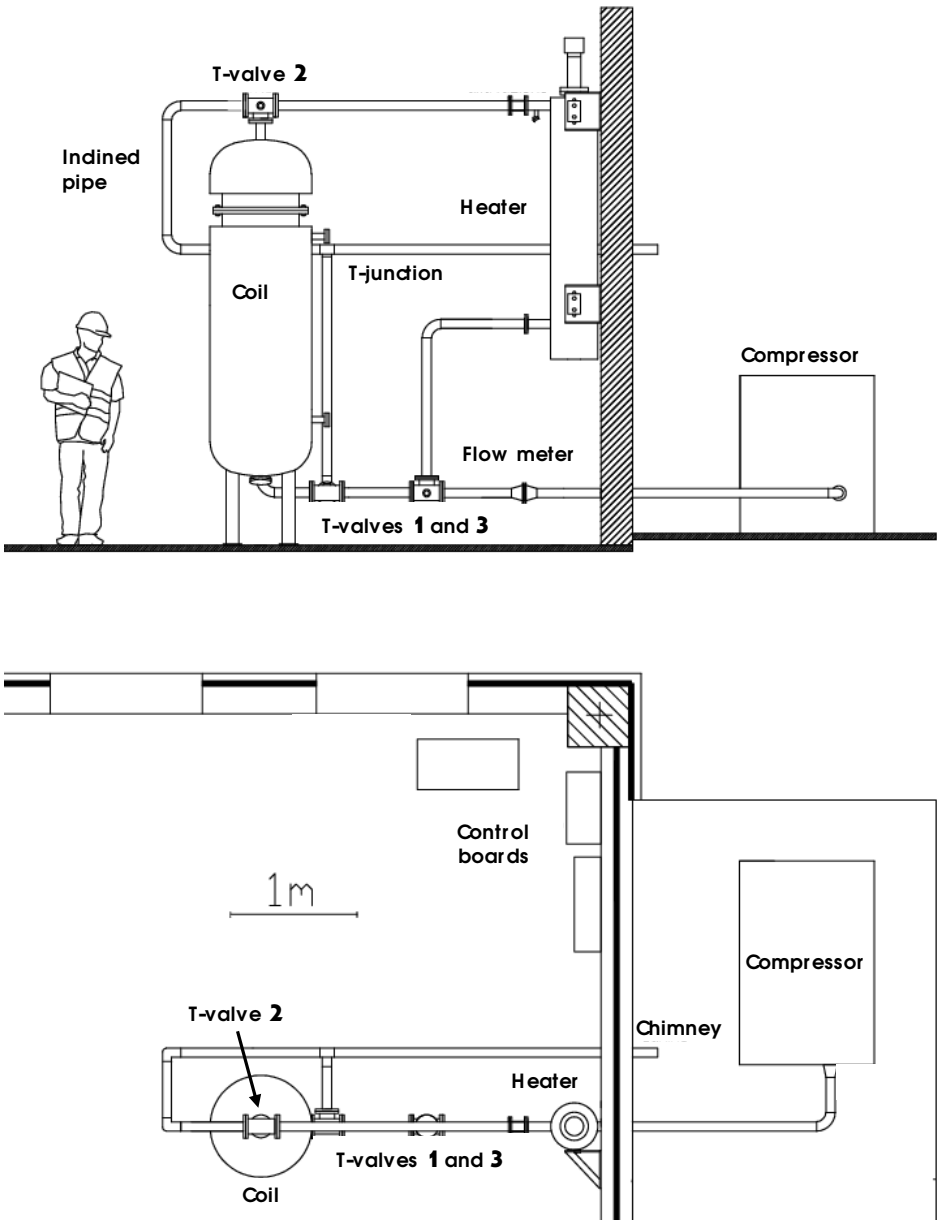

Figure 1.2: Laboratory plant at the Campus of Monserrato, Cagliari

Figure 1.3: SUPPORBIT: aluminum oxide balls

which prevent the balls from obstructing the pipes. Adjacent to the grid, diffusers are placed in order to spread the mass flow coming from the pipes to the entire section of the cylinder.

The storage material selected is called SUPPORBIT and is shown in picture **1.3**. It consists of small balls mostly composed of aluminum oxide. The chemical and thermophysical properties will be treated extensively in section **2.2**.

During the charging phase, a compressor, which is placed outdoors, pumps a specific mass flow into a pipe. There it is measured by a flow meter. The T-valve 3 redirects the flow to the heater, where the flow absorbs thermal energy and reaches a set temperature. The hot flow is then sent to the T-valve 2 which redirects it to the top of the thermally isolated cylinder. The cold flow that exits from the bottom of the coil is driven from the T-valve 1 to the chimney, which releases the fluid outdoors.

For discharging the system, the T-valves 1 and 3 are inverted. The compressor pumps the mass flow into the pipe, but instead of reaching the heater, it goes straight to the bottom of the cylinder. The hot flow that comes out of the upper part of the coil, is driven back to the heater through the T-valve 2. The same thermal energy, which was used to charge the coil, is now available.

In the next chapter, the thermophysical properties of the materials will be discussed in detail and the mathematical model of the Thermal Storage System is derived.

1.4 Performance

Considering the operation mode and the thermophysical properties exposed in the next chapter, the maximal energy capacity of this specific Thermal Storage System is

$$E_{max} = 37\,\mathbf{kWh}.^{1} \qquad (\mathbf{1.1})$$

To store this amount of energy in the cylinder, the entirety of the balls contained in it must be at the temperature of the hot fluid. As the axial direction of the coil has a temperature gradient, this can only happen if we allow that energy is wasted through the fluid that exits from the bottom of the cylinder.
On the other hand, if we stop the charging procedure as soon as the exit flow experiences some temperature change, only a minor percentage of the capacity can be exploited. Similarly, while discharging the system, we don't allow the exit flow to be colder than the fluid used to charge the coil. This reduces once more the energy that is available after one cycle.
If we perform more charging-discharging cycles on the system, these percentages decrease gradually. After a certain number of cycles the efficiency of the system is close to zero. In [2] the system is submitted to 10 consecutive cycles using a constant mass flow. During the first charging procedure, 62.19% of the cylinder can be charged. Unfortunately, it is only possible to recover 38.15% of the stored energy, that is 23.73% of the entire storage capacity. After 10 cycles, only 0.87% of the capacity is available to store energy and only the amount of energy corresponding to 0.8% of the entire storage ca-

[1] This data arises from:

$$E_{max} = C_{V,b}(T_{hot}) \cdot (T_{hot} - T_{cold}) \cdot V_b \cdot \rho_b$$

$$V_b = \epsilon \left(\frac{D}{2}\right)^2 \pi L$$

with all the parameters reported in **2.2**.

pacity can be recovered.

The ultimate aim of this master thesis is to improve the efficiency of the Thermal Storage System by means of control. In this case, only the charging phase of the system will be discussed.

Chapter 2

Modeling

In this chapter a mathematical model for the Thermal Storage System (TSS) is described. As there does not exist an analytical solution to the model equations, the simulation of the system will be based on a numerical calculation. The simulation results will be then compared to laboratory measurements.

2.1 Mathematical model of the TSS

As previously mentioned, the Thermal Storage System consists of a thermally isolated cylinder filled with small balls with a high thermal capacity. A hot liquid flows through the cylinder and conveys the thermal energy to the balls in order to be stored. As soon as the energy is requested, a cooler liquid is sent through the coil so that it absorbs the heat stored in the balls.
The mathematical model used is based on Schumann's equations for heat transfer in a porous prism [7]. It is a so-called LTNE (Local Thermal Non-Equilibrium model), which means that the temperature of the packed bed and of the fluid are considered to be unequal at each point. The problem is to find the distribution of the temperature of the packed bed and of the fluid for all time. To do so we assume that:

- The temperature gradient in radial direction is zero at each time
- The cylinder is perfectly thermally isolated
- The balls are small enough or have a sufficiently high thermal diffusivity to assure that each ball is at a uniform temperature at each time instant

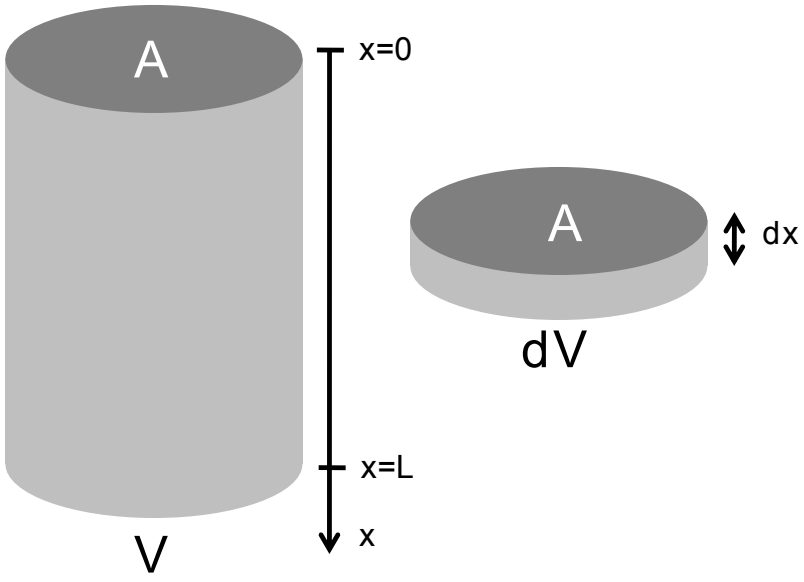

Figure 2.1: The axial direction x of the volume V is divided into infinitesimal $\mathrm{d}x$

- The conduction in the fluid itself and in the solid itself can be neglected with respect to the transfer of heat from fluid to solid
- The heat transfer rate from fluid to solid at any point is proportional to the average difference in temperature between fluid and solid at that point
- Thermal expansion can be neglected in both fluid and solid

As a start, the axial direction of the coil is divided into infinite disks as shown in figure **2.1**. Each disk has a volume equal to $\mathrm{d}V = A \cdot \mathrm{d}x$, where A is the base area of the cylinder and $\mathrm{d}x$ is an infinitesimal part of the height of the cylinder. Considering that x is the direction of the flow and $x = 0$ is set to the base of the cylinder, for each disk the law of conservation of energy can be written for both fluid and solid: [6]

$$\dot{q}_f = \rho_f \cdot A \cdot \mathrm{d}x \cdot \epsilon \cdot C_{V,f} \cdot \frac{\partial T_f}{\partial t} + \dot{m}_f \cdot C_{P,f} \cdot \frac{\partial T_f}{\partial x} \cdot \mathrm{d}x, \tag{2.1}$$

$$\dot{q}_b = \rho_b \cdot A \cdot \mathrm{d}x \cdot (1 - \epsilon) \cdot C_{V,b} \cdot \frac{\partial T_b}{\partial t}, \tag{2.2}$$

where

- the subscript b refers to the bed, hence the solid and the subscript f refers to the fluid
- $\dot{q}$ in $[\mathbf{W}]$ is the power exchanged between liquid and solid
- ρ in $[\mathbf{kg/m^3}]$ is the volumetric mass density
- A in $[\mathbf{m^2}]$ is the base of the cylinder
- $\epsilon \in [0, 1]$ is the porosity of the cylinder, i.e. the percentage of cylinder filled by air.
- C_V in $[\mathbf{J}/(\mathbf{kg} \cdot \mathbf{K})]$ is the specific heat at constant volume
- C_P in $[\mathbf{J}/(\mathbf{kg} \cdot \mathbf{K})]$ is the specific heat at constant pressure
- T in $[\mathbf{K}]$ is the temperature
- $\dot{m}_f$ in $[\mathbf{kg/s}]$ is the mass flow
- t in $[\mathbf{s}]$ is the time
- x in $[\mathbf{m}]$ is the distance in the axial direction

As the cylinder is thermally isolated it is obvious that power can only be exchanged between fluid and solid and the only supply of power is the mass flow $\dot{m}_f$.
Alternatively, the power exchange between liquid and solid can be written in terms of heat transfer:

$$\dot{q}_f = h \cdot N \cdot A_s(T_b - T_f), \tag{2.3}$$

$$\dot{q}_b = h \cdot N \cdot A_s(T_f - T_b), \tag{2.4}$$

where

- the subscript s refers to a single ball, hence A_s in $[\mathbf{m^2}]$ is the surface of a single ball
- h in $[\mathbf{W}/(\mathbf{m^2} \cdot \mathbf{K})]$ is the heat transfer coefficient
- N is the number of balls contained in the volume $\mathrm{d}V$

Notice that the power exchange is proportional to the temperature gap between liquid and solid in each point.
Combining (**2.1**) with (**2.3**) and (**2.2**) with (**2.4**) we get

$$\rho_f \cdot A \cdot \mathrm{d}x \cdot \epsilon \cdot C_{V,f} \cdot \frac{\partial T_f}{\partial t} + \dot{m}_f \cdot C_{P,f} \cdot \frac{\partial T_f}{\partial x} \cdot \mathrm{d}x = h \cdot N \cdot A_s (T_b - T_f), \quad (\mathbf{2.5})$$

$$\rho_b \cdot A \cdot \mathrm{d}x \cdot (1-\epsilon) \cdot C_{V,b} \cdot \frac{\partial T_b}{\partial t} = h \cdot N \cdot A_s (T_f - T_b). \quad (\mathbf{2.6})$$

Dividing both equations by $\mathrm{d}V = A \cdot \mathrm{d}x$ yields

$$\rho_f \cdot \epsilon \cdot C_{V,f} \cdot \frac{\partial T_f}{\partial t} + \dot{m}_f \cdot C_{P,f} \cdot \frac{\partial T_f}{\partial x} \frac{1}{A} = \underbrace{\frac{h \cdot N \cdot A_s}{\mathrm{d}V}}_{h_V} (T_b - T_f), \quad (\mathbf{2.7})$$

$$\rho_b \cdot (1-\epsilon) \cdot C_{V,b} \cdot \frac{\partial T_b}{\partial t} = \underbrace{\frac{h \cdot N \cdot A_S}{\mathrm{d}V}}_{h_V} (T_f - T_b). \quad (\mathbf{2.8})$$

Here h_V in $[\mathbf{W}/(\mathbf{m^3} \cdot \mathbf{K})]$ is the heat transfer coefficient per unit volume, which can be reduced to:

$$h_V = \frac{hNA_S}{\mathrm{d}V} \overset{(*)}{=} \frac{hNA_S(1-\epsilon)}{NV_S} \overset{(**)}{=} \frac{4\pi(d_S/2)^2}{\frac{4}{3}\pi(d_S/2)^3} h(1-\epsilon) = \frac{6h(1-\epsilon)}{d_S}, \quad (\mathbf{2.9})$$

where d_S in $[\mathbf{m}]$ is the average diameter of the balls. The term N has disappeared from the formula; this is essential as it's not possible to "count" the balls contained in an infinitesimal volume.
In (**2.9**) step (*) can be achieved considering that one single ball has a volume of V_S and that there are N balls in the volume V, hence

$$V_S \cdot N = \mathrm{d}V \cdot (1-\epsilon) \rightarrow \mathrm{d}V = \frac{V_S \cdot N}{(1-\epsilon)}. \quad (\mathbf{2.10})$$

Step (**) instead uses the formula for the volume and the surface of a ball with diameter d_S and radius r_S.[1]

The main difficulty in calculating h_V as shown in (**2.9**) is to specify the heat transfer coefficient h. To do so, one of the various empirical correlations

[1] $V_S = \frac{4}{3}\pi r_S^3$, $A_S = 4\pi r_S^2$

between the dimensionless Nusselt, Reynold and Prandtl numbers will be used.
The Reynolds number is defined as the ratio of momentum forces to viscous forces. A low Reynolds number is characteristic for smooth and constant flow motion. Turbulent flow with eddies and vortices occurs at high Reynolds numbers.
The Nusselt number is the ratio of convective to conductive heat transfer at the boundary between the solid and the fluid. A larger Nusselt number corresponds to more active convection, hence to a higher Reynolds number.
Finally, the Prandtl number defines the ratio of momentum diffusivity to thermal diffusivity.

Among the various relations between the three numbers, it is shown that for the specific case of a Thermal Storage System with packed bed, the more suitable is the following:

$$Nu_{DS} = 2 + 1.1 \cdot Pr^{1/3} \cdot Re_{DS}^{0.6}, \tag{2.11}$$

where the subscript DS refers to the diameter of the balls.
The expressions for each number are:

$$Nu_{DS} = \frac{h \cdot d_S}{k_f}, \tag{2.12}$$

$$Pr = \frac{C_{P,f} \cdot \mu_f}{k_f}, \tag{2.13}$$

$$Re_{DS} = \frac{\rho_f \cdot \upsilon \cdot d_S}{\mu_f}, \tag{2.14}$$

where

- k_f in $[\mathbf{W}/(\mathbf{m} \cdot \mathbf{K})]$ is the thermal conductivity of the fluid
- μ_f in $[\mathbf{Pa} \cdot \mathbf{s}]$ is the dynamic viscosity of the fluid
- υ in $[\mathbf{m}/\mathbf{s}]$ is the average velocity of the fluid inside the coil

We need to write the velocity υ in equation **2.14** in terms of the mass flow $\dot{m}_f$. To do so, let's define the superficial mass velocity G_f:

$$G_f := \rho_f \cdot \upsilon = \frac{\dot{m}_f}{A}. \tag{2.15}$$

Inserting (**2.12**), (**2.14**) and (**2.13**) into (**2.11**) and rearranging it, we get an expression for h that contains only known terms:

$$h = \frac{k_f}{d_S}\left(2 + 1.1 \cdot \left(\frac{C_{P,f} \cdot \mu_f}{k_f}\right)^{1/3} \cdot \left(\frac{\dot{m}_f \cdot d_S}{A \cdot \mu_f}\right)^{0.6}\right). \tag{2.16}$$

At this point, it is possible to summarize the entirety of differential equations that describe the trend of both temperatures T_f and T_b over time for an infinitesimal volume $\mathrm{d}V$:

$$\frac{\partial T_f}{\partial t} = -\frac{\dot{m}_f \cdot C_{P,f}}{\rho_f \cdot \epsilon \cdot C_{V,f} \cdot A}\frac{\partial T_f}{\partial x} + \frac{h_V}{\rho_f \cdot \epsilon \cdot C_{V,f}}(T_b - T_f), \tag{2.17}$$

$$\frac{\partial T_b}{\partial t} = \frac{h_V}{\rho_b \cdot (1-\epsilon) \cdot C_{V,b}}(T_f - T_b), \tag{2.18}$$

$$h_V = \frac{6h(1-\epsilon)}{d_S}, \tag{2.19}$$

$$h = \frac{k_f}{d_S}\left(2 + 1.1 \cdot \left(\frac{C_{P,f} \cdot \mu_f}{k_f}\right)^{1/3} \cdot \left(\frac{\dot{m}_f \cdot d_S}{A \cdot \mu_f}\right)^{0.6}\right). \tag{2.20}$$

Unluckily, this mathematical model cannot be solved analytically and therefore it is necessary to realize a numerical model for simulating the behavior of the Thermal Storage System. This will be the topic of the next sections.

2.2 Physical parameters of the model

Before creating the numerical model of the Thermal Storage System, it is important to fix the physical parameters used for both fluid and balls and the geometrical properties of the system.

Let's start with the geometrical properties of the cylinder and of the balls contained in it: [8][2]

- Length of the coil: $L = 1.8\,\mathbf{m}$
- Diameter of the coil: $D = 0.584\,\mathbf{m}$
- Average diameter of the balls $d_S = 8\,\mathbf{mm}$. (between $7\,\mathbf{mm}$ and $9\,\mathbf{mm}$)

- Porosity of the packed bed $\epsilon = 0.39$

The calculation of the porosity is quite difficult because not all the balls are the same size. Here it is based on a statistical approach as shown in [10].
In addition, there are some control specifications that must be satisfied. Due to the real system there are some restrictions on the mass flow:

- Lower limit for the mass flow: $\dot{m}_{f,MIN} = 0.7\,\mathbf{kg/s}$
- Upper limit for the mass flow: $\dot{m}_{f,MAX} = 0.26\,\mathbf{kg/s}$

The temperatures of the hot mass flow T_{hot} and of the cold mass flow T_{cold} (hence the temperature of the coil before starting the charging phase) can be chosen arbitrarily. For this master thesis the following temperatures are used:

- Hot mass flow: $T_{hot} = 237^o\mathbf{C}$
- Cold mass flow: $T_{cold} = 38^o\mathbf{C}$

Let's continue with the physical properties of the air, that is the fluid that we use for the Thermal Storage System. The choice of the material is based on the fact that it's cheap, permanently available and eco-sustainable. All thermophysical properties of the air are assumed to be dependent on the temperature T_f. Tabular **2.1** contains all the coefficients for various temperatures at atmospheric pressure ([4]). For all coefficients a polynomial fitting is done; the use of a look-up table is much slower and less efficient in *Simulink*. For the same reason the order of the polynomial should possibly be small. It is shown that for the various coefficients a polynomial of 1st or 2nd order is sufficient to achieve a good model. Figure **2.2** compares the interpolation of the values in tabular **2.1** to the polynomial fitting.
At that point, only the physical properties of the packed bed are missing. The material chosen for the balls is SUPPORBIT, which is mostly composed of aluminum oxide. See table **2.2** for the chemical composition. [8] The thermophysical properties of aluminum oxide (Al_2O_3) are used to model the SUPPORBIT ([4]). The density is assumed to be constant for all temperatures, the specific heat instead varies with the temperature (see table **2.3**). Figure **2.2** shows the data for the specific heat and the fitting with a polynomial of 2nd order.

Coefficient	$\rho_{\mathbf{f}}$	$\mathbf{C_{V,f}} \cdot 10^{-3}$	$\mathbf{C_{P,f}} \cdot 10^{-3}$	$\mathbf{k_f} \cdot 10^{2}$	$\mu_{\mathbf{f}} \cdot 10^{5}$
Unit	$[\mathbf{kg/m^3}]$	$[\mathbf{J/kg \cdot K}]$	$[\mathbf{J/kg \cdot K}]$	$[\mathbf{W/m \cdot K}]$	$[\mathbf{Pa/s}]$
1.85°C	1.2840	0.7167	1.0038	2.428	1.725
26.85°C	1.1770	0.7178	1.0049	2.624	1.846
51.85°C	1.0860	0.7192	1.0063	2.816	1.962
76.85°C	1.0090	0.7211	1.0082	3.003	2.075
101.85°C	0.9413	0.7235	1.0106	3.186	2.181
126.85°C	0.8824	0.7264	1.0135	3.365	2.286
176.85°C	0.7844	0.7335	1.0206	3.710	2.485
226.85°C	0.7060	0.7424	1.0295	4.041	2.670
276.85°C	0.6418	0.7527	1.0398	4.357	2.849
Pol. degree	2	2	2	1	1

Table 2.1: Thermophysical properties of air at various temperatures at atmospheric pressure

Material	Composition in [%]
Al_2O_3	≥ 89.5
SiO_2	≥ 6.5
$MgO + CaO$	≥ 5.2
BaO	≥ 0.6
Fe_2O_3	≥ 0.3
$Na_2O + K_2O$	≥ 0.6

Table 2.2: Chemical composition of the SUPPORBIT

Coefficient	$\mathbf{C_{V,b}} \cdot 10^{-3}$	$\rho_{\mathbf{b}}$
Unit	$[\mathbf{J/kg \cdot K}]$	$[\mathbf{kg/m^3}]$
24.9°C	0.7822	3600
26.9°C	0.7850	-
99.9°C	0.9014	-
126.9°C	0.9420	-
199.9°C	1.0168	-
226.9°C	1.0467	-
Pol. degree	2	-

Table 2.3: Thermophysical properties of aluminum oxide at various temperatures at atmospheric pressure

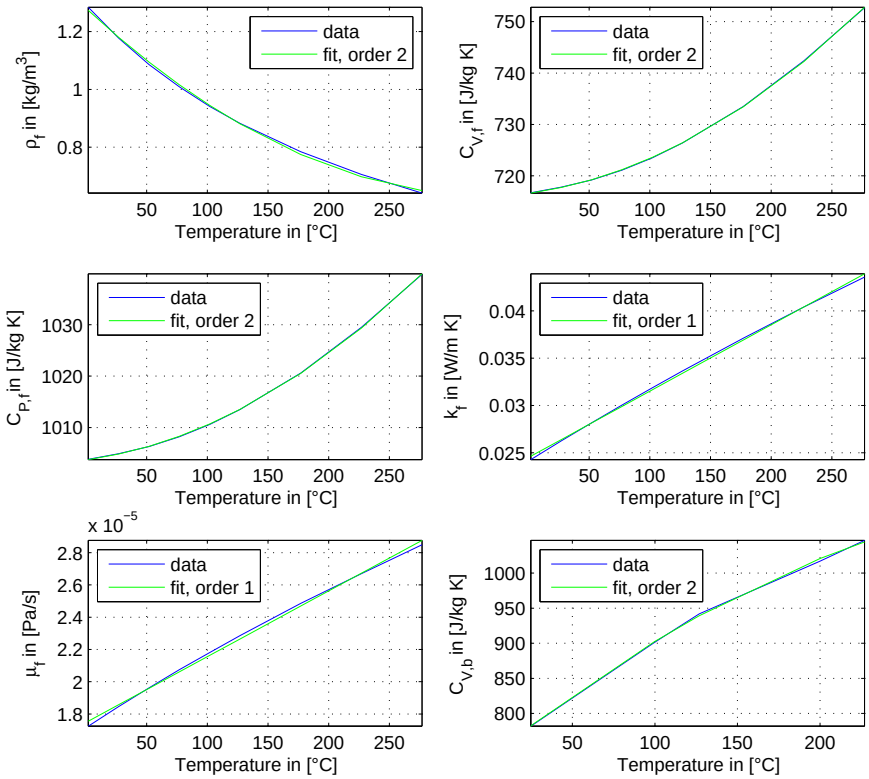

Figure 2.2: Thermophysical properties of air and aluminum oxide at atmospheric pressure. Actual data compared to a polynomial fitting

2.3 Numerical model of the system

The mathematical model for the trend of the temperatures T_f and T_b in the infinitesimal volume dV is described with the equations (**2.17**), (**2.18**), (**2.19**) and (**2.20**).
Extracting the following coefficients

$$C_1(\dot{m}_f, T_f) = \frac{\dot{m}_f \cdot C_{P,f}}{\rho_f \cdot \epsilon \cdot C_{V,f} \cdot A} \text{ in } \left[\frac{\mathbf{m}}{\mathbf{s}}\right], \tag{2.21}$$

$$C_2(\dot{m}_f, T_f) = \frac{h_V}{\rho_f \cdot \epsilon \cdot C_{V,f}} \text{ in } \left[\frac{\mathbf{1}}{\mathbf{s}}\right], \tag{2.22}$$

$$C_3(\dot{m}_f, T_f, T_b) = \frac{h_V}{\rho_b \cdot (1-\epsilon) \cdot C_{V,b}} \text{ in } \left[\frac{\mathbf{1}}{\mathbf{s}}\right], \tag{2.23}$$

the equations (**2.17**) and (**2.18**) can be rewritten as follows:

$$\frac{\partial T_f}{\partial t} = -C_1 \frac{\partial T_f}{\partial x} + C_2(T_b - T_f), \tag{2.24}$$

$$\frac{\partial T_b}{\partial t} = C_3(T_f - T_b). \tag{2.25}$$

The equation cannot be solved analytically because the coefficients C_1, C_2 and C_3 depend on T_f, T_b and $\dot{m}_f$.

