Model-based Controller Design Methods for Heating Systems

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Georg Pangalos

aus
Neu Wulmstorf

2016

1. Gutachter: Prof. Dr.-Ing. habil. Gerwald Lichtenberg
2. Gutachter: Prof. Dr. Herbert Werner
3. Gutachter: Prof. Gregor P. Henze, Ph.D., P.E.

Vorsitzender des Prüfungsausschusses:
Prof. Dr.-Ing. Günter Ackermann

Tag der mündlichen Prüfung: 17.07.2015

Impressum:

Verlag: Neopubli GmbH, Berlin
www.epubli.de
ISBN: 978-3-7418-2306-0
URN: urn:nbn:de:gbv:830-88213133

To my father

Abstract

Dynamical models of heating systems are derived taking into account the multiplication of temperature difference and flow rate, arising from the heat power balances. Multilinear time-invariant (MTI) models in tensor representation are shown to be adequate for heating systems. Tensor decomposition techniques are applied for rank reduction. A method to derive multilinear approximations of nonlinear models is presented. The multilinear property is used for the controller design. Further model-based controller designs for different heating system examples using nonlinear and switched affine models are discussed and evaluated.

Dynamische Modelle von Heizungsanlagen werden erstellt, wobei die Multiplikation von Temperaturdifferenz und Volumenstrom, die in den Wärmeleistungsbilanzen auftritt, berücksichtigt wird. Heizungsanlagen können als multilineare zeitinvariante (MTI) Modelle in Tensordarstellung modelliert werden, für die Methoden der Tensordekomposition zur Reduzierung des Ranges anwendbar sind. Eine Methode um multilineare Approximationen von nichtlinearen Modellen zu erstellen wird vorgestellt. Die multilineare Eigenschaft wird für den Reglerentwurf verwendet. Weitere modellbasierte Reglerentwürfe für nichtlineare und stückweise affine Modelle unterschiedlicher Heizungsanlagen werden diskutiert und bewertet.

Acknowledgement

Thank you, Gerwald Lichtenberg, for being my supervisor. You were the one who suggested to me to do work in the area of energy efficiency of heating systems. This was a really interesting journey. I am especially thankful to you not only because of the clearly helpful discussions on the topic but also for all those discussions that helped me a lot to broadening my mind. I always like how you can come from a topic, that lies far away from control back to control engineering.

Thank you, Herbert Werner, for the time in the institute of control systems and for your participation in the examination committee. With your lectures on control systems you inspired me to focus on control engineering, which is a decision that I have never regretted.

Thank you, Gregor P. Henze, for being part of the examination committee and for the fruitful discussions before and after and for helping me to appreciate the examination.

Thank you, Günter Ackermann, for leading the examination committee.

Thank you Max Schmidt, Timo Spengler, Fabian Rößner, Jan Seifert, Kai Kruppa and Sven Harder. You all wrote your Bachelor's thesis, Master's thesis, Studienarbeit or Diplomarbeit with my supervision. You all helped me a lot in conducting the research I wanted to do.

Thank you, Erik Sewe, for starting this journey with me, helping me with every implementation issue I had and for all the help that goes far beyond the project and this thesis.

Thank you, Annika Eichler, for charing the office with me for so long and being patient with explanations.

Thank you, Christian Hoffmann, for our blind understanding.

Thank you, Ahsan Ali, for countless extra curricular seminars and for numerous lessons on your cultural background.

Thank you, Dagmar Rokita, for making the change of my workplace a pleasant surprise.

Thank you, Kai Kruppa, not only for writing your master thesis with

my supervision but also for joining the team and helping me with the project.

Thank you, Sven Pfeiffer, for opening my mind to different time scales.

Thank you, Qin Liu, for the memorable subway ride.

Thank you Herwig Meyer, Klaus Baumgart, Birthe von Dewitz, Esteban Rosero, Siavash Ahmadi, Michael Heuer, Simon Wollnack, Ulf Pilz for being my colleagues.

And the final thanks in this acknowledgement is to my family:

Thank you!

Contents

1 Introduction

This introduction starts with the statement of the amount of consumed energy for heat supply of the buildings in Germany. Followed by some basics of heatings systems. The research question is stated, an overview of the current state of the art is given and the structure of the work is presented.

1.1 Energy

In Germany in 2012 about 34 % of the final energy was used for room heat and warm water supply, [1]. The use of modern boilers with condensing burners may reduce the required amount of energy to supply the building with heat by up to 20 %, [2]. Poorly tuned controllers waste 5 % to 30 % of the energy that could be used for heat supply. Thus, assuming the worst case for all controllers, up to 918 PJ of energy are wasted in the process of supplying buildings with heat.

1.2 Basics of heating systems

The main purpose of heating systems is the generation of heat, which can be done using different heat generation units. For example a boiler with an attached burner can heat up water by burning gas or oil in the burner. The boiler contains the water which is heated. Other examples are combined heat and power stations, where the gas is used for the generation of electrical power and the waste heat is used for heat supply of a building or solar collectors, which use solar radiation to heat up a carrier medium. Also widely used is district heating. Power plants use the waste heat to supply the surrounding buildings with heat. The heat is generally transferred with heat exchangers to hydraulically decouple the heating system of the building and the power plant. The heated water is then pumped through pipes to supply radiators or thermo-active building systems with heat, which in turn supply the building with heat. Thermo-active building systems are constructed by implementing pipes in the concrete structure of a building, having the advantage that lower temperatures can be used in comparison with radiators.

For all heating systems under investigation in this work, the carrier medium in the pipes is assumed to be water. This is without loss of generality, since just the density and specific heat coefficients would change with different carrier mediums.

The temperature of the water coming out of the heat generation unit will be called the supply temperature and denoted by T_s. The temperature of the water returning to the heat generation unit is called return temperature T_r. The consumer consists of all pipes, radiators, etc. in the building. If for the model of the heating system also a building model is used, this is also integrated in the consumer. If a three-way valve is implemented in front of the consumer, the supply temperature is not necessarily the consumer's supply temperature, which is the temperature of the water entering the consumer. The three-way valve can be used to cool down the supply temperature by adding water coming out of the consumer to the water going into the consumer. This can be used to supply the consumer with water of a specific temperature. Admixing pumps can be used to heat up the water going into the heat generation unit. Here the consumer's return temperature is increased by adding water from the heat generation output to the water going back into the heat generation unit. This can be beneficial when the boilers need a specified minimal return temperature. Pumps are used to generate a flow rate $\dot{V}$ in the pipes, such that the water circulates. Figure 1.1 illustrates these definitions. The difference between supply and return temperature will be denoted by temperature difference or temperature spread.

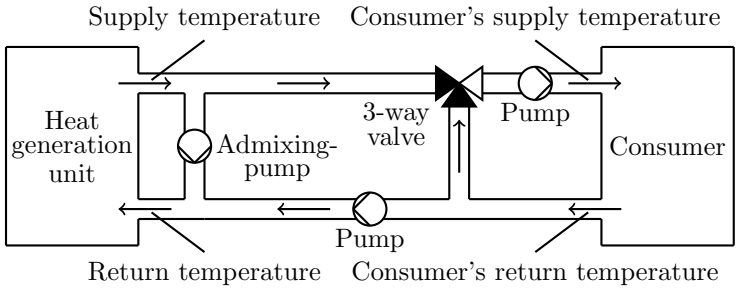


Figure 1.1: Scheme of a heating system

1.3 Research question

On the one hand, it has already been established, that improving the operation of heating systems can save a large amount of energy. The use of model-based controllers is widely used, therefore the investigation of these controller design methods for heating systems seems straight forward. On the other hand models of heating systems are quite challenging. First, multiplications of the states will occur and second also switching is part of the model. Thus, models of heating systems are often hybrid models, having continuous-valued and discrete-valued states. Since the continuous-valued dynamics are not linear, switched affine systems can not directly be used to model heating systems. The class of tensor systems is able to model these aspects.

The research question which was investigated in this work is:

What are the benefits of controllers synthesized using models of heating systems compared to existing controllers?

It is obvious, that not all advantages and disadvantages of the use of models can be stated. Nevertheless, the question is left somewhat imprecise because for different controller designs different aspects are regarded. For all designs a cost function is defined to specify the aspects that are under investigation. The model of the heating system was either used to design the controller, but no longer needed when the controller was designed or in the controller itself, such that the model was used during the control of the heating system. Different controller design methods were used for different buildings. In the design procedures different aspects were under investigation. Whereas the comfort always had to be met, the different buildings had different problems. For example in one building the switching rates of the boilers were too high or in another building the control of a hot water storage tank was suboptimal. For all models the abstraction level is similar and all models are created on the basis of heat power balances. Only the thermal part of the heating systems is investigated, no cooling systems are regarded and also no ventilation or air conditioning systems are investigated. The models take into account the multiplication of temperature difference and flow rate, arising from the heat power balances. This term prevents the use of lin-

earized models, since the operating regions, for which the linear model is valid is only very small.

1.4 State of the art

The basis of model-based controller design is the model. Therefore it is of interest to look at models of buildings and their heating systems. Approaches to building and heating system models are numerous, see e.g. [3, 4]. In most of those models a white box model of the building is generated, where floor plans, window areas etc. are measured. This leads to quite complex models, which are able to capture even smaller effects of the building operation, but are too extensive to be used on model-based control schemes like model predictive control.

Regarding the control of heating systems the most widely spread approach is entirely heuristic or even not set at all. Most of heating system components come with their own controllers, e.g. pumps which can be set to produce a constant pressure difference. Central building control systems are used to steer boilers such that some reference supply temperatures are tracked. The reference temperatures are generally adapted in dependence of the ambient temperature. More complex decisions e.g. the question when to charge or discharge a hot water storage tank are answered in a heuristic fashion or are a product of the plant operation.

For the control of heating systems several patents have been filed. Some of those patents will be described in the following. The German patent DE3410316A1 describes a possible control strategy for a heating system with several boilers. All but one boiler are hereby operated in a energy efficient way and only one boiler is switched on and off. In this patent no model of the heating system is used, but the energy efficiency of the entire system is not investigated. The European patent EP0445310A1 describes the use of a mixing equation to calculate the supply temperature of a multi boiler heating system depending on the boiler supply temperatures without having to measure it. The model of the admixing pump is a static model, where the incoming heat flow needs to be equal to the outgoing heat flow. The patent EP0874200A1 suggests the use of temperature sensors in a hydraulic separator to minimize the flow rate therein. If this flow is minimized the produced and consumed heat are

approximately the same which will enable an energy efficient operation of condensing boilers. In these patents static models are used, but the benefit of dynamic models is not exploited [1].

This thesis is dedicated to show the use of dynamic models of heating systems for the controller design. The controllers itself might incorporate a model of the heating system in the case of model predictive control. The models are grey-box models where the basic physical relationships are modeled and the parameters are estimated by measurement data. The abstraction level is way above the static models which where used e.g. in the patents described above but below the detailed models used in [4]. This abstraction level was found to be adequate, because the main effects of the heating system operation were captured but the models are not too complex for the controller design.

The abstract models of heating systems are characterized by belonging to the class of multilinear hybrid models. This property is used for the controller design. The findings in the domain of multilinear systems are not restricted to heating system, but can easily be extended to other domains, e.g. bio reactors where the reaction can be modeled as multilinear, [5].

In the scientific control engineering community, controller design for the energy efficient operation of heating systems is an active field of research. The IFAC world congress 2014 in Capetown offered an invited session to the topic *Control of Energy Efficient Buildings: Novel Control Strategies and Experimental Evaluations*, where e.g. the use of linear models of heating systems versus the computational effort of nonlinear models in model predictive controllers is discussed, [6]. The nonlinear properties of heating systems also were investigated in [7], where also ventilation and cooling is considered. None of the above publications use the multilinear property of heating systems, which will be in the center of this work.

[1] The literature survey on patents was performed by Timo Spengler

1.5 Outline

For heating systems several components are modeled which can be used to build models of heating systems. The model structures are investigated and hybrid multilinear discrete-time systems (HMDTS) can generally describe heating systems. Work is done in the area of multilinearization for this model class. The models are used for controller design using the Multi-Parametric Toolbox, nonlinear model predictive control, regions of attraction and feedback linearization. This is illustrated in figure 1.2.

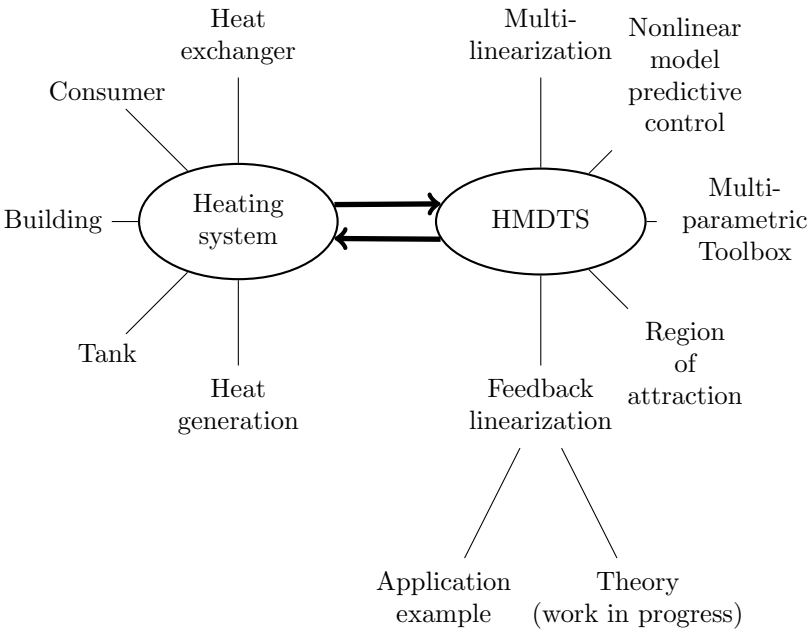


Figure 1.2: Structure of the work

Chapter 2 introduces the model classes investigated in this work. Starting from general nonlinear state space models, hybrid models are presented. The special class of tensor models concludes this chapter, which is followed by the description of models of heating systems in chapter 3.

The models are given first in the general nonlinear form, then as a hybrid model, given in piecewise affine form and finally the heating system is given as a tensor model. The following chapter 4 describes the different approaches to the controller design problem. First Boolean controllers are designed using a relaxation of algebraic normal forms. Nonlinear model predictive control is described and followed by model predictive control for a piecewise affine model of a heating system. Stabilizing controllers are designed guaranteeing ellipsoidal domains of attractions for multilinear systems and finally an approach for feedback linearization is presented. In the final chapter 5 a summary is given, conclusions are drawn and an outlook on possible future work is provided.

As usual there is more than one way through this work. The reader interested in nonlinear models of heating systems and their benefits is advised to jump from section 2.1 to section 3.1. Boolean controller design is then described in section 4.1 and model predictive control in section 4.2. The second line concerned with hybrid systems is started with section 2.2. Piecewise affine models are presented in section 3.2 and the use of these models in a predictive control scheme is presented in section 4.3. Multilinear models and tensor models are presented in 2.3. A heating system is modeled as a tensor model in section 3.3, where also a method of multilinearization and benefits of tensor models are described. Multilinear models are used in section 4.4, where stabilizing controllers are designed. Also tensor models are the basis of the investigations described in section 4.5.

2 Model classes

In this chapter the system classes under investigation in this work will be presented, starting with the most general, the nonlinear state space models. Subsequent the hybrid model class will be inspected by having a closer look at three different methods of modeling hybrid system. The chapter closes by introducing tensor models, which are also hybrid models and were found to be the most suitable models to represent heating systems having the property of being multilinear and hybrid.

2.1 Nonlinear state space models

Models of real systems have a great benefit for various reasons, e.g. tests of new concepts, safety, analysis, controller synthesis and many more. Let a system, as depicted in figure 2.1, have the inputs $\mathbf{u} \in \mathbb{D}^m$ and the outputs $\mathbf{y} \in \mathbb{D}^l$. Furthermore let the states $\mathbf{x} \in \mathbb{D}^n$ describe the current state of the system. The domain $\mathbb{D}$ is not fixed jet. It can stand for some continuous, discrete or even hybrid domain. The real domain will be denoted with $\mathbb{R}$, the Boolean domain with $\mathbb{B}$ and the hybrid domain with $\mathbb{H} = \mathbb{R} \times \mathbb{B}$.

Figure 2.1: System with input $\mathbf{u}$, output $\mathbf{y}$ and state $\mathbf{x}$

To describe the evolution of the state, differential equations in continuous time and difference equations in discrete time can be used. State space representations have been proven to be adequate for the description of the system given in figure 2.1. The general nonlinear continuous-time

state space model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.1}$$
$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.2}$$
$$\mathbf{x}(0) = \mathbf{x}_0, \tag{2.3}$$

where t denotes the time and $\mathbf{x}_0$ is the initial state. The discrete-time version is given by

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \tag{2.4}$$
$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) \tag{2.5}$$
$$\mathbf{x}(0) = \mathbf{x}_0, \tag{2.6}$$

where $k = 0, 1, 2, \ldots$ is the time index and $\mathbf{x}_0$ is the initial state. The information about the development of the state is contained in the function $\mathbf{f}$ and the output is given by the function $\mathbf{g}$. In the continuous-time case (2.1) will be called *state equation*, in the discrete-time case (2.4) will be called *next state equation* and in both cases (2.2) and (2.5) will be called *output equation*. The operator $\Phi(\mathbf{x})$ will be used to denote the time derivative $\dot{\mathbf{x}}(t)$ of $\mathbf{x}(t)$ in continuous time and the next state $\mathbf{x}(k+1)$ in discrete time.

2.2 Hybrid models

In most engineering tasks, models have to include not only continuous-valued signals, but also discrete-valued signals, e.g. switches. These models are called hybrid models. Several different modeling frameworks exist, which include continuous- and discrete-valued signals. Without loss of generality the discrete-valued signals will be assumed to be encoded Boolean. Therefore the terms Boolean and discrete will be used as synonyms. Switched systems introduce the idea of switching between different continuous-valued dynamics. Restricting the continuous-valued dynamics to be affine and assuming the discrete-valued dynamics to be very simple, describes the class of piecewise affine models. More general modeling frameworks are given with discrete hybrid automata and mixed logical dynamical systems. Both have the ability to represent discrete dynamics. The section will be concluded with a discussion of

equivalences of these hybrid models. Assumptions for the equivalences are given.

2.2.1 Switched systems

One approach to deal with hybrid systems is to investigate the continuous dynamics and not to emphasize the discrete dynamics. The hybrid system is then represented as several continuous-valued systems. The discrete-valued part is represented by switching between those systems. Often the switching is not represented in detail but e.g. stability is investigated for arbitrary switching, [8].

The continuous-valued dynamics can be represented by state space models given as

$$\dot{\mathbf{x}} = \mathbf{f}_i(\mathbf{x}). \tag{2.7}$$

To emphasize that several continuous-valued systems are investigated, the index $i \in \mathbb{N}$ is introduced. If the dynamics are linear, the set of systems can be given as

$$\dot{\mathbf{x}} = \mathbf{A}_i\mathbf{x}. \tag{2.8}$$

The switching between the systems can be thought of as state dependent or time dependent. For state dependent switching figure 2.2 shows a two dimensional state space with switching surfaces and resulting operating regions. For each operating region one continuous-valued system is defined. By adding a reset map, jumps of the state can be represented. If the state trajectory (thin line) hits on a switching surface (thick line), the reset map defines the new location of the state (dashed line). From there the system evolves according to the active dynamics. For time-dependent switching the index i in (2.7) or (2.8), respectively becomes a time-dependent function, e.g. figure 2.3 which depicts the selection of the active dynamics. A further distinction in the switching may be made between autonomous and controlled switching. In the above examples switching was autonomous and determined by either the state or the time. It is also possible to regard the switching as a control input. In this case the switching is controlled and may be adjusted during the evolution of the state to reach a certain performance.

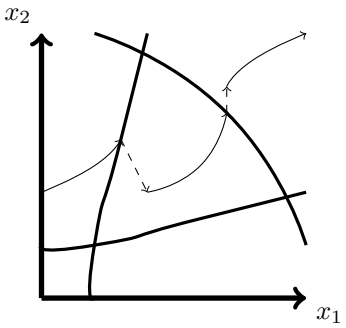


Figure 2.2: Spate space with switching surfaces (thick line) and an example state trajectory (thin line) with state jumps (dashed line).

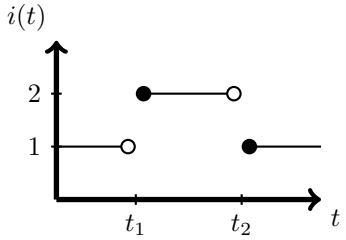


Figure 2.3: Time dependent switching function.

Switched systems may suffer from occurrences of Zeno behavior, i.e. there will be an infinite number of switching in a finite amount of time, if the state of a system converges to a switching surface.

2.2.2 Piecewise affine systems

Piecewise affine systems are a special form of switched systems, [9]. They arise when the continuous-valued states of an affine system are influenced by discrete states, which are assumed to be relatively simple. Polyhedral sets in the state-input space define the active continuous-valued dynam-

ics, i.e. switching is assumed to be state and input dependent. Discrete-time piecewise affine models will be presented such that specifics like Zeno behavior do not arise. The discrete-time system equations can be given as

$$\begin{aligned} \mathbf{x}(k+1) = &\quad \mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{u}(k) + \mathbf{f}_i \\ \mathbf{y}(k) = &\quad \mathbf{C}_i\mathbf{x}(k) + \mathbf{D}_i\mathbf{u}(k) + \mathbf{g}_i \end{aligned} \quad \text{for } \begin{pmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{pmatrix} \in \mathcal{X}_i, \tag{2.9}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ is the input and $\mathbf{y} \in \mathbb{R}^k$ is the output. The system matrices $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$, $\mathbf{D}_i$ and the offset vectors $\mathbf{f}_i$ and $\mathbf{g}_i$ have appropriate dimensions and define the systems behavior as long as the input and the state are in the input-state space partitions $\mathcal{X}_i$ for $i = 1, \ldots, s$. The polyhedral partitions can be defined using so called guards, i.e. the inequalities

$$\mathbf{H}_x^i\mathbf{x} + \mathbf{H}_u^i\mathbf{u} \leq \mathbf{K}_i. \tag{2.10}$$

Those inequalities have to hold for partition i to become active. Thus the sets $\mathcal{X}_i$ are defined as

$$\mathcal{X}_i = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix} \in \mathbb{R}^n \times \mathbb{R}^m \,\middle|\, \mathbf{H}_x^i\mathbf{x} + \mathbf{H}_u^i\mathbf{u} \leq \mathbf{K}_i \right\} \tag{2.11}$$

for $i = 1, \ldots, s$. For a piecewise affine system to be well posed, the output and the next state need to be uniquely defined, i.e. $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset, \forall i \neq j$ and $\cup_{i=1}^s \mathcal{X}_i = \mathbb{R}^{n+m}$, [10]. Piecewise affine models can not be used to model state jumps as defined by reset maps for switched systems, [8].