To find a numerical solution to the system of differential equations, a Finite Element Method is used. The system contains derivatives with respect to the time t and to the distance x, thus two different approaches of discretization are proposed in the next sections:

1. Discretize both t and $x \rightarrow$ Lax-Wendroff method

2. Discretize only x, while t is continuous

To discretize the function domain, all the derivatives can be substituted by a proper difference quotient. There are various ways to approximate, for example, a first order derivative:

- Forward finite difference: $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1}-u_i}{\Delta x}$
- Backward finite difference: $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_i-u_{i-1}}{\Delta x}$
- Centered finite difference: $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1}-u_{i-1}}{2\Delta x}$

In general, backward finite difference is used for derivatives with respect to the time, while for derivatives with respect to the distance usually a centered finite difference is used.

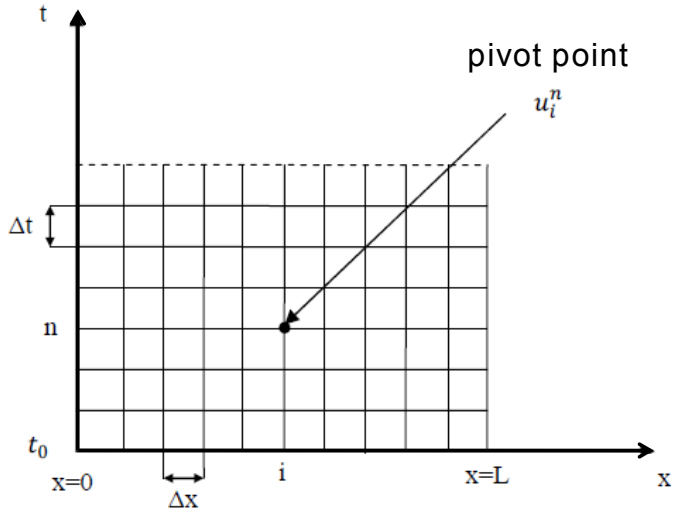

Figure 2.3: Grid for both x and t discretization

2.3.1 Lax-Wendroff method

To discretize the model temporary and spatially, the Lax-Wendroff method is used. This model has a second order accuracy for both t and x. Another advantage of this method is that stability is guaranteed unless a CFL condition does not hold.

Figure **2.3** shows the grid of discretization for both variables. Each node u of the grid carries two indexes:

$$\begin{aligned} &u_i^n = u(x = i \cdot \Delta x, t = n \cdot \Delta t), \\ &i \in [0, 1, ..., M], n \in [0, 1, ..., n_{end}], \end{aligned} \tag{2.26}$$

with boundary values

$$\begin{aligned} x_0 &= 0cm, \\ x_M &= L, \end{aligned} \tag{2.27}$$

and initial value

$$t_0 = 0s. \tag{2.28}$$

To derive the numerical model, let's start from the convection equation

$$\frac{\partial T_f}{\partial t} = -C_1 \frac{\partial T_f}{\partial x} + C_2(T_b - T_f) \tag{2.29}$$

and derive both parts with respect to t:

$$\frac{\partial^2 T_f}{\partial x \partial t} = -\frac{1}{C_1}\frac{\partial^2 T_f}{\partial t^2} + \frac{C_2}{C_1}\frac{\partial (T_b - T_f)}{\partial t} \tag{2.30}$$

and with respect to x, respectively:

$$\frac{\partial^2 T_f}{\partial t \partial x} = -C_1\frac{\partial^2 T_f}{\partial x^2} + C_2\frac{\partial (T_b - T_f)}{\partial x}. \tag{2.31}$$

Assuming that the function T_f is differentiable on its whole domain[2] , the right part of the two equations (**2.30**) and (**2.31**) can be combined to:

$$\frac{\partial^2 T_f}{\partial t^2} - C_1^2\frac{\partial^2 T_f}{\partial x^2} = +C_2\underbrace{\frac{\partial (T_b - T_f)}{\partial t}}_{\approx 0} - C_2 C_1\underbrace{\frac{\partial (T_b - T_f)}{\partial x}}_{\approx 0}. \tag{2.32}$$

Equation (**2.32**) is an inhomogeneous wave equation. Fo sake of simplification, it can be assumed that the difference between the temperature of the air and of the balls $(T_b - T_f)$ does not vary significantly (neither spatially nor temporary) during a the short time Δt. Hence, the right part of the equation can be set to zero and the wave equation is homogeneous now:

$$\frac{\partial^2 T_f}{\partial t^2} = C_1^2\frac{\partial^2 T_f}{\partial x^2}. \tag{2.33}$$

At this point, let's develop the function T_f at the point $(x, t + \Delta t)$ in Taylor Series:

$$T_f(x, t + \Delta t) = T_f(x, t) + \frac{\partial T_f(x, t)}{\partial t}\Delta t + \frac{\partial^2 T_f(x, t)}{\partial t^2}\frac{\Delta t^2}{2!} + ... \tag{2.34}$$

Inserting (**2.29**) and (**2.33**) in (**2.34**) and neglecting the terms of more than 2nd order leads to:

$$\begin{aligned} T_f(x, t + \Delta t) = T_f(x, t) &+ \left(-C_1\frac{\partial T_f}{\partial x} + C_2(T_b - T_f)\right)\Delta t \\ &+ \left(C_1^2\frac{\partial^2 T_f}{\partial x^2}\right)\frac{\Delta t^2}{2!}. \end{aligned} \tag{2.35}$$

[2]If a function $u(x,t), [x,t] \in D$ is differentiable in $(x_0, t_0) \in D$, then $u_{xt}(x_0, t_0) = u_{tx}(x_0, t_0)$

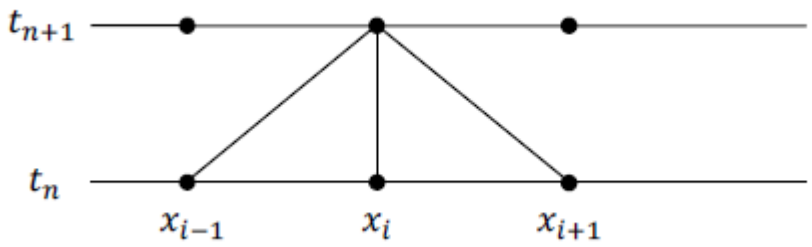

Figure 2.4: Iterative Lax-Wendroff procedure

The derivatives with respect to x can be replaced by the centered finite differences

$$(u_x)_i^n \approx \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}, \tag{2.36}$$

$$(u_{xx})_i^n \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}, \tag{2.37}$$

with the following result:

$$\begin{aligned} T_{f_i}^{n+1} = T_{f_i}^n + C_2 \cdot (T_{b_i}^n - T_{f_i}^n)\Delta t - C_1 \cdot \Delta t \frac{\left(T_{f_{i+1}}^n - T_{f_{i-1}}^n\right)}{2\Delta x} \\ + \frac{\Delta t^2 C_1^2}{2!} \cdot \frac{\left(T_{f_{i+1}}^n - 2T_{f_i}^n + T_{f_{i-1}}^n\right)}{\Delta x^2}. \end{aligned} \tag{2.38}$$

Here, the time t and the distance x have been replaced by the discrete values n and i. Equation (**2.38**) describes how T_f at some discretization point $(n+1, i)$ can be iteratively calculated from $T_{f_i}^n$, $T_{f_{i+1}}^n$, $T_{f_{i-1}}^n$ and $T_{b_i}^n$. Of course this is only possible if the initial conditions and the boundary conditions are known. See figure **2.4** for a graphical explanation of the procedure.

The calculation of the variable T_b is still missing. Therefore the forward finite difference

$$\left(\frac{\partial T_b}{\partial t}\right)_i^n \approx \frac{T_{b_i}^{n+1} - T_{b_i}^n}{\Delta t} \tag{2.39}$$

is used in order to discretize equation (**2.25**), which results in:

$$T_{b_i}^{n+1} = T_{b_i}^n + C_3 \Delta t (T_{f_i}^n - T_{b_i}^n). \tag{2.40}$$

For the pivot point corresponding to the last spatial discretization point $i = M$ with $x_M = L$, it is necessary to find a different correlation. The reason is that $i = M + 1$ does not exist, and hence the iteration for calculating $T_{f\,M}$

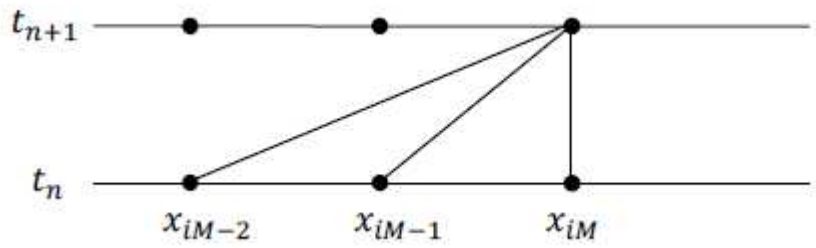

Figure 2.5: Iterative Lax-Wendroff procedure for $i = M$

and T_{bM} must depend only on T_f and T_b with $i \leq M$ (see figure **2.5**). The equation for T_f used in this case is[3]:

$$T_{f_M}^{n+1} = T_{f_M}^{n} + C_2 \cdot (T_{b_M}^{n} - T_{f_M}^{n})\Delta t - C_1 \cdot \Delta t \frac{\left(T_{f_{M-2}}^{n} - 4T_{f_{M-1}}^{n} + 3T_{f_M}^{n}\right)}{2\Delta x} + \frac{\Delta t^2 C_1^2}{2!} \cdot \frac{\left(T_{f_{M-2}}^{n} - 2T_{f_{M-1}}^{n} + T_{f_M}^{n}\right)}{\Delta x^2} \tag{2.41}$$

The equation for T_b instead is unchanged.
Before continuing with the modeling, the stability of the numerical model is briefly discussed.

Stability of the Lax-Wendroff method

If the Lax-Wendroff discretization is done on the system

$$\begin{aligned} \frac{\partial T_f}{\partial t} &= -C_1 \frac{\partial T_f}{\partial x} \\ \frac{\partial^2 T_f}{\partial t^2} &= C_1^2 \frac{\partial^2 T_f}{\partial x^2} \end{aligned} \tag{2.42}$$

there exists a test to assure the stability of the numerical model: the Courant-Friedrichs-Lewy number (CFL) must be smaller than 1: [12]

$$CFL = \frac{C_1 \Delta t}{\Delta x} \leq 1 \rightarrow \Delta t \leq \frac{\Delta x}{C_1}. \tag{2.43}$$

[3]Here, for the term $(u_x)_M^n$ a backward difference approximation with second order error is used. For the second derivative instead, it was assumed that $(u_{xx})_M^n \approx (u_{xx})_{M-1}^n$

Consider a wave moving across a discrete spatial grid. If we want to compute the amplitude of the wave at discrete time steps with constant sample time, then the sample time must be smaller than the time for the wave to reach an adjacent grid point.
This means that the discretization Δx can be chosen arbitrarily in order to achieve the accuracy needed. Instead Δt depends on the previous choice.
As mentioned previously, the temperature can be measured at 19 different points through the laboratory plant. The first sensor is placed at $x = 0\,\mathbf{cm}$ and the last one at $x = 180\,\mathbf{cm} = L$.[4] That means that there is a sensor each $10\,\mathbf{cm}$.
A good compromise between accuracy and model efficiency is the choice to discretize the gap between two sensors 4 times, i.e:

$$\Delta x = 2.5\,\mathbf{cm} \tag{2.44}$$

Thus, the model can be viewed as a state-space model with $(4 \cdot 18)$ state-variables for describing the temperature T_f and the same amount for T_b. Overall, there are 144 variables.

To determine the sample time Δt let's have a look at figure **2.6**. It shows how $C_1(\dot{m}_f, T_f)$ varies with the temperature of the fluid for different values of the mass flow, including the minimum and maximal flow. It is clear that C_1 is both proportional to the temperature and to the mass flow. That means that its maximum value is at:

$$C_{1,MAX} = C_1(\dot{m}_{f,MAX}, T_{hot}) = 5.03\,\mathbf{m/s} \tag{2.45}$$

To make sure that the CFL holds anytime, the following must be true:

$$\Delta t \leq \frac{\Delta x}{C_{1,MAX}} = 0.00497\,\mathbf{s} \tag{2.46}$$

After doing some experiments on the model, though, it was found that the discretization using the sample time $\Delta t = 0.004\,\mathbf{s}$ is not stable for high mass flows. This could be due to the fact that we are not considering the homogeneous system (**2.42**), but there is an additional term. All experiments held were stable using $\Delta t = 0.0025\,\mathbf{s}$ instead.

A possible choice for the discretization is therefore:

[4] Remember the assumption that the temperature does not vary in radial direction

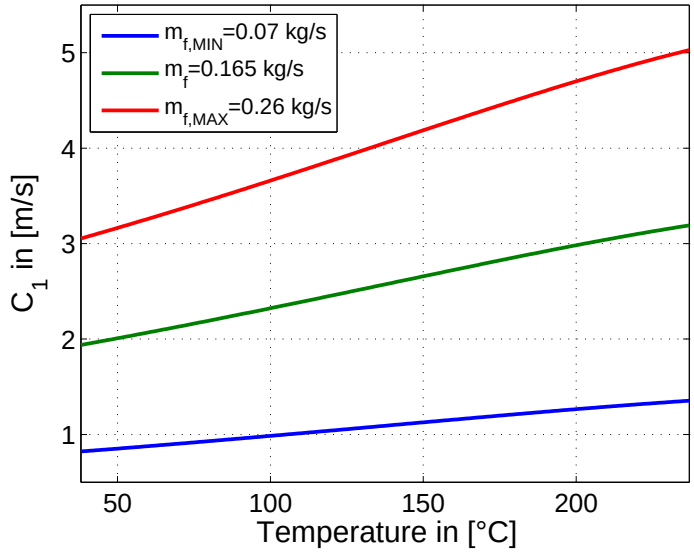

Figure 2.6: Coefficient C_1 over temperature T_f for various values of the mass flow $\dot{m}_f$

- $\Delta t = 0.0025\,\mathbf{s}$
- $\Delta x = 0.25\,\mathbf{cm}$

Initial conditions and Boundary conditions

For a *Simulink*-implementation of the Thermal Storage System there are still some specifications needed.

First of all, boundary conditions and initial conditions must be set. The initial conditions for the fluid and the packed bed are:

- $T_f(t = 0) = T_{cold}$
- $T_b(t = 0) = T_{cold}$

The only boundary condition that is needed to solve the iterative problem is $T_f(x = 0)$. All other boundaries are free.
It would seem quite obvious that the temperature of the fluid at the beginning of the coil is be equal to the temperature of the hot flow passing through the

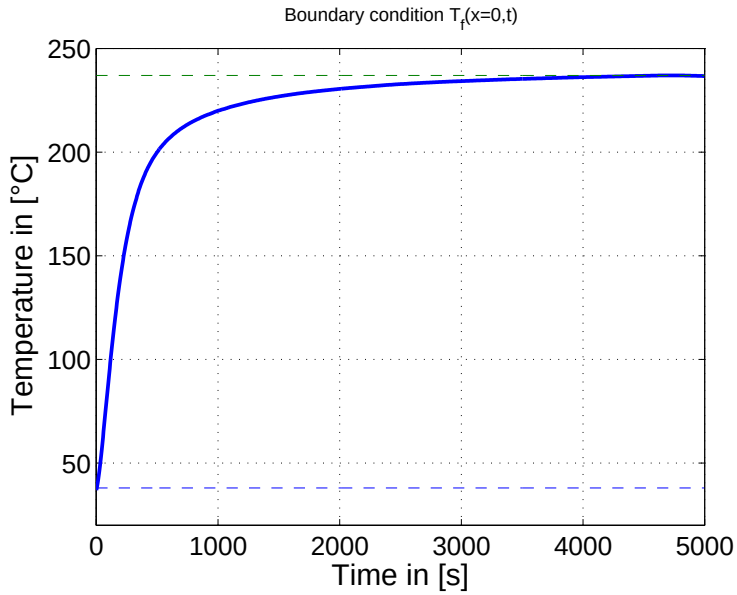

Figure 2.7: Boundary condition $T_f(x = 0, t)$

coil. The problem is that the flow, before reaching the coil, must pass through a diffuser that "delays" the reaching of the temperature T_{hot}. For this thesis we will assume that the delay does not depend on the mass flow and hence the boundary condition $T_f(x = 0, t)$ will be set equal to the temperature measured in a laboratory experiment from the sensor placed at $x = 0\,\mathbf{cm}$. Thus, the boundary condition is time-dependent(see figure **2.7**). It is clear that:

- $T_f(x = 0, t = 0) = T_{cold}$
- $T_f(x = 0, t \to \infty) = T_{hot}$

2.3.2 Continuous-time x-discretization

Here a different approach for solving the differential equations **2.24** and **2.25** is proposed. Instead of discretizing both variables x and t, here the time is continuous, while only x is divided into finite elements.
Each time-continuous $u(t)$ carries a single index:

$$\begin{aligned} &u_i(t) = u(x = i \cdot \Delta x, t) \\ &i \in [0, 1, ..., M], t \in [0, t_{end}] \end{aligned} \tag{2.47}$$

with boundary values

$$\begin{aligned} x_0 &= 0cm, \\ x_M &= L. \end{aligned} \tag{2.48}$$

Equation **2.25** does not contain any derivative with respect to x and remains hence unchanged. To discretize equation **2.24**, the derivative with respect to x is replaced by a centered finite difference, which results in

$$\frac{\partial T_{f_i}}{\partial t} = -C_1 \left(\frac{T_{f_{i+1}} - T_{f_{i-1}}}{2\Delta x} \right) + C_2(T_{b_i} - T_{f_i}) \tag{2.49}$$

For the last discretization point $i = M$, insted, a backward finite difference is used:

$$\frac{\partial T_{f_M}}{\partial t} = -C_1 \left(\frac{T_{f_M} - T_{f_{M-1}}}{\Delta x} \right) + C_2(T_{b_M} - T_{f_M}) \tag{2.50}$$

The initial conditions and boundary conditions are the same as for the Lax-Wendroff discretization.
The discretization Δx can be chosen arbitrarily and the model is always stable.

2.4 Model vs. plant

At this point the two numerical model can be used for testing their validity and accuracy.
This can be done by comparing the simulation results with the measurements of a laboratory experiment on the real plant for a certain input. Let's consider for example the charging procedure of the Thermal Storage System being at a temperature T_{cold}, using as an input the constant mass flow $\dot{m}_f = 0.15\,\mathbf{kg/s}$ at a temperature T_{hot}. We will assume that the sensors of the plant measure the temperature of the fluid, hence T_f.
Figure **2.8** shows for both numerical models with $\Delta x = 0.025\,\mathbf{m}$ the temperature measured over time by the 19 sensors placed all over the coil compared to the simulated temperature. The errors

$$\begin{aligned} e(T_0) &= T_{0sim} - T_{0lab} \\ e(T_1) &= T_{1sim} - T_{1lab} \\ e(T_2) &= ... \end{aligned} \tag{2.51}$$

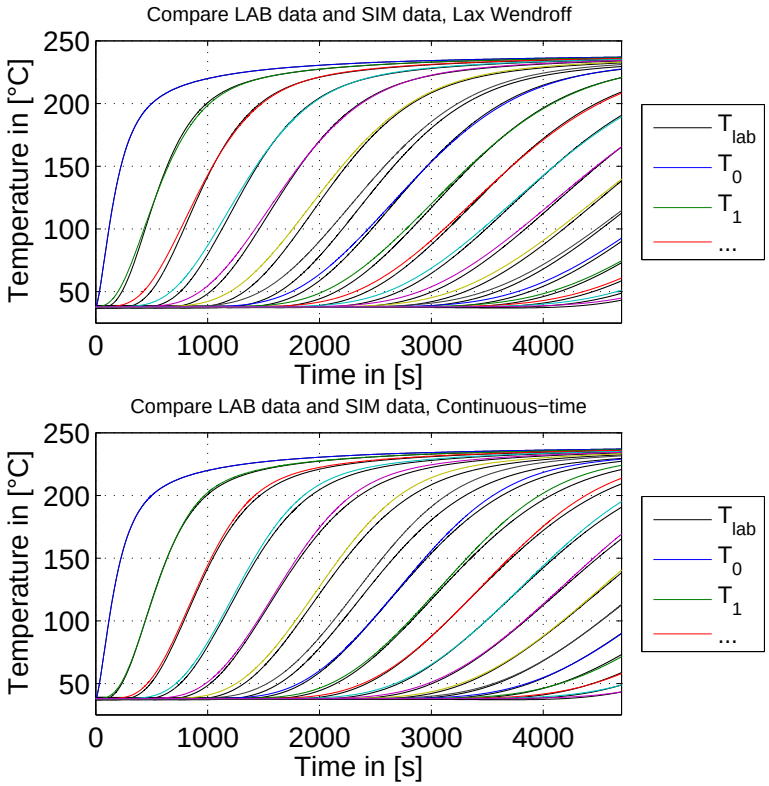

Figure 2.8: Comparison between laboratory measurements on the real plant and simulated model. Charging procedure of the Thermal Storage System with constant flow $\dot{m}_f = 0.15\,\mathbf{kg/s}$. First graph: Lax-Wendroff method. Second graph: Continuous-time x-discretization

are reported in figure **2.9**, respectively.
The error $e(T_0)$ is equal to zero because the laboratory measurement was used as the boundary condition. Looking at the first graph,corresponding to the Lax-Wendroff method, we can see that for the following measurement points the simulated temperature is a little bit higher than the actual temperature till they reach approximately $140\mathbf{^oC}$. Afterwords the simulated temperature is slightly lower than it should be, but quite accurate. This imprecisions could be due to quantization errors. The asymptotical behavior of the model is very good, as

$$e \to 0 \text{ for } t \to \infty. \tag{2.52}$$

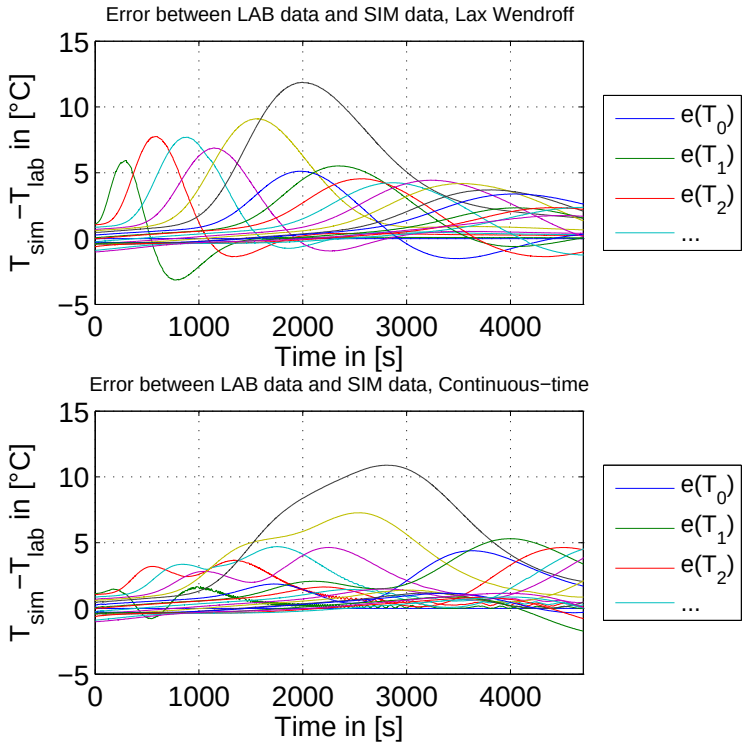

Figure 2.9: Error between laboratory measurements on the real plant and simulated model. Charging procedure of the Thermal Storage System with constant flow $\dot{m}_f = 0.15\,\mathbf{kg/s}$. First graph: Lax-Wendroff method. Second graph: Continuous-time x-discretization