2.2.3 Discrete hybrid automata

A more general description of a hybrid system can be given with the framework of a (time-) discrete hybrid (-valued) automaton. A discrete hybrid automaton arises when a switched affine system is connected with a finite state machine, [11]. For the connection an event generator and a mode selector are needed. Figure 2.4 shows the structure of a discrete hybrid automaton. To distinguish between discrete-valued signals and continuous-valued signals, underbars are used for discrete-valued signals and overtildes are used for continuous-valued signals. Depending on the

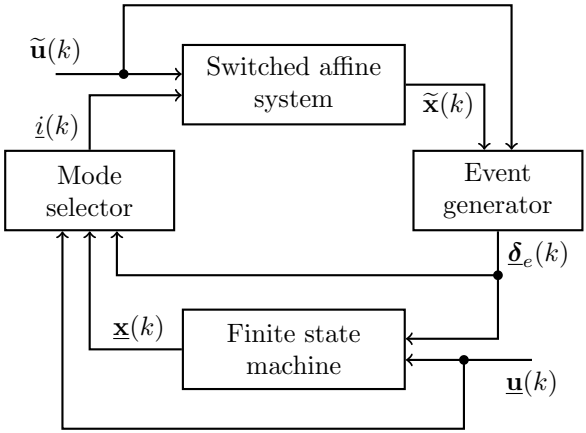


Figure 2.4: Structure of a discrete hybrid automaton.

continuous-valued states $\widetilde{\mathbf{x}}(k)$ of the switched affine system and the external continuous-valued inputs $\widetilde{\mathbf{u}}(k)$, discrete-valued events $\underline{\boldsymbol{\delta}}_e(k)$ are generated in the event generator. The discrete-valued next states $\underline{\mathbf{x}}(k)$ of the finite state machine are determined by its discrete-valued states, the discrete-valued events $\underline{\boldsymbol{\delta}}_e(k)$ and other external discrete-valued inputs $\underline{\mathbf{u}}(k)$. The mode selector chooses the active dynamics $\underline{i}(k)$ of the switched affine system in dependence of the discrete-valued states $\underline{\mathbf{x}}(k)$, external discrete-valued inputs $\underline{\mathbf{u}}(k)$ and the discrete-valued events $\underline{\boldsymbol{\delta}}_e(k)$. The continuous-valued states $\widetilde{\mathbf{x}}(k)$ then evolve according to the active dynamics of the switched affine system. Formally the discrete hybrid automaton can be described by a switched affine system, an event generator, a finite state machine and a mode selector. The switched affine system is given by

$$\begin{aligned} \widetilde{\mathbf{x}}(k+1) = &\quad \mathbf{A}_{i(k)}\widetilde{\mathbf{x}}(k) + \mathbf{B}_{i(k)}\widetilde{\mathbf{u}}(k) + \widetilde{\mathbf{f}}_{i(k)} \\ \widetilde{\mathbf{y}}(k) = &\quad \mathbf{C}_{\mathbf{i(k)}}\widetilde{\mathbf{x}}(k) + \mathbf{D}_{i(k)}\widetilde{\mathbf{u}}(k) + \widetilde{\mathbf{g}}_{i(k)} \end{aligned} \tag{2.12}$$

where $\widetilde{\mathbf{x}} \in \mathbb{R}^{\widetilde{n}}$ are the continuous-valued states, $\widetilde{\mathbf{u}} \in \mathbb{R}^{\widetilde{m}}$ are the continuous-valued inputs and $\widetilde{\mathbf{y}} \in \mathbb{R}^{\widetilde{l}}$ are the continuous-valued outputs. Note that with a slight violation of the notation, overtildes and underbars are also used for the dimensions of continuous-valued and discrete-valued

signals, respectively. The matrices $\mathbf{A}_{i(k)}$, $\mathbf{B}_{i(k)}$, $\mathbf{C}_{i(k)}$, and $\mathbf{D}_{i(k)}$ are real and have appropriate dimensions. The event generator is given as

$$\underline{\boldsymbol{\delta}}_e(k) = \mathbf{f}_e(\widetilde{\mathbf{x}}(k), \widetilde{\mathbf{u}}(k), \underline{k}), \tag{2.13}$$

where the function $\mathbf{f}_e : \mathbb{R}^{\widetilde{n}} \times \mathbb{R}^{\widetilde{m}} \times \mathbb{Z}_{\geq 0} \to \mathbb{B}^{\underline{o}}$ defines the events and $\mathbb{Z}_{\geq 0}$ being the non-negative integers. The finite state machine is given by

$$\underline{\mathbf{x}}(k+1) = \underline{\mathbf{f}}_a(\underline{\mathbf{x}}(k), \underline{\mathbf{u}}(k), \underline{\boldsymbol{\delta}}_e(k)) \tag{2.14}$$

where $\underline{\mathbf{x}} \in \mathbb{B}^{\underline{n}}$ are the discrete-valued states, $\underline{\mathbf{u}} \in \mathbb{B}^{\underline{m}}$ are the discrete-valued inputs and $\underline{\boldsymbol{\delta}}_e \in \mathbb{B}^{\underline{o}}$ are the discrete-valued events. By the discrete-valued function $\underline{\mathbf{f}}_a : \mathbb{B}^{\underline{n}} \times \mathbb{B}^{\underline{m}} \times \mathbb{B}^{\underline{o}} \to \mathbb{B}^{\underline{n}}$, the next states of the automaton are defined. Finally the mode selector, which is also called injector, takes the form

$$\underline{i}(k) = \underline{\mathbf{f}}_m(\underline{\mathbf{x}}(k), \underline{\mathbf{u}}(k), \underline{\boldsymbol{\delta}}_e(k)) \tag{2.15}$$

where the function $\underline{\mathbf{f}}_m : \mathbb{B}^{\underline{n}} \times \mathbb{B}^{\underline{m}} \times \mathbb{B}^{\underline{o}} \to \mathbb{B}$ defines the active dynamics of the switched affine system. The injector could be thought of as the output function of the finite state machine.

2.2.4 Mixed logical dynamical systems

The modeling framework introduced in [12] uses so called mixed logical dynamical systems to describe hybrid models. The model is represented by linear equations, which are solved subject to linear constraints, which include not only continuous-valued but also discrete-valued variables. A mixed logical dynamical system is described by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}(k) + \mathbf{B}_2\underline{\boldsymbol{\delta}}(k) + \mathbf{B}_3\widetilde{\mathbf{z}}(k) \tag{2.16}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}_1\mathbf{u}(k) + \mathbf{D}_2\underline{\boldsymbol{\delta}}(k) + \mathbf{D}_3\widetilde{\mathbf{z}}(k) \tag{2.17}$$

$$\mathbf{E}_2\underline{\boldsymbol{\delta}}(k) + \mathbf{E}_3\widetilde{\mathbf{z}}(k) \leq \mathbf{E}_1\mathbf{u}(k) + \mathbf{E}_4\mathbf{x}(k) + \mathbf{E}_5, \tag{2.18}$$

where the state $\mathbf{x} \in \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}}$, input $\mathbf{u} \in \mathbb{R}^{\widetilde{m}} \times \mathbb{B}^{\underline{m}}$ and output $\mathbf{y} \in \mathbb{R}^{\widetilde{l}} \times \mathbb{B}^{\underline{l}}$ are hybrid and $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$, $\mathbf{C}$, $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$, $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, $\mathbf{E}_4$, and $\mathbf{E}_5$ are matrices with appropriate dimensions. The discrete-valued auxiliary variable $\underline{\boldsymbol{\delta}} \in \mathbb{B}^{\underline{o}}$ and the continuous-valued auxiliary variable $\widetilde{\mathbf{z}} \in \mathbb{R}^{\widetilde{p}}$ are

introduced to take the role of the event generator, here called quantizer. Note that a certain structure is imposed on the matrices defining the next state and the output, such that both are hybrid [13, 14]. This means that, e.g. matrix $\mathbf{A}$ needs to have a lower left zero block. The injector or mode selector is defined implicitly by assuming that the logical TRUE corresponds to the real 1 and the logical FALSE to the real 0.

To give conditions for a mixed logical dynamical system to be well-posed the following sets need to be defined, [12].

Definition 2.1 *Let $\mathbb{I}$ be the set of indices i for which at least one of the i^{th} columns of $\mathbf{B}_2$ or $\mathbf{D}_2$ is nonzero and let $\mathbb{J}$ be the set of indices j for which at least one of the j^{th} columns of $\mathbf{B}_3$ or $\mathbf{D}_3$ is nonzero.*

In other words the set $\mathbb{I}$ contains the columns of $\mathbf{B}_2$ or $\mathbf{D}_2$ which have an impact on the next state and the set $\mathbb{J}$ contains the columns of $\mathbf{B}_3$ or $\mathbf{D}_3$ which have an impact on the output.

Let the discrete-valued part of the next state be denoted by $\underline{\mathbf{x}}(k+1)$. A mixed logical dynamical system is well-posed, if for all $k = 0, 1, \ldots$ and some $\underline{\boldsymbol{\delta}}(k)$, $\widetilde{\mathbf{z}}(k)$, and $\underline{\mathbf{x}}(k+1)$,

- there exist $\mathbf{x}(k)$ and $\mathbf{u}(k)$ such that (2.18) is satisfied and
- for all $i \in \mathbb{I}$ there exists a mapping $\boldsymbol{\Delta} : \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}} \times \mathbb{R}^{\widetilde{m}} \times \mathbb{B}^{\underline{m}} \to \mathbb{B}$ such that the i^{th} component of $\underline{\boldsymbol{\delta}}_i(k) = \boldsymbol{\Delta}(\mathbf{x}(k), \mathbf{u}(k))$ and
- for all $j \in \mathbb{J}$ there exists a mapping $\mathbf{Z} : \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}} \times \mathbb{R}^{\widetilde{m}} \times \mathbb{B}^{\underline{m}} \to \mathbb{R}$ such that the j^{th} component of $\widetilde{\mathbf{z}}_j(k) = \mathbf{Z}(\mathbf{x}(k), \mathbf{u}(k))$.

In other words, a mixed logical dynamical system is well-posed, if for fixed $\mathbf{x}(k)$ and $\mathbf{u}(k)$ the next state $\mathbf{x}(k+1)$ and the output $\mathbf{y}(k)$ are uniquely defined.

A mixed logical dynamical system is completely well-posed, if it is well-posed and the sets $\mathbb{I}$ and $\mathbb{J}$ are $\mathbb{I} = \{1, \ldots, \underline{o}\}$ and $\mathbb{J} = \{1, \ldots, \widetilde{p}\}$. In this case, all auxiliary variables are uniquely defined. See [12] for further details.

2.2.5 Equivalences of hybrid model classes

Relationships exist between the above described classes of hybrid models. A hybrid system, modeled as a well-posed piecewise affine system, can also be modeled as a well-posed discrete hybrid automaton, see Lemma 1 in [11]. The switched affine system of the discrete hybrid automaton (2.12) and the continuous-valued dynamics of the piecewise affine system (2.9) are equivalent. To establish the equivalence of the discrete hybrid automaton and the piecewise affine system, the mode selector of the discrete hybrid automaton needs to select the active dynamics according to the present position of state and input. Without discrete-valued states and inputs, the mode selector (2.15) becomes

$$\underline{i}(k) = \underline{\mathbf{f}}_m(\underline{\boldsymbol{\delta}}_e). \tag{2.19}$$

Since the event generator $\underline{\boldsymbol{\delta}}_e(k)(\widetilde{\mathbf{x}}(k), \widetilde{\mathbf{u}}(k))$ is a function of the continuous-valued state and input, it can divide the input-state space into polyhedral partitions. This results in the same behavior as piecewise affine systems.

Mixed logical dynamical systems are equivalent to discrete hybrid automata under the assumption, that both are well-posed, see Lemma 2 in [11]. Again the continuous dynamics in (2.12) and (2.16)-(2.17) are the same. Now the state automaton can be represented by the discrete part in (2.16)-(2.17). The auxiliary variables in (2.18) take the role of the event generator and the mode selector.

2.3 Tensor models

A survey of tensor models, which were introduced in [15], will be given in this section. Hybrid tensor models are also a framework to represent hybrid systems. The continuous-valued part of the system is in contrast to the ones introduced in the previous sections extended to include multiplications of states and inputs. All multilinear systems are included in the class of polynomial systems and include all bilinear and linear system, see figure 2.5. Note that the term *multilinear* is not unique. Here multilinear means that if all but one variable are fixed the function is linear. In other words multiplications of variables are allowed but no square terms are admissible.

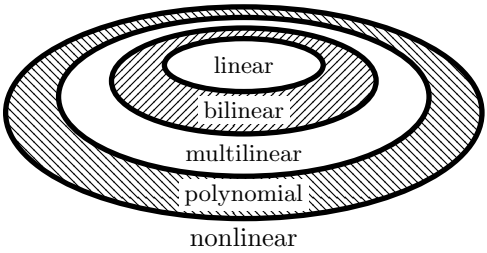


Figure 2.5: Linear, bilinear, multilinear and polynomial systems

After introducing some basics of tensor calculus, first Boolean (discrete-valued) state space models and then multilinear continuous-valued state space models are introduced. These models are combined to be multilinear hybrid models, i.e. tensor models. The section is concluded with a discussion on connecting tensor models.

2.3.1 Tensor calculus

The following standard definitions can be found, for example, in [16, 17]. They are given here to provide a self contained work.

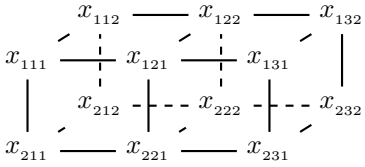
Definition 2.2 *A tensor* X *of order* n *is an* n*-way array*

$$\mathsf{X} \in \mathbb{D}^{I_1 \times I_2 \times \cdots \times I_n}. \tag{2.20}$$

The elements of X *are* $\mathbf{x}_{i_1,i_2,\ldots,i_n} \in \mathbb{D}$*, where the indices are* $i_j \in \{1, \ldots, I_j\}$ *for* $j = 1, \ldots, n$*.*

For a third order tensor this can be illustrated as a three dimensional construct as given in the following example, which is taken from [18].

Example 2.1 *The elements of a tensor* $\mathsf{X} \in \mathbb{R}^{2\times3\times2}$ *can be arranged as*

$\mathsf{X} \;=$ 

.

Tensor decompositions can be used to represent this tensor. The canonical polyadic (CP) decomposed form of tensors will be used later, therefore its definition is given.

Definition 2.3 *A tensor in CP decomposed form is given as*

$$\mathsf{K} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n] \cdot \lambda \in \mathbb{D}^{r_1 \times r_2 \times \cdots \times r_n}. \tag{2.21}$$

The elements of this tensor are given by the sums of the outer products of the columns of the so-called factor matrices $\mathbf{X}_i \in \mathbb{D}^{r_i \times r}$*, weighted by the elements of the so-called* weighting *or* parameter vector λ

$$\mathsf{K} = \sum_{i=1}^{r} \lambda_i \, (\mathbf{X}_1)_i \circ (\mathbf{X}_2)_i \circ \cdots \circ (\mathbf{X}_n)_i,$$

where $(\mathbf{X}_1)_i$ *is the* i^{th} *column of the matrix* $\mathbf{X}_1$*. The symbol* $\circ$ *denotes the outer product*[2]*. An element of the multidimensional tensor* K *is given by*

$$K_{jk\ldots p} = \sum_{i=1}^{r} \lambda_i \, (\mathbf{X}_1)_{ji} \;\; (\mathbf{X}_2)_{ki} \;\; \ldots (\mathbf{X}_n)_{pi},$$

where $(\mathbf{X}_1)_{ji}$ *is the element in the* j^{th} *row and* i^{th} *column of the matrix* $\mathbf{X}_1$ *and* r *is the rank of the tensor. When omitted, the parameter vector is assumed to be a vector of ones, i.e.* $\lambda = (1 \;\; 1 \;\; \ldots \;\; 1)^T$.

To multiply tensors the contracted tensor product is introduced.

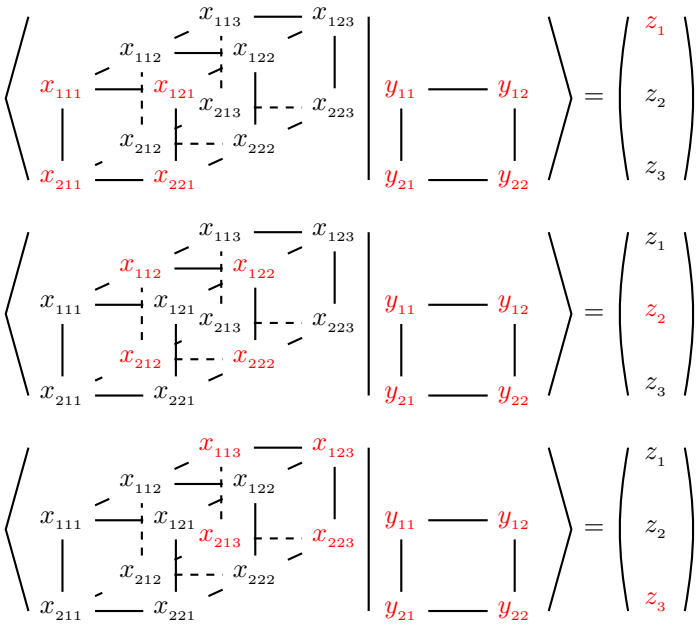
[2]The definition is given in appendix A

Definition 2.4 *The* contracted product *of two tensors* X *and* Y *with* $\mathsf{X} \in \mathbb{D}^{I_1 \times \ldots \times I_n \times I_{n+1} \times \ldots \times I_{n+m}}$ *and* $\mathsf{Y} \in \mathbb{D}^{I_1 \times \ldots \times I_n}$ *is a tensor of dimension* $I_{n+1} \times \ldots \times I_{n+m}$ *given as*

$$\langle \mathsf{X} \,|\, \mathsf{Y} \rangle (k_1, \ldots, k_m) = \sum_{i_1}^{I_1} \cdots \sum_{i_n}^{I_n} x_{i_1,\ldots,i_n,k_1,\ldots,k_m} y_{i_1,\ldots,i_n} \,. \tag{2.22}$$

To illustrate this product a third order example is given, which also is taken from [18].

Example 2.2 *Consider a tensor* $\mathsf{X} \in \mathbb{D}^{2\times2\times3}$ *and a tensor* $\mathsf{Y} \in \mathbb{D}^{2\times2}$. *To calculate the contracted product* $\mathsf{Z} = \langle \mathsf{X} \,|\, \mathsf{Y} \rangle \in \mathbb{D}^3$, *the matching dimensions have to be multiplied elementwise, as highlighted in red in the following illustrations.*



2.3.2 Boolean state space models

Boolean systems can be given in state space form as (2.4) - (2.6) with the domain of input, state and output being Boolean, i.e. $\underline{\mathbf{u}} \in \mathbb{B}^{\underline{m}}$, $\underline{\mathbf{x}} \in \mathbb{B}^{\underline{n}}$, and $\underline{\mathbf{y}} \in \mathbb{B}^{\underline{l}}$. The Boolean system is given as

$$\underline{\mathbf{x}}(k+1) = \underline{\mathbf{f}}(\underline{\mathbf{x}}(k)\underline{\mathbf{u}}(k)), \tag{2.23}$$

$$\underline{\mathbf{y}}(k) = \underline{\mathbf{f}}(\underline{\mathbf{x}}(k)\underline{\mathbf{u}}(k)), \tag{2.24}$$

$$\underline{\mathbf{x}}(0) = \underline{\mathbf{x}}_0, \tag{2.25}$$

where the next state function $\underline{\mathbf{f}} : \mathbb{B}^{\underline{n}} \times \mathbb{B}^{\underline{m}} \to \mathbb{B}^{\underline{n}}$ and the output function $\underline{\mathbf{g}} : \mathbb{B}^{\underline{n}} \times \mathbb{B}^{\underline{m}} \to \mathbb{B}^{\underline{l}}$ are Boolean functions, which could be represented by truth tables. Every Boolean function of n variables is determined, if for each combination of all variables the function value is given. Therefore there is a finite number of different Boolean functions. For n variables there exist 2^{2^n} different functions. All these functions can be represented, e.g. using the Boolean grammar AND, OR, and NOT.

Another way to represent Boolean functions is by Zhegalkin polynomials, [19]. Define the logic TRUE as 1 and the logic FALSE als 0. Then, every Boolean function can be given as a square free polynomial by replacing the the Boolean grammar by the algebraic operations as given in table 2.1.

Boolean function	algebraic function
NOT x_1	$1 - x_1$
x_1 AND x_2	$x_1 x_2$
x_1 OR x_2	$x_1 + x_2 - x_1 x_2$

Table 2.1: Boolean and corresponding algebraic function

Example 2.3 *The Boolean exclusive or function for the Boolean variables* $\underline{b}_1$ *and* $\underline{b}_2$ *is given with the truth table*

$\underline{b}_1$	$\underline{b}_2$	XOR$(\underline{b}_1, \underline{b}_2)$
0	*0*	*0*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

This function can be represented by the algebraic function

$$\underline{f}_b(\underline{b}_1, \underline{b}_2) = \underline{b}_1 + \underline{b}_2 - 2\underline{b}_1\underline{b}_2. \tag{2.26}$$

The symbol $\langle \mid \rangle^+$ indicates that every Boolean TRUE is converted to a real 1 and every Boolean FALSE is converted to a real 0. Restating the discrete-valued state space model (2.23) - (2.25) in form of contracted tensor products gives

$$\underline{\mathbf{x}}(k+1) = \langle\, \underline{\mathsf{F}} \mid \underline{\mathsf{M}}(\underline{\mathbf{x}}(k), \underline{\mathbf{u}}(k)) \,\rangle^+, \tag{2.27}$$

$$\underline{\mathbf{y}}(k) = \langle\, \underline{\mathsf{G}} \mid \underline{\mathsf{M}}(\underline{\mathbf{x}}(k), \underline{\mathbf{u}}(k)) \,\rangle^+, \tag{2.28}$$

where $\underline{\mathsf{M}}(\underline{\mathbf{x}}(k), \underline{\mathbf{u}}(k))$ is the monomial CP tensor

$$\underline{\mathsf{M}}(\underline{\mathbf{x}}(k), \underline{\mathbf{u}}(k)) = \left[\binom{1}{\underline{u}_{\underline{m}}(k)}, \ldots, \binom{1}{\underline{u}_1(k)}, \binom{1}{\underline{x}_{\underline{n}}(k)}, \ldots, \binom{1}{\underline{x}_1(k)} \right], \tag{2.29}$$

see, [15].

2.3.3 Multilinear continuous-valued models

The continuous-valued part of the tensor model includes all combinations of states and inputs. It is called multilinear because if all variables but one are fixed, the system is linear, due to the fact, that no square terms are allowed. All possible combinations of states and inputs, in vector form are given with the following definition.

Definition 2.1 *The* monomial vector *is defined as*

$$\widetilde{\mathbf{m}}(\widetilde{\mathbf{x}},\widetilde{\mathbf{u}})=\begin{pmatrix}1\\ \widetilde{u}_{\widetilde{m}}\end{pmatrix}\otimes\cdots\otimes\begin{pmatrix}1\\ \widetilde{u}_1\end{pmatrix}\otimes\begin{pmatrix}1\\ \widetilde{x}_{\widetilde{n}}\end{pmatrix}\otimes\cdots\otimes\begin{pmatrix}1\\ \widetilde{x}_1\end{pmatrix} \tag{2.30}$$

where $\widetilde{\mathbf{x}}\in\mathbb{R}^{\widetilde{n}}$ *is the state vector and* $\widetilde{\mathbf{u}}\in\mathbb{R}^{\widetilde{m}}$ *is the input vector.*

The symbol $\otimes$ is used to denote the Kronecker product, see appendix A for its definition.

Example 2.4 *For a model with one input and two states the monomial vector becomes*

$$\widetilde{\mathbf{m}}(\widetilde{\mathbf{x}},\widetilde{\mathbf{u}})=\begin{pmatrix}1\\ \widetilde{x}_1\\ \widetilde{x}_2\\ \widetilde{x}_1\widetilde{x}_2\\ \widetilde{u}\\ \widetilde{u}\widetilde{x}_1\\ \widetilde{u}\widetilde{x}_2\\ \widetilde{u}\widetilde{x}_1\widetilde{x}_2\end{pmatrix} \tag{2.31}$$

A multilinear model in matrix representation is then given as

$$\widetilde{\mathbf{x}}(k+1)=\widetilde{\mathbf{F}}\widetilde{\mathbf{m}}(\widetilde{\mathbf{x}}(k),\widetilde{\mathbf{u}}(k)) \tag{2.32}$$

$$\widetilde{\mathbf{y}}(k)=\widetilde{\mathbf{G}}\widetilde{\mathbf{m}}(\widetilde{\mathbf{x}}(k),\widetilde{\mathbf{u}}(k)), \tag{2.33}$$

where the state is $\widetilde{\mathbf{x}}\in\mathbb{R}^{\widetilde{n}}$, the input is $\widetilde{\mathbf{u}}\in\mathbb{R}^{\widetilde{m}}$ and the output is $\widetilde{\mathbf{y}}\in\mathbb{R}^{\widetilde{l}}$. The next state matrix is $\widetilde{\mathbf{F}}\in\mathbb{R}^{n\times 2^n}$ and the output matrix is $\widetilde{\mathbf{G}}\in\mathbb{R}^{l\times 2^n}$.

Example 2.5 *Again a model with one input and two states, as introduced in example 2.4 is considered. The next state vector for this model is assumed to be*

$$\begin{pmatrix}\widetilde{f}_{11}+\widetilde{f}_{12}\widetilde{x}_1+\widetilde{f}_{15}\widetilde{u}+\widetilde{f}_{16}\widetilde{u}\widetilde{x}_1\\ \widetilde{f}_{22}\widetilde{x}_1+\widetilde{f}_{23}\widetilde{x}_2+\widetilde{f}_{24}\widetilde{x}_1\widetilde{x}_2+\widetilde{f}_{26}\widetilde{u}\widetilde{x}_1+\widetilde{f}_{27}\widetilde{u}\widetilde{x}_2+\widetilde{f}_{28}\widetilde{u}\widetilde{x}_1\widetilde{x}_2\end{pmatrix}. \tag{2.34}$$

This model can be rewritten using the monomial vector as

$$\begin{pmatrix}\widetilde{\mathbf{x}}_1(k+1)\\ \widetilde{\mathbf{x}}_2(k+1)\end{pmatrix} = \begin{pmatrix}\widetilde{f}_{11} & \widetilde{f}_{12} & 0 & 0 & \widetilde{f}_{15} & \widetilde{f}_{16} & 0 & 0\\ 0 & \widetilde{f}_{22} & \widetilde{f}_{23} & \widetilde{f}_{24} & 0 & \widetilde{f}_{26} & \widetilde{f}_{27} & \widetilde{f}_{28}\end{pmatrix} \begin{pmatrix}1\\ \widetilde{x}_1(k)\\ \widetilde{x}_2(k)\\ \widetilde{x}_1(k)\widetilde{x}_2(k)\\ \widetilde{u}(k)\\ \widetilde{u}(k)\widetilde{x}_1(k)\\ \widetilde{u}(k)\widetilde{x}_2(k)\\ \widetilde{u}(k)\widetilde{x}_1(k)\widetilde{x}_2(k)\end{pmatrix}. \tag{2.35}$$

Linear transformations allow a numerical preconditioning of models. This can be used, for example, to bring the operating range of the states to the interval [0 1].