The error $e(T_6)$ is quite big, but it looks like this is due to a measurement error and not related to modeling. It is possible that the sensor was placed around $x = 63\,\mathbf{cm}$ instead of being at $x = 60\,\mathbf{cm}$. Therefore this error will not be taken into account in the following considerations.
The maximal error is

$$e_{MAX} = 9^{\mathbf{o}}\mathbf{C} \tag{2.53}$$

which results in a percentage error of

$$e_{MAX,\%} = \frac{e_{MAX}}{T_{hot} - T_{cold}} \cdot 100\% = 4.52\%. \tag{2.54}$$

However, the mean absolute error for a simulation of $4700s$ is:

$$e_{MEAN} = 1.3^{\mathbf{o}}\mathbf{C} \rightarrow e_{MEAN,\%} = \frac{e_{MEAN}}{T_{hot} - T_{cold}} \cdot 100\% = 0.65\%. \qquad (\mathbf{2.55})$$

Figure **2.10** shows the temperature distribution in the coil measured at $500\,\mathbf{s}$ time intervals. It is clear that the state of charge of the model is higher than the s.o.c. of the plant at each time instant. This can be due to the losses that were neglected in the model. However we can see that the model accuracy gets better after a certain settlement. Additionally, note the higher error at $x = 60\,\mathbf{cm}$.
The second graph of each pictures refers to the simulation results on the time-continuous model with x-discretization. We can see that in general this model is better than the previous one for lower temperatures, while it has a worse behavior for $t \rightarrow \infty$. Here the errors are:

$$e_{MAX} = 7.25^{\mathbf{o}}\mathbf{C} \rightarrow e_{MAX,\%} = 3.64\%, \qquad (\mathbf{2.56})$$

$$e_{MEAN} = 1.1^{\mathbf{o}}\mathbf{C} \rightarrow e_{MEAN,\%} = 0.55\%. \qquad (\mathbf{2.57})$$

All in all we can say that both models lead to a good result. For all the following experiments the first model will be used.

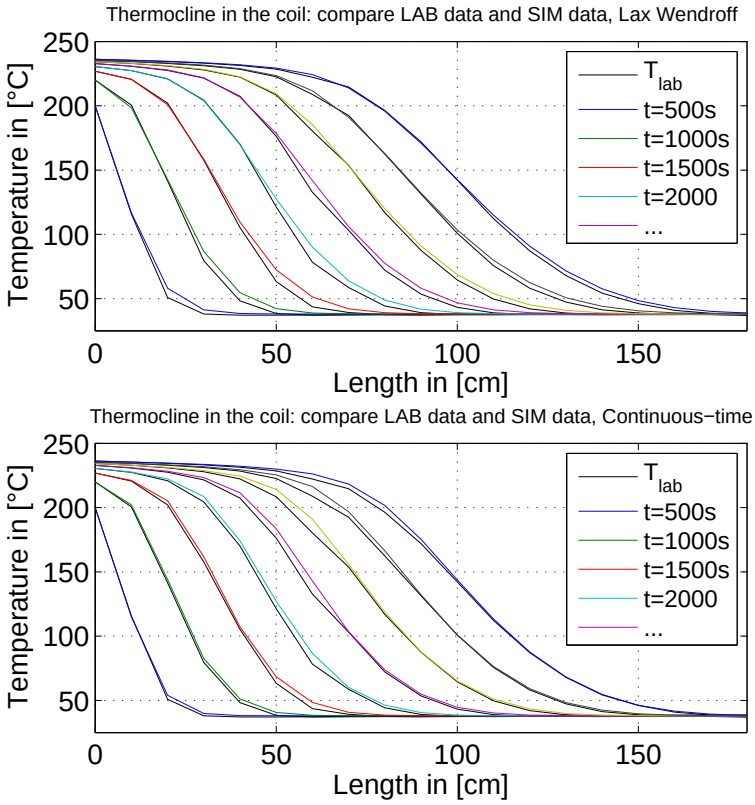

Figure 2.10: Temperature distribution in the coil at various time-steps. Comparison between laboratory measurements on the real plant and simulated model. Charging procedure of the Thermal Storage System with constant flow $\dot{m}_f = 0.15\,\mathbf{kg/s}$. First graph: Lax-Wendroff method. Second graph: Continuous-time x-discretization

Chapter 3

Why control?

3.1 Experiments on the model

The model built with the specifications exposed in chapter **2** can now be used for doing some experiments in order to understand the behavior of the system.

3.1.1 Experiments with constant inputs

As a start, some experiments with a constant input $\dot{m}_f$ are carried out. Figure **3.1** shows the evolution of the temperatures of the fluid and of the balls for each 10 **cm** for three different constant flows.

First of all let's do a consideration on the difference between the temperature of the fluid T_f and the temperature of the balls T_b. As we could imagine, T_b is always a little bit lower than T_f. It is understood that when the hot flow enters the coil, it takes some time for the balls to be heated. The temperature gap for each measurement point is graphically shown in figure **3.2**. Notice that the gap is higher for lower x and that they are asymptotically zero.
Getting back to figure **3.1**, notice that the temperatures can only increase and never drop. Even if the flow is lowered, it is never possible to drop the temperature, because the input-flow will always be at a constant temperature T_{hot}. In addition, we can clearly see that the final stable value does not depend on the flow, but is always equal to T_{hot}. We will see that these two

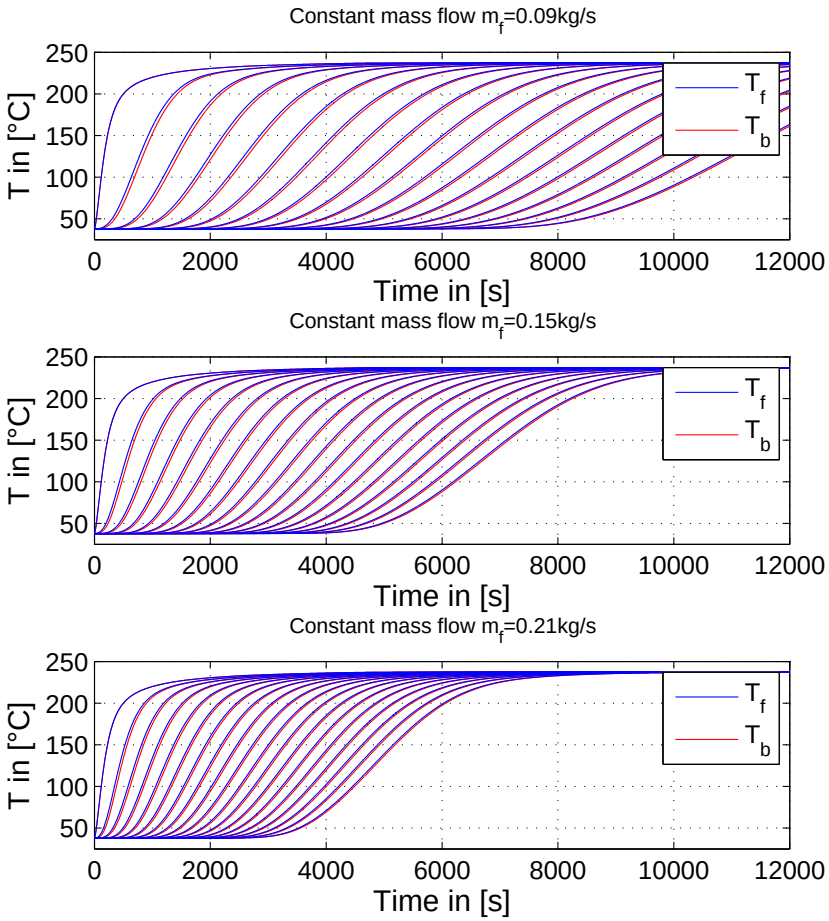

Figure 3.1: Evolution of the temperature of the fluid and of the balls for three different constant inputs: $\dot{m}_f = 0.09\,\mathbf{kg/s}$, $\dot{m}_f = 0.15\,\mathbf{kg/s}$ and $\dot{m}_f = 0.21\,\mathbf{kg/s}$

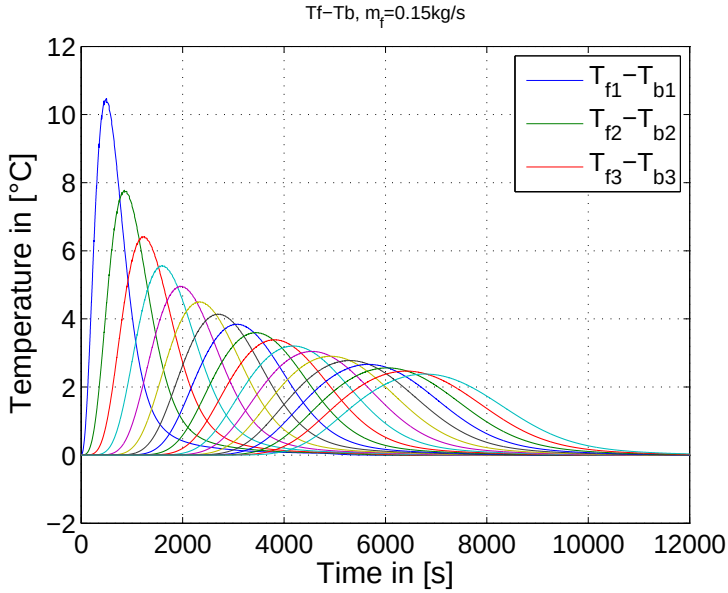

Figure 3.2: Temperature gap between the fluid and the balls for the constant input $\dot{m}_f = 0.15\,\mathbf{kg/s}$

properties of the system will make the control of the system a bit challenging. On the other hand, it is clear from the graphics that the duration of the charging phase is indirectly proportional to the strength of the flow. Throughout the whole thesis we will assume that the system is charged when the temperature of the fluid at the end of the cylinder ($x = 180\,\mathbf{cm}$) will rise by

$$\Delta T_{charged} = 7.5^{\mathbf{o}}\mathbf{C}. \tag{3.1}$$

With this assumption the charging-duration depending on the flow is depicted in figure **3.3**. For the minimum mass flow the charging procedure lasts more than three times longer than with the maximum mass flow ($10500\,\mathbf{s} \approx 3\,\mathbf{h}$ vs. $2700\,\mathbf{s} \approx 3/4\,\mathbf{h}$). The dependence can be approximated with a polynomial of 3rd degree.

At that point one could inadvertently conclude that the charging with a higher mass flow is in general favorable. The problem is that a higher mass flow leads to a worse charge distribution over the coil. As previously mentioned we assume the system to be fully charged as soon as $T_{f,18} \geq T_{cold} + \Delta T_{charged}$. At that time $t_{charged}$ we measure the percentage of charge

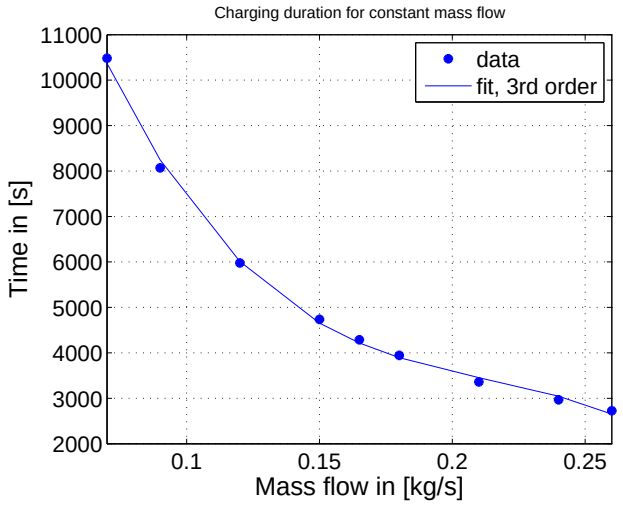

Figure 3.3: Charging duration of the model for different constant mass flows. The system is assumed to be charged as soon as $T_{f,18} \geq T_{cold} + \Delta T_{charged}$.

in the coil considering the temperature of the balls:

$$\%_{charged} = \left(\int_0^L T_b(t_{charged}, x)\, \mathrm{d}x - T_{cold} \right) \cdot \frac{100}{(T_{hot} - T_{cold}) \cdot L}. \qquad (\mathbf{3.2})$$

Figure **3.4** shows the charging percentage for various constant mass flows which can be approximated by a 2nd order polynomial. The percentage varies between 72.3% for the minimum mass flow and 63.8% for the maximum mass flow.
That means that there will always be a compromise between a better charge distribution and the charging duration.
All the pairs ($t_{charged}$, $\%_{charged}$) are summarized in table **3.1**.

3.1.2 Other experiments

As explained in the previous section the quality of an experiment is measured in terms of the duration and the charging percentage. The duration should be possibly short and the percentage high.
In addition to the experiments with constant mass flow, other experiments were held in order to try to achieve better results.

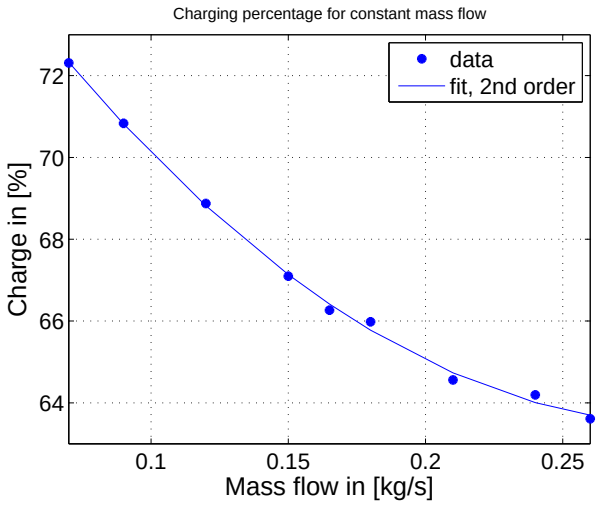

Figure 3.4: Charging percentage of the model for different constant mass flows. The system is assumed to be charged as soon as $T_{f,18} \geq T_{cold} + \Delta T_{charged}$.

	$\mathbf{t_{charged}}$ in $[s]$	$\%_{\mathbf{charged}}$
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.07\,kg/s}$	10482	72.3
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.09\,kg/s}$	8069	70.83
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.12\,kg/s}$	5978	68.87
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.15\,kg/s}$	4735	67.1
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.165\,kg/s}$	4286	66.27
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.18\,kg/s}$	3943	65.98
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.21\,kg/s}$	3359	64.56
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.24\,kg/s}$	2966	64.2
$\dot{\mathbf{m}}_\mathbf{f} = \mathbf{0.26\,kg/s}$	2727	63.61
Experiment 1	6840	66.69
Experiment 2	6579	70.19
Experiment 3	4390	68.5

Table 3.1: Duration of charging phase and percentage of charge for various experiments

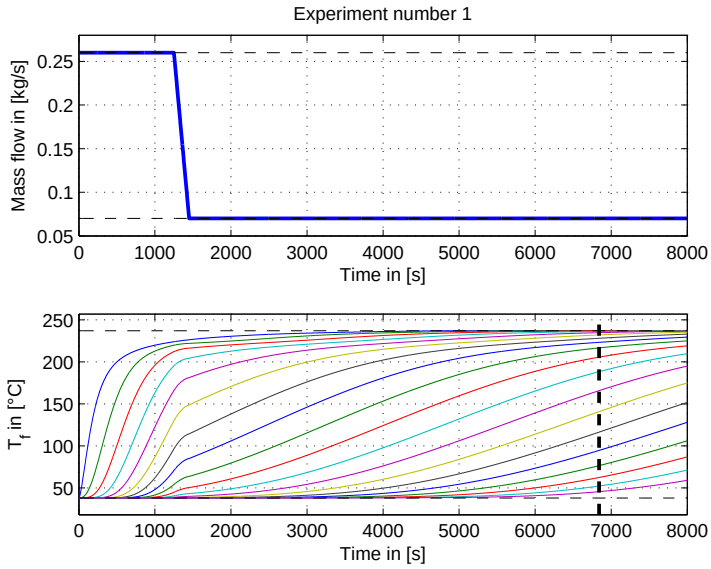

Figure 3.5: Experiment 1: input $\dot{m}_f$ and output T_f

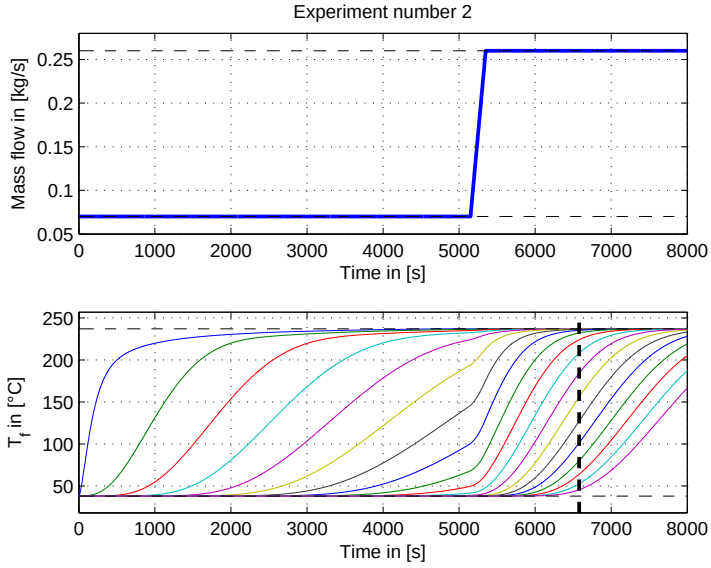

Figure 3.6: Experiment 2: input $\dot{m}_f$ and output T_f

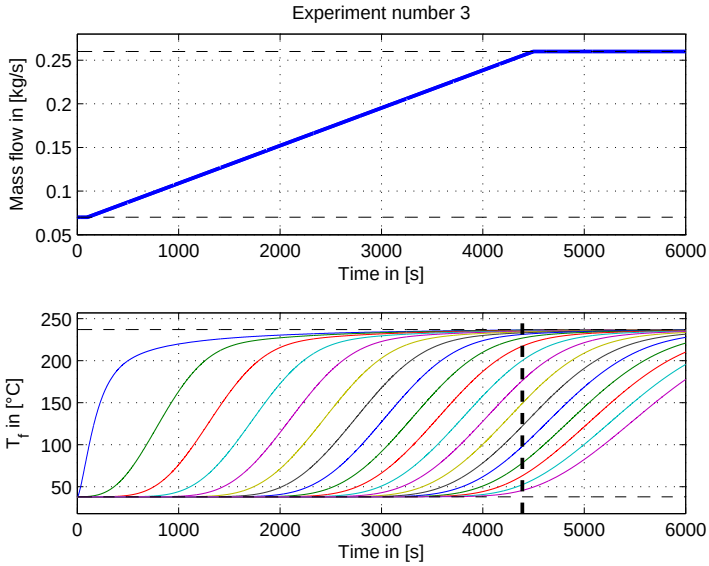

Figure 3.7: Experiment 3: input $\dot{m}_f$ and output T_f

Experiment 1 uses $\dot{m}_{f,MAX}$ for the first part of the experiment and then it decreases quite fast to $\dot{m}_{f,MIN}$, which is held till the end of the experiment. The switch between the two constant values occurs at

$$t_{switch} \approx t_{charge}(\dot{m}_{f,MAX})/2 \tag{3.3}$$

and lasts $200s$. See figure **3.5** for the input-output of the experiment.
On the contrary, Experiment 2 (see figure **3.6**) uses first the low values for the mass flow and then it increases to the higher value. Here the switch happens at

$$t_{switch} \approx t_{charge}(\dot{m}_{f,MIN})/2 \tag{3.4}$$

and lasts as well $200s$.
Experiment 3 (figure **3.7**) as well starts from the lower value and increases, but in that case the switch is at $100s$ and lasts almost till the end of the experiment ($4500s$).
All the pairs ($t_{charged}$, $\%_{charged}$) corresponding to the three experiments are reported in table **3.1**.

To measure if the experiments give better results than the experiments with constant input, figure **3.8** compares the pairs ($t_{charged}$, $\%_{charged}$) for con-

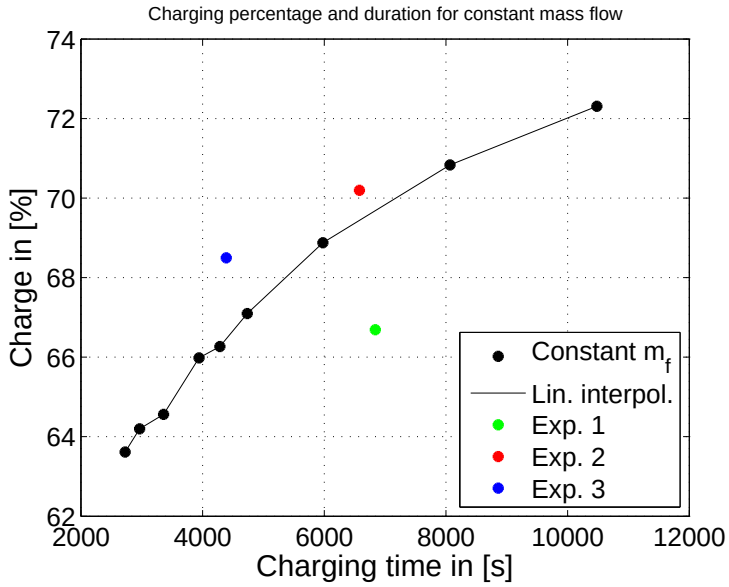

Figure 3.8: Compare the results of the experiments with constant mass flow (black line) with the three experiments of section **3.1.2**. The pairs that lie above the line are 'worse' experiments, the ones over it are 'better' experiments

stant mass flow with the pairs of the last three experiments. We can say that all the pairs that lie above the black line are 'worse' charging procedures and the ones that lie over it are 'better' results. We can clearly see that Experiment 1 is not suitable, as it takes $2340\,\mathbf{s} \approx 40\,\mathbf{min}$ more to achieve the same percentage as with constant mass flow.
Experiment 2 instead leads to good results. Here it takes $810\,\mathbf{s} \approx 15\,\mathbf{min}$ less to achieve the same percentage as with a constant input.
Experiment 3 gives even better results: it takes $1300\,\mathbf{s} \approx 22\,\mathbf{min}$ less to get the same percentage as with constant mass flow.
The experiments show that it should be avoided to decrease the mass flow during the charging phase. Instead, it is profitable to increase it gradually. The problem is that there is always a compromise between charging duration and distibution.

3.2 Control specification

All the experiments listed in the last section were performed in open-loop. That means that the the outputs or state-space variables do not affect the input during the experiment. The measurements of an experiment were merely used to decide the input of the next experiment.
In the next chapters various closed-loop control strategies are applied to the model in order to improve its performance. That means to achieve a higher charging percentage in a shorter time.