Lemma 2.1 *For multilinear models (2.32), linear state and input variable transformations*

$$\widetilde{\widetilde{x}}_i = \frac{\widetilde{x}_i - \widetilde{b}_i}{\widetilde{a}_i} \; \forall i = 1, \ldots, \widetilde{n} \tag{2.36}$$

$$\widetilde{\widetilde{u}}_i = \frac{\widetilde{u}_i - \widetilde{b}_{\widetilde{n}+i}}{\widetilde{a}_{\widetilde{n}+i}} \; \forall i = 1, \ldots, \widetilde{m} \tag{2.37}$$

are computable in closed form [18]. The transformed state transition matrix is given by

$$\widetilde{\widetilde{\boldsymbol{F}}} = \operatorname*{diag}_{i=1,\ldots,\widetilde{n}} \left(\frac{1}{\widetilde{a}_i}\right) \left(\widetilde{\boldsymbol{F}}\widetilde{\boldsymbol{T}} - \begin{pmatrix}\widetilde{\mathbf{b}} & 0_{\widetilde{n}\times 2^{(\widetilde{n}+\widetilde{m})}-1}\end{pmatrix}\right) \tag{2.38}$$

where diag *denotes the diagonal matrix with elements* $\frac{1}{\widetilde{a}_i}$,

$$\widetilde{\boldsymbol{T}} = \begin{pmatrix}1 & 0\\ \widetilde{b}_{\widetilde{n}+\widetilde{m}} & \widetilde{a}_{\widetilde{n}+\widetilde{m}}\end{pmatrix} \otimes \cdots \otimes \begin{pmatrix}1 & 0\\ \widetilde{b}_1 & \widetilde{a}_1\end{pmatrix} \tag{2.39}$$

is a transformation matrix, $\widetilde{\mathbf{b}}$ *is a column vector containing the* $\widetilde{b}_i$*'s and* $0_{\widetilde{n}\times 2^{\widetilde{n}+\widetilde{m}}-1}$ *denotes a zero matrix with dimension* $(\widetilde{n} \times 2^{\widetilde{n}+\widetilde{m}} - 1)$.

Proof: The proof follows from inserting (2.36) and (2.37) into (2.32).

□

To represent multilinear systems in tensor form the continuous-valued monomial CP tensor

$$\widetilde{\mathsf{M}}(\widetilde{\mathbf{x}}(k),\widetilde{\mathbf{u}}(k)) = \left[\begin{pmatrix}1\\ \widetilde{u}_{\widetilde{m}}(k)\end{pmatrix},\ldots,\begin{pmatrix}1\\ \widetilde{u}_{1}(k)\end{pmatrix},\begin{pmatrix}1\\ \widetilde{x}_{\widetilde{n}}(k)\end{pmatrix},\ldots,\begin{pmatrix}1\\ \widetilde{x}_{1}(k)\end{pmatrix}\right] \tag{2.40}$$

is introduced. Using the contracted product, the next state equation (2.32) and the output equation (2.33) can be given as

$$\widetilde{\mathbf{x}}(k+1) = \left\langle \widetilde{\mathsf{F}} \,\middle|\, \widetilde{\mathsf{M}}(\widetilde{\mathbf{x}}(k),\widetilde{\mathbf{u}}(k)) \right\rangle \tag{2.41}$$

$$\widetilde{\mathbf{y}}(k) = \left\langle \widetilde{\mathsf{G}} \,\middle|\, \widetilde{\mathsf{M}}(\widetilde{\mathbf{x}}(k),\widetilde{\mathbf{u}}(k)) \right\rangle, \tag{2.42}$$

where all parameters of the model are contained in the transition tensor $\widetilde{\mathsf{F}} \in \mathbb{R}^{\times^{(n+m)}2\times n}$ and the output tensor $\widetilde{\mathsf{G}} \in \mathbb{R}^{\times^{(n+m)}2\times l}$, [15]. Here a simple notation for tensor spaces is introduced

$$\mathbb{R}^{\times^{(n+m)}2} := \mathbb{R}^{\overbrace{2\times\ldots\times 2}^{n+m}}. \tag{2.43}$$

Once again continuing with the examples 2.4 and 2.5 a tensor model is shown to be equivalent to the multilinear system (2.35).

Example 2.6 *Using the state transition tensor in CP decomposed form*

$$\widetilde{\mathsf{F}} = \left[\widetilde{\boldsymbol{F}}_{\widetilde{u}},\widetilde{\boldsymbol{F}}_{\widetilde{x}_2},\widetilde{\boldsymbol{F}}_{\widetilde{x}_1},\widetilde{\boldsymbol{F}}_{\Phi}\right]\cdot\widetilde{\boldsymbol{\lambda}} \tag{2.44}$$

with the factor matrices

$$\widetilde{\boldsymbol{F}}_{\widetilde{x}_1} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{2.45}$$

$$\widetilde{\boldsymbol{F}}_{\widetilde{x}_2} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \tag{2.46}$$

$$\widetilde{\boldsymbol{F}}_{\widetilde{u}} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \tag{2.47}$$

$$\widetilde{\boldsymbol{F}}_{\Phi} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{2.48}$$

and the continuous-valued parameter vector

$$\widetilde{\boldsymbol{\lambda}} = \begin{pmatrix} \widetilde{f}_{11} & \widetilde{f}_{12} & \widetilde{f}_{15} & \widetilde{f}_{16} & \widetilde{f}_{22} & \widetilde{f}_{23} & \widetilde{f}_{24} & \widetilde{f}_{26} & \widetilde{f}_{27} & \widetilde{f}_{28} \end{pmatrix}^T \tag{2.49}$$

it is straight forward to calculate the next state as the contracted product of two tensors in CP decomposed form. This product can easily be calculated as

$$\widetilde{\mathbf{x}}(k+1) = \left\langle \widetilde{\mathsf{F}} \,\middle|\, \widetilde{\mathsf{M}}(\widetilde{\mathbf{x}}(k), \widetilde{\mathbf{u}}(k)) \right\rangle \tag{2.50}$$

$$= \widetilde{\boldsymbol{F}}_{\Phi} \left(\widetilde{\boldsymbol{\lambda}} \circledast \left(\widetilde{\boldsymbol{F}}_{\widetilde{u}}^T \begin{pmatrix} 1 \\ \widetilde{u} \end{pmatrix} \right) \circledast \left(\widetilde{\boldsymbol{F}}_{\widetilde{x}_2}^T \begin{pmatrix} 1 \\ \widetilde{x}_2 \end{pmatrix} \right) \circledast \left(\widetilde{\boldsymbol{F}}_{\widetilde{x}_1}^T \begin{pmatrix} 1 \\ \widetilde{x}_1 \end{pmatrix} \right) \right) \tag{2.51}$$

$$= \begin{pmatrix} \widetilde{f}_{11} + \widetilde{f}_{12}\widetilde{x}_1 + \widetilde{f}_{15}\widetilde{u} + \widetilde{f}_{16}\widetilde{u}\widetilde{x}_1 \\ \widetilde{f}_{22}\widetilde{x}_1 + \widetilde{f}_{23}\widetilde{x}_2 + \widetilde{f}_{24}\widetilde{x}_1\widetilde{x}_2 + \widetilde{f}_{26}\widetilde{u}\widetilde{x}_1 + \widetilde{f}_{27}\widetilde{u}\widetilde{x}_2 + \widetilde{f}_{28}\widetilde{u}\widetilde{x}_1\widetilde{x}_2 \end{pmatrix}. \tag{2.52}$$

The sysmbol $\circledast$ denotes the Hadamard product (element-wise product). The definition is given in appendix A.

The procedure to construct the factor matrices and the parameter vector is as follows: set the columns of the factor matrices to $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, if the summand contains the corresponding state or input and to $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ otherwise. The product of states and inputs is weighted by a constant given in the

parameter vector. The factor matrix $\widetilde{\mathbf{F}}_\Phi$ indicates to which next state element the summand is added.

This is shown exemplarily for the first summand of the Example. All first columns of the factor matrices are $\begin{pmatrix}1\\0\end{pmatrix}$, i.e. the first summand of the first next state does not depend on any current state or input but is equal to the first entry f_{11} of the parameter vector. Since the first four elements in the first row of $\widetilde{\mathbf{F}}_\Phi$ are 1 and all other elements in this row are 0, the first four summands construct the first next state $\widetilde{x}_1(k+1)$ and the remaining six summands construct the second next state $\widetilde{x}_2(k+1)$.

2.3.4 Multilinear hybrid models

A multilinear hybrid model is constructed by connecting a Boolean state space model with a continuous-valued multilinear model. Figure 2.6 shows this structure, which is quite similar to the structure of a discrete hybrid automaton. Since the continuous-valued multilinear system and the Boolean system both allow a direct feed-through of state and input, there are strong similarities of discrete hybrid automata and hybrid tensor model. The feedthrough terms can be used to model the external inputs of the quantizer and the injector.

Obviously the blocks *injector* and *quantizer* need further definitions, which are given in the following and are taken from [15].

Definition 2.5 *The function* $\alpha : \mathbb{H}^{I_1\times\cdots\times I_N} \to \mathbb{R}^{I_1\times\cdots\times I_N}$ *is called* standard injector*, which is given with index vector* $\mathbf{i} \in \mathbb{N}^N$ *for all elements by*

$$(\alpha(\mathbf{x}))_\mathbf{i} = \begin{cases} 1 \in \mathbb{R} & \text{if } x_\mathbf{i} = \textsc{True}\,, \\ 0 \in \mathbb{R} & \text{if } x_\mathbf{i} = \textsc{False}\,, \\ x_\mathbf{i} & \text{if } x_\mathbf{i} \in \mathbb{R}\,. \end{cases} \tag{2.53}$$

In case the input of the function α is a vector, the index vector $\mathbf{i}$ becomes a scalar.

Definition 2.6 *The function* $\beta : \mathbb{R}^{I_1\times\cdots\times I_N} \to \mathbb{B}^{I_1\times\cdots\times I_N}$ *is called* standard quantizer *and is given for all elements with index vector* $\mathbf{i} \in \mathbb{N}^N$

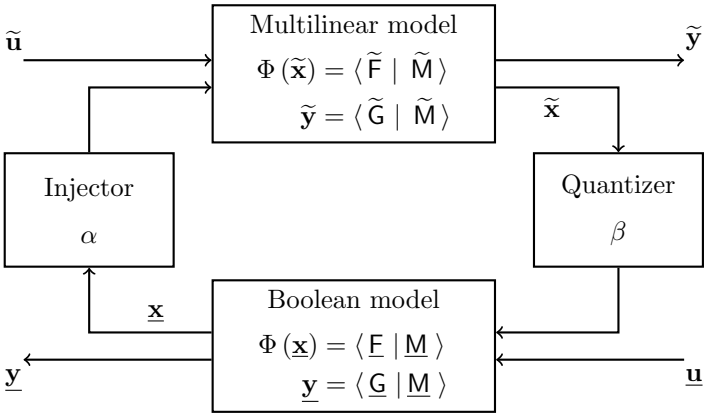


Figure 2.6: Structure of a multilinear hybrid model.

by

$$(\beta(\mathbf{x}))_{\mathbf{i}} = \sigma(x_{\mathbf{i}} - \frac{1}{2}) \,, \tag{2.54}$$

where the Heaviside *function is given as*

$$\sigma(x) = \begin{cases} 1 & \textit{if } x \geq 0 \,, \\ 0 & \textit{otherwise} \,. \end{cases} \tag{2.55}$$

To state the tensor model in a compact form one further definition is needed, which will define a hybrid contracted tensor product which includes the injector and the quantizer, see [15].

Definition 2.7 *The notation* $\langle \cdot \mid \cdot \rangle^{\boxplus}$ *is used for the* hybrid contracted tensor product*. This product maps all values to the correct domains, which are given by the domains of the tensor on the left side of the con-*

tracted product using the standard quantizer β and the injector α

$$\left\langle \underline{\mathsf{F}} \,\middle|\, \widetilde{\mathsf{M}}(\mathbf{x}) \right\rangle^{\boxplus} = \left\langle \underline{\mathsf{F}} \,\middle|\, \widetilde{\mathsf{M}}(\beta(\mathbf{x})) \right\rangle \tag{2.56}$$

$$\left\langle \widetilde{\mathsf{F}} \mid \underline{\mathsf{M}}(\mathbf{x}) \right\rangle^{\boxplus} = \left\langle \widetilde{\mathsf{F}} \mid \alpha(\underline{\mathsf{M}}(\mathbf{x})) \right\rangle \tag{2.57}$$

$$\left\langle \widetilde{\mathsf{F}} \,\middle|\, \widetilde{\mathsf{M}}(\mathbf{x}) \right\rangle^{\boxplus} = \left\langle \widetilde{\mathsf{F}} \,\middle|\, \widetilde{\mathsf{M}}(\mathbf{x}) \right\rangle \tag{2.58}$$

$$\langle \underline{\mathsf{F}} \mid \underline{\mathsf{M}}(\mathbf{x}) \rangle^{\boxplus} = \langle \underline{\mathsf{F}} \mid \underline{\mathsf{M}}(\mathbf{x}) \rangle \ . \tag{2.59}$$

If the tensor on the left hand side of the contracted product is hybrid valued, the operations are applied to the subtensors. With this definition the hybrid tensor system can be given by

$$\Phi(\mathbf{x}) = \langle \mathsf{F} \mid \mathsf{M}(\mathbf{x}, \mathbf{u}) \rangle^{\boxplus} \ . \tag{2.60}$$

$$\mathbf{y} = \langle \mathsf{G} \mid \mathsf{M}(\mathbf{x}, \mathbf{u}) \rangle^{\boxplus}, \tag{2.61}$$

where the hybrid state is $\mathbf{x} = \begin{pmatrix} \widetilde{\mathbf{x}} \\ \underline{\mathbf{x}} \end{pmatrix} \in \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}} = \mathbb{H}^n$, the hybrid input is $\mathbf{u} = \begin{pmatrix} \widetilde{\mathbf{u}} \\ \underline{\mathbf{u}} \end{pmatrix} \in \mathbb{H}^m$ and the hybrid output is $\mathbf{y} = \begin{pmatrix} \widetilde{\mathbf{y}} \\ \underline{\mathbf{y}} \end{pmatrix} \in \mathbb{H}^l$, [15].

2.3.5 Connecting tensor models

Series connection

Consider the series connection of hybrid tensor models, as given in figure 2.7.

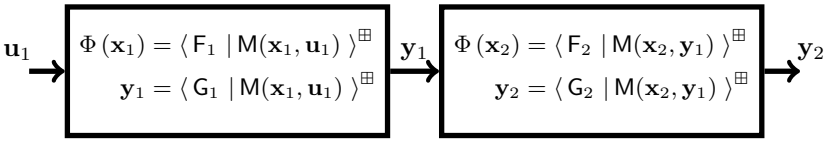


Figure 2.7: Series connection of two tensor systems.

Lemma 2.2 *For two hybrid SISO tensor models, the series connection still gives a tensor model.*

Proof: The output of the first system can only contain multilinear combinations of input and the states of the first system $y = \mathbf{g}_1\mathbf{m}(\mathbf{x}_1, \mathbf{u}_1)$. If this output is scalar and the above connection is used, i.e. the output of the first system is used as input of the second system, multilinear terms of the second systems states and the output of the first system occur $\Phi(\mathbf{x_2}) = \mathbf{F}_2\mathbf{m}(\mathbf{x_2}, \mathbf{gm}(\mathbf{x}_1, \mathbf{u}_1)) = \ldots \mathbf{m}(\mathbf{x_2}, \mathbf{x}_1, \mathbf{u}_1)$. I.e. only multilinear terms of the first systems input, the first systems state and the second systems state are possible and no square or higher order terms can be constructed. □

The following example illustrates the series connection of two discrete-time continuously valued SISO tensor models.

Example 2.7 *Assume the models connected in series are*

$$\mathbf{x}_1(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}u(k) + f_{14}u(k)x_1(k)x_2(k) \end{pmatrix} \tag{2.62}$$

$$\mathbf{y}_1(k) = g_{11}x_1(k) + g_{12}x_1(k)x_2(k) \tag{2.63}$$

and

$$\mathbf{x}_2(k+1) = \begin{pmatrix} x_3(k+1) \\ x_4(k+1) \end{pmatrix} = \begin{pmatrix} f_{21}x_3(k) + f_{22}y_1(k) \\ f_{23}x_3(k)x_4(k) \end{pmatrix} \tag{2.64}$$

$$\mathbf{y}_2(k) = g_{21}x_4(k). \tag{2.65}$$

The series connection is

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}u(k) + f_{14}u(k)x_1(k)x_2(k) \\ f_{21}x_3(k) + f_{22}g_{11}x_1(k) + f_{22}g_{12}x_1(k)x_2(k) \\ f_{23}x_3(k)x_4(k) \end{pmatrix} \tag{2.66}$$

$$\mathbf{y}_2(k) = g_{21}x_4(k). \tag{2.67}$$

This still contains only multilinear terms and therefore can be rewritten

as

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{pmatrix} = \left\langle \mathsf{F} \middle| \mathsf{M} \left(\begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{pmatrix}, \mathbf{u}_1 \right) \right\rangle^{\boxplus} \tag{2.68}$$

with $\mathsf{F} = [\mathbf{F}_{x_1}, \mathbf{F}_{x_2}, \mathbf{F}_{x_3}, \mathbf{F}_{x_4}, \mathbf{F}_\Phi] \cdot \lambda_f$, $\mathsf{G} = [\mathbf{G}_{x_1}, \mathbf{G}_{x_1}, \mathbf{G}_{x_1}, \mathbf{G}_{x_1}, \mathbf{G}_\Phi] \cdot \lambda_g$ *and the factor matrices and parameter vectors*

$$\begin{aligned}
\mathbf{F}_{x_1} &= \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, & \mathbf{G}_{x_1} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\
\mathbf{F}_{x_2} &= \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}, & \mathbf{G}_{x_2} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\
\mathbf{F}_{x_3} &= \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, & \mathbf{G}_{x_3} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\
\mathbf{F}_{x_4} &= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, & \mathbf{G}_{x_4} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \\
\mathbf{F}_\Phi &= \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, & \mathbf{G}_\Phi &= 1,
\end{aligned} \tag{2.69}$$

$$\lambda_f = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{21} & f_{22}g_{11} & f_{22}g_{12} & f_{23} \end{pmatrix}^T, \quad \lambda_g = g_{21}.$$

which is a CP tensor model.

The series connection of two MIMO tensor models is not necessarily a tensor model. If the first system has multiple outputs, one way to avoid multiplications, that are not possible in tensor systems, is to ensure, that for the construction of the outputs states and inputs can be used only once. Otherwise it is possible that the inputs of the second system are multiplied with each other. If two inputs of the second system contain, for example, the first state of the first system, a square term would occur, if these inputs are multiplied with each other in the second system. This means, that in CP decomposed form, per factor matrix only one column

is allowed to be $\begin{pmatrix}0\\1\end{pmatrix}$. If at least one state is used in two different outputs, the series connection might no longer be a tensor model, but a polynomial system as illustrated in the following example.

Lemma 2.3 *The series connection of two MIMO tensor models is a tensor model, if the input and states of the first system are present only once in the outputs of the first system.*

Proof: It is easy to see, that if the inputs of the second system only contain different inputs and states of the first system, their multilinear combinations are still multilinear, since no squares or higher oder terms can be constructed. □

This is similar to the SISO case where just one multilinear combination of first input and first state enter the second system.

Example 2.8 *Consider the systems*

$$\mathbf{x}_1(k+1) = x_1(k+1) = f_{11}x_1(k) + f_{12}x_1(k)u_1(k) \tag{2.70}$$

$$\mathbf{y}_1(k) = \begin{pmatrix}y_1(k)\\y_2(k)\end{pmatrix} = \begin{pmatrix}g_{11}x_1(k)\\g_{12}x_1(k)u_1(k)\end{pmatrix} \tag{2.71}$$

and

$$\mathbf{x}_2(k+1) = x_2(k+1) = f_{21}x_2(k) + f_{22}y_1(k)y_2(k) \tag{2.72}$$

$$\mathbf{y}_2(k) = g_{21}x_2(k). \tag{2.73}$$

Connecting them in series gives

$$\begin{pmatrix}x_1(k+1)\\x_2(k+1)\end{pmatrix} = \begin{pmatrix}f_{11}x_1(k) + f_{12}x_1(k)u_1(k)\\f_{21}x_2(k) + f_{22}g_{11}g_{12}x_1^2(k)u_1(k)\end{pmatrix} \tag{2.74}$$

$$\mathbf{y}_2(k) = g_{21}x_2(k). \tag{2.75}$$

This is no longer a tensor system because term $f_{22}g_{11}g_{12}x_1^2(k)u_1(k)$ contains the square of x_1, which can not be modeled directly in this framework. One possible way to still model this system as a tensor model could be to augment the system state and model the first state twice. If both x_1 have the same initial condition, it would be possible to model them as a tensor model.

Parallel connection For the parallel connection of two tensor systems no restrictions are needed. The parallel connection is given in figure 2.8.

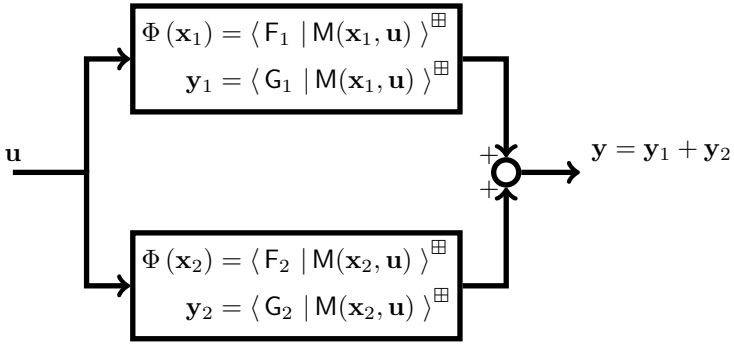


Figure 2.8: Parallel connection of two tensor systems

Lemma 2.4 *The parallel connection of two MIMO tensor models is a tensor model.*

Proof: The state equations are combined by appending them to each other. The output is constructed by adding the output of the first and the second system. In each component of the output terms are added due to the plus sign but no multiplications can occur. This still gives a tensor model. □

Feedback connection For the feedback connection of two tensor systems the restrictions are even more strict. The feedback structure is given in figure 2.9.

First an example where the feedback connected system can no longer be represented as a tensor model is given to point out the problem.

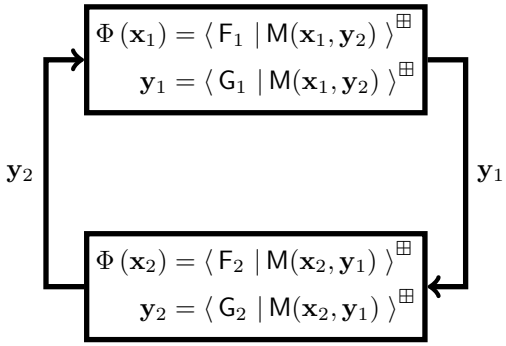


Figure 2.9: Feedback connection of two tensor systems

Example 2.9 *Consider the systems*

$$\mathbf{x}_1(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}y_2(k) + f_{14}y_2(k)x_1(k)x_2(k) \end{pmatrix} \tag{2.76}$$

$$\mathbf{y}_1(k) = y_1(k) = g_{12}x_1(k)y_2(k) \tag{2.77}$$

and

$$\mathbf{x}_2(k+1) = \begin{pmatrix} x_3(k+1) \\ x_4(k+1) \end{pmatrix} = \begin{pmatrix} f_{21}x_3(k) + f_{22}x_4(k) \\ f_{23}y_1(k)x_3(k)x_4(k) \end{pmatrix} \tag{2.78}$$

$$\mathbf{y}_2(k) = y_2(k) = g_{21}x_4(k). \tag{2.79}$$

The closed loop system then becomes

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}g_{21}x_4(k) + f_{14}g_{21}x_1(k)x_2(k)x_4(k) \\ f_{21}x_3(k) + f_{22}x_4(k) \\ f_{23}g_{12}g_{21}x_1(k)x_3(k)x_4^2(k) \end{pmatrix} \tag{2.80}$$

This can not be modeled as a tensor model without cloning the states because of the term $f_{23}g_{12}g_{21}x_1(k)x_3(k)x_4^2(k)$.

If in addition to the restriction on the outputs given for series connections, the systems are input affine, i.e. there are no multiplications of

inputs or of input and state, this problem can no longer occur. Another possibility is to look at systems with no direct feed-through, i.e. with an output function, that does not depend on the input. Using such an output function also no polynomial terms can occur and therefore the feedback system can be represented as a tensor model.

In the following we will use input affine multilinear systems which are therefore introduced.