3.2.1 Input, output and state-vector

In section **2.3** the x- and t-discretized model for the Thermal Storage System is deduced. The iterative model obtained is rewritten here[1]:

$$T_{f_i}^{n+1} = T_{f_i}^{n} + C_2 \cdot (T_{b_i}^{n} - T_{f_i}^{n})\Delta t - C_1 \cdot \Delta t \frac{\left(T_{f_{i+1}}^{n} - T_{f_{i-1}}^{n}\right)}{2\Delta x} + \frac{\Delta t^2 C_1^2}{2!} \cdot \frac{\left(T_{f_{i+1}}^{n} - 2T_{f_i}^{n} + T_{f_{i-1}}^{n}\right)}{\Delta x^2}, \tag{3.5}$$

$$T_{b_i}^{n+1} = T_{b_i}^{n} + C_3 \Delta t (T_{f_i}^{n} - T_{b_i}^{n}). \tag{3.6}$$

As previously explained, choosing $\Delta x = 2.5\,\mathbf{cm}$, it can be viewed as a state-space model with 144 state-space variables describing T_b and T_f at the discretization points.

To perform a control it is important to know the inputs of the system which are manipulable. This inputs are called control variables. In this case the only **control variable** is the **mass flow $\dot{\mathbf{m}}_\mathbf{f}$**.[2] The mass flow does not appear directly in equations (**3.5**) and (**3.6**), but the coefficients C_1, C_2 and C_3 depend on it.

The only **outputs of the system** are the **temperatures of the fluid $\mathbf{T_f}$ measured every 10cm**. The control variable can depend on this output. The temperature T_b instead is not available for the feedback because

[1] Initial conditions and boundary conditions are needed

[2] Remember that the temperature of the mass flow during the charging phase is fixed at T_{hot} and cannot vary.

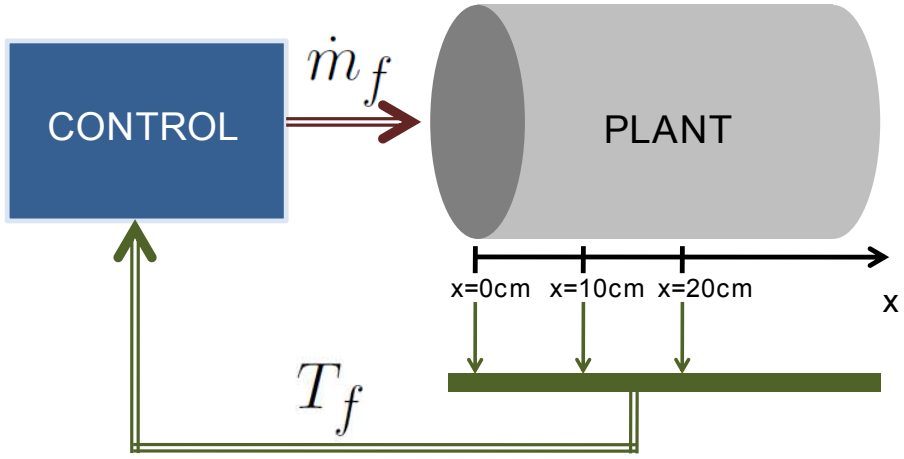

Figure 3.9: General control scheme of the Thermal Storage System

it cannot be measured on the real plant, as well as T_f at the remaining discretization points. See figure **3.9** for a general control scheme.

The main difficulty performing the control is that there is only one output that must control 19 outputs. Such a system is called SIMO-system (Single Input, Multiple Output). Actually the input must control an infinite number of outputs, because in fact the variable T_f is continuous over the length of the coil. As there are only a finite number of sensors, the continuous variable will be controlled by controlling only the samples available.

Another difficulty of controlling the system is announced in the following. Let $\mathbf{x}_k \in \mathbb{R}^{n\text{x}1}$ be the state-vector, $\mathbf{u}_k \in \mathbb{R}^{m\text{x}1}$ the input and $\mathbf{y}_k \in \mathbb{R}^{p\text{x}1}$ the output of the system. It is not possible to write the system given by (**3.5**) and (**3.6**) in the following way, which is required for many control methods:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k)\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) \end{aligned} \tag{3.7}$$

Instead, the model is of this type:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) \end{aligned} \tag{3.8}$$

We will see in the next chapters that this fact will hamper a lot the control procedure.

3.2.2 Reference temperature

To perform the control of a system, it is important to know how we want the model to behave. In that case the temperature T_f at the measurement points should be possibly similar to a set reference temperature.
Consider the graphs in figure **3.10**. The graphs on the left show the temperature T_f[3] over time with the respective distibution over the coil after the charging phase. All experiments have the same charging duration.
The graphs on the first line refer to an experiment with constant mass flow, which was treated extensively in the previous section.
The graphs on the second line show the ideal charging of the Thermocline. We can clearly see that first of all the temperature T_0 reaches gradually its peak value T_{hot}. Only afterwords, the temperature T_2 starts to increase. Similarly, T_4 increases only once T_2 is set to T_{hot}. In that way at each time step one part of the coil is 'hot' and the remaining part is 'cold'; the transition between the two parts is prompt. In this case the coil can be fully charged and the percentage of charge is close to 100%.
As this is quite unrealistic, what we want to reach is a behavior that is similar to the graph in the third line. Here all the temperatures increase in a very steep way and there is only a small overlap between the time intervals where the various temperatures increase. We see that this leads to a very good charging distribution over the coil. In the experiment with a constant mass flow instead, the temperatures with a higher index increase with a small velocity and have a very high overlap. Thus, the reference temperatures used for the various control theories in this thesis will be similar to the ones in the third line of figure **3.10**.

[3]For all the graphs only the outputs with an even index are used: i.e. T_0, T_2, T_4, etc.

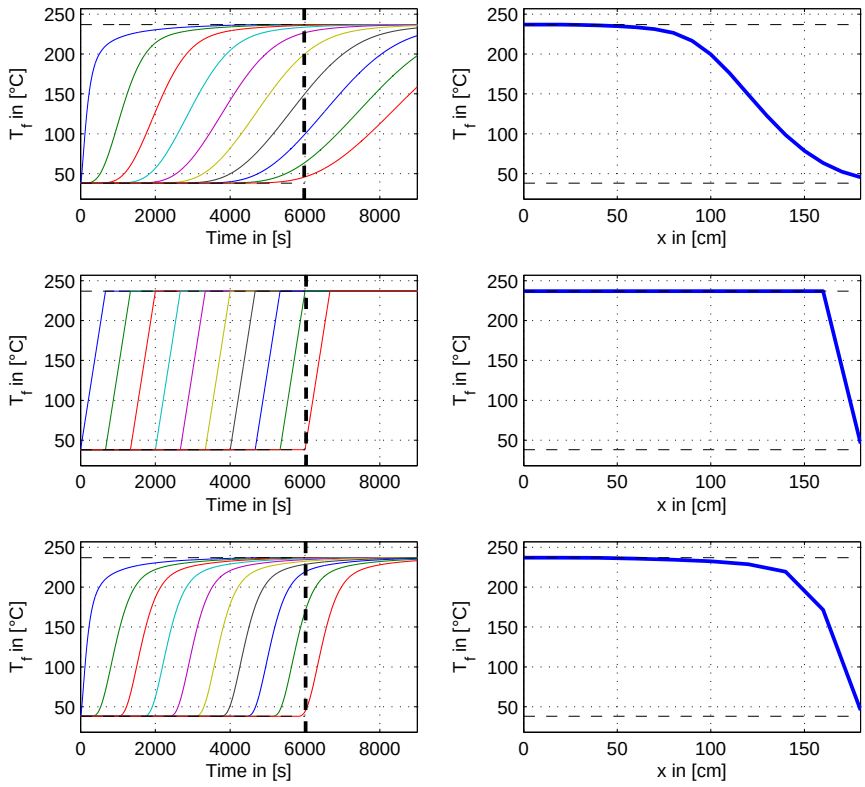

Figure 3.10: The figures on the left show the temperature T_f over time with, on the right, the respective T_f distribution after the charging phase. The charging duration is the same for all three cases. First line: constant mass flow. Second line: ideal case. Third line: sub-optimal ideal case

Chapter 4

PI controller

In this chapter, first of all the system is simplified to a linear model containing only stable poles and constant delays. The poles vary with the input and are identified using a "visual" approach.
This model is then controlled by a PI controller combined with a Smith-predictor with variable delay. As there is only one input but multiple outputs, the control feedback switches between the output variables.
Unfortunately this approach does not lead to a good result, because one control variable cannot simultaneously control the spatial and temporal behavior of more than one output.

4.1 First-order identification

As previously mentioned, the temperature of the Thermal Storage System can be measured by 19 sensors placed through the axial direction of the coil. The first sensor, which measures the temperature T_{f0}, is placed at $x = 0\,\mathbf{cm}$, while the last sensor for T_{f18} is placed at $x = 180\,\mathbf{cm}$. The distance between two neighboring sensors is always $10\,\mathbf{cm}$.[1] To identify a model that can be used for the continuous-time PI control, each volume between two sensors is modeled as a separate transfer function as shown in **4.1**.

The transfer function $\mathbf{P_1}(\mathbf{s})$ relating the input $\dot{m}_f$ to $(T_1 - T_{cold})$ will be

[1] Throughout this chapter the subscript f will be omitted, hence, $T_i = T_{f\,i}$.

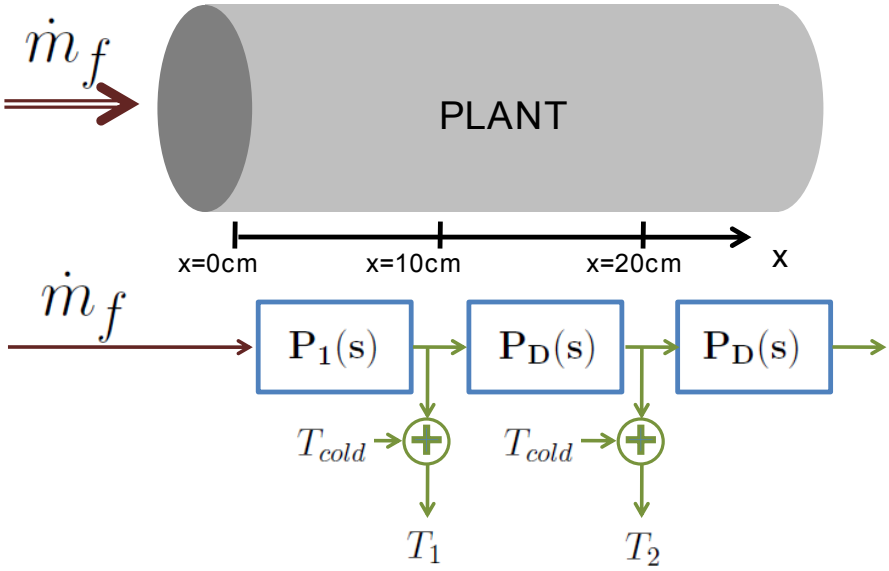

Figure 4.1: Scheme of model identification for the PI-control

identified as a first-order system with constant delay:

$$\mathbf{P_1(s)} = \underbrace{G(\dot{m}_f)\frac{1}{1+s\tau(\dot{m}_f)}}_{\mathbf{P_0(s)}} e^{-s\tau_{P_1}} \tag{4.1}$$

with parameters $G(\dot{m}_f)$ and $\tau(\dot{m}_f)$ depending on the mass flow, while τ_{P_1} is assumed to be constant for the whole range of the mass flow.
The following transfer functions $\mathbf{P_i(s)}$ will be modeled as:

$$\mathbf{P_D(s)} = \mathbf{P_i(s)} = \frac{1}{1+s\tau_D(\dot{m}_f)} \quad \forall i = 2, 3, ...18 \tag{4.2}$$

with parameter $\tau_D(\dot{m}_f)$ depending on the mass flow. It is trivial that the gain of these transfer functions must be unitary.
Hence, the relation between the input $\dot{m}_f$ and a temperature T_i with $i = 1, 2, ..., 18$ is defined as:

$$T_i(s) = \mathbf{P_{\dot{m}_f \to T_i}(s)}\dot{m}_f + T_{cold} = \mathbf{P_1(s)P_D(s)}^{i-1}\dot{m}_f + T_{cold}. \tag{4.3}$$

The parameters of the transfer functions were first of all identified for the constant inputs

$$\dot{m}_f \in \{0.07,\ 0.12,\ 0.15,\ 0.18,\ 0.24,\ 0.26\}\ \mathbf{kg/s} \tag{4.4}$$

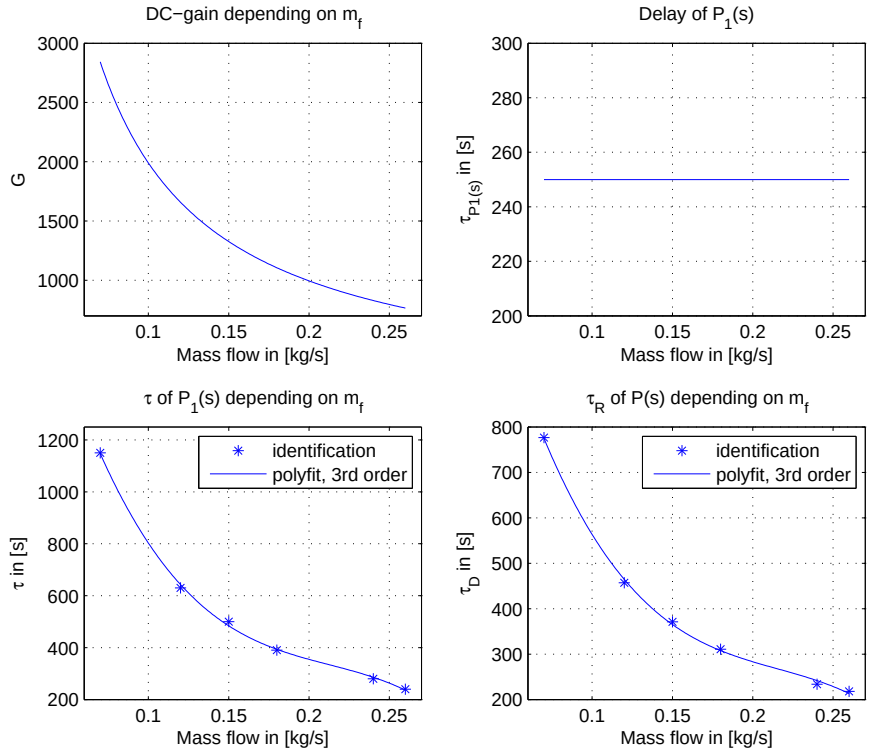

Figure 4.2: Parameters of the identified first-order transfer functions with delay. The dependance of the parameters on the input $\dot{m}_f$ can be described analytically or approximated by a polynomial of 3rd order.

using the simulation results of the discrete-time numerical model derived in **2.3.1**. Then a polynomial fit of 3rd order was performed on the single parameters in order to obtain an identified model which is valid on the whole input-range.

The single parameters, which are reported in figure **4.2**, were identified as explained in the following points.

Delay of the transfer function $\mathbf{P_1(s)}$ $\rightarrow$ τ_{P_1}

This delay is assumed to be constant for all inputs, i.e.:

$$\tau_{P_1} = 250\,\mathbf{s}. \tag{4.5}$$

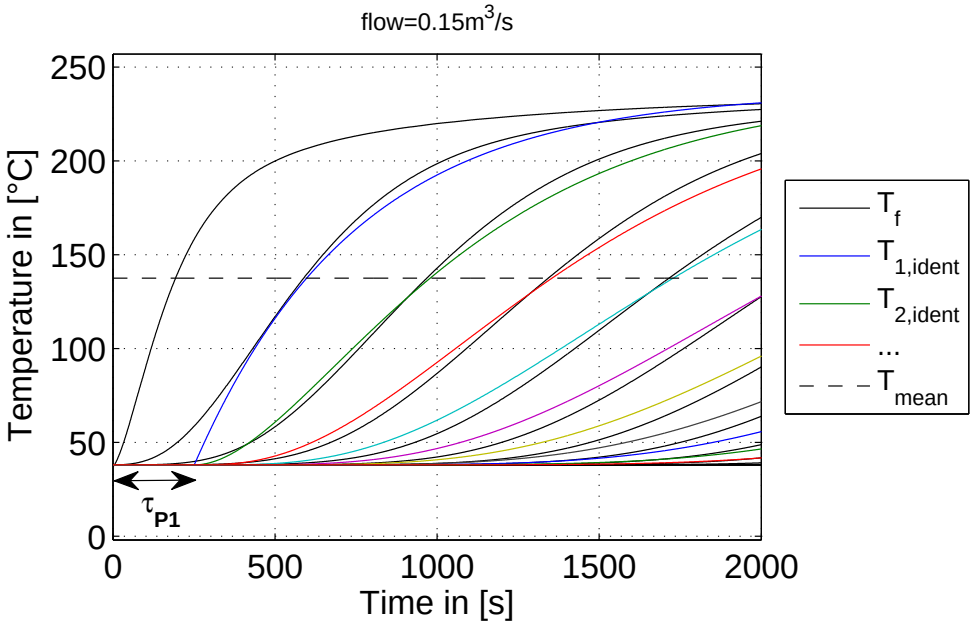

Figure 4.3: Comparison between the identified model and the numerical model for the constant input $\dot{m}_f = 0.15\,\mathbf{kg/s}$

DC-gain of the transfer function $\mathbf{P_1}(s)$ $\rightarrow$ $G(\dot{m}_f)$

As mentioned in the previous chapter, for all inputs the output-temperatures increase from the value T_{cold} to T_{hot}. This means that the the DC-gain can be written analytically as:

$$G(\dot{m}_f) = \frac{(T_{hot} - T_{cold})}{\dot{m}_f} = \frac{(199\mathbf{^oC})}{\dot{m}_f}. \tag{4.6}$$

Time constant of the transfer function $\mathbf{P_1}(s)$ and time constant of the transfer functions $\mathbf{P_D}(s)$ $\rightarrow$ $\tau(\dot{m}_f)$ and $\tau_D(\dot{m}_f)$

For all constant inputs both time constants were chosen so that the temperatures T_i of the identified model are exactly equal to the respective temperature of the numerical model when they both reach the value

$$T_{mean} = \frac{T_{hot} - T_{cold}}{2} + T_{cold}. \tag{4.7}$$

As an example, figure **4.3** compares the temperature behavior of the numerical model and of the identified model for the constant input $\dot{m}_f = 0.15\,\mathbf{kg/s}$.

In this case the time constants are

$$\tau(\dot{m}_f = 0.15\,\mathbf{kg/s}) = 500\,\mathbf{s}, \tag{4.8}$$

$$\tau_D(\dot{m}_f = 0.15\,\mathbf{kg/s}) = 371\,\mathbf{s}. \tag{4.9}$$

We can clearly see how the constant delay τ_{P1} affects the temperatures of the identified model. Furthermore, notice that all the temperatures of the identified model and of the numerical model intersect each other exactly at T_{mean}.
The identified values of $\tau(\dot{m}_f)$ and $\tau_D(\dot{m}_f)$ depending on the mass flow are shown in figure **4.2**. Both dependencies can be approximated to a 3rd order polynomial.

Stability

The stability of the identified model is guaranteed because it contains only poles lying in the left-half plane.

4.2 PI controller and Smith-predictor

As we have seen, the system is stable for the whole input range. In addition, the maximal stable value depends only on the choice for the temperature T_{cold} and is the same for all inputs. Therefore, in this case, the PI controller doesn't have to stabilize the system nor to track a reference signal.
For the control it is assumed that the transfer functions $\mathbf{P_D(s)}$ identified in the previous section can be approximated by a time delay:

$$\mathbf{P_D(s)} = \frac{1}{1 + s\tau_D(\dot{m}_f)} \approx e^{-s\tau_D(\dot{m}_f)}. \tag{4.10}$$

The transfer functions relating $\dot{m}_f$ to T_i become then, for $i = 1, 2, ..., 18$:

$$\begin{aligned} \mathbf{P_{\dot{m}_f \to T_i}(s)} &= \mathbf{P_1(s)P_D(s)}^{i-1} \\ &\approx \underbrace{G(\dot{m}_f)\frac{1}{1 + s\tau(\dot{m}_f)}}_{\mathbf{P_0(s)}} e^{-s\left(\tau_{P_1} + (i-1)\tau_D(\dot{m}_f)\right)}. \end{aligned} \tag{4.11}$$

The transfer function (**4.11**) is a first-order system with delay, where the gain, the pole and the delay depend on the input.

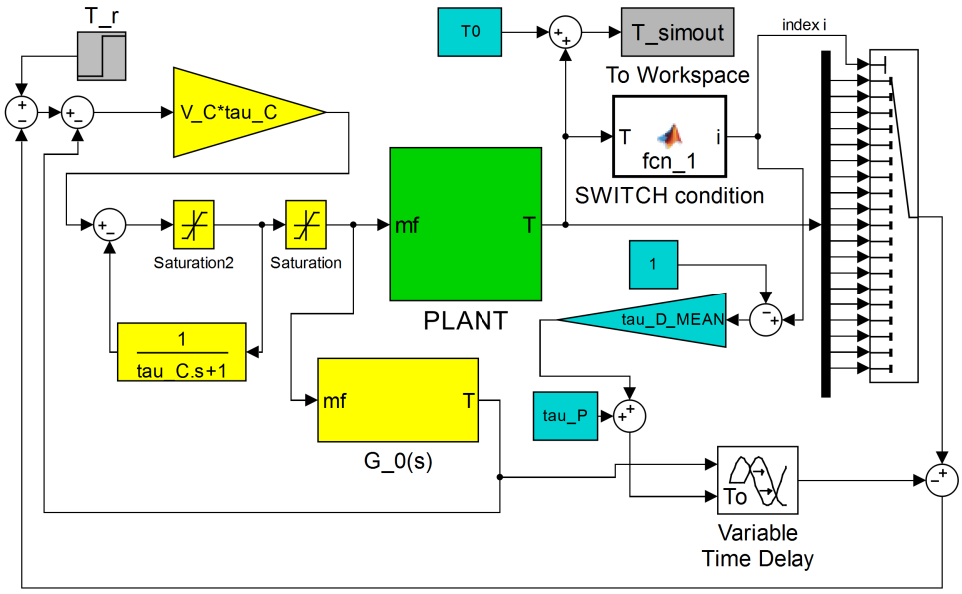

Figure 4.4: Block diagram in *Simulink* for the PI control combined with a Smith predictor

To control the system, a PI controller combined with a Smith predictor is used. At the beginning, the feedback variable for the control is the temperature T_1. In this case the delay of the model is equal to τ_{P_1}. As soon as the feedback variable satisfies some condition, such as having reached a certain rising percentage, the feedback switches to the next control variable, in this case T_2. At that point the system is controlled with the same PI controller, but the delay that must be taken into account by the Smith predictor is now equal to $\tau_{P_1} + \tau_D(\dot{m}_f)$. This procedure lasts until the system is fully charged. See figure **4.4** for the *Simulink* block-diagram used to perform the PI control combined with a Smith Predictor.

Parameters for the PI and for the Smith predictor

The PI controller that has to control the transfer function

$$\mathbf{P_0(s)} = G(\dot{m}_f)\frac{1}{1 + s\tau(\dot{m}_f)} \tag{4.12}$$

is of the type:

$$\mathbf{C(s)} = V_C \frac{\tau_C s + 1}{c}. \tag{4.13}$$

A good choice for placing the zero is

$$\tau_C = \tau(\dot{m}_f). \tag{4.14}$$

Due to causality the controller cannot depend on its output. Therefore it was chosen to use the value of $\tau(\dot{m}_f)$ linearized around the mean value of the input

$$\dot{m}_{f,MEAN} = \frac{\dot{m}_{f,MAX} - \dot{m}_{f,MIN}}{2} + \dot{m}_{f,MIN} = 0.165 kg/s \tag{4.15}$$

which results in

$$\tau_C = \tau(\dot{m}_{f,MEAN}) = 450 s. \tag{4.16}$$

The same linearization will be done for setting the gain of (**4.13**). The cutoff frequency will be equal to

$$\omega_c = G(\dot{m}_{f,MEAN}) G_C. \tag{4.17}$$

Choosing the desired rise time t_r, the gain of the controller can be calculated as follows:

$$G_C = \frac{1.5}{t_r \cdot G(\dot{m}_{f,MEAN})} = \frac{1.5}{t_r \cdot 1206}. \tag{4.18}$$

Similarly, the delay considered by the Smith predictor is

$$d_{i,SMITH} = \tau_{P_1} + (i-1)\tau_D(\dot{m}_{f,MEAN}) = [250 + (i-1)370] \ \mathbf{s} \tag{4.19}$$

where i is the index of the temperature being currently the feedback variable. The parameters of the transfer function $\mathbf{P_0(s)}$ depend on the input and its non-linearity will be included into the block diagram.
Furthermore, a simple Anti-Wind-Up was integrated into the PI-controller.
A huge variety of switching conditions could be used to perform the control, for example:

1. SWITCH as soon as the current feedback variable reaches 70% or 80%
2. SWITCH as soon as the next feedback variable reaches 30% or 20%

In addition, it is possible to choose to increase the index i always by 1, or for example by 2, skipping half of the outputs.