Definition 2.8 *Input affine multilinear systems in matrix representation can be given as*

$$\Phi(\mathbf{x}) = \mathbf{F}\mathbf{m}(\mathbf{x})^{\boxplus} + \mathbf{B}\mathbf{u}^{\boxplus} \tag{2.81}$$

$$\mathbf{y} = \mathbf{G}\mathbf{m}(\mathbf{x})^{\boxplus} + \mathbf{D}\mathbf{u}^{\boxplus}, \tag{2.82}$$

where $^{\boxplus}$ *indicates that the the standard quantizer and injector are used to map discrete and continuous valued states and inputs to the correct domain defined by the corresponding domain of the state or output.*

Lemma 2.5 *The feedback connection of two tensor models is a tensor model if*

- *the inputs and states are present only once in the outputs of the systems*

and one of the following conditions holds

- *the systems are input affine or*
- *the systems have no direct feed-through.*

Proof: The first part follows directly from lemma 2.3. Looking at the feedback connection of input affine systems, the outputs contain multilinear terms of the states and linear terms of the inputs. Since the output of one system acts as input of the other system no multilinear terms of the outputs occur. Therefore no square or higher order terms are possible. Looking at systems with no direct-feedthrough, the inputs are multilinear combinations of the states of the other system. Taking

multilinear combinations with the states of the system still gives only multilinear terms. □

Starting from example 2.9 first the systems are altered to be input affine and then without direct feed-through.

Example 2.10 *Consider the input affine systems*

$$\mathbf{x}_1(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}y_2(k) + f_{14}x_1(k)x_2(k) \end{pmatrix} \tag{2.83}$$

$$\mathbf{y}_1(k) = y_1(k) = g_{12}y_2(k) \tag{2.84}$$

and

$$\mathbf{x}_2(k+1) = \begin{pmatrix} x_3(k+1) \\ x_4(k+1) \end{pmatrix} = \begin{pmatrix} f_{21}x_3(k) + f_{22}x_4(k) \\ f_{23}y_1(k) \end{pmatrix} \tag{2.85}$$

$$\mathbf{y}_2(k) = y_2(k) = g_{21}x_4(k). \tag{2.86}$$

The closed loop can be given as

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}g_{21}x_4(k) + f_{14}x_1(k)x_2(k) \\ f_{21}x_3(k) + f_{22}x_4(k) \\ f_{23}g_{12}g_{21}x_4(k) \end{pmatrix}, \tag{2.87}$$

which contains only multilinear terms and can therefore be modeled as a tensor model.

Example 2.11 *Now, consider a systems, which is not input affine, but without direct feed-through given as*

$$\mathbf{x}_1(k+1) = \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}y_2(k) + f_{14}y_2(k)x_1(k)x_2(k) \\ y_2(k) \end{pmatrix} \tag{2.88}$$

$$\mathbf{y}_1(k) = y_1(k) = g_{12}x_1(k)x_3(k) \tag{2.89}$$

and

$$\mathbf{x}_2(k+1) = \begin{pmatrix} x_4(k+1) \\ x_5(k+1) \end{pmatrix} = \begin{pmatrix} f_{21}x_4(k) + f_{22}x_5(k) \\ f_{23}y_1(k)x_4(k)x_5(k) \end{pmatrix} \tag{2.90}$$

$$\mathbf{y}_2(k) = y_2(k) = g_{21}x_5(k). \tag{2.91}$$

The closed loop system then becomes

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \end{pmatrix} = \begin{pmatrix} f_{11}x_1(k) + f_{12}x_1(k)x_2(k) \\ f_{13}g_{21}x_5(k) + f_{14}y_2(k)x_1(k)x_2(k) \\ g_{21}x_5(k) \\ f_{21}x_4(k) + f_{22}x_5(k) \\ f_{23}g_{12}x_1(k)x_3(k)x_4(k)x_5(k) \end{pmatrix}, \tag{2.92}$$

which can be represented with tensor models. In comparison to the model in example 2.9 the term y_2 in the output $\mathbf{y}_1$ is delayed by one sample by adding another state.

3 Modeling heating systems using thermal balances

Thermal balances will be used in this chapter to describe models of heating system components. Starting with first principle equations, grey box models for various components of heating systems are derived. First these models are given as general nonlinear state space models. Then, some components are given as switched affine models and finally they are rewritten as tensor models.

The basis of the following models is the law of conservation of energy, which states that no energy can be generated or lost but it can change its form. The heating system components supply other components with heat (the burner supplies the boiler, the boiler supplies the radiators, the radiators supply the building and the building looses heat to the ambiance). Therefore only thermal energy will be of interest and all other forms of energy like potential or kinetic energy are assumed to have negligible effects.

The thermal energy, i.e. the heat, stored in a system, e.g. the system depicted in figure 3.1, can be given as

$$Q(t) = \rho\, c\, V(t)\, T(t), \tag{3.1}$$

where ρ is the density, c the specific heat coefficient, $V(t)$ the volume and $T(t)$ the temperature of the medium in the system.

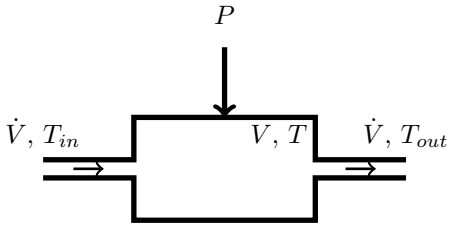


Figure 3.1: Component of a heating system

Taking the derivative of the heat gives the heat flow

$$\dot{Q}(t) = \rho c(\dot{V}(t)T(t) + V(t)\dot{T}(t)), \tag{3.2}$$

which is a power. If the volume of the system does not change, which is true for almost all heating system components, the heat flow reduces to

$$\dot{Q}(t) = \rho c V \dot{T}(t). \tag{3.3}$$

If there is a flow $\dot{V}(t)$ with the temperature $T_{in}(t)$ into the system, energy is transported into the system. In the time interval Δt the amount of energy being transported is equal to $\Delta Q_{in} = \rho c \Delta V T_{in}$, assuming the temperature is constant in the short time interval and $\Delta V = \int_t^{t+\Delta t} \dot{V}(\tau)\mathrm{d}\tau$. If the time interval is reduced to be infinitesimal, the energy transported into the system is also infinitesimal. This is in contrast to the heat flow $\dot{Q}_{in}(t)$, which is derived by taking the limit of the difference quotient for $\Delta t \to 0$, given as

$$\begin{aligned} \dot{Q}_{in}(t) = \lim_{\Delta t \to 0} \frac{\Delta Q_{in}(t)}{\Delta t} = \\ \lim_{\Delta t \to 0} \left(\rho c \Delta V(t) \frac{\Delta T_{in}(t)}{\Delta t} + \rho c \frac{\Delta V(t)}{\Delta t} T_{in}(t) \right) = \rho c \dot{V}(t) T_{in}(t). \end{aligned} \tag{3.4}$$

The first part of the chain rule in equation (3.4) is neglected, because the temperature $T_{in}(t)$ is assumed to be constant during the infinitesimal time interval i.e. $\dot{T}_{in}(t) = 0$, also the volume $\Delta V(t)$ is infinitesimal. Similarly, if there is a flow out of the system, energy is transported out of the system, thus for the heat and the heat flow out of the system the equations $\Delta Q_{out}(t) = \rho c \Delta V(t) T_{out}(t)$ and $\dot{Q}_{out}(t) = \rho c \dot{V}(t) T_{out}(t)$ are obtained.

To model the dynamical process heat flow balances are used. For the components all heat flows into the system are accumulated; heat flows out of the system are counted negative. The sum of heat flows augments or decreases the amount of energy stored in the device, i.e. it equals the change of stored energy. The heat flow balance for the component depicted in figure 3.1 is

$$\dot{Q}(t) = \dot{Q}_{in}(t) - \dot{Q}_{out}(t) + P(t) \tag{3.5}$$

$$\rho c V(t) \dot{T}(t) = \rho c \dot{V}(t) T_{in}(t) - \rho c \dot{V}(t) T_{out}(t) + P(t), \tag{3.6}$$

where $P(t)$ is an additional thermal power brought into the system e.g. by a burner. In general a last assumption will be made, which is to assume that the medium in the component is stirred immediately. This means that the temperature of the component $T(t)$ equals the temperature of the flow out of the component $T_{out}(t) = T(t)$. With this assumption a differential equation for the temperature T_{out} out of the component can be given, for the above example as

$$\dot{T}_{out}(t) = \frac{1}{V}\dot{V}(t)T_{in}(t) - \frac{1}{V}\dot{V}(t)T_{out}(t) + \frac{1}{\rho c V}P(t). \tag{3.7}$$

To obtain a difference equation for a time-discrete application, the first order Euler forward method is used, resulting in

$$T_{out}(k+1) = T_{out}(k)(1 - \frac{dt}{V}\dot{V})\frac{dt}{V}\dot{V}T_{in}(k) + \frac{dt}{\rho c V}P(k), \tag{3.8}$$

where dt is the sampling period and $k = 1, 2, \ldots$ are the sampling instances. Since in general the systems under investigation are slow, i.e. a sampling time of one minute is sufficient, this method was found to give tolerable results.

3.1 Nonlinear models of heating system components

In this section nonlinear models of heating system components are given. Starting with heat generation followed by a storage unit and a consumer model. Finally, an entire heating system will be modeled.

3.1.1 Heat generation unit

The investigated heat generation unit is a burner which is attached to a boiler. Note that the presented equations are easily extendable for combined heat and power plants, since only the thermal processes are investigated.

In figure 3.2 a boiler and a burner are illustrated. The heat flow consists of three parts. Water with a flow rate of $\dot{V}(t)$ and the return temperature $T_r(t)$ enters the boiler and water with the same flow rate and the supply temperature $T_s(t)$ leaves the boiler. The burner supplies the

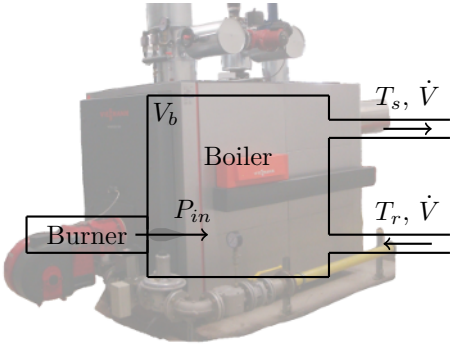


Figure 3.2: Boiler and burner of a heating system

boiler with a heat flow denoted as $P_{in}(t)$. With the volume V_b of the boiler and the assumption that the water is stirred immediately, the heat flow balance can be given as

$$\rho c V_b \dot{T}_s(t) = \rho c \dot{V}(t) T_r(t) - \rho c \dot{V}(t) T_s(t) + P_{in}(t). \tag{3.9}$$

The differential equation for the supply temperature is therefore

$$\dot{T}_s(t) = \frac{1}{V_b}\dot{V}(t) T_r(t) - \frac{1}{V_b}\dot{V}(t) T_s(t) + \frac{1}{\rho c V_b} P_{in}(t). \tag{3.10}$$

3.1.2 Heat exchanger

Power plants often use the waste heat to supply buildings, which are close by, with heat. To decouple the power plants pipeline network from the pipes in the building, heat exchanger are used. They are also used within a heating system, if the heating substance needs to be separated (e.g. the water in solar panels is usually mixed with substances which avoid congelation or the water, used in the radiators, needs to be separated from hot tap water).

The basic principle of a heat exchanger is depicted in figure 3.3. The model of the heat exchanger is based on [20, 21, 22] and references

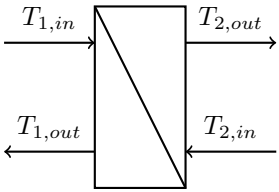


Figure 3.3: Scheme of a heat exchanger

therein. The water from the primary side (subscript 1) is used to heat up the water on the secondary side (subscript 2). Here a counter flow heat exchanger is modeled. I.e. the water on the primary side flows in the opposite direction as the water on the secondary side. This has the advantage, that the water entering the heat exchanger from the primary side with temperature $T_{1,in}$ will heat up the already heated water exiting the heat exchanger on the secondary side with temperature $T_{2,out}$. But the cooled down water with temperature $T_{1,out}$ exiting the primary side will still be able to heat up the cold water, with temperature $T_{2,in}$ entering the heat exchanger on the secondary side. In between the inputs and outputs the water is heated analogously.

The thermal processes can be summarized as follows. First, heat is transferred from the water on the primary side to the wall separating primary and secondary side. Then the heat is transferred through the wall and third the heat is transferred from the wall to the water on the secondary side. These three steps can be given in form of the equations

$$\dot{Q}_1 = \alpha_1 A \left(T_1 - T_{w,1}\right), \tag{3.11}$$

$$\dot{Q}_2 = \lambda \frac{A}{d} \left(T_{w,1} - T_{w,2}\right), \tag{3.12}$$

$$\dot{Q}_3 = \alpha_2 A \left(T_{w,2} - T_2\right), \tag{3.13}$$

where $T_{w,i}$, $i = 1, 2$ are the temperatures of the wall on the primary and secondary side. Combining these equations gives

$$\dot{Q} = kA(T_1 - T_2) \text{ with} \tag{3.14}$$

$$\frac{1}{k} = \frac{1}{\alpha_1} + \frac{d}{\lambda} + \frac{1}{\alpha_2}. \tag{3.15}$$

In the above equations A is the area of the wall under investigation (assumed to be equal for the three steps), α_i are the heat transfer coefficients on the primary (subscript 1) and secondary (subscript 2) side. The thermal conductivity and the thickness of the wall are denoted by λ and d. Note, that k is assumed to be constant, where in reality it is flow dependent, leading to a model valid in a certain operating area.

Constructing heat flow balances for the primary and the secondary side gives

$$\rho c \dot{V}_1 \left(T_{1,in} - T_{1,out}\right) - kA\Delta T_m = \rho c V_1 \dot{T}_{1,out}, \tag{3.16}$$

$$\rho c \dot{V}_2 \left(T_{2,in} - T_{2,out}\right) + kA\Delta T_m = \rho c V_2 \dot{T}_{2,out}, \tag{3.17}$$

where $kA\Delta T_m$ is the heat transferred from the primary to the secondary side over the whole length of the heat exchanger. Here, the mean temperature difference

$$\Delta T_m = \frac{\left(T_{2,out} - T_{2,in}\right) - \left(T_{1,in} - T_{1,out}\right)}{\ln\left(\frac{T_{1,out} - T_{2,in}}{T_{1,in} - T_{2,out}}\right)}, \tag{3.18}$$

is used, where the temperature drop in the heat exchanger is assumed to depend linearly on the position and not on the time, see [20] for details.

Heat exchangers suffer from fouling, i.e. the deposit of pollutant from the water on the walls. This will lead to a decrease of the heat transfer from one side to the other. The thickness of the pollutant will stop increasing after some time, when the deposit of new pollutant and the detachment of old pollutant, due to the flow, will be balanced. The heat exchanger model integrates fouling by having a time-dependent heat transfer coefficient

$$k(t) = \left(R_f + R_{f\infty}\left(1 - e^{\beta t}\right)\right)^{-1}, \tag{3.19}$$

where R_f is the thermal resistivity, $R_{f\infty}$ the final thermal resistivity for $t \to \infty$ and $\beta > 0$ the fouling coefficient. This will only be valid for turbulent flows.

The differential equations for the supply temperatures read

$$\dot{T}_{1,out} = \frac{1}{V_1}\dot{V}_1 \left(T_{1,in} - T_{1,out}\right) - \frac{k(t)A}{\rho c V_1}\Delta T_m, \tag{3.20}$$

$$\dot{T}_{2,out} = \frac{1}{V_2}\dot{V}_2 \left(T_{2,in} - T_{2,out}\right) + \frac{k(t)A}{\rho c V_2}\Delta T_m. \tag{3.21}$$

3.1.3 Tank model

Several heating systems include hot water storage tanks. These tanks are used to ease the operation. Switching of the boiler is avoided since the tanks can be charged before switching off a boiler and they can be discharged before the boiler needs to be switched on again. Unfortunately, tanks are generally not controlled efficiently but are charged and discharged according to the plant operation.

The model of the hot water storage tank was developed in [23] which in turn is based on the tank model developed in [24]. It is parametrizable in the number of identical tanks connected, number of layers per tank, in which the temperature is assumed constant, i.e. the water in the layer is assumed to be perfectly stirred, the position of in- and outflow and the position of temperature sensors.

The model includes the pipes as shown in figure 3.4, such that hot water from the heat generation unit flows into the series of tanks with temperature $T_{1,in}$ and flow rate $\dot{V}_1$ and flows back to the heat generation unit with temperature $T_{1,out}$ and flow rate $\dot{V}_1$. On the consumer side water with temperature $T_{2,out}$ and flow rate $\dot{V}_2$ comes out of the tank series and goes back in with the same flow rate but temperature $T_{2,in}$. In this way, the flow rates to other components have a fixed direction and only within the model the flow rate - and therefore the causal direction - changes.

This described the outer setting of the tank series. In the following one single tank is under investigation. Hot water is injected at the top of a tank, cold water is taken from the bottom. If the temperature difference between top and bottom is high, the tank can work efficiently. Due to the difference in the density, hot water will stay at the top and cold water at the bottom. Each tank consists of n layers, each having its own

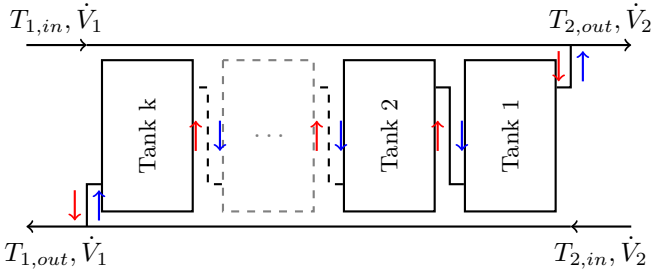


Figure 3.4: Series of tanks (red arrows: charging mode, blue arrows: discharging mode)

temperature T_i. The temperature in a layer is assumed constant, such that a trade off between accuracy and number of layers (which increases the number of states and therefore the complexity) occurs. The tank is assumed to be cylindrical with diameter D_t, and having a height of H_t. The sensor temperatures are denoted by $T_{s,i}$.

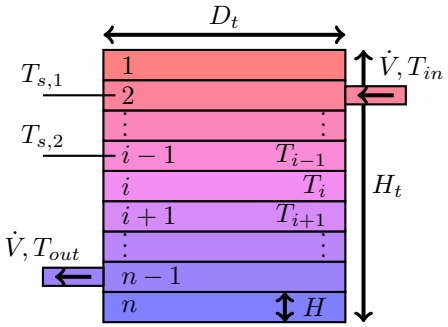


Figure 3.5: Single tank (in charging mode)

To create a nonlinear state space model of the tank series, heat flow balances are established for the layers, which consist of three parts. The major influence is given by the flow through the layers. Assuming the

flow rate through the tank is $\dot{V}_t$ and the tank is charged this gives

$$\dot{Q}_f = \rho c \dot{V}_t \left(T_{i-1} - T_i\right). \tag{3.22}$$

Since two layers have a contact surface $A_t = \pi \left(\frac{D_t}{2}\right)^2$ there is a heat transfer between these layers (from layer i to layer $i+1$), giving

$$\dot{Q}_t = \lambda_l A_t \frac{T_i - T_{i+1}}{H}. \tag{3.23}$$

And finally there are heat losses to the ambiance which can be given as

$$\dot{Q}_l = U A_d \left(T_i - T_a\right), \tag{3.24}$$

where U is the heat transfer coefficient, $A_d = \pi D_t H$ is the contact area to the ambiance and T_a is the ambient temperature.

Combining all three heat transfer mechanisms, a differential equation for the temperature of the layers can be given as

$$\begin{aligned} \rho c V_i \dot{T}_i = &\rho c \dot{V}_t \left(T_{i-1} - T_i\right) - U A_d \left(T_i - T_a\right) + \\ &+ \frac{\lambda_l A_t}{H} \left(T_{i-1} - T_i\right) - \frac{\lambda_l A_t}{H} \left(T_i - T_{i+1}\right), \end{aligned} \tag{3.25}$$

where V_i is the volume of layer i. This balance is depicted in figure 3.6. Equation (3.25) holds true for layers that do not have an input or out-

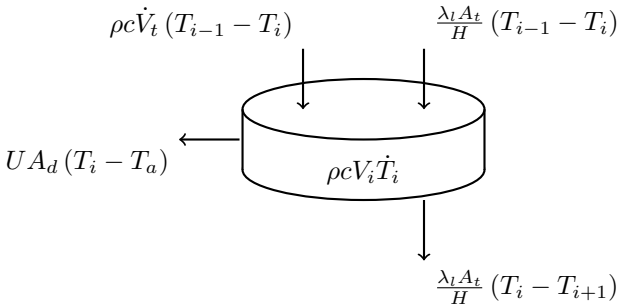


Figure 3.6: Heat power balance of layer i (in charging mode)

put and are located between the input and output. For the layers having an input or an output the heat power balance changes accordingly. For layers above the upper pipe connection and below the lower pipe connection a mixing algorithm is implemented. This algorithm models the effect that the water is mixed, if the water in the lower layers has a higher temperature than the water in the upper layers. The mixing algorithm is simple. The mean value of hot lower layer and the upper connecting layers is taken. The number of layer which are mixed is dependent on the temperature. The algorithm will use as many layers such that the mean temperature is lower or at least the same as the next connecting layer. Figure 3.7 illustrates the mixing algorithm step by step. Here three layers are needed such that the mixing is complete.

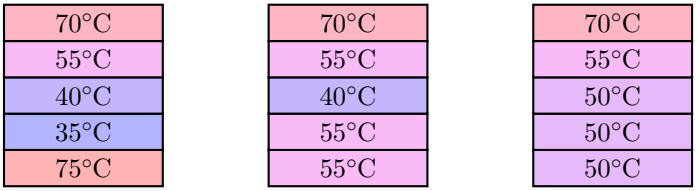


Figure 3.7: Mixing algorithm

3.1.4 Consumer model

Radiators are widely used to supply buildings with heat. Nowadays thermo-active building systems are more and more used. They are build by installing the pipes within the concrete structure of the building. This leads to quite large delay-times, which often pose problems with the installed controllers. The consumer model, will be validated with data from a building with radiators. The extension of the model to represent thermo-active building systems would lead to time delays which will not be investigated in this work.

The consumer model described in this section is not usable for diagnostics within the building but sufficient to evaluate controller designs.

The model was developed in [18] and is based on a one zone model with simple wall models, which corresponds to an RC-circuit in an electric analog. The extension to multi-zone building models, see [4] is straight forward, but will not be discussed since the resulting model would lead to an undesired complexity.

The consumer model consists of two parts. The first one are the radiators. All radiators will be combined to one. Let the flow rate through this fictive radiator be $\dot{V}$, the temperature entering the consumer T_s and the temperature leaving the consumer T_r. Denoting the heat demand by $\dot{Q}_d$ a heat power balance can be given as

$$\rho c V_c \dot{T}_r = \rho c \dot{V} T_s - \rho c \dot{V} T_r - \dot{Q}_d, \tag{3.26}$$

where V_c is the volume of the consumer. This heat power balance is depicted in figure 3.8.

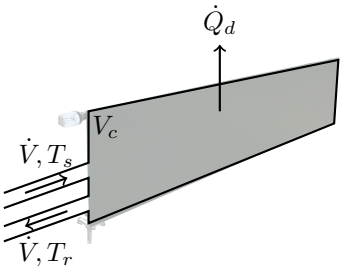


Figure 3.8: Consumer model

In the second part, a building model is used to determine the heat demand. Therefore the entire building is modeled as one room with temperature T_b. The radiator is assumed to have the temperature T_r. This corresponds to the previously used assumption that the water in the radiator is stirred. The building has the temperature T_b and outside the building there is a temperature T_a. The heat power balance can be given as

$$c_b \dot{T}_b = k_{r,b} \left(T_r - T_b \right) - k_{b,a} \left(T_b - T_a \right), \tag{3.27}$$

where c_b is the thermal capacity of the building, $k_{r,b}$ is the heat transfer coefficient from the radiator to the building and $k_{b,a}$ is the heat transfer coefficient from the building to the ambiance. The heat power balance is depicted in figure 3.9.

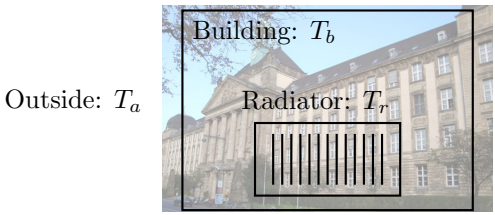


Figure 3.9: One zone building model

The heat demand $\dot{Q}_d$ is equal to the heat power taken from the radiator, i.e.

$$\dot{Q}_d = k_{r,b} \left(T_r - T_b \right). \tag{3.28}$$

The consumer model has two states T_r and T_b, three inputs $\dot{V}$, T_s and T_a and one output T_r. As shown in [18] good simulation results were achieved using the building model, which justifies the use of just one zone and simple wall models for the building under consideration. The simulation results are given in section 3.1.7.

The state equations of the nonlinear state space model read

$$\dot{T}_r = \frac{1}{V_c} \dot{V} T_s - \frac{1}{V_c} \dot{V} T_r - \frac{k_{r,b}}{\rho c V_c} \left(T_r - T_b \right), \tag{3.29}$$

$$\dot{T}_b = \frac{k_{r,b}}{c_b} \left(T_r - T_b \right) - \frac{k_{b,a}}{c_b} \left(T_b - T_a \right). \tag{3.30}$$

3.1.5 Flow rate collector

Consider nodes, where multiple pipes come together, see figure 3.10. The resulting flow rate will obviously be the sum of all flow rates coming

together,

$$\dot{V} = \sum_{i=1}^{n} \dot{V}_i. \tag{3.31}$$

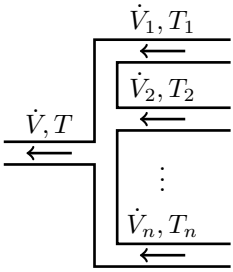


Figure 3.10: Node with several pipe connections

Using the notation of figure 3.10, the resulting temperature can be given by summing up the heat flows. Since no volume is to be considered, the sum of the heat flows is zero. Arriving heat flows will be counted positive, leaving heat flows negative. The heat flow balance is

$$0 = \rho c \sum_{i=1}^{n} \left(\dot{V}_i T_i \right) - \rho c \dot{V} T \tag{3.32}$$

and therefore the outflow temperature becomes

$$T = \frac{\sum_{i=1}^{n} \left(\dot{V}_i T_i \right)}{\dot{V}}. \tag{3.33}$$

3.1.6 Hydraulics

In the operation of heating systems lots of problems are related to hydraulics. Parts of the building may not be supplied with heat sufficiently only because of bad hydronic balances in the heating system. This topic is quite omnipresent in existing buildings as well as in the design of new

buildings. Nevertheless for the optimization of the control of heating systems it is necessary to deal with buildings that are already hydraulically balanced. For the case of a building with adequate hydronic balance, a simple model for the hydraulic system could be found, see [18]. The rough representation of the hydraulics uses the fact, that in the measurement data an upper and a lower limit of the flow rate could be identified. Using the building model and thus the fictitious room temperature a linear dependence between the building temperature and the flow rate could be found. The structure of this connection is depicted in figure 3.11.

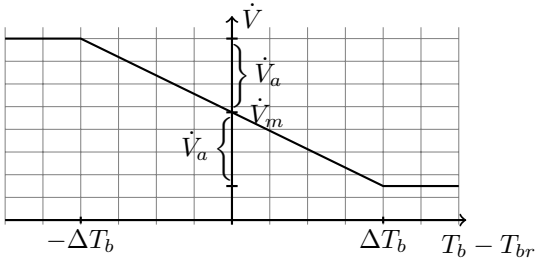


Figure 3.11: Flow rate over control error of the room temperature.

The hydraulic dependency is modeled in a component named pump. The pump has the fictitious room temperature as input and the flow rate

$$\dot{V}(t) = \begin{cases} -\frac{\dot{V}_a}{\Delta T_b} T_b(t) + \frac{\dot{V}_a}{\Delta T_b} T_{br} + \dot{V}_m & & |T_b(t) - T_{br}| < \Delta T_b, \\ \dot{V}_m + \dot{V}_a & \text{for} & T_b(t) \leq T_{br} - \Delta T_b, \\ \dot{V}_m - \dot{V}_a & & T_b(t) \geq T_{br} + \Delta T_b, \end{cases} \tag{3.34}$$

as output. The buildings reference temperature is denoted as T_{br}, the linear temperature amplitude is given as ΔT_b, the mean flow rate $\dot{V}_m$ is and the flow rate amplitude is $\dot{V}_a$, see figure 3.11 for illustration.

3.1.7 Heating system model

Using the previously established models of heating system components, a discrete-time hybrid-valued model of a heating system is constructed, see [18]. The discrete-valued signals are introduced to model the modes of the flow rate. Since the slow processes in the heating system allow the use of a simple method for the temporal discretization, the Euler forward method is chosen.

The building model parameters and the hydraulic model parameters are identified using measurement data from the building of the District Council in Düsseldorf. For heat generation a boiler is used. A scheme of the heating system is given in figure 3.12.

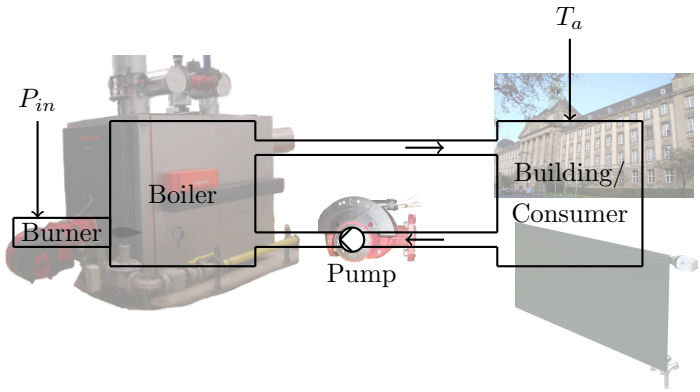


Figure 3.12: Scheme of a heating system

The inputs to the model are the ambient temperature T_a and the thermal power P_{in} to be generated by the boiler. The states of the model are the supply temperature T_s, the return temperature T_r, the building temperature T_b and two discrete states X_l and X_u, used to model the hydraulics.

The difference equations for the used components will be stated in the following, starting with the pump. The discrete-valued states are defined

as

$$X_l = \begin{cases} 1 & \text{if} \quad T_b > T_{br} - \Delta T_b, \\ 0 & \text{else}, \end{cases} \tag{3.35}$$

$$X_u = \begin{cases} 1 & \text{if} \quad T_b > T_{br} + \Delta T_b, \\ 0 & \text{else}. \end{cases} \tag{3.36}$$

The flow rate can be given in dependence of the states (3.35), (3.36) and the building temperature T_b as

$$\begin{aligned} \dot{V} = \dot{V}_m + \dot{V}_a + \left(\frac{T_{br}\dot{V}_a}{\Delta T_b} - \dot{V}_a \right) X_l - \left(\frac{T_{br}\dot{V}_a}{\Delta T_b} + \dot{V}_a \right) X_u - \\ - \frac{\dot{V}_a}{\Delta T_b} T_b X_l + \frac{\dot{V}_a}{\Delta T_b} T_b X_u \, . \end{aligned} \tag{3.37}$$

The next state equation for the boiler, using equation (3.37) reads

$$\begin{aligned} \Phi\left(T_s\right) = T_s \left(1 - \frac{t_s(\dot{V}_a + \dot{V}_m)}{V_b} \right) + T_r \frac{t_s(\dot{V}_a + \dot{V}_m)}{V_b} + \\ + T_s X_l \frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_b} - T_r X_l \frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_b} + \\ + T_b T_s X_l \frac{\dot{V}_a t_s}{\Delta T_b V_b} - T_b T_r X_l \frac{\dot{V}_a t_s}{\Delta T_b V_b} + \\ + T_s X_u \frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_b} - T_r X_u \frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_b} - \\ - T_b T_s X_u \frac{\dot{V}_a t_s}{\Delta T_b V_b} + T_b T_r X_u \frac{\dot{V}_a t_s}{\Delta T_b V_b} + P_{in} \frac{t_s}{V_b c \rho}. \end{aligned} \tag{3.38}$$

The next state equation for the consumer is

$$\begin{aligned}\Phi\left(T_r\right) = T_s\frac{t_s(\dot{V}_a+\dot{V}_m)}{V_c} + T_r\left(1-\frac{t_s(k_{r,b}+c\dot{V}_a\rho+c\dot{V}_m\rho)}{V_c c\rho}\right)+\\ + T_b\frac{k_{r,b}t_s}{V_c c\rho} - T_s X_l\frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_c} + T_r X_l\frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_c} -\\ - T_b T_s X_l\frac{\dot{V}_a t_s}{\Delta T_b V_c} + T_b T_r X_l\frac{\dot{V}_a t_s}{\Delta T_b V_c} - T_s X_u\frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_c} +\\ + T_r X_u\frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_c} + T_b T_s X_u\frac{\dot{V}_a t_s}{\Delta T_b V_c} - T_b T_r X_u\frac{\dot{V}_a t_s}{\Delta T_b V_c}.\end{aligned} \tag{3.39}$$

And finally the next state equation for the building temperature reads

$$\Phi\left(T_b\right) = T_b\left(1-\frac{t_s(k_{r,b}+k_{b,a})}{c_b}\right) + T_r\frac{t_s k_{r,b}}{c_b} + T_a\frac{t_s k_{b,a}}{c_b}. \tag{3.40}$$

The model parameters of the heating system components are given in table 3.1. In addition to these parameters, table 3.2 lists the material constants of water.

With this model the behaviour over one day as well as the behaviour over the whole year are captured. The following figures validate the building model and justify the use of the simple hydraulic model. First the flow rate is given for an entire year. One can see that the trend over the year is captured by the model, see figure 3.13. But also the

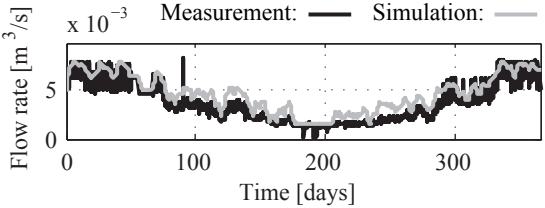


Figure 3.13: Flow rate of one year.

Component	Parameter	Symbol	Value
Boiler	Volume boiler	V_b	$1.05\,\mathrm{m}^3$
	Nominal power of the boiler	P_n	$1.1\,\mathrm{MW}$
Consumer	Volume consumer	V_c	$5\,\mathrm{m}^3$
	Thermal capacity of the building	c_b	$1 \times 10^{10}\ \frac{\mathrm{W\,s}}{\mathrm{K}}$
	Total heat transfer coefficient (radiator building)	$k_{r,b}$	$2.5 \times 10^4\ \frac{\mathrm{W}}{\mathrm{K}}$
	Total heat transfer coefficient (building ambiance)	$k_{b,a}$	$4.7338 \times 10^4\ \frac{\mathrm{W}}{\mathrm{K}}$
Pump	Mean flow rate	$\dot{V}_m$	$4.75 \times 10^{-3}\ \frac{\mathrm{m}^3}{\mathrm{s}}$
	Flow rate amplitude	$\dot{V}_a$	$3.25 \times 10^{-3}\ \frac{\mathrm{m}^3}{\mathrm{s}}$
	Linear temperature amplitude of the thermostat	ΔT_b	$4\,\mathrm{K}$
	Building reference temperature	T_{br}	$21\,^\circ\mathrm{C}$

Table 3.1: Parameters of the heating model.

Density of water	ρ	$1000\ \frac{\mathrm{kg}}{\mathrm{m}^3}$
Specific heat capacity of water	c	$4182\ \frac{\mathrm{W\,s}}{\mathrm{kg\,K}}$

Table 3.2: Material constants of water.

behaviour of one day is captured, see figure 3.14. Supply and return temperatures from measurements and from the simulation are given in figure 3.15. The slight mismatch in the return temperature probably results from a bypass, which was not taken into account in the model and was later removed from the system. And finally the supply and return temperatures for two days are given in figure 3.16. In addition

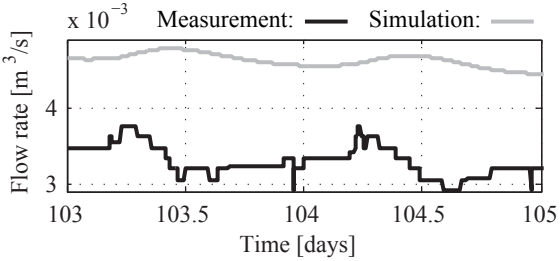


Figure 3.14: Flow rate of two days.

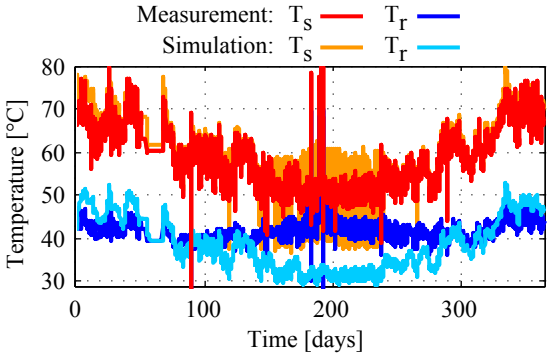


Figure 3.15: Supply and return temperatures for one year.

to these qualitative validations of the model table 3.3 gives the root mean square error as well as the standard deviation for the differences in supply, return and building temperature of measurement data and simulation. The accuracy of the model was assumed to be sufficient to evaluate the controller design methods.

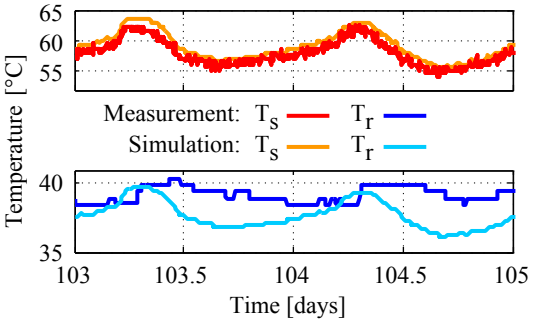


Figure 3.16: Supply and return temperatures for two days.

Temperature	Root mean square error	Standard deviation
Supply temperature	4.13 °C	3.93 °C
Return temperature	7.58 °C	2.79 °C
Flow rate	9.54×10^{-4} $\frac{m^3}{h}$	6.54×10^{-4} $\frac{m^3}{h}$

Table 3.3: Root mean square error and standard deviation of the simulation error

3.2 Piecewise affine models of heating system components

In this section a component of a heating system will be modeled as a piecewise affine system, followed by the description of a model of an entire heating system.

3.2.1 Model of a heating system component

Hybrid systems can be represented using piecewise affine models, as long as the continuous-valued dynamics of the affine models are linear. If the continuous-valued dynamics include the multiplication of two inputs,

input and state, or two states, multiple affine systems have to be given to model this effect.

We shall have a closer look at a boiler. The inputs to the boiler are the return temperature T_r, the heat power to be produced by the boiler P_{in} and the flow rate $\dot{V}$. The state and output is the supply temperature T_s.

In discrete time the state equation can be given as

$$T_s(k+1) = T_s(k) - \frac{\Delta t}{V_b}\dot{V}(k)T_s(k) + \frac{\Delta t}{V_b}\dot{V}(k)T_r(k) + \frac{\Delta t}{\rho c V_b}P_{in}(k). \tag{3.41}$$

Problems arise from the second and the third addend where a multiplication of input and state or input and input has to be calculated. To cope with this problem, the flow rate is discretized using a step size of $\Delta\dot{V}$. For each interval of the flow rate a new affine model is given. The guard function then selects the active dynamics depending on the input $\dot{V}$.

The piecewise affine model (a piecewise linear model in this case) can be given as

$$T_s(k+1) = \left(1 - \frac{\Delta t}{V_b}\dot{V}_i\right) T_s(k) + \begin{bmatrix} \frac{\Delta t}{V_b}\dot{V}_i & \frac{\Delta t}{\rho c V_b} & 0 \end{bmatrix} \begin{bmatrix} T_r(k) \\ P_{in}(k) \\ \dot{V}(k) \end{bmatrix} \tag{3.42}$$

or using $\mathbf{x} = T_s$ and $\mathbf{u} = \begin{bmatrix} T_r & P_{in} & \dot{V} \end{bmatrix}^T$ as

$$\mathbf{x}(k+1) = \underbrace{\left(1 - \frac{\Delta t}{V_b}\dot{V}_i\right)}_{\mathbf{A}_i} \mathbf{x}(k) + \underbrace{\begin{bmatrix} \frac{\Delta t}{V_b}\dot{V}_i & \frac{\Delta t}{\rho c V_b} & 0 \end{bmatrix}}_{\mathbf{B}_i} \mathbf{u}(k), \tag{3.43}$$

where the constant $\dot{V}_i$ is different for each model and equals the mean value of the flow rate $\dot{V}$ for which this model is active. The guard function

$$\underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\mathbf{G_x}} x(k) + \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}}_{\mathbf{G_u}} \mathbf{u}(k) \leq \underbrace{\begin{bmatrix} \dot{V}_i + \frac{\Delta\dot{V}}{2} \\ -\dot{V}_i + \frac{\Delta\dot{V}}{2} \end{bmatrix}}_{\mathbf{G}_{ci}} \tag{3.44}$$

selects the model to be active. Depending on the discretization size and the upper and lower bound on the flow rate, more or less models have to be defined. The fewer models are defined, the bigger the discretization error of the flow rate will be. Here the classical trade-off between accuracy and complexity has to be made.

Using this procedure, other component models given in section 2.1 could be rewritten as switched affine models. This modeling framework can also be used to model heating systems that include switching dynamics. An example is given in the following section.

3.2.2 Model of a heating system

In this section a piecewise affine model of the heating system of the state office building for nature, environment and consumerism in Düsseldorf is described. The model was developed in [25]. Figure 3.17 shows a scheme of the heating system consisting of three boilers in parallel, a pump and a consumer. The consumer represents all radiators and pipes of all three buildings that are connected to the heating system. The power to be produced by the boilers can be set and in case of no produced power a lid in the pipe to the boiler can be closed or opened. If the boiler is producing heat, the lid needs to be opened. The inputs of the model are the power to be produced by the boilers and the decision which boiler has to be switched on, subject to some constraints given in the following.

The main goal of this model is to investigate the operation mode, i.e. which boiler is switched on. Therefore, no sophisticated consumer model is needed. The heat power demand $\dot{Q}_d$ and the flow rate $\dot{V}$ are disturbance inputs to the model, which is justified as long as the supply temperature and its reference are close to each other. In this case, the signals produced by the consumer model will match the measurement data very closely. The reference is calculated in dependence on the ambient temperature T_a, which is also a disturbance input to the system. The dependence of ambient and reference supply temperature is depicted in figure 3.18.

The supply temperature of the heating system is composed of the supply

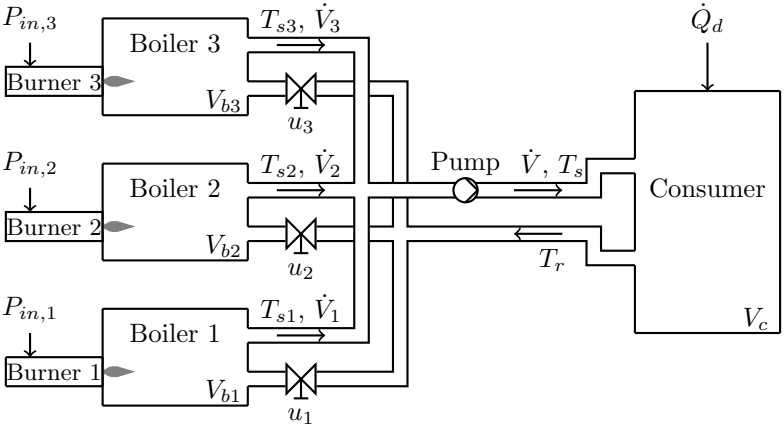


Figure 3.17: Scheme of the heating system.

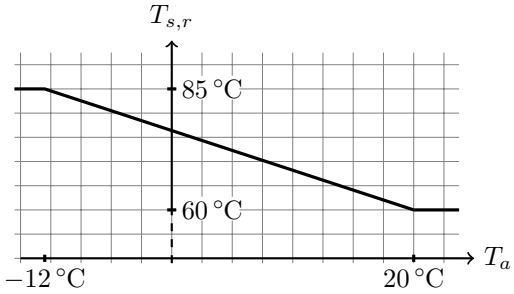


Figure 3.18: Reference supply temperature in dependence on the ambient temperature.

temperatures of the three boilers and the flow rates through the boilers

$$T_s = \frac{\dot{V}_1 T_{s1} + \dot{V}_2 T_{s2} + \dot{V}_3 T_{s3}}{\dot{V}}. \tag{3.45}$$

The decision which lid is opened in dependence on the flow rate is reached

by an underlying controller. Possible settings of the lids are given in table 3.4. As one can see, there are intervals of the flow rate where two different lid settings are feasible. In this case the limits are used as limits of a hysteresis function. Only when the upper limit of the interval is crossed, the *higher* lid setting is used and only if the flow rate is below the lower limit, the *lower* lid setting is used. The *higher* and *lower* lid setting are the lid settings corresponding to the higher and lower flow rates, respectively. The limits on the flow rate are based on hydraulically

L	$\frac{\dot{V}_1}{\dot{V}}$	$\frac{\dot{V}_2}{\dot{V}}$	$\frac{\dot{V}_3}{\dot{V}}$	$\dot{V}_{min}\left[\frac{1}{\mathrm{L}}\right]$	$\dot{V}_{max}\left[\frac{1}{\mathrm{L}}\right]$
1	1	0	0	0	3.1
2	0	1	0	2.2	6.4
3	0.33	0.67	0	5.3	9.4
4	0.2	0.4	0.4	8.1	∞

Table 3.4: Distribution of the flow rate depending on the opened lids with the corresponding lid setting L.

considerations[3].

To control the lids a control strategy is chosen, which tries to avoid switching, i.e. as long as the flow rate is in the corresponding admissible range the lid configuration is not changed. The control law is given as

$$\mathrm{L}(k) = \begin{cases} 1, & \text{if} \quad \dot{V}(k) \leq 2.2\,, \\ & \text{or} \quad 2.2 < \dot{V}(k) \leq 3.1 \quad \wedge \quad \mathrm{L}(k-1) = 1\,. \\ 2, & \text{if} \quad 2.2 < \dot{V}(k) \leq 3.1 \quad \wedge \quad \mathrm{L}(k-1) \neq 1\,, \\ & \text{or} \quad 3.1 < \dot{V}(k) \leq 5.3\,, \\ & \text{or} \quad 5.3 < \dot{V}(k) \leq 6.4 \quad \wedge \quad \mathrm{L}(k-1) \in \{1,2\}\,. \\ 3, & \text{if} \quad 5.3 < \dot{V}(k) \leq 6.4 \quad \wedge \quad \mathrm{L}(k-1) \notin \{1,2\}\,, \\ & \text{or} \quad 6.4 < \dot{V}(k) \leq 8.1\,, \\ & \text{or} \quad 8.1 < \dot{V}(k) \leq 9.4 \quad \wedge \quad \mathrm{L}(k-1) \neq 4\,. \\ 4, & \text{if} \quad 8.1 < \dot{V}(k) \leq 9.4 \quad \wedge \quad \mathrm{L}(k-1) = 4\,, \\ & \text{or} \quad 9.4 < \dot{V}(k)\,. \end{cases} \tag{3.46}$$

[3]The numerical values have been found by our project partners from Plenum Ingenieurgesellschaft für Planung, Energie, Umwelt mbH

Within the lid settings different boiler configurations are possible, i.e. different combinations of the boilers can be switched on. Table 3.5 lists all boiler configurations with the corresponding coding B. If $b_i = 1$ the boiler i is switched on, if $b_i = 0$ the boiler i is switched off.

B	b_1	b_2	b_3
1	1	0	0
1.5	0	0	0
2	0	1	0
2.5	1	1	0
3	1	1	1

Table 3.5: Boiler configuration and corresponding boilers.

Now, the decision which boiler has to be switched on (which is an input to the heating system) can be given in terms of the different boiler configurations.

This model will be used in an online model predictive control with short time horizons, see section 4.3. Since the flow rate and the heat power demand are changing slowly they can be assumed to be constant for the horizons. This has the advantage, that no discretization of the flow rate is needed and therefore no additional system matrices have to be defined. In each time instant a different switched affine system may be used for the controller design. The lid setting is determined by the current flow rate and therefore all feasible boiler configurations, i.e. which boiler is switched on. For each feasible boiler configuration a system and its guard function is defined.

The state equations in continuous time are

$$\dot{T}_{si}(t) = \frac{1}{V_{bi}}\dot{V}_i T_r(t) - \frac{1}{V_{bi}}\dot{V}_i T_{si}(t) + \frac{1}{\rho c V_{bi}} P(t) \ , \quad i = 1, 2, 3 \tag{3.47}$$

$$\dot{T}_r(t) = c\frac{1}{V_c}\dot{V} T_s(t) - \frac{1}{V_c}\dot{V} T_r(t) - \frac{1}{\rho c V_c}\dot{Q}_d. \tag{3.48}$$

Rewriting the state equations in state space form gives

$$\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \\ \dot{T}_3 \\ \dot{T}_{rl} \end{bmatrix} = \tag{3.49}$$

$$\begin{bmatrix} -\frac{v_1\dot{V}}{V_{K1}} & 0 & 0 & \frac{v_1\dot{V}}{V_{K1}} \\ 0 & -\frac{v_2\dot{V}}{V_{K2}} & 0 & \frac{v_2\dot{V}}{V_{K2}} \\ 0 & 0 & -\frac{v_3\dot{V}}{V_{K3}} & \frac{v_3\dot{V}}{V_{K3}} \\ \frac{v_1\dot{V}}{V_V} & \frac{v_2\dot{V}}{V_V} & \frac{v_3\dot{V}}{V_V} & -\frac{\dot{V}}{V_V} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_{rl} \end{bmatrix} + \begin{bmatrix} 0 & \frac{b_1 P_{n,1}}{c\rho V_{K1}} \\ 0 & \frac{b_2 P_{n,2}}{c\rho V_{K2}} \\ 0 & \frac{b_3 P_{n,3}}{c\rho V_{K3}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} B \\ P \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{\dot{Q}_{ab}}{c\rho V_V} \end{bmatrix}.$$

Introducing $\mathbf{x} = \begin{bmatrix} T_{s1} & T_{s2} & T_{s3} & T_r \end{bmatrix}^T$ and $\mathbf{u} = \begin{bmatrix} B & P \end{bmatrix}^{\mathbf{T}}$, this can be rewritten in a short notation as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{f} \tag{3.50}$$

Using the zero-order-hold method for temporal discretization leads to the following system matrices in discrete time

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) + \mathbf{f}_d. \tag{3.51}$$

Adding the two previous boiler configurations to the state vector, such that during the design of the controller, switching of the boilers can be part of the cost function, gives

$$\mathbf{x}_f(k+1) = \begin{bmatrix} \mathbf{x}(k+1) \\ x_5(k+1) \\ x_6(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_d & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_d \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} \mathbf{f}_d \\ 0 \\ 0 \end{bmatrix}, \tag{3.52}$$

where $\mathbf{0}$ is a vector or a matrix with all entries 0 and appropriate dimensions.