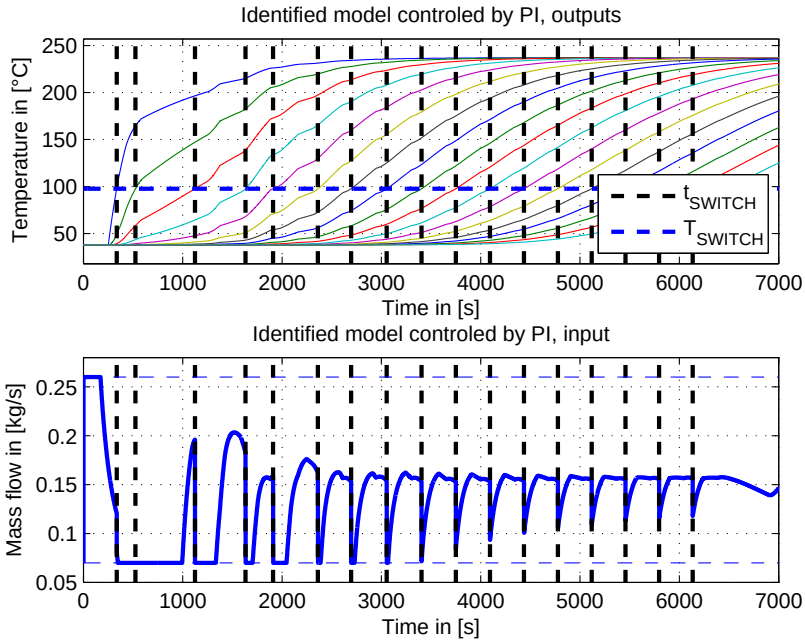

Figure 4.5: Simulation experiment 1 with PI controller and Smith predictor. Identified model

4.2.1 Simulation examples

Two different simulation experiments are shown in this section.

1. The results of the first experiment, which was held on the identified model, are shown in figure **4.5** . The parameters used for the control are the following:

   - Rise time: $t_r = 150\,\mathbf{s}$
   - Switch as soon as the next feedback variable reaches 30%
   - Increase the index of the feedback variable by 1

2. The second simulation was performed on the numerical model of the Thermal Storage System (figure **4.6**). The parameters used for the control are the following:

   - Rise time: $t_r = 100\,\mathbf{s}$

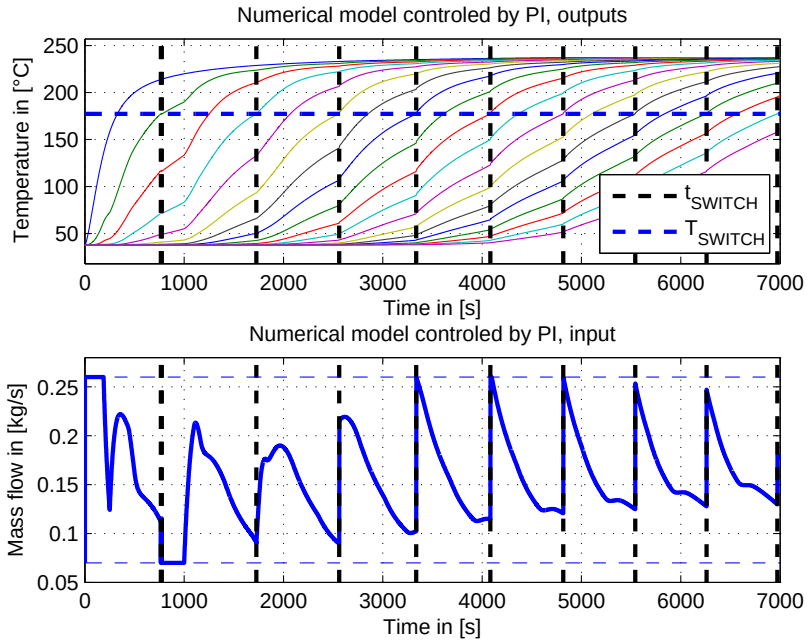

Figure 4.6: Simulation experiment 2 with PI controller and Smith predictor. Continuous time numerical model

- Switch as soon as the current feedback variable reaches 70%
- Increase the index of the feedback variable by 2

The charging performance of this simulation is:

$$(t_{charged}, \%_{charged}) = (4447\,\mathbf{s}, 65.8\%) \qquad (\mathbf{4.20})$$

4.2.2 Restrictions of controlling the Thermal Storage System using a PI controller

As we have seen in the previous section **4.2.1**, the control did not increase the charging performance. The problem is that, as we are dealing with a SIMO (Single Input, Multiple Output) system, only one input has to control more than one independent output. Hence, the PI controller cannot simultaneously control the temporal and the spatial behavior of the system. When one output is connected to the input through a feedback, the PI can control temporary this single input. The PI cannot, instead, control what happens

to the remaining outputs; this is the spatial behavior of the system.
We can clearly see, in figure **4.6**, that at the first switching point, the output T_1 was highered very fast. In the mean time, though, the following outputs and as well T_3, which is the next feedback variable, have already increased significantly, while they should instead still be close to T_{cold}.
Therefore, in the next chapter, an optimal control was performed. The optimal control can include in the cost function both temporal and spatial behavior of the system.

Chapter 5

Model predictive control

In this chapter the procedure for performing a linear Model Predictive Control (MPC) is outlined.
Then, ARMAX-identifications are done for various input-output pairs, in order to create an ARMAX model with parameters varying with the input.
This model is then used for the linear MPC of the system. Unluckily, the results are not as expected because there is a significant difference between prediction and actual behavior of the model.

5.1 MPC - theory

Model predictive control is a discrete time optimal control. The controller relies on a dynamic model of the plant which is usually determined by identification. Usually the identified model provides an adequate control of the system.
In each sampling instant, the MPC optimizes a finite time-horizon, but implements only the current timeslot. Then, for the next time step, the horizon is shifted by one time step and a new optimization is performed. The optimization tries to minimize the error between some reference function (for the output or for the state vector) and the predicted behavior of the system based on the identified model. Hence, the control is based on a prediction, but for each time step, the current behavior of the system is taken into account. This must be done due to possible identification errors.

5.1.1 Linear MPC

Here the approach for the LINEAR Model Predictive Control is described. First of all a time dependent reference $\mathbf{r}_k$ for each output of the identified model must be defined. Then a prediction horizon n_p must be chosen, which defines the number of time steps on which the error is optimized. The optimization variables are the inputs for each time step of the prediction horizon. Additionally, in order to decrease the computational effort, it can be chosen to set an optimization horizon n_o, after which the optimization variables are held constant. After each optimization, only the first value of the optimization variables is injected into the system, while a new optimization is performed. Be aware that the optimization must be performed within a single sampling interval.

Let

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k \end{aligned} \tag{5.1}$$

be a linear discrete-time ARMAX system where

- $\mathbf{x}_k \in \mathbb{R}^{n\text{x}1}$ is the state space vector of the system
- $\mathbf{u}_k \in \mathbb{R}^{m\text{x}1}$ is the input of the system
- $\mathbf{y}_k \in \mathbb{R}^{p\text{x}1}$ is the output of the system
- $A \in \mathbb{R}^{n\text{x}n}$
- $B \in \mathbb{R}^{n\text{x}m}$
- $C \in \mathbb{R}^{p\text{x}n}$

Defining

$$\Delta\mathbf{u}_k := \mathbf{u}_k - \mathbf{u}_{k-1}, \tag{5.2}$$

the first iteration step of the system can be rewritten as:

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k \\ &= A\mathbf{x}_k + B\mathbf{u}_{k-1} + B\Delta\mathbf{u}_k \\ \mathbf{y}_{k+1} &= C\mathbf{x}_{k+1}. \end{aligned} \tag{5.3}$$

It was chosen to use the $\Delta\mathbf{u}_i$ with $i = 1, 2, ..., n_o$ as optimization variables. The result (**5.3**) is then used for calculating the next iteration step:

$$\begin{aligned}
\mathbf{x}_{k+2} &= A\mathbf{x}_{k+1} + B\mathbf{u}_{k+1} \\
&= A\mathbf{x}_{k+1} + B\mathbf{u}_{k-1} + B\Delta\mathbf{u}_k + B\Delta\mathbf{u}_{k+1} \\
&= A^2\mathbf{x}_k + AB\mathbf{u}_{k-1} + AB\Delta\mathbf{u}_k + B\mathbf{u}_{k-1} + B\Delta\mathbf{u}_k + B\Delta\mathbf{u}_{k+1} \\
&= A^2\mathbf{x}_k + (AB + B)\mathbf{u}_{k-1} + (AB + B)\Delta\mathbf{u}_k + B\Delta\mathbf{u}_{k+1} \\
\mathbf{y}_{k+2} &= C\mathbf{x}_{k+2}.
\end{aligned} \tag{5.4}$$

We can see that the output $\mathbf{y}_{k+2}$ only depends on the known system matrices, the current state $\mathbf{x}_k$, the initial input-value $\mathbf{u}_{k-1}$ and on the optimization variables.
The same procedure is done for n_o steps with the final result:

$$\begin{aligned}
\mathbf{x}_{k+n_o} &= A^{n_o}\mathbf{x}_k + (A^{n_o-1} + A^{n_o-2} + ... + I)B\mathbf{u}_{k-1} + \\
&\quad + \sum_{j=1}^{n_o}(A^{n_o-j} + A^{n_o-j-1} + ... + A + I)B\Delta\mathbf{u}_{k+j-1} \\
\mathbf{y}_{k+n_o} &= C\mathbf{x}_{k+n_o}.
\end{aligned} \tag{5.5}$$

Setting $\Delta\mathbf{u}_i = 0$ for $i \geq n_o$, after $(n_p - n_o)$ more steps we get:

$$\begin{aligned}
\mathbf{x}_{k+n_p} &= A^{n_p}\mathbf{x}_k + (A^{n_p-1} + A^{n_p-2} + ... + I)B\mathbf{u}_{k-1} + \\
&\quad + \sum_{j=1}^{n_o}(A^{n_p-j} + A^{n_p-j-1} + ... + A + I)B\Delta\mathbf{u}_{k+j-1} \\
\mathbf{y}_{k+n_p} &= C\mathbf{x}_{k+n_p}.
\end{aligned} \tag{5.6}$$

At that point we want to summarize all n_p iteration steps and rewrite them in a matrix form. To do so let's define the vector of the outputs and of the optimization variables:

$$\underbrace{\overline{\mathbf{y}}_{k+1}}_{(p\cdot n_p \,\mathrm{x}\, 1)} := \begin{bmatrix} \mathbf{y}_{k+1} \\ \mathbf{y}_{k+2} \\ \vdots \\ \mathbf{y}_{k+n_p} \end{bmatrix} = \begin{bmatrix} C\mathbf{x}_{k+1} \\ C\mathbf{x}_{k+2} \\ \vdots \\ C\mathbf{x}_{k+n_p} \end{bmatrix}, \quad \underbrace{\overline{\Delta\mathbf{u}}_k}_{(m\cdot n_o \,\mathrm{x}\, 1)} := \begin{bmatrix} \Delta\mathbf{u}_k \\ \Delta\mathbf{u}_{k+1} \\ \vdots \\ \Delta\mathbf{u}_{k+n_o-1} \end{bmatrix}.$$

The entirety of the iteration steps can be summarized as follows:

$$\overline{\mathbf{y}}_{k+1} = \underbrace{F\mathbf{x}_k + G\mathbf{u}_{k-1}}_{=\mathbf{g}_k} + H\overline{\Delta\mathbf{u}}_k, \tag{5.7}$$

being F, G and H known matrices as they contain only the known system matrices A, B and C:

$$F_{(n_p p \,\mathrm{x}\, n)} := \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{n_p} \end{bmatrix}, \quad G_{(n_p p \,\mathrm{x}\, m)} := \begin{bmatrix} C(I)B \\ C(A+I)B \\ \vdots \\ C\left(\sum_{i=0}^{n_p-1} A^i\right) B \end{bmatrix}$$

$$\underbrace{H}_{(n_p p \,\mathrm{x}\, n_o m)} := \left[\begin{array}{c|c|c|c} C(I)B & \mathbf{0} & \cdots & \mathbf{0} \\ C(A+I)B & C(I)B & \mathbf{0} & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ C\left(\sum_{i=0}^{n_o-1} A^i\right) B & C\left(\sum_{i=0}^{n_o-2} A^i\right) B & \cdots & C(I)B \\ \hline C\left(\sum_{i=0}^{n_o} A^i\right) B & C\left(\sum_{i=0}^{n_o-1} A^i\right) B & \cdots & C(A+I)B \\ \vdots & \vdots & \ddots & \vdots \\ C\left(\sum_{i=0}^{n_p-1} A^i\right) B & C\left(\sum_{i=0}^{n_p-2} A^i\right) B & \cdots & C\left(\sum_{i=0}^{n_p-n_o} A^i\right) B \end{array}\right].$$

In equation (**5.7**)

- $\overline{\mathbf{y}}_{k+1}$, $\mathbf{x}_k$ and $\mathbf{u}_{k-1}$ are known or measurable $\rightarrow$ $\mathbf{g}_k$ is known
- $\overline{\Delta \mathbf{u}}_k$ is to be searched $\rightarrow$ optimization vector

In order to perform the MPC optimization, for the vector $\overline{\mathbf{y}}_{k+1}$ a corresponding reference output $\overline{\mathbf{r}}_{k+1}$ must be defined:

$$\underbrace{\overline{\mathbf{r}}_{k+1}}_{(p \cdot n_p \,\mathrm{x}\, 1)} := \begin{bmatrix} \mathbf{r}_{k+1} \\ \mathbf{r}_{k+2} \\ \vdots \\ \mathbf{r}_{k+n_p} \end{bmatrix}.$$

At that point, let's define the (known) error $\overline{\mathbf{e}}_k$ as the difference between the system behavior for $\overline{\Delta \mathbf{u}}_k = 0$[1] and the reference output:

$$\overline{\mathbf{e}}_k := \overline{\mathbf{g}}_k - \overline{\mathbf{r}}_{k+1}. \tag{5.8}$$

[1] This corresponds to the system if it is not controlled

Choosing the symmetric and positive definite matrices Q and R

$$Q_{(n_p p \,\mathrm{x}\, n_p p)} := \begin{bmatrix} \overline{Q}_{(p \,\mathrm{x}\, p)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \overline{Q}_{(p \,\mathrm{x}\, p)} & \ddots \\ \mathbf{0} & \ddots & \ddots \end{bmatrix} \tag{5.9}$$

$$R_{(n_o m \,\mathrm{x}\, n_o m)} := \begin{bmatrix} \overline{R}_{(m \,\mathrm{x}\, m)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \overline{R}_{(m \,\mathrm{x}\, m)} & \ddots \\ \mathbf{0} & \ddots & \ddots \end{bmatrix} \tag{5.10}$$

the scalar function to be minimized can, for example, be written as:

$$J(\overline{\Delta \mathbf{u}}_k) = (\overline{\mathbf{y}}_{k+1} - \overline{\mathbf{r}}_{k+1})^T Q (\overline{\mathbf{y}}_{k+1} - \overline{\mathbf{r}}_{k+1}) + (\overline{\Delta \mathbf{u}}_k)^T R (\overline{\Delta \mathbf{u}}_k). \tag{5.11}$$

Here the difference between system output and reference output is minimized, as well as the variation of the control variable $\mathbf{u}_k$. The matrices Q and R define the weigths on the errors.
Using (**5.7**) and (**5.8**), the (**5.11**) becomes:

$$\begin{aligned} J(\overline{\Delta \mathbf{u}}_k) &= (\overline{\mathbf{e}}_k + H\overline{\Delta \mathbf{u}}_k)^T Q (\overline{\mathbf{e}}_k + H\overline{\Delta \mathbf{u}}_k) + (\overline{\Delta \mathbf{u}}_k)^T R (\overline{\Delta \mathbf{u}}_k) \\ &= \overline{\mathbf{e}}_k^T Q \overline{\mathbf{e}}_k + (H\overline{\Delta \mathbf{u}}_k)^T Q \overline{\mathbf{e}}_k + \overline{\mathbf{e}}_k^T Q H \overline{\Delta \mathbf{u}}_k \\ &\quad + (H\overline{\Delta \mathbf{u}}_k)^T Q (H\overline{\Delta \mathbf{u}}_k) + (\overline{\Delta \mathbf{u}}_k)^T R (\overline{\Delta \mathbf{u}}_k). \end{aligned} \tag{5.12}$$

As the function is scalar, its summands can arbitrarily be transposed. In this case

$$\overline{\mathbf{e}}_k^T Q H \overline{\Delta \mathbf{u}}_k \xrightarrow{scalar} (\overline{\mathbf{e}}_k^T Q H \overline{\Delta \mathbf{u}}_k)^T = (H\overline{\Delta \mathbf{u}}_k)^T Q \overline{\mathbf{e}}_k \tag{5.13}$$

leads to

$$J(\overline{\Delta \mathbf{u}}_k) = \overline{\Delta \mathbf{u}}_k^T (H^T Q H + R) \overline{\Delta \mathbf{u}}_k + 2\overline{\Delta \mathbf{u}}_k (H^T Q \overline{\mathbf{e}}_k). \tag{5.14}$$

The term $\overline{\mathbf{e}}_k^T Q \overline{\mathbf{e}}_k$ was removed because it does not depend on the optimization variables.

Constraints

The main advantage of MPC is that constraints on input and on state-vector can easily be integrated into the optimization problem.
Here we will only consider the constraints on the input, such as:

$$\mathbf{u}_{MIN} \leq \mathbf{u}_k \leq \mathbf{u}_{MAX} \quad \forall \quad k \geq 0. \tag{5.15}$$

For all steps of the prediction horizon, the constraints can be rewritten in terms of $\overline{\Delta \mathbf{u}_k}$ like

$$\begin{aligned}
&\mathbf{u}_{MIN} \leq \mathbf{u}_k = \mathbf{u}_{k-1} + \Delta \mathbf{u}_k \leq \mathbf{u}_{MAX} \\
&\mathbf{u}_{MIN} \leq \mathbf{u}_{k+1} = \mathbf{u}_{k-1} + \Delta \mathbf{u}_k + \Delta \mathbf{u}_{k+1} \leq \mathbf{u}_{MAX} \\
&\qquad \vdots \\
&\mathbf{u}_{MIN} \leq \mathbf{u}_{k+n_o-1} = \mathbf{u}_{k-1} + \Delta \mathbf{u}_k + ... + \Delta \mathbf{u}_{k+n_o-1} \leq \mathbf{u}_{MAX},
\end{aligned} \tag{5.16}$$

what can be summarized as follows:

$$\overline{\mathbf{u}_{MIN}} \leq L\mathbf{u}_{k-1} + M\Delta \mathbf{u}_k \leq \overline{\mathbf{u}_{MAX}}, \tag{5.17}$$

where

$$\underbrace{L}_{(n_o m \,\mathrm{x}\, m)} := \begin{bmatrix} I_{(m \,\mathrm{x}\, m)} \\ \vdots \\ I_{(m \,\mathrm{x}\, m)} \end{bmatrix}, \quad \underbrace{M}_{(n_o m \,\mathrm{x}\, n_o m)} := \begin{bmatrix} I_{(m \,\mathrm{x}\, m)} & \mathbf{0} & \cdots & \mathbf{0} \\ I_{(m \,\mathrm{x}\, m)} & I_{(m \,\mathrm{x}\, m)} & \mathbf{0} & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ I_{(m \,\mathrm{x}\, m)} & I_{(m \,\mathrm{x}\, m)} & \cdots & I_{(m \,\mathrm{x}\, m)} \end{bmatrix}.$$

Equation **5.17** can be rewritten as linear matrix inequality (LMI)

$$W\overline{\Delta \mathbf{u}}_k \leq \overline{\omega}, \tag{5.18}$$

being

$$W_{(2n_o m \,\mathrm{x}\, n_o m)} := \begin{bmatrix} -M \\ M \end{bmatrix}, \quad \omega_{(2n_o m \,\mathrm{x}\, 1)} := \begin{bmatrix} -\overline{\mathbf{u}_{MIN}} + L\mathbf{u}_{k-1} \\ \overline{\mathbf{u}_{MAX}} - L\mathbf{u}_{k-1} \end{bmatrix}. \tag{5.19}$$

The optimization problem that must be solved for each step of the MPC can finally be established:

$$\boxed{\begin{aligned} \underset{\overline{\Delta \mathbf{u}_k}}{\text{minimize}} \quad & \overline{\Delta \mathbf{u}_k}^T (H^T Q H + R)\overline{\Delta \mathbf{u}_k} + 2\overline{\Delta \mathbf{u}_k}(H^T Q \overline{\mathbf{e}}_k) \\ \text{subject to} \quad & W\overline{\Delta \mathbf{u}_k} \leq \overline{\omega} \end{aligned}} \tag{5.20}$$

This type of optimization problem is a so called Quadratic Programming (QP):

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^T A x + 2x^T b \\ \text{subject to} \quad & Cx \leq y. \end{aligned} \tag{5.21}$$

The huge advantage of such a problem is that the function that must be minimized, as well as the domain, is convex. This means that any minimum of the function is as well the global minimum. The computational effort is relatively low and there exist a huge variety of methods which can easily solve the optimization.

5.2 ARMAX-Identification

As mentioned in the previous section, in order to perform the MPC, a linear ARMAX-model of the system is needed:

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k. \end{aligned} \tag{5.22}$$

In chapter **2**, a nonlinear discrete-time numerical model of the laboratory plant was derived. It can be rewritten in the form:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_k &= C\mathbf{x}_k. \end{aligned} \tag{5.23}$$

This model cannot be used for the linear MPC. Therefore we must perform an estimation on the model in order to get a model like (**5.22**). It is clear that the estimation cannot be a linear model which describes the system behavior for all possible inputs, therefore the aim is to find an ARMAX-model with the matrix parameters varying with the input, i.e:

$$\begin{aligned} \mathbf{x}_{k+1} &= A(\mathbf{u}_k)\mathbf{x}_k + B(\mathbf{u}_k)\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k. \end{aligned} \tag{5.24}$$

To do so, a black-box approach is used; that means that the estimation is only based on input-output pairs and not on some previous knowledge of the model. In this case the MATLAB-function *armax* was used.[2] The parameters used for the estimation are extensively explained in the following sections.

[2]The principle behind the *armax* function is to do a least square minimization of the error between measured output and predicted output.
The estimator $\hat{y}$ for the output y is defined as:

$$\hat{y}(k) = A(z)y(k) + B(z)u(k)$$

5.2.1 Definition of model parameters for the linear systems

To get the matrices $A(\mathbf{u}_k)$ and $B(\mathbf{u}_k)$ depending on the input $\mathbf{u}_k$, first of all the range of the input $[\mathbf{u}_{MIN}, \mathbf{u}_{MAX}]$ is divided into r parts. For each part a linear systems is estimated that is assumed to be valid in the entire interval. Then, an analytical relation (or at least a piecewise-defined function) that describes how the parameters vary with $\mathbf{u}_k$ must be found.

Inputs

First of all, $r = 9$ different estimations were performed on input-output pairs with inputs:

$$\mathbf{u}_{k,q} = \delta_{-1,k} \cdot q + \vartheta_k \quad q \in Q.^3 \tag{5.25}$$

where ϑ_k is a triangle wave with period $p = 2000\,\mathbf{s}$ and amplitude $a = 0.01\,\mathbf{kg/s}$, defined as:

$$\vartheta_k := \left| 2 \left(\frac{t}{p} - \left\lfloor \frac{t}{p} + \frac{1}{2} \right\rfloor \right) \right|.^4 \tag{5.26}$$

The constant value q is chosen from the ordered set Q containing $r = 9$ elements:

$$\begin{aligned} Q &= \{q_1 < q_2 < ... < q_r\} = \\ &= \{0.07,\ 0.095,\ 0.12,\ 0.15,\ 0.165,\ 0.18,\ 0.21,\ 0.24,\ 0.26\}\ \mathbf{kg/s}. \end{aligned} \tag{5.27}$$

The values of Q are almost equispatially distributed and cover the whole domain of $\mathbf{u}_k$, including both maximum and minimum value.