Finally the guard function

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_f(k) + \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \mathbf{u}(k) \leq \begin{bmatrix} B \\ -B \end{bmatrix} \tag{3.53}$$

selects the active dynamics. This model will be used in section 4.3 in model predictive controller.

3.3 Tensor models of heating system components

In the thermal heat power balances of the nonlinear state-space models introduced in section 3.1, the nonlinearities arise mainly from the multiplication of two states or a state and an input. These multiplications can be represented exactly by a tensor model. For this reason, tensor models of the components introduced in section 3.1 will be derived in the following. Note that the factor matrices are denoted with the same symbols, in the following subsections, even though they are not the same for the different components.

3.3.1 Heat generation units

The heat power balance for a boiler with attached burner (3.10) leads to the next state equation for the supply temperature given as

$$T_s(k+1) = T_s(k) - \frac{t_s}{V_b}\dot{V}(k)T_s(k) + \frac{t_s}{V_b}\dot{V}(k)T_r(k) + \frac{t_s}{\rho c V_b}P_{in}(k), \tag{3.54}$$

using the Euler-forward method for temporal discretization.

For this model the inputs are the return temperature T_r, the flow rate $\dot{V}$ and the input heat power P_{in}. The state is the supply temperature T_s. Therefore the monomial tensor in CP decomposed form can be given as

$$\mathsf{M}_b(T_s, T_r, \dot{V}, P_{in}) = \left[\begin{pmatrix}1\\P_{in}\end{pmatrix}, \begin{pmatrix}1\\\dot{V}\end{pmatrix}, \begin{pmatrix}1\\T_r\end{pmatrix}, \begin{pmatrix}1\\T_s\end{pmatrix}\right]. \tag{3.55}$$

In order to express equation (3.54) with the help of a contracted tensor product, the factor matrices are stated. In CP decomposed form this becomes

$$\mathsf{F}_b = [\mathbf{F}_{P_{in}}, \mathbf{F}_{\dot{V}}, \mathbf{F}_{T_r}, \mathbf{F}_{T_s}, \mathbf{F}_{\Phi}] \cdot \lambda_b, \tag{3.56}$$

where

$$\mathbf{F}_{T_s} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}, \tag{3.57}$$

$$\mathbf{F}_{T_r} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{3.58}$$

$$\mathbf{F}_{\dot{V}} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \tag{3.59}$$

$$\mathbf{F}_{P_{in}} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.60}$$

$$\mathbf{F}_{\Phi} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}, \tag{3.61}$$

$$\lambda_b = \begin{pmatrix} 1 & -\frac{t_s}{V_b} & \frac{t_s}{V_b} & \frac{t_s}{\rho c V_b} \end{pmatrix}^T. \tag{3.62}$$

Thus the next state equation is given by

$$\Phi\left(T_s\right) = \left\langle \, \mathsf{F}_b \, \middle| \, \mathsf{M}_b(T_s, T_r, \dot{V}, P_{in}) \, \right\rangle. \tag{3.63}$$

3.3.2 Heat exchanger

From the equations (3.20) and (3.21) the difference equations

$$T_{1,out}(k+1) = T_{1,out}(k) + \frac{t_s}{V_1}\dot{V}_1(k)\left(T_{1,in}(k) - T_{1,out}(k)\right) - \frac{kAt_s}{\rho c V_1}\Delta T_m(k), \tag{3.64}$$

$$T_{2,out}(k+1) = T_{2,out}(k) + \frac{t_s}{V_2}\dot{V}_2(k)\left(T_{2,in}(k) - T_{2,out}(k)\right) + \frac{kAt_s}{\rho c V_2}\Delta T_m(k), \tag{3.65}$$

are derived using the Euler forward method and assuming that $k(t)$ is constant. To establish the tensor model, the term $\Delta T_m(k)$ needs further investigation. Recalling

$$\Delta T_m = \frac{(T_{2,out} - T_{2,in}) - (T_{1,in} - T_{1,out})}{\ln\left(\frac{T_{1,out} - T_{2,in}}{T_{1,in} - T_{2,out}}\right)} \tag{3.66}$$

and that multilinearity is needed for tensor models, it is obvious that a multilinear approximation for this term needs to be found. Substituting

$$x_1 = \frac{T_{1,out} - T_{2,in}}{T_{1,in} - T_{2,out}} \tag{3.67}$$

$$x_2 = T_{1,in} - T_{2,out} \tag{3.68}$$

gives

$$\Delta T_m(k) = \frac{x_1 x_2 - x_2}{\ln(x_1)}. \tag{3.69}$$

The choice of the variables x_1 and x_2 represents the temperature difference at the input of the primary side and the ratio of the temperature differences at the connection points. With the linear assumed temperatures over the length of the heat exchanger the four variables can be reduced to two on a power level. Using the approximation method, which will be introduced in section 3.3.6, a multilinear approximation of equation (3.69) could be found for the intervals $x_1 \in [0.004 \ldots 0.04]$ and $x_2 \in [30\,\mathrm{K} \ldots 45\,\mathrm{K}]$, i.e.

$$\Delta\hat{T}_m = a_1 + a_2 (T_{1,in} - T_{2,out}) + a_3 (T_{1,out} - T_{2,in}), \tag{3.70}$$

with $a_1 \approx -1.6692\,\mathrm{K}$, $a_2 \approx 0.1847$ and $a_3 \approx 3.0351$. This function is in fact linear in the original variables but has a multilinear term in the transformed variables x_1 and x_2. Simulation results of the heat exchanger model with the nonlinear original function and the approximation are given in figure 3.19. Inserting equation (3.70) in (3.64) and (3.65) gives

$$\begin{aligned} T_{1,out}(k+1) = {} & T_{1,out}(k) + \frac{t_s}{V_1}\dot{V}_1(k)\,(T_{1,in}(k) - T_{1,out}(k)) - \\ & - \frac{kAt_s}{\rho c V_1}\,(a_1 + a_2\,(T_{1,in}(k) - T_{2,out}(k)) + a_3\,(T_{1,out}(k) - T_{2,in}(k)))\,, \end{aligned} \tag{3.71}$$

$$\begin{aligned} T_{2,out}(k+1) = {} & T_{2,out}(k) + \frac{t_s}{V_2}\dot{V}_2(k)\,(T_{2,in}(k) - T_{2,out}(k)) + \\ & + \frac{kAt_s}{\rho c V_2}\,(a_1 + a_2\,(T_{1,in}(k) - T_{2,out}(k)) + a_3\,(T_{1,out}(k) - T_{2,in}(k)))\,. \end{aligned} \tag{3.72}$$

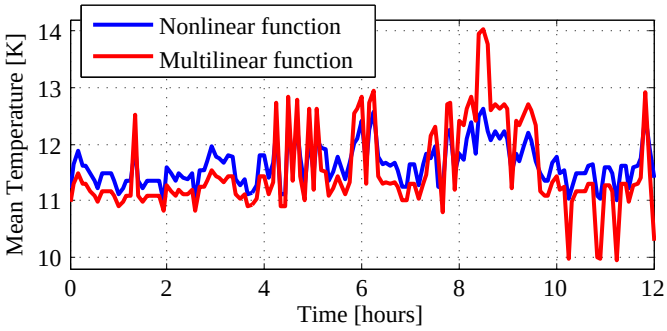


Figure 3.19: Original and approximation of the temperature defining the heat exchange.

The inputs for this model are the intake temperatures on the primary and secondary side of the heat exchanger $T_{1,in}$ and $T_{2,in}$ and the flow rates on the primary and secondary side $\dot{V}_1$ and $\dot{V}_2$. The states are the outflow temperatures of both sides $T_{1,out}$ and $T_{2,out}$. Thus the monomial tensor in CP decomposition can be given as

$$\mathsf{M}_e(T_{1,out}, T_{2,out}, T_{1,in}, T_{2,in}, \dot{V}_1, \dot{V}_2) = \left[\begin{pmatrix}1\\ \dot{V}_2\end{pmatrix}, \begin{pmatrix}1\\ \dot{V}_1\end{pmatrix}, \begin{pmatrix}1\\ T_{2,in}\end{pmatrix}, \begin{pmatrix}1\\ T_{1,in}\end{pmatrix}, \begin{pmatrix}1\\ T_{2,out}\end{pmatrix}, \begin{pmatrix}1\\ T_{1,out}\end{pmatrix}\right]. \tag{3.73}$$

With the factor matrices

$$\mathbf{F}_{T_{1,out}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.74}$$

$$\mathbf{F}_{T_{2,out}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{3.75}$$

$$\mathbf{F}_{T_{1,in}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \tag{3.76}$$

$$\mathbf{F}_{T_{2,in}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad (3.77)$$

$$\mathbf{F}_{\dot{V}_1} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (3.78)$$

$$\mathbf{F}_{\dot{V}_2} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad (3.79)$$

$$\mathbf{F}_{\Phi} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad (3.80)$$

and the parameter vector

$$\begin{aligned} \lambda_e = \Bigg(& -a_1 \frac{kAt_s}{\rho c V_1}, \quad 1 - a_3 \frac{kAt_s}{\rho c V_1}, \quad -\frac{t_s}{\mathrm{V}_1}, \quad a_2 \frac{kAt_s}{\rho c \mathrm{V}_1}, \\ & -a_1 \frac{kAt_s}{\rho c \mathrm{V}_1}, \quad \frac{t_s}{\mathrm{V}_1}, \quad a_3 \frac{kAt_s}{\rho c V_1}, \quad a_1 \frac{kAt_s}{\rho c V_2}, \quad a_3 \frac{kAt_s}{\rho c V_2}, \\ & 1 - a_2 \frac{kAt_s}{\rho c V_2}, \quad -\frac{t_s}{V_2}, \quad a_2 \frac{kAt_s}{\rho c V_2}, \quad -a_3 \frac{kAt_s}{\rho c V_2}, \quad \frac{t_s}{V_2} \Bigg)^T, \end{aligned} \quad (3.81)$$

the next state tensor is given as

$$\mathsf{F}_e = \left[\mathbf{F}_{\dot{V}_2}, \mathbf{F}_{\dot{V}_1}, \mathbf{F}_{T_{2,in}}, \mathbf{F}_{T_{1,in}}, \mathbf{F}_{T_{2,out}}, \mathbf{F}_{T_{1,out}}, \mathbf{F}_{\Phi} \right] \cdot \lambda_e. \quad (3.82)$$

Thus, the next state equation is given by

$$\Phi \left(\begin{pmatrix} T_{1,out} \\ T_{2,out} \end{pmatrix} \right) = \left\langle \mathsf{F}_e \;\middle|\; \mathsf{M}_e(T_{1,out}, T_{2,out}, T_{1,in}, T_{2,in}, \dot{V}_1, \dot{V}_2) \right\rangle. \quad (3.83)$$

3.3.3 Tank model

Recalling equation (3.25) and rearranging gives

$$\begin{aligned} \dot{T}_i = & \frac{1}{V_i} \dot{V}_t T_{i-1} - \frac{1}{V_i} \dot{V}_t T_i + \frac{UA_d}{\rho c V_i} T_a + \frac{\lambda_l A_t}{\rho c V_i H} T_{i-1} - \\ & - \left(2 \frac{\lambda_l A_t}{\rho c V_i H} + \frac{UA_d}{\rho c V_i} \right) T_i + \frac{\lambda_l A_t}{\rho c V_i H} T_{i+1}, \end{aligned} \quad (3.84)$$

with the temperatures T_j for the layers $j = i-1, i, i+t$, the ambient temperature T_a and the flow rate through the tank $\dot{V}_t$. The constants

in this equation are the volume V_i of the layer i, the heat transfer coefficient U, the exterior surface A_d of each layer, the contact area of the layers A_t, the height of the tank H and the heat transfer coefficient λ_l between the layers.

In discrete time this reads

$$\begin{aligned}\Phi\left(T_i\right) =& \frac{t_s}{V_i}\dot{V}_t T_{i-1} - \frac{t_s}{V_i}\dot{V}_t T_i + \frac{t_s U A_d}{\rho c V_i} T_a + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i-1} + \\ &+ \left(1 - 2\frac{t_s \lambda_l A_t}{\rho c V_i H} - \frac{t_s U A_d}{\rho c V_i}\right) T_i + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i+1}. \end{aligned} \tag{3.85}$$

These are the next state equations for layers between the layers with a pipe connection. In layers above or below the pipe connection the next state equation becomes

$$\begin{aligned}\Phi\left(T_i\right) =& \frac{t_s U A_d}{\rho c V_i} T_a + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i-1} + \\ &+ \left(1 - 2\frac{t_s \lambda_l A_t}{\rho c V_i H} - \frac{t_s U A_d}{\rho c V_i}\right) T_i + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i+1}. \end{aligned} \tag{3.86}$$

For the top layer the second term and for the bottom layer the last term vanishes from equation (3.86). Using the notation for the temperatures given in figure 3.5, for the charging and the discharging mode, the following two equations can be stated. The next state equation for the temperature in the upper layer with pipe connection in charging mode is

$$\begin{aligned}\Phi\left(T_i\right) =& \frac{t_s}{V_i}\dot{V}_t T_{in} - \frac{t_s}{V_i}\dot{V}_t T_i + \frac{t_s U A_d}{\rho c V_i} T_a + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i-1} + \\ &+ \left(1 - 2\frac{t_s \lambda_l A_t}{\rho c V_i H} - \frac{t_s U A_d}{\rho c V_i}\right) T_i + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i+1}. \end{aligned} \tag{3.87}$$

The next state equation for the temperature in the lower layer with pipe connection in discharging mode is

$$\begin{aligned}\Phi\left(T_i\right) =& \frac{t_s}{V_i}\dot{V}_t T_{i+1} - \frac{t_s}{V_i}\dot{V}_t T_{out} + \frac{t_s U A_d}{\rho c V_i} T_a + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i-1} + \\ &+ \left(1 - 2\frac{t_s \lambda_l A_t}{\rho c V_i H} - \frac{t_s U A_d}{\rho c V_i}\right) T_i + \frac{t_s \lambda_l A_t}{\rho c V_i H} T_{i+1}. \end{aligned} \tag{3.88}$$

To switch between these modes a discrete state c is introduced. This state is 1 in charging mode and 0 otherwise

$$\Phi(c) = \begin{cases} 1 & \text{if } \dot{V}_t > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.89}$$

To model the mixing due to temperature inversion discrete states are introduced

$$d_i = \begin{cases} 1 & \text{if } T_{i+1} > T_i. \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i = 1, \ldots, n-1. \tag{3.90}$$

The next state equations with mixing then become

$$\begin{aligned} \Phi\left(\hat{T}_i\right) =& \Phi(T_i)\left(1 + \frac{1}{2}d_{i-1} - \frac{1}{2}d_i + \frac{1}{3}d_{i-1}d_i\right) + \\ &+ \Phi(T_{i-1})\left(\frac{1}{2}d_{i-1} - \frac{1}{6}d_{i-1}d_i\right) + \Phi(T_{i+1})\left(\frac{1}{2}d_i - \frac{1}{6}d_{i-1}d_i\right), \end{aligned} \tag{3.91}$$

where $\hat{T}_i$ is introduced as symbol for the temperature with mixing.

Finally the three-way valves at the top and at the bottom need to be modeled. The function of these valves changes in dependence on the mode. If the tank is charged the upper acts as a distribution valve (two outflows, one inflow) and the lower valve acts as a collector (two inflows, one outflow). In discharging mode this is inverted.

The temperatures $T_{2,out}$ and $T_{1,out}$ can be given as

$$T_{2,out} = T_{1,in}v_1 + T_t(v_1 - 1)(1 - c) + T_{1,in}c \tag{3.92}$$

$$T_{1,out} = T_{2,in}v_2 + T_t(1 - v_2)c + T_{2,in}(1 - c), \tag{3.93}$$

with the ratios $v_1 = \frac{\dot{V}_1}{\dot{V}_2}$ and $v_2 = \frac{\dot{V}2}{\dot{V}1}$ and T_t being the temperature that comes out of the tank either at the top (in discharging mode) or at the bottom (in charging mode). These temperatures can be given as an output function of the tensor system, which so far was not used. See [15] for details. The tank model is not used in an approach using tensor models, therefore no factor matrices are given.

3.3.4 Consumer model

The input of the consumer model is the supply temperature. The outputs are the return temperature and the flow rate, which depends on the building temperature and the discrete states X_u and X_l indicating the building temperature above an upper or below a lower limit as given in equations (3.35) and (3.36). This relation is depicted in figure 3.11. The continuous-valued states are the return temperature and the building temperature the discrete-valued states are X_u and X_l. For better readability the discrete-time next state equations are restated here. The next state of the return temperature (3.39) is

$$\begin{aligned}\Phi\left(T_r\right) = T_s\frac{t_s(\dot{V}_a+\dot{V}_m)}{V_c} + T_r\left(1-\frac{t_s(k_{r,b}+c\dot{V}_a\rho+c\dot{V}_m\rho)}{V_c c\rho}\right)+\\ + T_b\frac{k_{r,b}t_s}{V_c c\rho} - T_s X_l\frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_c} + T_r X_l\frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_c} -\\ - T_b T_s X_l\frac{\dot{V}_a t_s}{\Delta T_b V_c} + T_b T_r X_l\frac{\dot{V}_a t_s}{\Delta T_b V_c} - T_s X_u\frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_c} +\\ + T_r X_u\frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_c} + T_b T_s X_u\frac{\dot{V}_a t_s}{\Delta T_b V_c} - T_b T_r X_u\frac{\dot{V}_a t_s}{\Delta T_b V_c}\end{aligned} \tag{3.94}$$

and the next state of the building temperature (3.40) is

$$\Phi\left(T_b\right) = T_b\left(1-\frac{t_s(k_{r,b}+k_{b,a})}{c_b}\right) + T_r\frac{t_s k_{r,b}}{c_b} + T_a\frac{t_s k_{b,a}}{c_b}. \tag{3.95}$$

The equations contain multilinear terms, but no other nonlinearities. The conversion to a tensor model is therefore straight forward.

The monomial tensor is

$$\mathsf{M}_c(T_r, T_b, X_u, X_l, T_s, T_a) = \left[\begin{pmatrix}1\\T_a\end{pmatrix}, \begin{pmatrix}1\\T_s\end{pmatrix}, \begin{pmatrix}1\\X_l\end{pmatrix}, \begin{pmatrix}1\\X_u\end{pmatrix}, \begin{pmatrix}1\\T_b\end{pmatrix}, \begin{pmatrix}1\\T_r\end{pmatrix}\right]. \tag{3.96}$$

With the factor matrices and the parameter vector

$$\mathbf{F}_{T_r} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.97}$$

$$\mathbf{F}_{T_b} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \tag{3.98}$$

$$\mathbf{F}_{X_u} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.99}$$

$$\mathbf{F}_{X_l} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.100}$$

$$\mathbf{F}_{T_s} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.101}$$

$$\mathbf{F}_{T_a} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.102}$$

$$\mathbf{F}_{\Phi} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{3.103}$$

$$\begin{aligned} \lambda_c = \Bigg(& \frac{t_s(\dot{V}_a + \dot{V}_m)}{V_c}, \quad 1 - \frac{t_s(k_{r,b} + c\dot{V}_a\rho + c\dot{V}_m\rho)}{V_c c\rho}, \quad \frac{k_{r,b}t_s}{V_c c\rho}, \\ & -\frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_c}, \quad \frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_c}, \quad -\frac{\dot{V}_a t_s}{\Delta T_b V_c}, \quad \frac{\dot{V}_a t_s}{\Delta T_b V_c}, \\ & -\frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_c}, \quad \frac{\dot{V}_a t_s(\Delta T_b + T_{br})}{\Delta T_b V_c}, \quad \frac{\dot{V}_a t_s}{\Delta T_b V_c}, \quad -\frac{\dot{V}_a t_s}{\Delta T_b V_c}, \\ & 1 - \frac{t_s(k_{r,b} + k_{b,a})}{c_b}, \quad \frac{t_s k_{r,b}}{c_b}, \quad \frac{t_s k_{b,a}}{c_b}, \quad 1, \quad -T_{br} + \Delta T_b + 0.5, \\ & 1, \quad -T_{br} - \Delta T_b + 0.5 \Bigg)^T, \end{aligned} \tag{3.104}$$

the next state tensor in CP decomposed form

$$\mathsf{F}_c = [\mathbf{F}_{T_a}, \mathbf{F}_{T_s}, \mathbf{F}_{X_l}, \mathbf{F}_{X_u}, \mathbf{F}_{T_b}, \mathbf{F}_{T_r}, \mathbf{F}_{\Phi}] \cdot \lambda_c \tag{3.105}$$

can be created. Using this tensor, the next state equations for the return and the building temperature as well as for the discrete states X_u and X_l can be given by the hybrid contracted tensor product

$$\Phi \left(\begin{pmatrix} T_r \\ T_b \\ X_u \\ X_l \end{pmatrix} \right) = \langle \, \mathsf{F}_c \mid \mathsf{M}_c(T_r, T_b, X_u, X_l, T_s) \, \rangle^{\boxplus} . \tag{3.106}$$

3.3.5 Heating system model

The heating system described in section 3.1.7 will be translated into a tensor model. The heating system is extended to include a controller and a signal generator for the ambient temperature. Both components are held simple. The controller will switch on the boiler if a lower threshold of the supply temperature is underrun and it will switch off the boiler if an upper limit of the supply temperature is exceeded. The signal generator of the ambient temperature is an oscillator with two frequencies. One for the temperature oscillation over the days and one for the oscillations over a year. This model was published in [18, 26] and is depicted in figure 3.20.

The continuous-valued states of the system are the supply temperature T_s, which is the state of the boiler, the return temperature T_r and the building temperature T_b, which are the states of the consumer model. The signal generator models the day oscillation and the year oscillation of the ambient temperature with four continuous-valued states $T_{a1}, \ldots, T_{a4}$, two for each oscillation. Two Boolean states X_l and X_u of the consumer determine whether or not the building temperature is in the interval where the thermostatic valve acts linear or are in the lower or upper saturation. The controller has three further Boolean states. The states H_l and H_u determine, if the supply temperature is above an upper limit or below a lower limit and the Boolean state B states, if the boiler is running or not. The state vector is given by

$$\mathbf{x} = [T_s \;\; T_r \;\; T_b \;\; T_{a1} \;\; T_{a2} \;\; T_{a3} \;\; T_{a4} \;\; X_l \;\; X_u \;\; H_l \;\; H_u \;\; B]^T \in \mathbb{R}^7 \times \mathbb{B}^5 . \tag{3.107}$$

The flow rate is not a state in this model since the dependence of flow rate on building temperature and the discrete states X_l and X_u (3.37) is assumed.

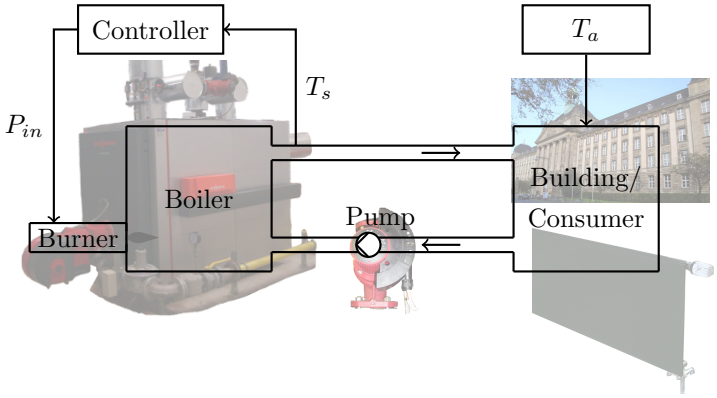


Figure 3.20: Scheme of a heating system with controller and signal generator.