---

with

$$A(z) = a_1 z^{-1} + ... + a_{n_a} z^{n_a}, \quad B(z) = b_1 z^{-1} + ... + b_{n_b} z^{n_b}.$$

Note that the regressor of the estimator is not the estimator itself, but the output y. The model parameters $\mathbf{p} = \begin{bmatrix} a_1 & \cdots & a_{n_a} & b_1 & \cdots & b_{n_b} \end{bmatrix}^T$ are found by minimizing the function

$$f(\mathbf{p}) = \sum_{k=1}^{N} (\hat{y}(k) - y(k))^2$$

[3] The function $\delta_{-1,k}$ is the step function defined as: $\delta_{-1,k} = \begin{cases} 1 & \text{for } k \geq 1 \\ 0 & \text{elsewhere} \end{cases}$

[4] The function $f(x) = \lfloor x \rfloor$ is the floor function and is defined as $\max\{k \in \mathbb{Z} | k \leq x\}$

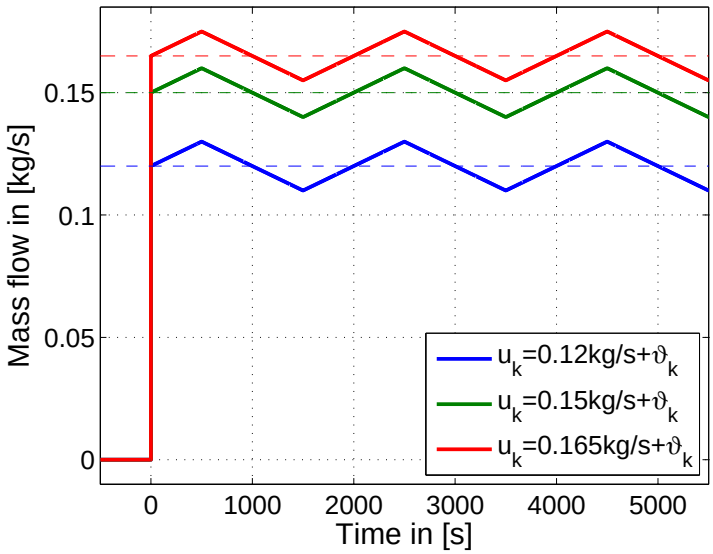

Figure 5.1: Input functions $u_{k,q}$ with $q = 0.12\,\mathbf{kg/s}$, $q = 0.15\,\mathbf{kg/s}$ and $q = 0.165\,\mathbf{kg/s}$ used for ARMAX-identification

The triangle wave was added to the step functions to perform an estimation that is valid in a neighborhood of the constant value q.
Figure **5.1** shows the three inputs (**5.25**) with $q = 0.12\,\mathbf{kg/s}$, $q = 0.15\,\mathbf{kg/s}$ and $q = 0.165\,\mathbf{kg/s}$, respectively.

Outputs

As a compromise between the order of the ARMAX-model and control accuracy, not the entirety of the outputs[5] was used for the estimation, but only the 7 outputs placed at:

$$x_{armax} = \begin{bmatrix} 10 & 30 & 60 & 900 & 120 & 150 & 180 \end{bmatrix}^T \mathbf{cm}. \qquad (\mathbf{5.28})$$

These outputs will be called $T_{i,ax}$ with $i = 1, 2, ..., 7$. The output measured at $x = 0\,\mathbf{cm}$ was not used because it represents the boundary condition which is the same for all inputs. It is highly important, instead, that the temperature measured at $x = 180\,\mathbf{cm}$ is included, because it is the temperature that determines when the coil is assumed to be fully charged.

[5]The outputs are the temperatures of the fluid flowing through the coil, hence T_f

Sample time

The sample time used is $T_s = 5\,\mathbf{s}$. It was deliberately chosen to use a low sampling frequency, because in general the system dynamics are slow and the sufficient time for the optimization must be assured.

Order of the ARMAX-model

The ARMAX-model has the following form:

$$\left(I_{(7\,\mathrm{x}\,7)} + \underbrace{\begin{bmatrix} A_{11}(z) & A_{12}(z) & \cdots & A_{17}(z) \\ A_{21}(z) & A_{22}(z) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A_{71}(z) & \cdots & \cdots & A_{77}(z) \end{bmatrix}}_{\mathbf{A}(z)} \right) \begin{bmatrix} T_{1,ax} \\ T_{2,ax} \\ \vdots \\ T_{7,ax} \end{bmatrix} = \underbrace{\begin{bmatrix} B_1(z) \\ B_2(z) \\ \vdots \\ B_7(z) \end{bmatrix}}_{\mathbf{B}(z)} \dot{m}_f. \tag{5.29}$$

The parameters and the order n of all the components of the matrices $\mathbf{A}(z)$ and $\mathbf{B}(z)$ are defined as:

$$\begin{aligned} A_{ij}(z) &= a_{ij,1}z^{-1} + a_{ij,2}z^{-2} + ... + a_{ij,n}z^{-n}, \\ B_i(z) &= b_{i,1}z^{-1} + b_{i,2}z^{-2} + ... + b_{i,n}z^{-n}. \end{aligned} \tag{5.30}$$

After various experiments, the orders chosen for both matrices are:

$$n_{\mathbf{A}(z)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}, \qquad n_{\mathbf{B}(z)} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{5.31}$$

First of all, note that all the outputs $T_{i,ax}$ can only depend on the outputs $T_{j,ax}$ with $j \leq i$. It is clear that a temperature at the end of the coil can depend on a temperature at the beginning of the coil, but not vice versa. Secondly, it was chosen to obtain a system without feedthrough.[6]

[6]The system

$$\begin{aligned} x[n+1] &= Ax[n] + Bu[n] \\ y[n] &= Cx[n] + Du[n] \end{aligned}$$

Realization and State variables

The minimal realization of the ARMAX system obtained has 14 state-space variables and could be easily computed using MATLAB. The problem concerning the minimal realization is that we don't know what the single state-space variables (except from the ones that are the outputs) describe. In section **5.1**, though, it was explained that for each step of the MPC the vector $\mathbf{x}_k$ must be known or measurable even on the numerical model or on the plant.
Therefore a realization with 15 state-space variables is used, where each variable is measurable or can be obtained by the previous measurements. Using the parameters of (**5.29**) which are defined as (**5.30**), the realization is:

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k, \end{aligned} \tag{5.32}$$

with

$$A = \left[\begin{array}{cccc:cccc:c} a_{11,1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & b_{1,2} \\ a_{21,1} & a_{22,1} & \ddots & \vdots & a_{21,2} & a_{22,2} & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 & \vdots \\ a_{71,1} & a_{72,1} & \cdots & a_{77,1} & a_{71,2} & a_{72,2} & \cdots & a_{77,2} & 0 \\ \hdashline 1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots & & & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & 0 \\ \hdashline 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 \end{array}\right], \tag{5.33}$$

$$B = \left[\begin{array}{c} b_{1,1} \\ b_{2,1} \\ \vdots \\ b_{7,1} \\ \hdashline 0 \\ \vdots \\ \vdots \\ 0 \\ \hdashline 1 \end{array}\right], \quad C = \left[\begin{array}{c:c} \mathbf{I}_{(7\,\mathrm{x}\,7)} & \mathbf{0}_{(7\,\mathrm{x}\,8)} \end{array}\right], \quad \mathbf{x}_k = \left[\begin{array}{c} T_{1,ax} \\ T_{2,ax} \\ \vdots \\ T_{7,ax} \\ \hdashline x_{1,ax} \\ x_{2,ax} \\ \vdots \\ x_{7,ax} \\ \hdashline x_{u,ax} \end{array}\right][k]. \tag{5.34}$$

with $D = 0$ is a system without feedthrough

The first 7 states are directly measurable, because they are at the same time system outputs. The next 7 states are measurable as well, because they are the first 7 states with a unit delay:

$$\begin{bmatrix} x_{1,ax} \\ x_{2,ax} \\ \vdots \\ x_{7,ax} \end{bmatrix}[k+1] = \begin{bmatrix} T_{1,ax} \\ T_{2,ax} \\ \vdots \\ T_{7,ax} \end{bmatrix}[k]. \tag{5.35}$$

The last state instead is the input with a unit delay:

$$x_{u,ax}[k+1] = u[k]. \tag{5.36}$$

Nomenclature

All linear ARMAX models obtained will be referred to as $AX_q = ss(A_q, B_q, C, D)$, for example $AX_{0.07}$. The respective matrices of the system will be $A_{0.07}$ and $B_{0.07}$.

5.2.2 Identification results of the linear systems

The accuracy of the identified linear ARMAX models is discussed here. All ARMAX models were obtained like explained in **5.2.1** using a triangle function as an input. Figure **5.2** shows three tests performed on the linear models $AX_{0.07}$, $AX_{0.165}$ and $AX_{0.26}$ using the corresponding CONSTANT test functions. The graphs compare the ARMAX model outputs with the output of the numerical model described in section **2.3**.

The main difficulty of the identification procedure was to define parameters for the identification which give good results for the whole range of the input. We can see for example that for low mass flows the errors are higher than for higher mass flows, whereas the undershoot is almost negligible for low mass flows, but significant for higher $\dot{m}_f$. For higher mass flows the outputs have undesired oscillations while they should instead be constant at T_{cold}. For the whole range overshoot phenomena do not occur. Table **5.1** summarizes some important properties of the identification error, defined as

$$e_{ident} = T_{ax} - T_{f,num}, \tag{5.37}$$

such as:

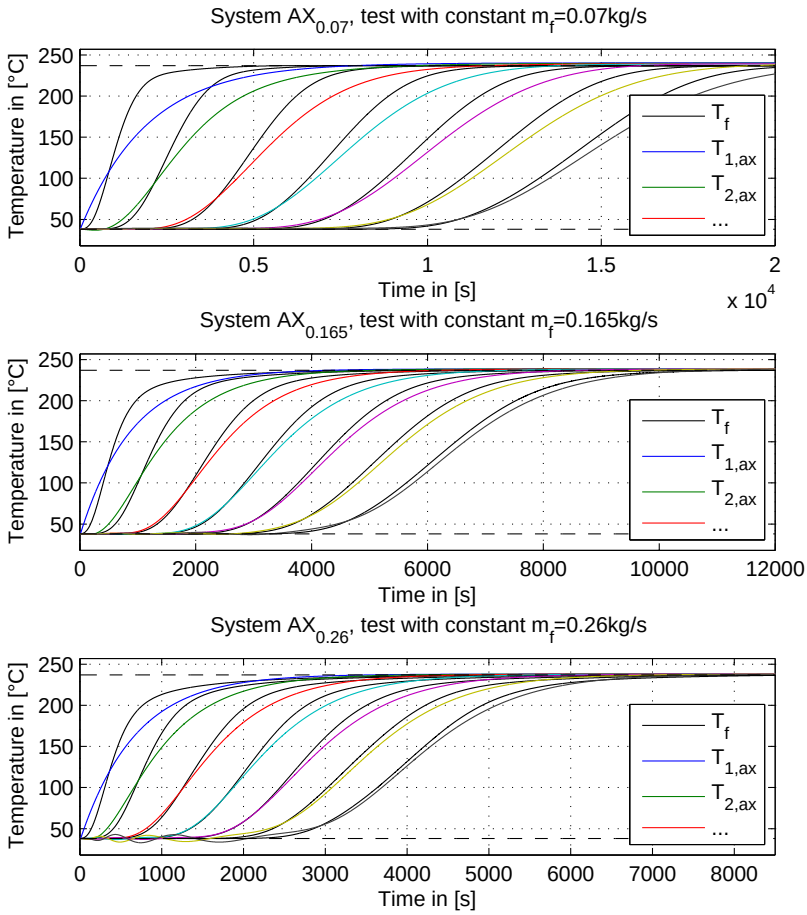

Figure 5.2: Test on the ARMAX models $AX_{0.07}$, $AX_{0.165}$ and $AX_{0.26}$. The ARMAX models are tested with the constant inputs $\dot{m}_f = 0.07\,\mathbf{kg/s}$, $\dot{m}_f = 0.165\,\mathbf{kg/s}$ and $\dot{m}_f = 0.26\,\mathbf{kg/s}$, respectively. The results are compared to the numerical model described in section **2.3**.

	$\mathbf{e_{MAX}}$ in [%]	$\mathbf{e_{MEAN}}$ in [%]	$\mathbf{e_{\infty,MEAN}}$ in [%]	**us** in [%]
$\mathbf{AX_{0.07}}$	25.55	1.77	1.38	0.55
$\mathbf{AX_{0.165}}$	14.87	0.7178	0.86	0.49
$\mathbf{AX_{0.26}}$	13.54	1.67	0.52	2.56

Table 5.1: ARMAX Identification: error properties

- The maximal percentage error: $e_{MAX} = \frac{\max|e|}{T_{hot}-T_{cold}} \cdot 100\%$
- The mean absolute value of the error: $e_{MEAN} = \frac{\text{mean}|e|}{T_{hot}-T_{cold}} \cdot 100\%$
- The percentage error for $t \to \infty$: $e_{\infty,MEAN} = \frac{\text{mean}|e_{t\to\infty}|}{T_{hot}-T_{cold}} \cdot 100\%$
- The undershoot percentage: $us = \frac{\min(T_{ax})-T_{hot}}{T_{hot}-T_{cold}} \cdot 100\%$

In general the errors are negative; that means that the ARMAX models are slower than the numerical model. This could be solved by using a higher order for the ARMAX model, which would unfortunately lead to stronger oscillations.
The biggest error is the one corresponding to the first output. This error is similar to a delay and probably this delay is transmitted to all the following outputs.

5.2.3 Combination of the linear systems

After estimating the 9 linear ARMAX models that describe the behavior of the model for some set input, it is necessary to find a relation that describe how the parameters of the matrices A and B vary with the input $\dot{m}_f$.[7] Each parameter of A and B, except from the ones that are equal to 0 or 1, can be written as a $(9\,\text{x}\,1)$-vector:

$$\mathbf{a}_{ij,ax} = \left[\; a_{0.07} \quad a_{0.095} \quad a_{0.12} \quad a_{0.15} \quad a_{0.165} \quad a_{0.18} \quad a_{0.21} \quad a_{0.24} \quad a_{0.26} \;\right]^T_{ij,ax}$$

$$\mathbf{b}_{l,ax} = \left[\; b_{0.07} \quad b_{0.095} \quad b_{0.12} \quad b_{0.15} \quad b_{0.165} \quad b_{0.18} \quad b_{0.21} \quad b_{0.24} \quad b_{0.26} \;\right]^T_{l,ax}.$$

[7] The matrix C is the same for all inputs

In figure **5.3**, as an example, the four vectors corresponding to the parameters $a_{55,1}$, $a_{75,2}$, $b_{1,1}$ and $b_{5,1}$, respectively, are described. The vectors of the first two graphs show a "regular" dependence on the mass flow. In the third and fourth graph, the elements of the vectors are almost randomly distributed.
Three different procedures to describe some correlation in the vectors $\mathbf{a}_{ij,ax}$ and $\mathbf{b}_{l,ax}$ are proposed in the following.

Combination by polynomial fitting → UNSTABLE

The first attempt to describe the parameters depending on the input was done using a polynomial fitting.
To do so the parameters were divided into three groups:

- Parameters with a small mean value:

  $$|\text{mean}(\mathbf{a}_{ij,ax})| \leq \epsilon_{mean}, \quad |\text{mean}(\mathbf{b}_{l,ax})| \leq \epsilon_{mean}.$$

  These parameters are assumed to be constant and equal to the mean value.

- Parameters with a small variance:

  $$\text{var}(\mathbf{a}_{ij,ax}) \leq \epsilon_{var}, \quad \text{var}(\mathbf{b}_{l,ax}) \leq \epsilon_{var}.$$

  These parameters are as well assumed to be constant and equal to the mean value.

- Parameters that do not belong to neither of the previous groups are fitted with a polynomial of nth order.

The variables ϵ_{mean}, ϵ_{var} and n can be chosen arbitrarily.
A polynomial fitting of 4th order of some of the vectors is shown in figure **5.3**. For the vectors $\mathbf{a_{55,1,ax}}$ and $\mathbf{b_{1,1,ax}}$ the fit is quite accurate. For the vectors $\mathbf{a_{75,2,ax}}$ and $\mathbf{b_{5,1,ax}}$, instead, the vector can badly be described by a polynomial function. Unlike the vector $\mathbf{b_{5,1,ax}}$, the vector $\mathbf{a_{75,2,ax}}$ has a high variance and the polynomial fit leads to major errors.
It is clear that the system obtained can be described analytically and would therefore be useful for various control methods.
This procedure, though, did not lead to a stable system for the majority of the inputs. Even setting $\epsilon_{mean} = \epsilon_{var} = 0$ and choosing a high order n, it was not possible to obtain a stable system. The poles of the various linear ARMAX models are very close to the unity circle and hence even a very slight perturbation of the parameters can make the system unstable.

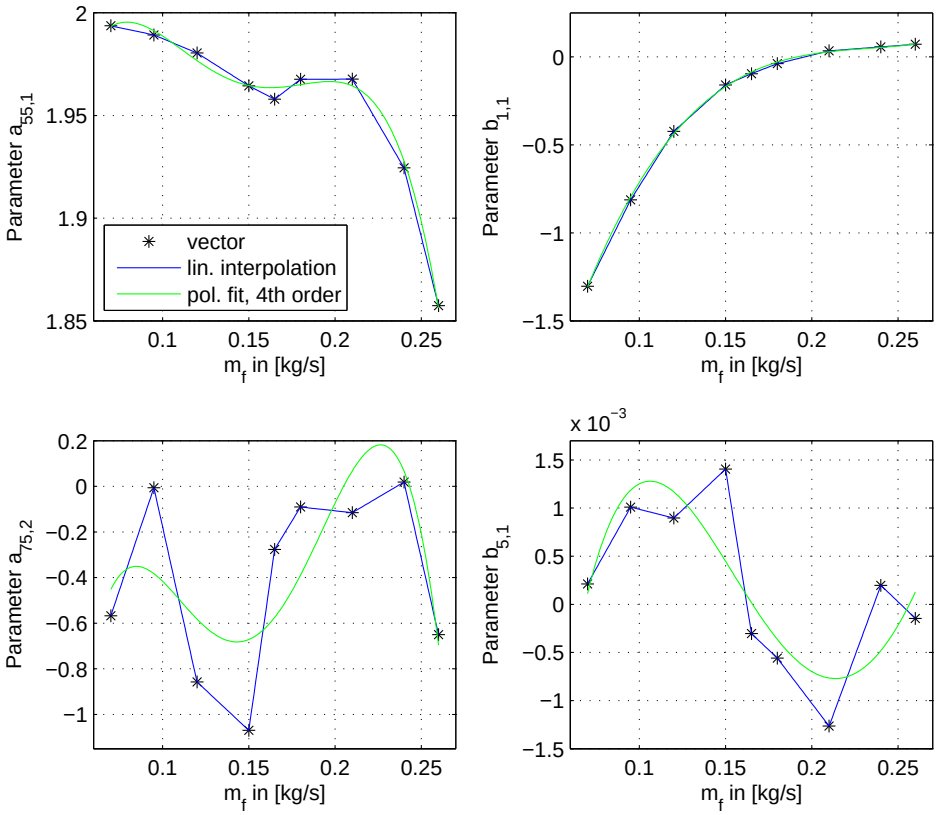

Figure 5.3: Example: the four vectors corresponding to the parameters $a_{55,1}$, $a_{75,2}$, $b_{1,1}$ and $b_{5,1}$ with respective linear interpolation and polynomial fitting of 4th order

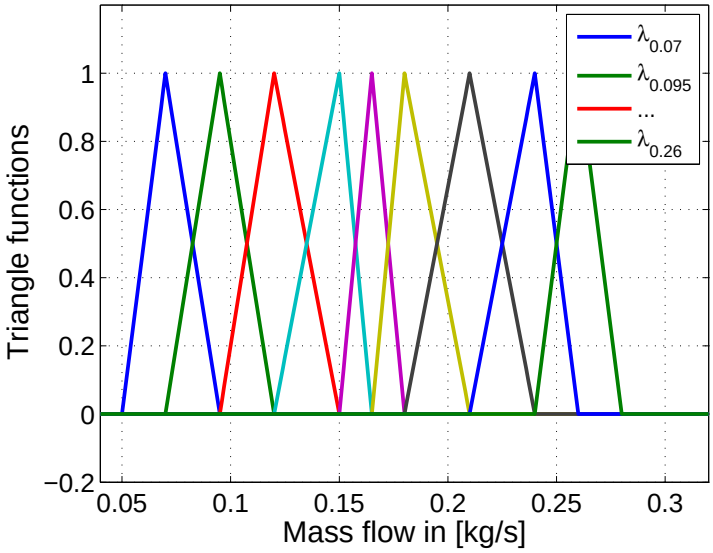

Figure 5.4: Triangle functions used for linear interpolation

Combination by interpolation → ACCURATE

The second procedure for combining the linear ARMAX models was to perform a linear interpolation for all the vectors $\mathbf{a}_{ij,ax}$ and $\mathbf{b}_{l,ax}$. The problem behind this procedure is that the model obtained is only piecewise analytical. The triangle functions $\lambda_q(\dot{m}_f)$ used for linear interpolation are shown in figure **5.4**, being

$$q \in Q = \{0.07,\, 0.095,\, 0.12,\, 0.15,\, 0.165,\, 0.18,\, 0.21,\, 0.24,\, 0.26\}\,.$$

The nonlinear ARMAX-model can now be written as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \left[\sum_{q\in Q} A_q \cdot \lambda_q(\mathbf{u}_k)\right]\mathbf{x}_k + \left[\sum_{q\in Q} B_q \cdot \lambda_q(\mathbf{u}_k)\right]\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k. \end{aligned} \tag{5.38}$$

In this case it is clear that, $\forall g \in Q$:

$$\left[\sum_{q\in Q} A_q \cdot \lambda_q(\mathbf{u}_g)\right] = A_g, \quad \left[\sum_{q\in Q} B_q \cdot \lambda_q(\mathbf{u}_g)\right] = B_g. \tag{5.39}$$

An example of linear interpolation is shown in figure **5.3**.
To do a stability test the input domain $[u_{MIN}, u_{MAX}]$ was divided into $N = 1000$ parts and an eigenvalue test was done for each discretization point. As the system was stable for each discretization point, we can assume that the model is stable on the whole domain.
However, in general, it is not assured that this approach leads to a stable model. In appendix **A**, a Monte Carlo method was held in order to compute the probability of obtaining a stable model with this approach.
For each discretization point the final stable value of all outputs was computed and is shown in figure **5.5**. The actual final value of the laboratory plant and of the numerical system is T_{hot}. The dashed vertical lines are placed at the $\dot{m}_f$ which correspond to a liner ARMAX model AX_q, $q \in Q$. It is clear that the gain is a little bit overestimated even for these values; the values between the points $q \in Q$ are even more overestimated. In general the error is smaller for higher mass flows and for $T_{i,ax}$ with a small i. All in all it's a good result because the percentage errors

$$e_{t\to\infty,\%} = \frac{T(t \to \infty)}{T_{hot} - T_{cold}} \cdot 100 \tag{5.40}$$

are inside the range of $[0.5\%, 2.2\%]$.

Combination by Gaussian interpolation → ANALYTICAL

Another possible way of combining the linear ARMAX models is to perform a Gaussian interpolation on the parameters. The procedure is similar to the linear interpolation, but here Gaussian functions are used instead of triangle functions. The functions, which are shown in figure **5.6**, are defined as:

$$\gamma_q(\dot{m}_f) = \exp\left\{-\frac{1}{2}\left(\frac{\dot{m}_f - q}{\sigma}\right)^2\right\} \quad \forall q \in Q, \tag{5.41}$$

with variance

$$\sigma = \frac{1}{r-1}\sum_{i=1}^{r-1}(q_{i+1} - q_i) \quad \text{with} \quad Q = \{q_1 < q_2 < \cdots < q_r\}. \tag{5.42}$$

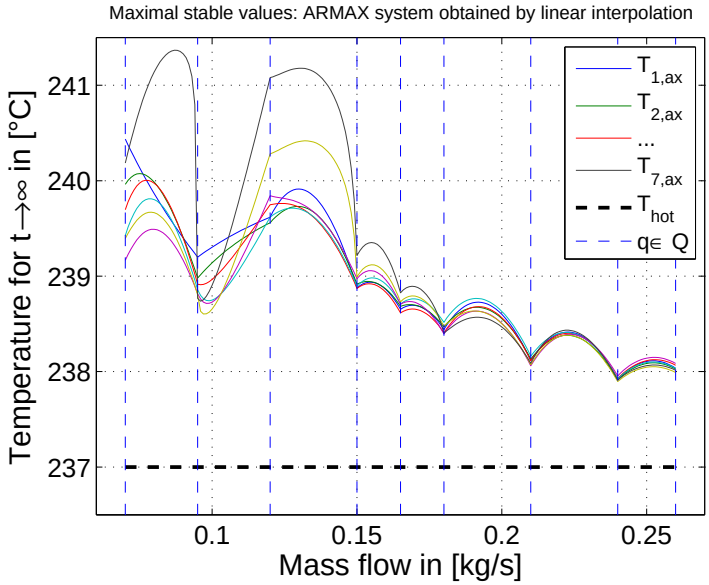

Figure 5.5: Final stable value of the outputs of the ARMAX model obtained by linear interpolation

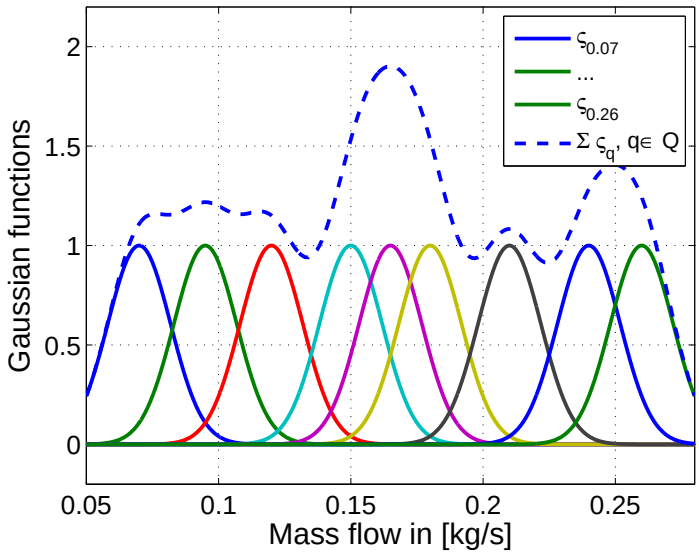

Figure 5.6: Gauss functions used for Gaussian interpolation

The nonlinear ARMAX-model can be written analytically for the whole input range as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \frac{\sum_{q\in Q} A_q \cdot \gamma_q(\mathbf{u}_k)}{\sum_{q\in Q} \gamma_q(\mathbf{u}_k)}\mathbf{x}_k + \frac{\sum_{q\in Q} B_q \cdot \gamma_q(\mathbf{u}_k)}{\sum_{q\in Q} \gamma_q(\mathbf{u}_k)}\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k. \end{aligned} \tag{5.43}$$

Unlike the case with linear interpolation, in this case, $\forall g \in Q$:

$$\frac{\sum_{q\in Q} A_q \cdot \gamma_q(\mathbf{u}_g)}{\sum_{q\in Q} \gamma_q(\mathbf{u}_g)} \neq A_g, \quad \frac{\sum_{q\in Q} B_q \cdot \gamma_q(\mathbf{u}_g)}{\sum_{q\in Q} \gamma_q(\mathbf{u}_g)} \neq B_g. \tag{5.44}$$

Here as well the stability was confirmed by the eigenvalue test on $N = 1000$ discretization points. The final stable values at these points are shown in figure **5.7**. We can clearly see that for some inputs the gain is overestimated, while for other inputs it is underestimated. The percentage error defined in (**5.40**) varies for the whole output range between 0% and 4.5%. This error is much bigger than the one given by the linear interpolation; in general this model has a worst accuracy.

5.3 Linear MPC for the Thermal Storage System

In section **5.1**, the procedure for MPC applied to a linear ARMAX-model is explained. It is highly important that the model is linear, because in that case the optimization problem is a convex program and can hence be easily solved.

The identification of the Thermal Storage System proposed in section **5.2** leads to an ARMAX-model with parameters varying with the input:

$$\begin{aligned} \mathbf{x}_{k+1} &= A(\mathbf{u}_k)\mathbf{x}_k + B(\mathbf{u}_k)\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k. \end{aligned} \tag{5.45}$$

The matrices of the ARMAX-model were constructed in a way that assures that the state-vector can be obtained by knowing the output, which is measurable.

Even though it cannot be described analytically, the model obtained by performing a linear interpolation led to the most accurate model and will therefore be used for the MPC.

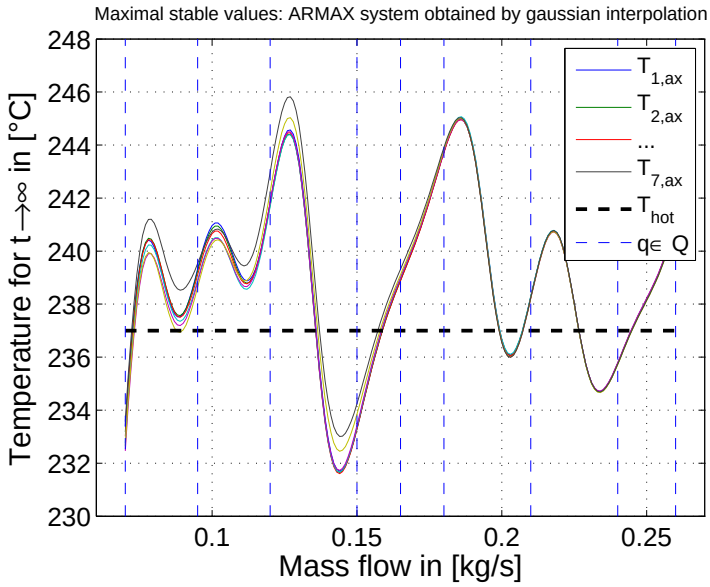

Figure 5.7: Final stable value of the outputs of the ARMAX model obtained by Gaussian interpolation

For the control of the Thermal Storage System, for each step the optimization was done on the linear ARMAX-model obtained by linearizing the model (**5.45**) around the current input. This means that the MPC is based on a linear prediction, but for each step the linear model used is updated.
Algorithm **1** explains the whole procedure used for performing the MPC. The algorithm was implemented using *Simulink*: see figure **5.8** for the block diagram. The optimization problem (**5.46**) was solved using the MATLAB-Toolbox YALMIP. [13]

5.3.1 Control specifications

Before showing some control examples using linear MPC, some control specifications are still needed.

Reference temperature

First of all the reference temperature must be set. In section **3.2.2**, it was exposed how the reference must be set in order to achieve a good charging

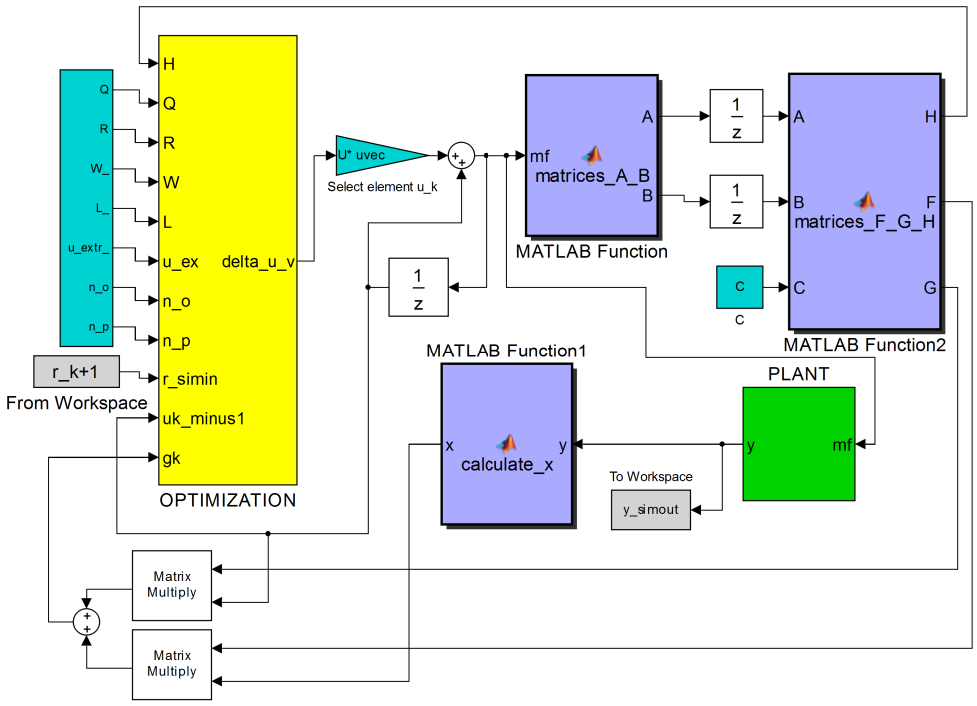

Figure 5.8: Block diagram in *Simulink* for the linear MPC with system update at each step

performance. For the MPC, for example, a piecewise linear function for $\mathbf{r}_k$ is used like shown in the first graph of figure **5.9**. Each element of the vector function $\mathbf{r}_k$ starts at T_{cold} and reaches finally T_{hot}. The degrees of freedom for this functions are:

- The time at which the function starts to increase: $t_{start,i}$, $i = 1, 2, ..., 7$. It is necessary that:

$$t_{start,i-1} \leq t_{start,i} \leq t_{start,i+1}.$$

  The parameter $t_{start,7}$ will approximately be equal to $t_{charged}$. Therefore this parameter must be set somewhere between the minimal time of charge and the maximal time of charge. As explained in section **3.1**, these time of charge occur if the thermocline is charged with the constant mass flow $\dot{m}_{f,MIN}$ and $\dot{m}_{f,MAX}$, respectively.
  Defining

$$\Delta t_{charged,i} := t_{start,i+1} - t_{start,i}, \tag{5.47}$$

Algorithm 1 Linear MPC with system update at each step

Require: $\overline{r_k}$, C, Q, R, W, $\overline{\omega}$, n_o, n_p, k_{end}, $n = 15$, $m = 1$, $p = 7$, $T_s = 5$

1: **Initialize:** u_{k-1}, $\mathbf{x}_k = \begin{bmatrix} T_{cold} & \cdots & T_{cold} & u_{k-1} \end{bmatrix}^T$
2: Calculate $A(u_{k-1})$ and $B(u_{k-1})$ using the estimated ARMAX-models combined by linear interpolation
3: Calculate the matrices H, F and G from $A(u_{k-1})$, $B(u_{k-1})$ and C
4: Calculate $\mathbf{g}_k = F\mathbf{x}_k + G\mathbf{u}_{k-1}$ and $\overline{\mathbf{e}}_k = \overline{\mathbf{g}}_k - \overline{\mathbf{r}}_{k+1}$
5: **while** $t < Ts$ **do**
6: Get $\overline{\Delta u_k}$ by solving the optimization problem:

$$\begin{aligned} \underset{\overline{\Delta u_k}}{\text{minimize}} \quad & \overline{\Delta u_k}^T (H^T Q H + R)\overline{\Delta u_k} + 2\overline{\Delta u_k}(H^T Q \overline{\mathbf{e}}_k) \\ \text{subject to} \quad & W\overline{\Delta u_k} \leq \overline{\omega} \end{aligned} \tag{5.46}$$

7: **end while**
8: **if** $t = T_s$ **then**
9: $\overline{\Delta u_k} = \overline{\Delta u_{k-1}}$
10: **end if**
11: Select the element Δu_k and calculate $u_k = u_{k-1} + \Delta u_k$
12: Measure $\mathbf{y}_{k+1}$ $\rightarrow$ Calculate $\mathbf{x}_{k+1}$
13: Wait until $t = T_s$ $\rightarrow$ set u_k as the input of the plant
14: **Set:** $k = k + 1$, $t = 0$
15: **if** $k \leq k_{end}$ **then**
16: go to **2**
17: **end if**
18: **return**

in order to make the reference temperature more realistic, it is advised to set

$$\Delta t_{charged,i} \geq \Delta t_{charged,i+1} \geq \cdots \quad \text{for } i = 1, 2, ..., 6 \tag{5.48}$$

- The slope of the function: s_i, $i = 1, 2, ..., 7$. A possible choice is to use the same value for all functions

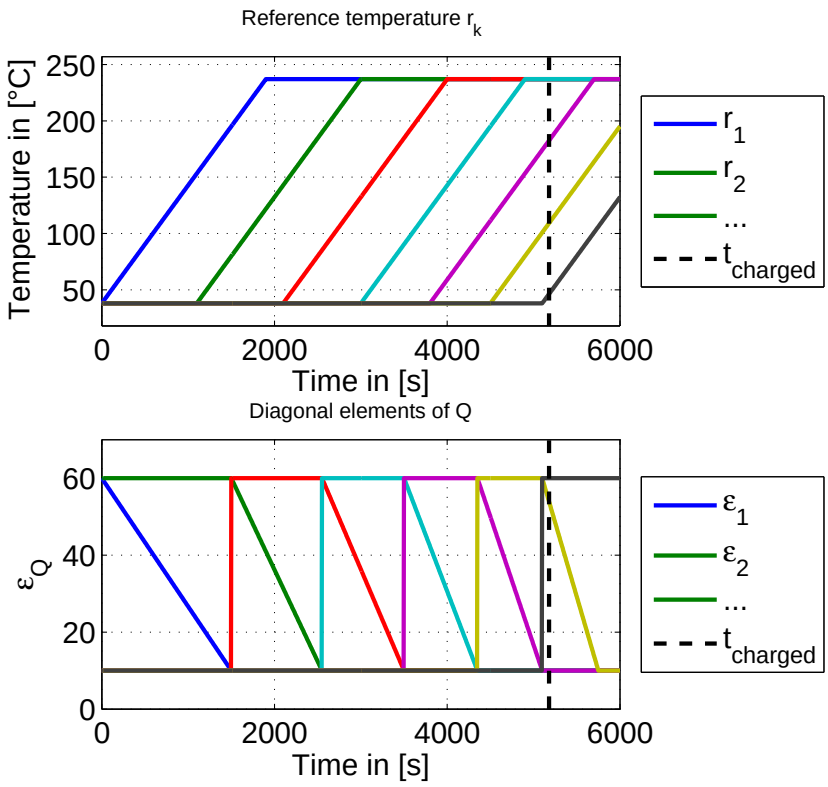

Figure 5.9: Example for the reference temperature with respective weighting errors used in MPC

Matrix Q

The weighting matrix $\overline{Q}_{(p\,\mathrm{x}\,p)}$ must be chosen positive definite. A good choice can be to use a diagonal matrix with positive, time-variant elements:

$$\overline{Q}_{(p\,\mathrm{x}\,p)} = \begin{bmatrix} \epsilon_{1,k} & 0 & \cdots & 0 \\ 0 & \epsilon_{2,k} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \epsilon_{3,k} \end{bmatrix}. \tag{5.49}$$

A possible choice for the functions $\epsilon_{i,k}$ is shown in the second graph of figure **5.9**. Here, trapezoidal curves were used, but it could be a triangular function or a Gaussian function instead.
It is important that an error $e_i = y_i - r_i$ is weighted more when the reference

function is about to raise. Therefore, the value of ϵ_i is higher (up to ϵ_{MAX}) in a neighborhood of the corresponding $t_{start,i}$. Afterwords the function should possibly decrease to a nonzero value ϵ_{MIN}.

Matrix R

Being $m = 1$, the matrix R, which weights the Δu_k, is a scalar: $\overline{R} = \epsilon_R$. Due to a correct optimization, it is important to choose a proper order of magnitude for the ϵ_R. For example, if we want to weight the Δu_k d times less than a single error $e_i = y_i - r_i$, the following must be true:

$$\hat{e_i}^2 \cdot \epsilon_{MAX,Q} \approx d \cdot \hat{\Delta u}^2 \cdot \epsilon_R \cdot, \tag{5.50}$$

where $\hat{e_i}$ and $\hat{\Delta u}$ are approximate guesses of the errors.

Horizons n_p and n_o

The prediction horizon n_p should possibly be high enough to include at least the dynamics of two neighboring reference temperature. For the reference temperature in figure **5.9**, for example, this is assured if:

$$n_p \geq \frac{1100}{T_s} = 220. \tag{5.51}$$

The optimization horizon n_o can be smaller than the prediction horizon, for example equal to $n_p/2$ or $n_p/4$.

5.3.2 Control examples

The first experiments of MPC were done on the nonlinear ARMAX-model instead of on the numerical model. Figure **5.10** shows the simulation for the case

$$\begin{aligned} n_o &= 50 \\ n_p &= 400 \\ [\epsilon_{Q,MIN}, \epsilon_{Q,MAX}] &= [2, 60] \\ \epsilon_R &= 4 \cdot 10^6, \end{aligned} \tag{5.52}$$

whereas figure **5.11** refers to the case

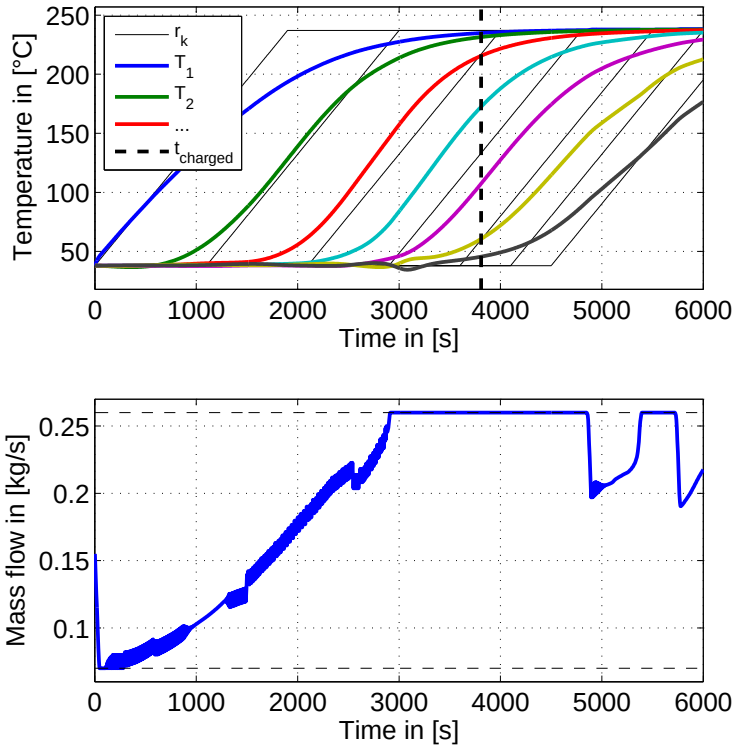

Figure 5.10: MPC applied to the nonlinear ARMAX-model. Simulation ex. 1

$$\begin{aligned} n_o &= n_p = 50 \\ [\epsilon_{Q,MIN}, \epsilon_{Q,MAX}] &= [2, 60] \\ \epsilon_R &= 4 \cdot 10^7. \end{aligned} \tag{5.53}$$

We see that by setting the ϵ_R higher, the chattering of the input has disappeared.
The charging time and charging percentage of the two experiments are

$$(t_{charged}, \%_{charged}) = (3810\,\mathbf{s}, 58.56\%) \tag{5.54}$$

and

$$(t_{charged}, \%_{charged}) = (3980\,\mathbf{s}, 63.4\%), \tag{5.55}$$

respectively. Unfortunately, both charging performances are worse than the ones obtained for the experiments with constant mass flow. Even looking

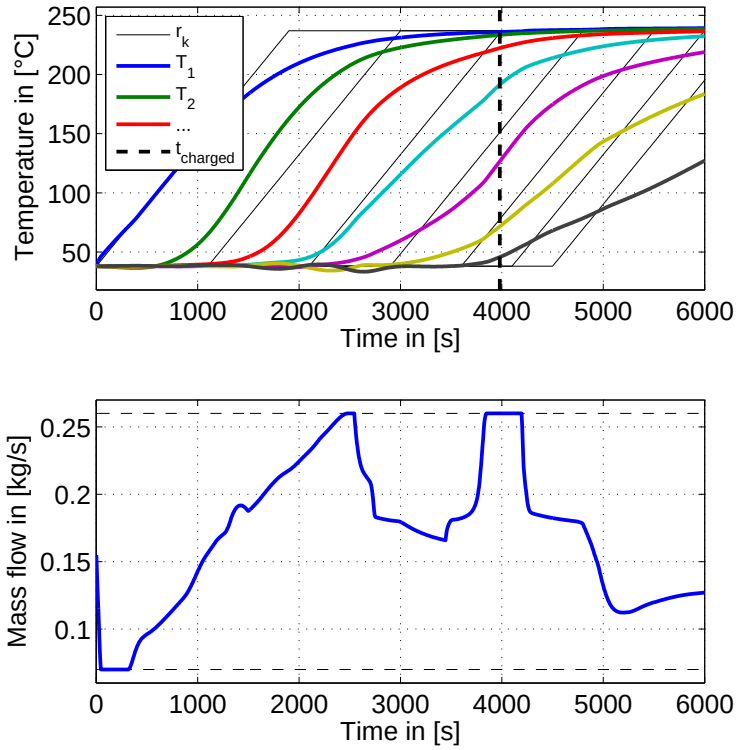

Figure 5.11: MPC applied to the nonlinear ARMAX-model. Simulation ex. 2

at the simulation results we can see that the output does not track the reference output. It seems, instead, that the evolution of the output happens randomly. In the next section, it is explained why the MPC did not improve the charging performance of the system.

5.3.3 Restrictions of controlling the Thermal Storage System using MPC

The linear MPC does not work for this ARMAX-model because there is a significant difference between the predicted behavior of the system and its actual behavior.

See the signals in the first graph of figure **5.12**. The temperature $T_{\dot{m}_f=0.15\,\mathbf{kg/s}}$ is the output of the nonlinear ARMAX-model if the input $\dot{m}_f = 0.15\,\mathbf{kg/s}$ is

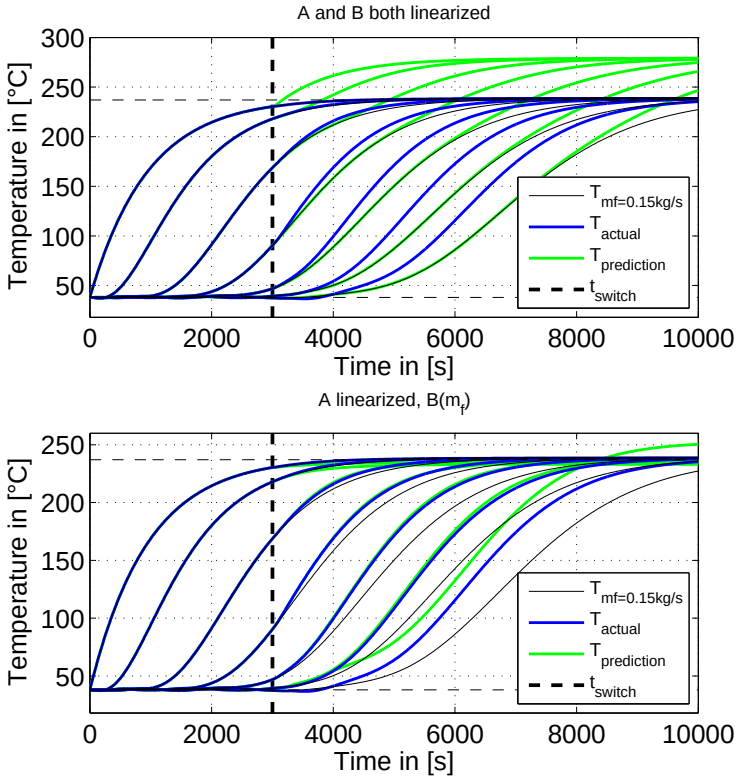

Figure 5.12: First graph: System output for an abrupt change of the input compared to its predicted value for A and B both linearized. Second graph: System output for an abrupt change of the input compared to its predicted value for A linearized and $B(\dot{m}_f)$ nonlinear.

held constant through the experiment. The signal T_{actual} shows how the systems behaves if at the time t_{switch} the input changes abrupt to $\dot{m}_f = 0.18\,\mathbf{kg/s}$ in order to be kept there. The signal $T_{prediction}$ is the prediction of what happens if the input is switched. The MPC performs the optimization only considering this prediction, that is obviously wrong. First of all the predicted final stable value is significantly wrong. Secondly, the prediction does not know that the temperatures are bound to increase faster if the input is suddenly switched to a higher input.
We can affirm that the linear MPC will not be able to control the system with such a wrong prediction.

This means that in order to get better results, the non-linearity of the ARMAX-model must be taken into account while performing the optimization; that is a nonlinear MPC. Probably it would be enough to consider the non-linearity of the matrix $B(\dot{m}_f)$, while using the linearized matrix A. The prediction in this case is shown in the second graph of figure **5.12**. We can see that, except for the last output, the prediction is quite accurate and satisfactory.
Unfortunately, as explained in the next section, if we have to take into account the nonlinearity of B, the optimization problem is no longer convex.

5.4 Nonlinear MPC for the Thermal Storage System

Some attempts to control the Thermal Storage System with nonlinear MPC methods were done during this thesis, but are only briefly discussed here. In general we can affirm that it is difficult to apply such a method to this system, because the two available stable nonlinear ARMAX-models are not suitable.
The nonlinear ARMAX model obtained by linear interpolation is only piecewise analytical and would therefor lead to a piecewise defined non-convex optimization problem.
On the other hand, the nonlinear ARMAX model obtained by gaussian interpolation can be described analytically, but the dependence on the optimization variables is highly nonlinear, which leads to a highly nonlinear optimization problem.