The signal generator is defined as

$$\begin{pmatrix} \dot{T_{a1}} \\ \dot{T_{a2}} \\ \dot{T_{a3}} \\ \dot{T_{a4}} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\omega_d^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_y^2 & 0 \end{pmatrix} \begin{pmatrix} T_{a1} \\ T_{a2} \\ T_{a3} \\ T_{a4} \end{pmatrix}, \tag{3.108}$$

where $\omega_d = \left(\frac{2\pi}{86400}\right) \frac{\text{rad}}{\text{s}}$ and $\omega_y = \left(\frac{2\pi}{31536000}\right) \frac{\text{rad}}{\text{s}}$. The continuous-time state update function is temporally discretized using the MATLAB command `c2d` with the zero-order hold method. Here the first order Euler forward method gave unsatisfactory results when simulation for an entire year. The ambient temperature can then be calculated as

$$T_a = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} T_{a1} \\ T_{a2} \\ T_{a3} \\ T_{a4} \end{pmatrix} + T_{a,m}. \tag{3.109}$$

Note that for a tensor system no coordinate transformation is needed but the constant term $T_{a,m} = 12.5\,^{\circ}\text{C}$ can be modeled.

To describe the controller the discrete states H_l and H_u need to be defined as

$$\Phi(H_l) = \begin{cases} 1 & \text{if} \quad T_s > 75\,^\circ\text{C} \\ 0 & \text{else} \end{cases} \tag{3.110}$$

$$\Phi(H_u) = \begin{cases} 1 & \text{if} \quad T_s > 95\,^\circ\text{C} \\ 0 & \text{else.} \end{cases} \tag{3.111}$$

With these definitions the next state of B (boiler is running or not) is determined by

$$\Phi(B) = \begin{cases} B & \text{if} \quad H_l(1-H_u) = 1 \\ 1 & \text{if} \quad (1-H_l)(1-H_u) = 1 \\ 0 & \text{if} \quad H_l H_u = 1 \end{cases} . \tag{3.112}$$

It is straight forward to translate the next state equations for the supply temperature (3.38) and the return temperature (3.39) into factor matrices and parameter vectors. By inserting equation (3.109) in the next state equation of the building temperature (3.40), this equation is only dependent on the states making the conversion to factor matrices and parameter vector possible. The same conversion is done for the next state equations of the signal generator (3.108) as well as for the discrete next state equations (3.35), (3.36), (3.110), (3.111) and (3.112). This leads to an overall tensor model

$$\Phi(\mathbf{x}) = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x}) \,\rangle^{\boxplus} \tag{3.113}$$

with 12 factor matrices of dimension $\mathbb{B}^{2\times 46}$, one factor matrix of dimension $\mathbb{B}^{46\times 12}$ and a parameter vector with 46 real entries, given in the appendix B. The initial state vector is

$$\mathbf{x}(0) = \left(253 \quad 323 \quad 294 \quad 0 \quad -\frac{5\pi}{86400} \quad -12.5 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1\right)^T . \tag{3.114}$$

3.3.6 Multilinearization

In this section a method to generate a tensor model from a nonlinear state space model is presented, [5, 27]. The method is a numerical one, i.e. the next states or the derivatives of the states need to be computed. Here the state space model is assumed to be given as a Simulink model, such that Simulink can be used to do these calculations, but the method could be extended, such that it could use measurement data from a real system to estimate a tensor model. First an approximation method for nonlinear functions is presented. This is then extended to nonlinear systems.

A multilinear function can be given in terms of the contracted tensor product as

$$h(\mathbf{x}) = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x}) \,\rangle = \sum_{i=1}^{2^n} \varphi_{\mathbf{i}_i} \mu_{\mathbf{i}_i}(\mathbf{x}) = \tag{3.115}$$

$$\varphi_{\mathbf{i}_1} \cdot 1 + \varphi_{\mathbf{i}_2} \cdot x_1 + \varphi_{\mathbf{i}_3} \cdot x_2 + \varphi_{\mathbf{i}_4} \cdot x_1 x_2 + \ldots + \varphi_{\mathbf{i}_{2^n}} \cdot x_1 x_2 \ldots x_n,$$

which is a function in the variables $\mathbf{x} \in \mathbb{R}^n$. The tensor $\mathsf{F} \in \mathbb{R}^{\times^{(n)} 2}$ contains the factor matrices and the parameter vector. The scalar elements of a tensor of dimension $\mathbb{R}^{\times^{(n)} 2}$ are indexed by a subscript vector $\mathbf{i} = \begin{pmatrix} i_1 & \cdots & i_n \end{pmatrix}$. Since every element of $\mathbf{i}$ can take 2 values, i.e. 1 or 2, there are 2^n different index vectors possible, denoted by $\mathbf{i}_i,\ i = 1, \ldots, 2^n$. Thus the elements of $\mathsf{M}(\mathbf{x})$ are given by $\mu_{\mathbf{i}_i}(\mathbf{x})$ and the elements of F are given by $\varphi_{\mathbf{i}_i}$.

Approximation problem The goal is to find a multilinear function $h(\mathbf{x})$, which is the best approximation of the nonlinear function $f(\mathbf{x})$. Obviously the term *best* needs further definition. The approximation will be valid in a domain $D \subset \mathbb{R}^n$, which is bounded by the intervals of the variables

$$x_i \in [x_{i,l}, x_{i,u}], i = 1, \ldots, n.$$

The best multilinear approximation $h(\mathbf{x})$ of a nonlinear function $f(\mathbf{x})$ minimizes the distance between approximation and original function. For this measure the norm

$$\|f(\mathbf{x})\|_w := \sqrt{\langle f(\mathbf{x}), f(\mathbf{x}) \rangle_w} \tag{3.116}$$

is introduced, where the inner product of $f_1(\mathbf{x}) \in \mathcal{N}$ and $f_2(\mathbf{x}) \in \mathcal{N}$ is given as

$$\langle f_1(\mathbf{x}), f_2(\mathbf{x}) \rangle_w := \int_D f_1(\mathbf{x}) f_2(\mathbf{x}) w(\mathbf{x})\, \mathrm{d}\mathbf{x},\ w(\mathbf{x}) > 0. \tag{3.117}$$

Here $\mathcal{N}$ denotes the inner product space of the nonlinear functions and $w(\mathbf{x})$ is a weighting function used to increase the importance of a good approximation in a subset of the domain D. All multilinear functions are in the inner product space $\mathcal{N}$ of nonlinear functions and therefore define a subset $\mathcal{M} \subset \mathcal{N}$. The approximation problem can be formulated as a minimization problem

$$\min_{h(\mathbf{x}) \in \mathcal{M}} \|h(\mathbf{x}) - f(\mathbf{x})\|_w \ . \tag{3.118}$$

The uniqueness of the solution to the minimization problem is given, since $\mathcal{N}$ is an inner product space and therefore strict convex, [28]. The polynomial $h(\mathbf{x})$ is the solution to (3.118), if and only if the orthogonality condition

$$\left\langle \hat{h}(\mathbf{x}), h(\mathbf{x}) - f(\mathbf{x}) \right\rangle_w = 0, \ \forall \hat{h}(\mathbf{x}) \in \mathcal{M} \tag{3.119}$$

is fulfilled, [29]. The base $\{\mu_{\mathbf{i}_1}(\mathbf{x}), \ldots, \mu_{\mathbf{i}_{2^n}}(\mathbf{x})\}$ of $\mathcal{M}$ is given by the elements of the monomial tensor $\mathsf{M}(\mathbf{x})$ and is called monomial base in the following. The function $\hat{h}(\mathbf{x})$ can be given as

$$\hat{h}(\mathbf{x}) = \left\langle \hat{\mathsf{F}} \mid \mathsf{M}(\mathbf{x}) \right\rangle = \sum_{k=1}^{2^n} \hat{\varphi}_{\mathbf{i}_k} \mu_{\mathbf{i}_k}(\mathbf{x}), \tag{3.120}$$

with $\hat{\mathsf{F}} \in \mathbb{R}^{\times^{(n)}2}$. Inserting (3.120) in (3.119) gives

$$\left\langle \sum_{k=1}^{2^n} \hat{\varphi}_{\mathbf{i}_k} \mu_{\mathbf{i}_k}(\mathbf{x}) \ , \ h(\mathbf{x}) - f(\mathbf{x}) \right\rangle_w = 0 \tag{3.121}$$

and leads to a system of linear equations

$$\langle \mu_{\mathbf{i}_k}(\mathbf{x}), h(\mathbf{x}) - f(\mathbf{x}) \rangle_w = 0, \quad k = 1, \ldots, 2^n. \tag{3.122}$$

The unknowns of these equations are the coefficients $\varphi_{\mathbf{i}_i}$ of the function $h(\mathbf{x})$. Inserting (3.115) in (3.122) and rearranging gives

$$\sum_{i=1}^{2^n} \varphi_{\mathbf{i}_i} \langle \mu_{\mathbf{i}_k}(\mathbf{x}), \mu_{\mathbf{i}_i}(\mathbf{x}) \rangle_w = \langle \mu_{\mathbf{i}_k}(\mathbf{x}), f(\mathbf{x}) \rangle_w, k = 1, \ldots, 2^n. \tag{3.123}$$

Rewriting (3.123) as matrix equation gives

$$\begin{pmatrix} \langle \mu_{\mathbf{i}_1}(\mathbf{x}), \mu_{\mathbf{i}_1}(\mathbf{x}) \rangle_w & \cdots & \langle \mu_{\mathbf{i}_1}(\mathbf{x}), \mu_{\mathbf{i}_{2^n}}(\mathbf{x}) \rangle_w \\ \vdots & \ddots & \vdots \\ \langle \mu_{\mathbf{i}_{2^n}}(\mathbf{x}), \mu_{\mathbf{i}_1}(\mathbf{x}) \rangle_w & \cdots & \langle \mu_{\mathbf{i}_{2^n}}(\mathbf{x}), \mu_{\mathbf{i}_{2^n}}(\mathbf{x}) \rangle_w \end{pmatrix} \begin{pmatrix} \varphi_{\mathbf{i}_1} \\ \vdots \\ \varphi_{\mathbf{i}_{2^n}} \end{pmatrix} = \begin{pmatrix} \langle f(\mathbf{x}), \mu_{\mathbf{i}_1}(\mathbf{x}) \rangle_w \\ \vdots \\ \langle f(\mathbf{x}), \mu_{\mathbf{i}_{2^n}}(\mathbf{x}) \rangle_w \end{pmatrix}. \tag{3.124}$$

It is difficult to solve (3.124) reliably, since the matrix on the left hand side is ill conditioned. By choosing an appropriate base of $\mathcal{M}$ this problem can be overcome.

Orthonormalization An orthonormal monomial base is constructed, using the orthogonalization algorithm of Gram-Schmidt, [30]. The base is constructed iteratively with

$$\begin{aligned}
\overline{\mu}_{\mathbf{i}_1}(\mathbf{x}) =& \frac{\mu_{\mathbf{i}_1}(\mathbf{x})}{\|\mu_{\mathbf{i}_1}(\mathbf{x})\|_w}, \\
\overline{\mu}_{\mathbf{i}_k}(\mathbf{x}) =& \frac{\mu_{\mathbf{i}_k}(\mathbf{x}) - \sum_{i=1}^{k-1} \left\langle \mu_{\mathbf{i}_k}(\mathbf{x}), \overline{\mu}_{\mathbf{i}_i}(\mathbf{x}) \right\rangle_w \overline{\mu}_{\mathbf{i}_i}(\mathbf{x})}{\left\| \mu_{\mathbf{i}_k}(\mathbf{x}) - \sum_{i=1}^{k-1} \left\langle \mu_{\mathbf{i}_k}(\mathbf{x}), \overline{\mu}_{\mathbf{i}_i}(\mathbf{x}) \right\rangle_w \overline{\mu}_{\mathbf{i}_i}(\mathbf{x}) \right\|_w}, k = 2, \ldots, 2^n.
\end{aligned} \tag{3.125}$$

The orthonormal base $\{\overline{\mu}_{\mathbf{i}_1}(\mathbf{x}), \ldots, \overline{\mu}_{\mathbf{i}_{2^n}}(\mathbf{x})\}$ of $\mathcal{M}$ has the property

$$\left\langle \overline{\mu}_{\mathbf{i}_i}(\mathbf{x}), \overline{\mu}_{\mathbf{i}_j}(\mathbf{x}) \right\rangle_w = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases} \quad \forall i, j = 1, \ldots, 2^n. \tag{3.126}$$

Using this property, it is easy to see that all off-diagonal elements of the left hand side matrix in (3.124) are zero and the diagonal elements are equal to one. Therefore the matrix equation reads

$$\mathbf{I}_{2^n} \begin{pmatrix} \overline{\varphi}_{\mathbf{i}_1} \\ \vdots \\ \overline{\varphi}_{\mathbf{i}_{2^n}} \end{pmatrix} = \begin{pmatrix} \left\langle f(\mathbf{x}), \overline{\mu}_{\mathbf{i}_1}(\mathbf{x}) \right\rangle_w \\ \vdots \\ \left\langle f(\mathbf{x}), \overline{\mu}_{\mathbf{i}_{2^n}}(\mathbf{x}) \right\rangle_w \end{pmatrix}, \tag{3.127}$$

where $\mathbf{I}_{2^n}$ denotes a $2^n \times 2^n$ identity matrix. Solving this system of equations results in the approximation

$$h(\mathbf{x}) = \sum_{i=1}^{2^n} \overline{\varphi}_{\mathbf{i}_i} \overline{\mu}_{\mathbf{i}_i}(\mathbf{x}) = \left\langle \, \overline{\mathsf{F}} \, \middle| \, \overline{\mathsf{M}}(\mathbf{x}) \, \right\rangle, \tag{3.128}$$

where the tensor $\overline{\mathsf{M}}(\mathbf{x}) \in \mathbb{R}^{\times^{(n)}2}$ has the same structure as $\mathsf{M}(\mathbf{x})$ but the orthonormal base elements $\overline{\mu}_{\mathbf{i}_i}(\mathbf{x})$, $i = 1, \ldots, 2^n$ instead of the original ones. The approximation was created with the orthogonal projection of the nonlinear function $f(\mathbf{x})$ on the orthonormal base polynomials, [29]. Since the orthonormal and the monomial base span the same space $\mathcal{M}$, all elements of the orthonormal base can be expressed by a linear combination of the monomial base elements

$$\begin{pmatrix} \overline{\mu}_{\mathbf{i}_1}(\mathbf{x}) \\ \vdots \\ \overline{\mu}_{\mathbf{i}_{2^n}}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \vartheta_{1,1} & \cdots & \vartheta_{1,2^n} \\ \vdots & \ddots & \vdots \\ \vartheta_{2^n,1} & \cdots & \vartheta_{2^n,2^n} \end{pmatrix} \begin{pmatrix} \mu_{\mathbf{i}_1}(\mathbf{x}) \\ \vdots \\ \mu_{\mathbf{i}_{2^n}}(\mathbf{x}) \end{pmatrix}. \tag{3.129}$$

Inserting (3.129) in (3.128) leads to the function

$$h(\mathbf{x}) = \sum_{i=1}^{2^n} \varphi_{\mathbf{i}_i} \mu_{\mathbf{i}_i}(\mathbf{x}) = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x}) \,\rangle\,, \tag{3.130}$$

which is the approximation expressed with respect to the monomial base, which minimizes (3.118).

Extension to nonlinear systems The approximation method for nonlinear functions is used in this section to find a multilinear approximation

$$\dot{\mathbf{x}} = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x}, \mathbf{u}) \,\rangle = \begin{pmatrix} h_1(\mathbf{x}, \mathbf{u}) & \cdots & h_n(\mathbf{x}, \mathbf{u}) \end{pmatrix}^T \tag{3.131}$$

of a given nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} f_1(\mathbf{x}, \mathbf{u}) & \cdots & f_n(\mathbf{x}, \mathbf{u}) \end{pmatrix}^T \tag{3.132}$$

with n states $\mathbf{x}$ and m inputs $\mathbf{u}$. The best approximation of the nonlinear system for the domain D, bounded by the intervals of the operating range of the variables

$$x_i \in [x_{i,l}, x_{i,u}]\,, i = 1, \ldots, n, \tag{3.133}$$

$$u_j \in [u_{j,l}, u_{j,u}]\,, j = 1, \ldots, m, \tag{3.134}$$

is to be found. The interval boundaries $x_{i,l} < x_{i,u}$ and $u_{j,l} < u_{j,u}$ of the operation range are usually given by the engineering application. Because of the $n + m$ variables of the function $\mathbf{f}(\mathbf{x}, \mathbf{u})$, the tensor $\mathsf{M}(\mathbf{x}, \mathbf{u})$ has dimension $\mathsf{M} \in \mathbb{R}^{\times^{(n+m)}2}$ and therefore there exist 2^{n+m} index vectors. The approximation method is applied to each of the functions $f_j(\mathbf{x}, \mathbf{u}),\ j = 1, \ldots, n$ in (3.132) and the approximations $h_j(\mathbf{x}, \mathbf{u})$ are computed elementwise

$$h_j(\mathbf{x},\mathbf{u}) = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x},\mathbf{u}) \,\rangle(j) = \sum_{i=1}^{2^{n+m}} \varphi_{\mathbf{i}_i,j} \mu_{\mathbf{i}_i}(\mathbf{x},\mathbf{u}), \forall j = 1, \ldots, n, \tag{3.135}$$

with the state transition tensor $\mathsf{F} \in \mathbb{R}^{\times^{(n+m)}2\times n}$ with elements $\varphi_{\mathbf{i}_i,j}$. The state transition tensor has a further dimension, since the factor matrix $\mathbf{F}_\Phi$ assigning the factors to the elements of the state vector is taken into account. For each function $f_j(\mathbf{x}, \mathbf{u}), j = 1, \ldots, n$ the approximation is computed by solving the minimization problem

$$\min_{h_j(\mathbf{x},\mathbf{u}) \in \mathcal{M}} \left\| h_j(\mathbf{x}, \mathbf{u}) - f_j(\mathbf{x}, \mathbf{u}) \right\|_w \ , \forall j = 1, \ldots, n\,. \tag{3.136}$$

The tensor system

$$\dot{\mathbf{x}} = \langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x},\mathbf{u}) \,\rangle = \left\langle\, \overline{\mathsf{F}} \;\middle|\; \overline{\mathsf{M}}(\mathbf{x},\mathbf{u}) \,\right\rangle, \tag{3.137}$$

with the orthonormal base $\left\{\overline{\mu}_{\mathbf{i}_i}(\mathbf{x},\mathbf{u})\right\}$ minimizes (3.136), if the scalar elements of $\overline{\mathsf{F}}$ are equal to

$$\overline{\varphi}_{\mathbf{i}_i,j} = \left\langle f_j(\mathbf{x},\mathbf{u}), \overline{\mu}_{\mathbf{i}_i}(\mathbf{x},\mathbf{u}) \right\rangle_w, \forall i = 1,\ldots,2^{n+m}, \forall j = 1,\ldots,n. \tag{3.138}$$

Using (3.129) the elements $\varphi_{\mathbf{i}_i,j}$ of F can be computed, which define the tensor system, which is the best approximation of the nonlinear system (3.132), with respect to the norm (3.116). In general F can be a full tensor. For real systems in many cases lower rank approximations might be found.

Numerical implementation The analytic multilinearization might not be suitable, if e.g. the system for which a tensor model should be created already exists in Simulink. Therefore it will be shown, how the introduced approximation method can be implemented numerically.

The MATLAB Simulink model needs to be evaluated for each of the variables $x_1,\ldots,x_n$ and $u_1,\ldots,u_m$ in a sufficiently large number of points in the intervals, defining the operation range. This gives the dataset

$$(\mathbf{x}_i, \mathbf{u}_j, \mathbf{f}(\mathbf{x}_i,\mathbf{u}_j)), \forall i = 1,\ldots,N^n,\ \forall j = 1,\ldots,N^m, \tag{3.139}$$

where each $\mathbf{x}_i$ and $\mathbf{u}_j$ stands for a vector containing all states and inputs, respectively. Since the nonlinear model is only known in the sampling points, numerical integration has to be used to compute the inner product (3.117), e.g. by using the trapezoidal rule, [31].

Because of the inaccuracies introduced by numerical integration the calculation of the parameters $\varphi_{\mathbf{i}_i,j}$ will contain some errors. These errors gain influence, if arbitrary large intervals are used for the variables $\mathbf{x}$ and $\mathbf{u}$. Another source of errors are large offsets in the intervals (3.133) and (3.134), e.g. if the states or inputs represent temperatures in Kelvin. To avoid such numerical problems it is necessary to scale the variables to fixed intervals resulting in the scaled data set

$$(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_j, \mathbf{f}(\tilde{\mathbf{x}}_i,\tilde{\mathbf{u}}_j)),\ \forall i = 1,\ldots,N^n,\ \forall j = 1,\ldots,N^m, \tag{3.140}$$

using lemma 2.1. Computing the multilinear approximation of the system in the scaled variables $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{u}}$ gives the tensor system $\dot{\tilde{\mathbf{x}}} = \left\langle\, \tilde{\mathsf{F}} \mid \mathsf{M}(\tilde{\mathbf{x}},\tilde{\mathbf{u}}) \,\right\rangle$. Using the inverse transformation gives the system $\langle\, \mathsf{F} \mid \mathsf{M}(\mathbf{x},\mathbf{u}) \,\rangle$ with variables $\mathbf{x}$ and $\mathbf{u}$ in the desired intervals (3.133) and (3.134).

Problems due to the exponentially growing number of parameters to be identified, if the number of states and input increases can be softened be restricting the maximal number of states and inputs to be multiplied with each other, called multilinear order.

Definition 3.1 *The* multilinear order *describes the number of variables that are multiplied in a multilinear monomial.*

Example 3.1 *The term $x_1 \cdot x_2 \cdot \ldots \cdot x_k$ has the multilinear order k.*

A function with maximal multilinear order k has monomials with an order lower or equal to k. Bounding the maximal multilinear order for the approximation leads to less parameters to be identified. The parameters concerning the higher order terms are set to 0.

To find the best approximation of a nonlinear system given as a Simulink model perform the following steps:

Algorithm 3.1 *Multilinearization*

- *Choose a maximal multilinear order,*
- *Specify the operation range D of the system,*
- *Choose the number N of sampling points for $\mathbf{x}$ and $\mathbf{u}$*
- *Evaluate the Simulink model in the sampling points to get the dataset (3.139),*
- *Orthonormalize the monomial base of chosen order*
- *Scale the determined data to the interval $[-1, 1]$,*
- *Calculate the parameters of the best multilinear approximation $\dot{\tilde{\mathbf{x}}} = \left\langle \tilde{\mathsf{F}} \mid \mathsf{M}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \right\rangle$ of the scaled system*
- *Rescale the tensor $\tilde{\mathsf{F}}$ to get the tensor F and thus the best multilinear approximation (3.131) of the nonlinear model in the desired operating range D.*

Discrete-time tensor model If a discrete-time tensor model

$$\mathbf{x}(k+1) = \langle\, \mathsf{F}_d \mid \mathsf{M}(\mathbf{x}(k), \mathbf{u}(k)) \,\rangle \tag{3.141}$$

is to be identified, the above method can be used, [27]. In case of the numerical integration, a time discrete dataset is used for the identification of the

continuous-time tensor model. To avoid high offsets in the states, it is advisable to work with differences between states in the discrete-time case instead of the absolute values of the states

$$\frac{1}{t_s}\left(\mathbf{x}(k+1)-\mathbf{x}(k)\right). \tag{3.142}$$

The resulting dataset contains the same information as in the continuous-time case, therefore a continuous-time tensor model F_c can be identified, which then is discretized using

$$\frac{1}{t_s}\left(\mathbf{x}(k+1)-\mathbf{x}(k)\right) = \langle\, \mathsf{F}_c \mid \mathsf{M}(\mathbf{x},\mathbf{u}) \,\rangle\,. \tag{3.143}$$

The result is a discrete-time tensor model

$$\mathbf{x}(k+1) = \langle\, \mathsf{F}_d \mid \mathsf{M}(\mathbf{x}(k),\mathbf{u}(k)) \,\rangle\,, \tag{3.144}$$

where $\mathsf{F}_d = t_s\mathsf{F}_c + \Theta$ and

$$\Theta(\mathbf{i}) = \begin{cases} 1 & \text{if } i_j = 2 \wedge i_k = 1, \forall k \neq j \\ 0 & \text{otherwise} \end{cases} \quad \forall j = 1,\ldots,n, \tag{3.145}$$

assuming that $\mathbf{i} = i_1,\ldots,i_n$ is the index vector of the tensor Θ.

Application to a model of a heating system The approximation method will be applied to a Simulink model of the heating system described in section 3.1.7. Contrary to the described system, continuous-time equations will be used, the ambient temperature will be generated by an oscillator and the dependency of room temperature to flow rate will be governed by

$$\dot{V}(t) = \dot{V}_m + \kappa_1 \arctan\left(\frac{T_b - T_{br}}{\kappa_2}\right), \tag{3.146}$$

where $\kappa_1 = -2.325\times10^{-3}\,\frac{\mathrm{m}^3}{\mathrm{s}}$ and $\kappa_2 = 2.2\,\mathrm{K}$. This application example is taken from [27]. The dependency of the flow rate on the building temperature is depicted in figure 3.21.