Chapter 6

Fuzzy control

In this chapter the procedure of applying a Fuzzy controller to the Thermal Storage System is exposed.
As this control method is not model-based there is no need to perform an identification: a good knowledge of the black-box behavior of the system is sufficient.
Usually this control method is suitable for systems with the same amount of inputs and outputs. Thus, in this case the control input will be calculated as an average of the control terms of all the multiple outputs.

6.1 Fuzzy control of the Thermal Storage System

The Fuzzy control of the Thermal Storage System was done by using the MATLAX *Fuzzy Toolbox*. In order to decrease the degrees of freedom, the method used is the so called Sugeno-Tagaki Fuzzy control.
First of all, we need to know if the output of the system is proportional to the input or not. This information can usually be obtained by simple experiment or by an overall knowledge of the system behavior.
Usually the Fuzzy control is performed on discrete-time SISO-systems or on MIMO-systems with uncoupled outputs. The system to be controlled has only one input, but 19 outputs. To overcome this hindrance, the control input will be calculated as a weighted average of the control terms of the multiple outputs, or of a part of the outputs. That means that each output can separately send some sort of command to the input, e.g. 'raise', 'lower'

or 'keep'. The input will consider all the commands and obey to what the majority of the outputs told him to do.

6.1.1 Outputs and output reference

Only 7 of the 19 available outputs were used for the control. These are the outputs placed at:

$$x_{fy} = \begin{bmatrix} 10 & 30 & 60 & 900 & 120 & 150 & 180 \end{bmatrix}^T \textbf{ cm} \qquad (\textbf{6.1})$$

These outputs will be referred to as $T_{fy,i}$, with $i = 1, 2, ..., 7$.

The reference for these outputs could be similar to the sub-ideal behavior shown in the third graph of figure **3.10**. Alternatively we could use the same one as for the MPC, as exposed in section **5.3.1**.
Generally, the experiments have shown that the reference temperature must be "realistic". If the reference temperature is chosen too ideal, the system won't be able to track the reference.

6.1.2 Mapping rules

Defining the tracking errors between reference temperature and actual temperature as

$$\mathbf{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_7 \end{bmatrix} = \begin{bmatrix} T_{r,1} \\ \vdots \\ T_{r,7} \end{bmatrix} - \begin{bmatrix} T_{fy,1} \\ \vdots \\ T_{fy,7} \end{bmatrix} \qquad (\textbf{6.2})$$

for both first order and second order Fuzzy control, some mapping rules can be set.

First order Fuzzy control

We know that if the input of our system is highered, then all the outputs are bound to increase faster. On the contrary, if the output is lowered, then all the outputs keep increasing, but slower.[1] This means that if an output $T_{fy,i}$ is lower (higher) than its reference value $T_{r,i}$, then the output must be

[1] The outputs can only grow, but never drop.

increased (decreased).

Fist of all let's define

$$u_{k+1} = u_k + \Delta u_k. \tag{6.3}$$

Choosing proper membership functions for the error, for each of the 7 outputs the mapping rules of the Fuzzy control can be described as:

1. **If** e_i is negative **then** $\Delta u_{i,1} = u_N$
2. **If** e_i is small **then** $\Delta u_{i,2} = 0$
3. **If** e_i is positive **then** $\Delta u_{i,3} = u_P$

where $u_N < 0$ and $u_P > 0$.
The Δu_i is then

$$\Delta u_i = \frac{s_1 \cdot \Delta u_{i,1} + s_2 \cdot \Delta u_{i,2} + s_3 \cdot \Delta u_{i,3}}{s_1 + s_2 + s_3} \tag{6.4}$$

where s_j with $j = 1, 2, 3$ are the truth contents of the mapping rules 1, 2 and 3.
The global result is thus calculated as a weighted average of all Fuzzy outputs, like:

$$\Delta u = \frac{\sum_{i=1}^{7} \Delta u_i \epsilon_i}{\sum_{i=1}^{7} \epsilon_i}, \tag{6.5}$$

where ϵ_i are the weights that can be time-dependent.
The input is then calculated for each time step as (**6.3**).

Second order Fuzzy control

To perform a second order Fuzzy control, even the variation of the input must be taken into account, i.e. the variation of the tracking error.
As the Fuzzy control is a discrete-time control, the variation is assumed to be equal to the backward finite difference of the error:

$$\dot{e} \approx \frac{e_k - e_{k-1}}{T_s} \tag{6.6}$$

Being $u_{NN} < u_N < 0$ and $u_{PP} > u_P > 0$, the mapping rules of the second order Fuzzy control are:

1. **If** e_i is negative **and** $\dot{e}_i$ is negative **then** $\Delta u_{i,1} = u_{NN}$
2. **If** e_i is negative **and** $\dot{e}_i$ is small **then** $\Delta u_{i,2} = u_N$
3. **If** e_i is negative **and** $\dot{e}_i$ is positive **then** $\Delta u_{i,3} = 0$
4. **If** e_i is small **and** $\dot{e}_i$ is negative **then** $\Delta u_{i,4} = u_N$
5. **If** e_i is small **and** $\dot{e}_i$ is small **then** $\Delta u_{i,5} = 0$
6. **If** e_i is small **and** $\dot{e}_i$ is positive **then** $\Delta u_{i,6} = u_P$
7. **If** e_i is positive **and** $\dot{e}_i$ is negative **then** $\Delta u_{i,7} = 0$
8. **If** e_i is positive **and** $\dot{e}_i$ is small **then** $\Delta u_{i,8} = u_P$
9. **If** e_i is positive **and** $\dot{e}_i$ is positive **then** $\Delta u_{i,9} = u_{PP}$

Here, it was chosen to calculate the **and** using the product-function.
The Δu_i and the Δu are computed like for the first order case.

6.1.3 Simulation example

A huge amount of experiments have shown that in general the second order Fuzzy control leads to better results than the first order control. In addition, it was observed that it is better to set all the ϵ equal and constant.
A simulation example with these properties is exposed in the following. The experiment was held on the discrete-time model obtained by Lax-Wendroff discretization with a sample time $dt = 0.0025\,\mathbf{s}$ for the model and $T_s = 1s$ for the control. Other control specifications are:

- The reference temperatures are obtained considering an experiment on the numerical model with constant input $\dot{m}_f = 0.12\,\mathbf{kg/s}$. The first output $T_{r,1}$ will be set equal to the measured output at $x = 20\,\mathbf{cm}$. For the following reference temperatures a shifted version of it is used.
- The membership functions for e and $\dot{e}$ were chosen trapezoidal and are shown in figure **6.1**.
- $u_{PP} = 0.01$, $u_P = 0.005$, $u_N = -u_P$ and $u_{NN} = -u_{PP}$.

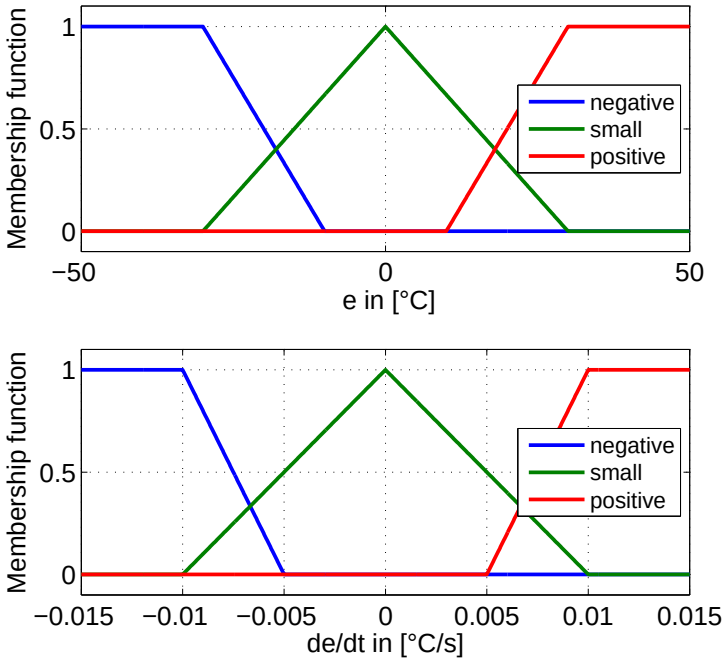

Figure 6.1: Membership functions of the tracking error and its derivative used to perform a second order Fuzzy control

The graphical results of the simulation are shown in figure **6.2**. We can see that till 2500 **s** the tracking is satisfactory. Afterwords, the tracking errors increase significantly. This happens because after some time there are too many Fuzzy subsystems sending contradictory commands to the input. For example, at $3500s$, the error e_3 is positive and would like to higher the input, while e_4 is negative and tries to lower the input.
The charging performance in this case is

$$(t_{charged}, \%_{charged}) = (6496\,\mathbf{s}, 69.72\%) \tag{6.7}$$

Let's compare this result with the charging performances for experiments with constant input in figure **3.8**. We can assert that this specific experiment led to a better charging distribution (+0.25%) for the same charging duration. Similarly, to achieve the same percentage it takes 215 **s** less.
It is obvious that somehow the tracking was successful, but personally I don't think that the control has contributed into achieving a (very slightly) better

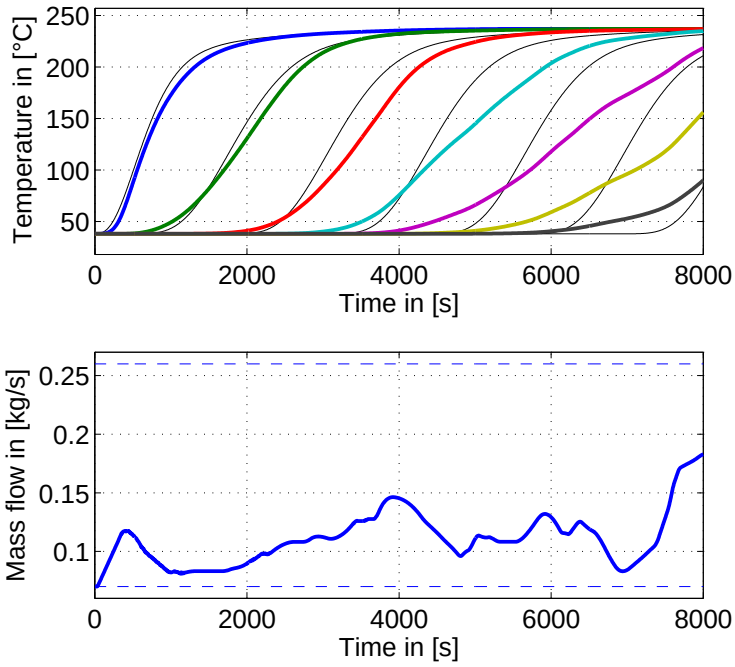

Figure 6.2: Simulation example: Second order Fuzzy control applied to the numerical model obtained by Lax-Wendroff discretization

charging performance. It is more likely that this improvement is given by the fact that the average derivative of the mass flow in figure **6.2** is positive. In section **3.1.2**, some open-loop experiments were held on the numerical model, which have shown that in general a gradually increasing input is profitable.

Conclusion

In the introduction of this master thesis, the importance of using energy storage systems for developing renewable energy systems was expound. In electricity systems based on renewable energy, the sources are only intermittently available, while the system must satisfy a constant electricity demand. Therefore, an efficient usage of energy storage systems is essential to provide a back-up to intermittent renewable energy.

The Packed Bed Energy Storage System treated in this master thesis, offers a relatively cheap way of storing energy. Its storage capacity merely depends on the choice of the storage medium and on the size of the system itself. We have seen, though, that only a part of its storage capacity can be exploited during the charging phase, and only a fraction of the stored energy can be fully recovered. That means that the charging and discharging procedure are non completely reversible, which dramatically affects the storage capacity for a higher number of charging-discharging cycles.
Another crucial requirement on the system are the thermal losses to be negligible. To derive a mathematical model of the system, this was assumed to be true. The simulation results were then compared to real laboratory measurements and it was shown that the model is quite accurate. This means that the thermal losses of the laboratory plant are negligible for short-therm storing.

The principal aim of this work was to improve the charging performance, which means to exploit the available storage capacity, of such a system by means of control.
To investigate the system behavior, a mathematical model of the system was derived. As it was not possible to solve the model equations analytically, the solution is based on a numerical approach.

Experiments held on the mathematical model using different constant mass flows have shown that by using a lower mass flow the capacity can be best exploited, with the huge disadvantage that the charging procedure is much slower. Using a higher mass flow the charging duration is shorter, but the charging distribution in the system is much worse.
However, it has been shown by numerical investigations that non constant flow profiles can improve the charging performance compared to the constant-flow scenario. In general, it is profitable to start the experiment with a lower mass flow and increase gradually the input throughout the charging procedure.

While open-loop experiments have shown some improvements of the charging performance, was the attempt to control the system rather unsuccessful. The main problem arises because the mass flow alone is supposed to control the temperature distribution in a volume. Basically, one single control variable must control a very high number of outputs.
Furthermore, the available model of the system is highly nonlinear and non-affine in the control.
It was shown that the PI controller is not suitable for such a problem, because it cannot control simultaneously the temporal and spacial behavior of the system. For the same reason, a Fuzzy controller is not capable to perform a satisfactory tracking of a reference signal; there are too many Fuzzy subsystems sending contradictory commands to a single control variable.
To take into account the temporal and spacial behavior of the system, model-based optimal control strategies such as the Model Predictive Control appear to be a promising class of approaches to be considered. Unfortunately, the linear MPC did not give a good result because there was a mismatch between the prediction and the actual behavior of the system. On the other hand, nonlinear MPC strategies are difficult to implement because for this specific system the optimization problems to be solved are highly nonlinear or only piecewise defined.

Finally, we can say that the control results did not meet the expectations and could not improve the system behavior. This is due to the complex system dynamics and to the fact that there is only a single control variable available to steer the system.

Appendix A

Empirical theory: Linear interpolation of parameters of characteristic polynomial

In section **5.2.3**, a nonlinear ARMAX-model for the Thermal Storage System was estimated in two steps:

1. For some constant values of the input, linear ARMAX-models of the same order were estimated
2. The nonlinear ARMAX-model was built by linear interpolation between the parameters of the matrices of the linear-ARMAX models.

The nonlinear ARMAX-model resulted to be stable for the whole input range. Here we want to find out if this is always valid.

Problem

Let

$$a(z) = z^n + a_{n-1}z^{n-1} + ... + a_0, \quad b(z) = z^n + b_{n-1}z^{n-1} + ... + b_0 \qquad (\mathbf{A.1})$$

be the characteristic polynomial of the discrete-time BIBO-stable systems sys_a and sys_b, respectively. Both systems have the same order.
Consider the system with characteristic polynomial

$$c(z) = z^n + c_{n-1}z^{n-1} + ... + c_0 \qquad (\mathbf{A.2})$$

where the parameters are given by interpolating the parameters of the two systems sys_a and sys_b, i.e.

$$c_i(u) = \frac{b_i - a_i}{u_b - u_a} u - \frac{u_b a_i - u_a b_i}{u_b - u_a}, \quad \forall i = 0, 1, ...n-1, u \in [u_a, u_b]\,. \qquad (\mathbf{A.3})$$

Without loss of generality, we will set $u_a = 0$, $u_b = 1$ and hence the parameters of $c(z)$ can be rewritten as:

$$c_i(u) = (b_i - a_i)\, u - a_i, \quad \forall i = 0, 1, ...n-1, u \in [0, 1] \qquad (\mathbf{A.4})$$

Is the system with characteristic polynomial (**A.2**) BIBO-stable?

Empirical solution

A Monte-Carlo approach was used to find out the probability that the procedure of interpolating two linear systems lead to a stable system.
To do so, two non-correlated characteristic polynomials of order n containing the following roots were built:

- $\left\lfloor \frac{n}{2} \right\rfloor$ complex roots $r_i = |r_i| e^{-j\rho_i}$ with $0 \leq |r_i| < 1$ and $0 \leq \rho_i < 2\pi$ randomly chosen $\forall i$
- the respective complex conjugate roots
- $n - 2\left\lfloor \frac{n}{2} \right\rfloor$ real roots $-1 \leq r_i \leq 1$ were randomly chosen $\forall i$

At that point, the characteristic polynomial $c(z)$ is built according to (**A.2**). For $N = 1000$ discretization points in the interval $u \in [0, 1]$, an eigenvalue test was done to see if the system is stable.
This procedure was repeated $M = 200$ times and the probability of obtaining a stable system with this approach can be assumed to be equal to the percentage of experiments which led to a stable system $\forall u \in [0, 1]$. The computed probabilities are shown in figure **A.1** as *Experiment 1.* It is shown that for a system of 2nd order stability is guaranteed, while the probability decreases for higher orders. The percentage corresponding to the order $n = 14$ of the ARMAX-models identified for the Thermal Storage System is 43.5%. As 9 linear interpolations were performed, the probability that the nonlinear ARMAX-model is stable on the whole input range is around 0.0558%. At that point we could ask ourselves if we have just been "lucky" that the nonlinear ARMAX-model of the Thermal Storage System turned

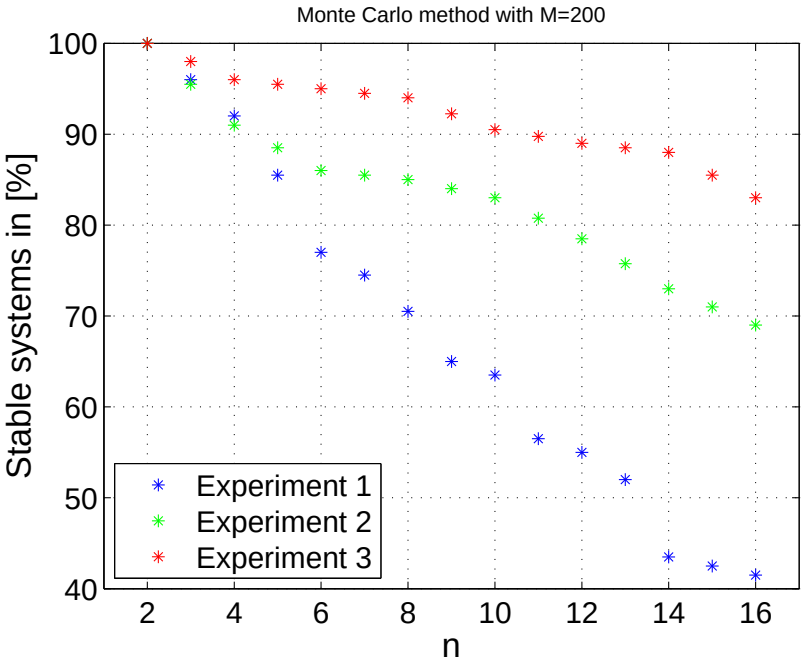

Figure A.1: Probability of getting a BIBO-stable system by linear interpolation of the parameters of two BIBO-stable characteristic polynomial. The probability is expressed as a percentage and is computed for various orders n of the characteristic polynomial.

out to be stable. We can assume, though, that there is some likelihood or correlation between two neighboring linear ARMAX-models. That could be the reason for the system to be stable.

Therefore, two more experiments were performed where there is some correlation between the characteristic polynomials $a(z)$ and $b(z)$. Here, the roots of $a(z)$ were chosen randomly. The roots of $b(z)$, instead, are the roots of $a(z)$ with a variation $(\Delta r_i, \Delta\rho_i)$ chosen randomly for each root. Of course it was assured that the variation does not make the single roots unstable. For *Experiment 2* in figure **A.1** the maximal variation was:

$$\begin{aligned} \Delta r_{i,MAX} &= 0.2 \\ \Delta\rho_{i,MAX} &= \frac{1}{8}\pi rad \end{aligned} \qquad \textbf{(A.5)}$$

whereas for *Experiment 3* it was:

$$\begin{aligned} \Delta r_{i,MAX} &= \frac{\Delta r_{i,MAX}}{2} = 0.1 \\ \Delta \rho_{i,MAX} &= \frac{\Delta \rho_{i,MAX}}{2} = \frac{1}{16}\pi rad \end{aligned} \tag{A.6}$$

We can see that the probability of stability increases significantly if there is some likelihood between the roots of the two characteristic polynomials. Therefore, if the interpolated ARMAX-model turns out to be unstable, a possible approach can be to increase the distance between two neighboring linear ARMAX-models in order to increase the likelihood between two models that are to be interpolated.

Bibliography

[1] Daniela Dejaco, Alessandro Pisano, Elio Usai, Martin Horn, Giorgio Cau, Pierpaolo Puddu, Fabio Serra. *Numerical Investigation of Packed-Bed Thermal Energy Storage Systems with Varying Heat Transfer Fluid Flow.* ICAMIA2015 - 2015 International Conference on Advanced Manufacturing and Industrial Application Pukhet, December 2015.

[2] Mario Cascetta. *Modellazione di un impianto solare termodinamico operante con fluidi termovettori gassosi ad alta temperatura* . (Italian) [*Modeling of a solar thermal power plant using heat transfer fluids at high temperatures*] Doctoral thesis, University of Cagliari, Italy, 2010.

[3] Alberto Randaccio. *Accumulo termico per impianti solari termodinamici operanti con CO_2 ad alta temperatura.* (Italian) [*Thermal storage system for solar thermal power plant using CO_2 at high temperatures*] Master thesis, University of Cagliari, Italy, 2012.

[4] Incropera, DeWitt, Bergman and Lavine. *Fundamentals of Heat and Mass Transfer.* Wiley, 1981.

[5] Alberto Varone. *The Storage System.* Center for advanced studies, research and development in Sardinia.

[6] Hank Price, Jim Pacheco and Bruce Kelly. *Thermocline analysis.* National Renewable Energy Laboratory (NREL).

[7] T. E. W. Schumann. *Heat transfer: a liquid flowing through a porous prism.* J. Franklin Inst. 208, 405-416, 1929.

[8] Giorgio Cau for estate LAB. *Realizzazione dell'impianto di accumulo termico in scala di laboratorio.* (Italian) [*Realization of the laboratory proto-*

type of a Thermal Storage System] Department of Mechanical Engineering, University of Cagliari, Italy, 2011.

[9] M. Horn and N. Dourdoumas. *Regelungstechnik*. (German) [*Automatic Control*] Pearson, 2004.

[10] A.C. Hoffmann, H.J. Finkers. *A relation for the void fraction of randomly packed particle beds*. Powder Technology, 1994.

[11] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation theory*. Prentice Hall, 1993.

[12] Luciano Rezzolla. *Numerical Methods for the Solution of Partial Differential Equations*. Lecture Notes for the COMPSTAR School on Computational Astrophysics, Caen, France, 1993.

[13] Johan Löfberg. *YALMIP : A Toolbox for Modeling and Optimization in MATLAB*. In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004..

[14] European Commission, Directorate-General for Energy *The future role and challenges of Energy Storage*. 2014.

Acknowledgements

Innanzittutto vorrei esprimere la mia gratitudine verso tutto il gruppo di automazione dell'Università degli studi di Cagliari, e qui vorrei citare in particolar modo il mio relatore Elio Usai e Alessandro Pisano. Vorrei ringraziarvi di avermi dato questa bellissima opportunità di collaborare con voi durante il mio periodo di tesi e di avermi supportata sempre. Mi sono trovata molto bene nell'ambiente di lavoro e mi avete aiutato a capire qual'è la mia strada.
In secondo luogo vorrei ringraziare i miei 'amici del pranzo' per la bella compagnia e per aver risvegliato in me l'interesse verso il mondo della ricerca.
Infine ringrazio di cuore i miei amici conosciuti a Cagliari, in particolar modo Petra e Benny, per avermi dato l'opportunità di esplorare insieme a loro un'isola fantastica, creando un bellissimo sfondo al lavoro di tesi.

Anschliessend möchte ich mich bei meinem Betreuer Martin Horn für die Unterstützung und Hilfsbereitschaft bei der Masterarbeit bedanken. Mein Dankeschön geht auch an das gesamte Regelungstechnikinstitut für die guten Vorlesungen und für all das was ich in diesen Jahren gelernt habe.

Inoltre vorrei ringraziare gli amici di sempre: sia quelli che ci sono stati in questi anni, sia quelli che che c'erano solo virtualmente ma pur sempre nei miei pensieri. Grazie.
In modo particolare voglio ringraziare Enrico, che c'è sempre stato in questi anni e che mi ha sempre supportata. Vorrei ringraziarti perché nel breve periodo in cui pensavo di non farcela più mi hai incorraggiata tantissimo, e quindi se ho ultimato questa tesi lo devo soprattutto a te. Grazie di cuore.

Pur stlü jö uressi ciamó ringrazié döta la mia familia pur m avëi dé la poscibilité da ti jí dô a chësta strada che me á dé en gröm de sodisfaziuns y pur avëi três cherdü t me. Sënza d os ne fossl nia sté poscibile. Dër bel dilan.