The ambient temperature will be modeled by

$$\begin{pmatrix}\dot{T}_{a1}\\ \dot{T}_{a2}\end{pmatrix} = \begin{pmatrix}0 & 1\\ -\omega_d^2 & 0\end{pmatrix}\begin{pmatrix}T_{a1}\\ T_{a2}\end{pmatrix} \tag{3.147}$$

$$T_a = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}T_{a1}\\ T_{a2}\end{pmatrix}, \tag{3.148}$$

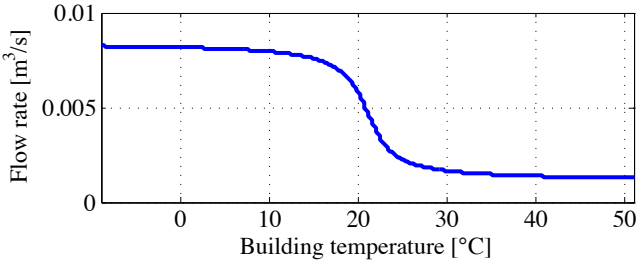


Figure 3.21: Dependency of room temperature and flow rate.

where $\omega_d = 7.27 \times 10^{-5}\,\frac{1}{\mathrm{s}}$. The states of the model are

$$\mathbf{x} = \begin{pmatrix} T_b & T_r & T_s & T_{a1} & T_{a2} \end{pmatrix}^T \tag{3.149}$$

The domain of the input and states for which a good approximation is needed are chosen to be

$$T_b = [288\,\mathrm{K}, 298\,\mathrm{K}]\,, \qquad T_r = [323\,\mathrm{K}, 333\,\mathrm{K}]\,, \qquad T_s = [353\,\mathrm{K}, 368\,\mathrm{K}]\,, \tag{3.150}$$

$$T_{a1} = [-2.5\,\mathrm{K}, 2.5\,\mathrm{K}]\,, \quad T_{a2} = \left[-1.818 \times 10^{-4}\,\frac{\mathrm{K}}{\mathrm{s}}, 1.818 \times 10^{-4}\,\frac{\mathrm{K}}{\mathrm{s}}\right], \tag{3.151}$$

$$P_{in} = [0.5P_n, P_n]\ . \tag{3.152}$$

Using a multilinear order of 2, a weighting function $w(\mathbf{x}) = 1$ and $N = 10$ values per variable a tensor model

$$\dot{\mathbf{x}} = \left\langle\, \hat{\mathsf{F}} \mid \mathsf{M}(\mathbf{x}, u) \,\right\rangle \tag{3.153}$$

could be approximated. Simulation results of the supply temperature, the return temperature and the building temperature of the original nonlinear and the approximated multilinear systems are given in figure 3.22. These simulation results show a good fit of the temperatures of the nonlinear and the multilinear model. The approximation of the dependency of the flow rate on the building temperature can be approximated with a multilinear function. Note that contrary to the tensor model of the heating system described in the last subsection no discrete-valued signals are used.

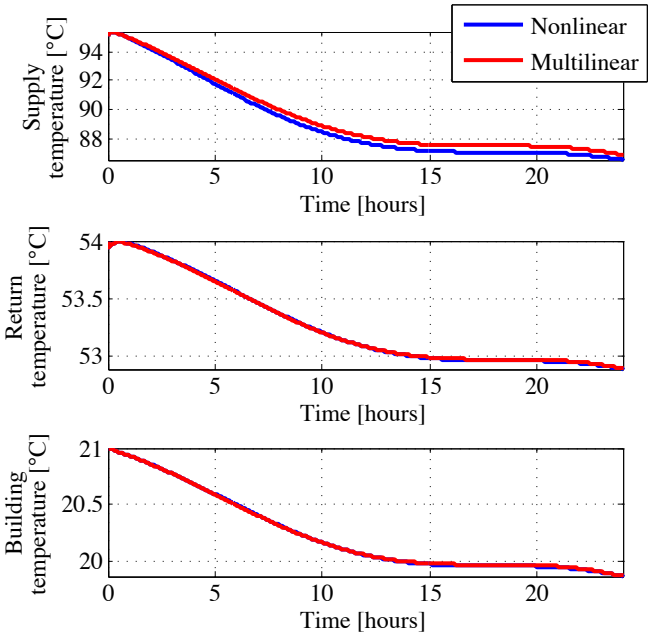


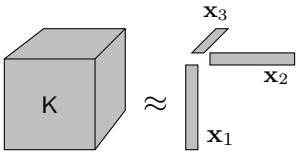
Figure 3.22: Simulation results of the original nonlinear and the approximated multilinear system.

The multilinearization procedure is obviously not restricted to heating systems. The multilinear representation is also interesting for other fields. In [5] the benefits of multilinear modeling were shown for a chemical reaction with very promising results.

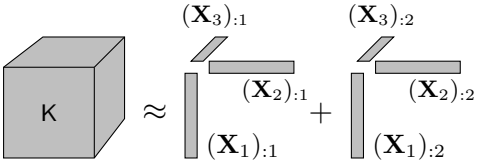
3.3.7 Tensor decomposition

In this section standard tensor decomposition techniques, see [16], are applied to the next state tensor F of the tensor model (3.113), to find an approximation with smaller factor matrices, i.e. a lower tensor rank. The following example illustrates the meaning of a tensor rank.

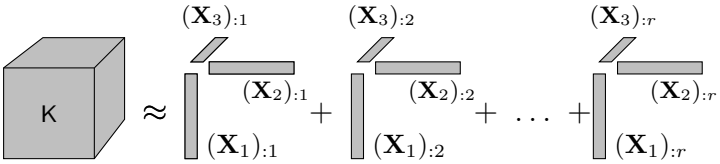
Example 3.2 *A third-order tensor* $\mathsf{K} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ *can be approximated by a rank* 1 *CP tensor* $\mathsf{K} \approx [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3] \cdot \lambda$ *with* $\mathbf{x}_i \in \mathbb{R}^{r_i}$, $i = 1, 2, 3$. *Graphically this can be illustrated as*



where the block stands for the full tensor and the entries are approximated by multiplying the corresponding entries in the vectors $\mathbf{x}_1$, $\mathbf{x}_2$ *and* $\mathbf{x}_3$. *The approximation might be improved by adding another set of vectors as given in* $\mathsf{K} \approx [\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3] \cdot \lambda$ *with* $\mathbf{x}_i \in \mathbb{R}^{r_i \times 2}$, $i = 1, 2, 3$. *Illustrating the rank* 2 *approximation graphically gives*



where $(\mathbf{X}_n)_{:i}$ *is used to denote the* i^{th} *column of the matrix* $\mathbf{X}_n$, $n = 1, 2, 3$. *The rank can be further increased to find a better approximation. For a rank* R *CP tensor this gives* $\mathsf{K} \approx [\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3] \cdot \lambda$ *with* $\mathbf{x}_i \in \mathbb{R}^{r_i \times r}$, $i = 1, 2, 3$, *and graphically*



Tensors might have structurally a specific rank, in this case increasing the rank would not give a better approximation.

The next state tensor F of the tensor model (3.113) was constructed to have a tensor rank of 46, [18]. Using the Tensor Toolbox, [32] and the command `cp_als`, the rank of the system tensor is reduced to 23, which has a

fit of over 99%, i.e. the sum of the differences in the corresponding entries is very small. This leads to a lower rank approximation tensor model

$$\Phi(\mathbf{x}) = \langle\, \mathsf{F}_a \mid \mathsf{M}(\mathbf{x}) \,\rangle^{\boxplus}. \tag{3.154}$$

For comparison with the originally constructed factor matrices, see appendix B, the first five entries of the first two factor matrices of F_a are given as

$$\mathbf{F}_1 = \begin{pmatrix} -0.999 & 0.999 & 1.000 & 1.000 & 1.000 & \dots \\ 0.045 & -0.054 & 0.009 & 0.007 & 0.000 & \dots \end{pmatrix} \tag{3.155}$$

$$\mathbf{F}_2 = \begin{pmatrix} 0.988 & 0.984 & 0.999 & 1.000 & 1.000 & \dots \\ -0.152 & -0.179 & 0.040 & 0.031 & 0.000 & \dots \end{pmatrix}. \tag{3.156}$$

The corresponding entries in the parameter vector are

$$\lambda = \begin{pmatrix} 10.443 & 6.967 & 4.322 & 2.598 & 2.587 & \dots \end{pmatrix}^T. \tag{3.157}$$

As opposed to the originally constructed entries, the entries of the new factor matrices contain also values different to 0 and 1. The good fit and the simulation results (see figure 3.23) suggest that there are dependencies in the next state tensor, which were not used in the construction of the factor matrices but lead to a lower rank next state tensor.

Figure 3.23 shows the simulation results of the tensor model (3.113) and its lower rank approximation (3.154). The main system dynamics are captured with both systems. The supply temperature drops to 75 °C if 95 °C is reached and rises again. Also the building and the return temperature show similar behavior for both state transition functions.

As discussed in the previous section, the value of the states has a big impact on the results. For this reason a state transformation was performed using lemma 2.1 such that all states of the transformed system are in the interval $[0\ \ 0.5]$. Since a 12^{th} order system is under investigation and the multiplication of all 12 states is possible in a tensor model, a very small factor for this multiplication of all states can give a next state many orders of magnitude away from the original next state – in the original system the maximal multilinear order was 3. This effect can be reduced significantly by normalizing the states to the interval $[0\ \ 0.5]$. The following example illustrates this issue.

Example 3.3 *Assume* $T_s = T_r = T_b = 300$ *and that for the next state of* T_s *the multiplication of all three temperatures* $\Phi(T_s) = \dots + 0 \cdot T_s T_r T_b + \dots$ *is not used. Assume also that a CP decompostion is performed and the factor for the multiplication of the three temperatures is found to be* 0.001*. The fit for the tensor is still very good, but the next state of* T_s *is* $300^3 \cdot 0.001 = 27 \cdot 10^3$

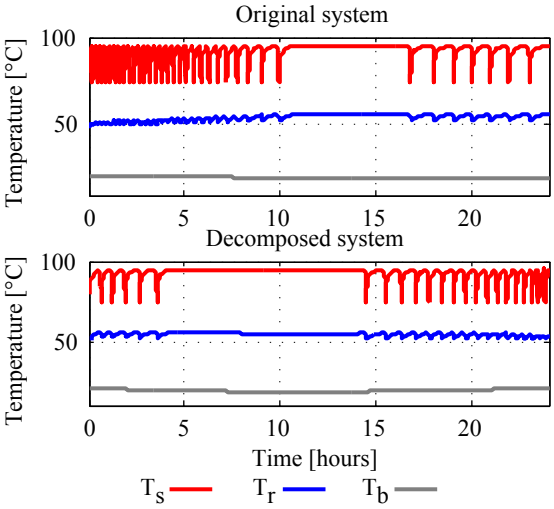


Figure 3.23: Original and decomposed system.

different to that of the original system. Now consider a transformed system. The states are assumed to be in the interval $[280 \quad 300]$ *and therefore transformed with* $\tilde{\vec{x}} = \frac{\vec{x}-280}{40}$ *leading to* $T_s = T_r = T_b = 0.5$*. To compare the sensitivity to small errors assume, that the factor for the multiplication of all three states is again* 0.001*. Calculating the error of the next state in this case gives* $0.5^3 \cdot 0.001 = 125 \cdot 10^{-6}$ *transforming this error back to the original coordinates gives an error of* $5 \cdot 10^{-3}$ *in the temperature.*

4 Model-based controller design

In this section five different controller design methods for heating systems are presented. All methods have in common, that models of the heating systems are needed. For the predictive controllers these models are used in the controller, i.e. during the control of the system. In the other controller design procedures, the model of the heating system is used in the process of designing the controller. In the controller itself, no model of the heating system is needed.

The section is organized as follows. Starting with two design methods for nonlinear models, the class of investigated models is restricted to switched affine models and finally to tensor models. The controller design methods for nonlinear systems are Boolean controller design and nonlinear model predictive control. A design method for a switched affine model of a heating system is given, then a design method for a multilinear system is presented. Finally, an approach for feedback linearization of nonlinear systems is used to linearize a multilinear system.

4.1 Boolean controller design by algebraic relaxation

In this section an approach for the design of a Boolean controller via algebraic relaxation of a Boolean optimization problem is given, [33]. The relaxation gives a continuous-valued optimization problem, which is easier to solve.

In every complex heating system with more than one heat generation unit, the decision at which time which unit contributes to the heat generation has to be reached. This decision can be expressed as a Boolean signal. If this decision is made in dependence of quantized measurements, e.g. thresholds on temperatures or flows, a Boolean controller is needed. Nowadays, most of these controllers are designed heuristically, which leads to suboptimal performance, i.e. in the application example a high switching rate.

For quantized systems a similar controller design problem was described for linear systems already in [34] and for linear hybrid systems, described as mixed logical dynamical (MLD) systems in [35]. For discrete event systems alternative approaches are given e.g. in [36]. The basic idea of the Boolean controller design approach is similar to linear program relaxation in [37], where the Boolean problem is assumed to be linear. None of these approaches is able to solve the Boolean controller design problem for the class of heating systems characterized as hybrid multilinear systems.

In the case of gene dynamics modelling, the use of algebraic normal forms of Boolean functions has been proven to be beneficial, see [38]. The use of Zhegalkin polynomials as algebraic normal forms for controller design was already proposed in [39], but was only linking a special subclass of problems to pole placement methods of linear systems.

4.1.1 Model of the heating system

The heating system under investigation in this section is supplying the building of the state office for nature, environment and consumerism, a large administration building in Düsseldorf, Germany with heat. The nonlinear model of this building, [33], is constructed using thermal balances as given in section 3.1. In addition to the basic equations given there, the model constructed here includes a variety of nonlinear phenomena of the real plant as emergency shutdowns of the boilers, if an upper temperature limit is surpassed or downtime after switching a boiler off, due to the necessity of flushing. The heating system can be divided into a heat generation unit and a consumer which is divided into six circuits. The heat generation unit consists of three boilers with burners which supply a multistory building (two circuits), two three-story buildings and a shop building (three circuits) as well as the warm tap water (one circuit) with heat, see figure 4.1.

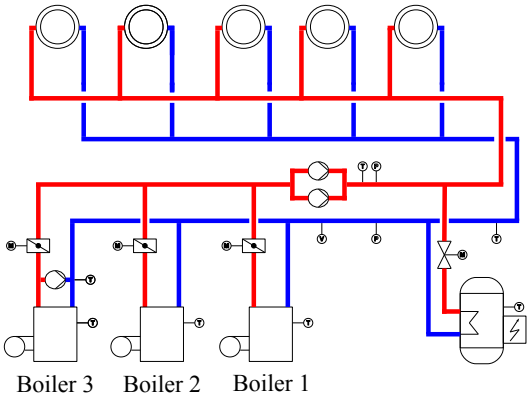


Figure 4.1: Scheme of the heating system

For the three boilers the decision which boiler is switched on has to be made. The boilers 1 and 2 are able to produce a fraction of their nominal power in a specified range (see table 4.1).

	Technology	Nominal Heating Power	Minimal Modulation
Boiler 1	cendensing	$P_{n,1} = 370\,\text{kW}$	ca. 44 %
Boiler 2	cendensing	$P_{n,2} = 720\,\text{kW}$	ca. 15 %
Boiler 3	low temperature	$P_{n,3} = 780\,\text{kW}$	no modulation

Table 4.1: Parameters of the Boilers

The boilers are steered by the modulation signals $\varphi_i(k) \in [0,\, 1],\ i = 1, 2, 3$ and $k = 0, 1, \ldots, T$, i.e. a power of $P(k) = \varphi_i(k) P_{n,i},\ i = 1, 2, 3$ has to be generated, if possible.

An example of the operation of the heating system before altering the controller is given in figure 4.2. The measurements of the supply temperature T_s from July 4th, 2010 are given together with the switching signal of boiler 2. The heating system is running with a very small heat demand coming from the warm water supply unit in this time of the year.

Between 12 am and 2:15 pm boiler 2 is switched on and off ten times. Desired are 8-10 switchings per day, i.e. the switching rate is too high. This results in the goal to reduce the number of switchings while maintaining the comfort requirements. To state this as an optimization problem define the cost function

$$J = \sum_{i=1}^{3} \sum_{k=2}^{T} \text{XOR}(\bar{\varphi}_i(k-1), \bar{\varphi}_i(k)), \tag{4.1}$$

where XOR stands for the logical operator "exclusive or" and

$$\bar{\varphi}_i(k) = \begin{cases} 1 & \text{if } \varphi_i(k) > 0 \\ 0 & \text{else}\,. \end{cases}$$

The Boolean controller design problem at hand has the following properties:

- the controller computes a Boolean output signal indicating the heat sources which are 'on',
- heating system behaviors in general are nonlinear,
- the implemented controller has fixed sampling time,

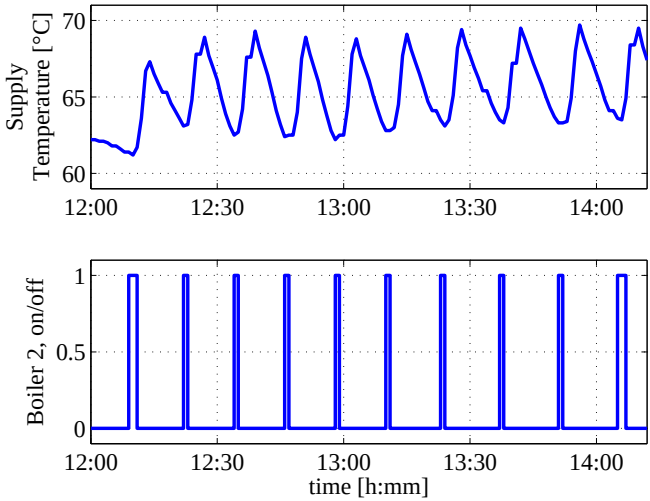


Figure 4.2: Supply temperature and boiler switching signal.

- the goal is to ensure the long term performance with different power demands depending on seasons, weekdays and time during the day.

4.1.2 Algebraic normal forms

A Boolean function b of n variables is defined as a mapping $b : \mathbb{B}^n \to \mathbb{B}$, formally. As discussed in section 2.3 a commonly used possibility to represent Boolean functions is to provide its truth table, e.g. table 4.2.

If the order of the variables in the 2^n rows of the truth table are fixed, here the rows are given in lexicographical order, it suffices to give the n-element Boolean vector

$$\underline{b} = (b_{00\cdots00}, b_{00\cdots01}, \ldots, b_{11\cdots11}) \tag{4.2}$$

to represent the Boolean function b. This vector is called truth vector, will be assumed to be a row vector in the following and is denoted as $\underline{b} \in \mathbb{B}^{2^n}$.

The number of entries in a truth vector and the number of rows in a truth table increase with 2^n, thus it becomes cumbersome to deal with truth tables and truth vectors. Therefore more compact representations of Boolean

x_n	...	x_i	...	x_3	x_2	x_1	$\underline{b}(x_1, ..., x_n)$
0	...	0	...	0	0	0	$b_{00\cdots00}$
0	...	0	...	0	0	1	$b_{00\cdots01}$
0	...	0	...	0	1	0	$b_{00\cdots10}$
0	...	0	...	0	1	1	$b_{00\cdots11}$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	...	1	...	1	1	1	$b_{11\cdots11}$

Table 4.2: Truth table of the Boolean function b

functions are used, such as *logical expressions*, or *ternary vector lists*, see [40]. Another possibility, known since the beginning of the last century, are algebraic representations of Boolean functions, [19], which are introduced in the following.

Definition 4.1 *Define the vector of literals, [41]*

$$\mathbf{L}(\mathbf{x}) = \begin{pmatrix} \bar{x}_n \\ x_n \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} \bar{x}_1 \\ x_1 \end{pmatrix} \in \mathbb{R}^{2^n} \tag{4.3}$$

for all $\mathbf{x} \in \mathbb{R}^n$*, given by the Kronecker product of the literals*

$$\mathbf{l}(x) := \begin{pmatrix} \bar{x} \\ x \end{pmatrix} \in \mathbb{R}^2 \tag{4.4}$$

where $\bar{x} = 1 - x, \forall x \in \mathbb{R}$.

For Boolean values ($x \in \mathbb{B}$) $\bar{x}$ gives their negation. The vector of literals can be used to represent a Boolean function as a Zhegalkin polynomial.

Definition 4.2 *A Zhegalkin polynomial ([19])*

$$f(\mathbf{x}) = \mathbf{b}\mathbf{L}(\mathbf{x}) \tag{4.5}$$

of order n is a polynomial of all combinations of the entries in $\mathbf{x}$ *but without square and higher order terms. The polynomial is given by the scalar product of a Boolean truth vector* $\mathbf{b} \in \mathbb{B}^{2^n}$ *and a vector of literals* $\mathbf{L}(\mathbf{x}) \in \mathbb{R}^{2^n}$*. Zhegalkin Polynomoials are sometimes also referred to as* Reed-Muller Forms *or* Algebraic Normal Forms *(ANF).*

In the following example of order $n = 2$, taken from [33], a Boolean function is given as a Zhegalkin polynomial.

Example 4.1 *The Boolean function $b = \neg(x_1 \wedge x_2)$ with truth table*

x_2	x_1	b
0	*0*	1
0	*1*	1
1	*0*	1
1	*1*	0

can be given as a Zhegalkin polynomial by

$$\begin{aligned} f(\mathbf{x}) = \mathbf{bL}(\mathbf{x}) &= \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_2\bar{x}_1 \\ \bar{x}_2 x_1 \\ x_2\bar{x}_1 \\ x_2 x_1 \end{pmatrix} \\ &= (1-x_1)(1-x_2) + x_1(1-x_2) + (1-x_1)x_2 \\ &= 1 - x_1 x_2 \ . \end{aligned}$$

It is straight forward to check, that inserting Boolean values for x_1 and x_2 results in the Boolean values given in the truth table with truth vector $\mathbf{b}$.

The connection of the truth vector with the Zhegalkin polynomial is true for all Boolean functions as given in the following proposition.

Proposition 4.1 *([39]) The Zhegalkin polynomial $f(\mathbf{x}) = \mathbf{bL}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ gives the same results as the Boolean function represented by the truth vector $\mathbf{b}$, if $\mathbf{x} \in \mathbb{B}^n$ is Boolean. But it is a continuous differentiable function $f \in C^\infty : \mathbf{x} \to f(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^n$.*

4.1.3 Boolean controller design problem

To state the Boolean controller design problem, first the system is introduced for which the controller is to be designed. The system is assumed to be represented as a hybrid deterministic discrete-time state space model for $k = 0, 1, \ldots$ given as

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \tag{4.6}$$

$$\underline{\mathbf{y}}(k) = \mathbf{g}(\mathbf{x}(k)) \tag{4.7}$$

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{4.8}$$

with the hybrid state vector $\mathbf{x}(k) \in \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}}$, the hybrid input vector

$$\mathbf{u}(k) = \begin{pmatrix} \widetilde{\mathbf{u}}(k) \\ \underline{\mathbf{u}}(k) \end{pmatrix}, \tag{4.9}$$

where $\widetilde{\mathbf{u}}(k) \in \mathbb{R}^{\widetilde{m}}$ is a continuous disturbance input, $\underline{\mathbf{u}}(k) \in \mathbb{B}^{\underline{m}}$ is a Boolean control input and $\underline{\mathbf{y}}(k) \in \mathbb{B}^{\underline{l}}$ is the Boolean output vector. The initial state is denoted as $\mathbf{x}_0$ and assumed to be fixed in the following. The transition function $\mathbf{f} : \mathbb{R}^{(\widetilde{n}+\widetilde{m})} \times \mathbb{B}^{\underline{n}+\underline{m}} \to \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}}$ and the output function $\mathbf{g} : \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}} \to \mathbb{B}^{\underline{l}}$ hold all information about the dynamics.

To evaluate a controller, a cost function has to be defined. This is done in dependence of a hybrid input vector sequence

$$\mathbf{U} = (\mathbf{u}(0), \ldots, \mathbf{u}(T-1)) \tag{4.10}$$

and the corresponding hybrid state sequence

$$\mathbf{X} = (\mathbf{x}(1), \ldots, \mathbf{x}(T)) \,, \tag{4.11}$$

as well as the Boolean output sequence

$$\underline{\mathbf{Y}} = (\underline{\mathbf{y}}(1), \ldots, \underline{\mathbf{y}}(T)) \tag{4.12}$$

computed with (4.6)-(4.8). These input and state vector sequences are used to define the cost function $J_x(\mathbf{X}, \mathbf{U})$ given as

$$J_x : \mathbb{R}^{(\widetilde{n}+\widetilde{m})\times T} \times \mathbb{B}^{(\underline{n}+\underline{m})\times T} \to \mathbb{R} \tag{4.13}$$

according to the goals of plant operation. Since for heating systems the cost function often depends on non quadratic terms like the number of switching, this function is used and no quadratic cost function is defined.

An automaton with state $\underline{\mathbf{x}}_a(k) \in \mathbb{B}^{\underline{n}_a}$, $k = 0, \ldots, T$ will be used as controller, which maps the Boolean output $\underline{\mathbf{y}}$ to the Boolean control input $\underline{\mathbf{u}}$. The controller design problem can be formulated as the search for the transition function $\underline{\mathbf{f}}_a : \mathbb{B}^{\underline{l}+\underline{n}_a} \to \mathbb{B}^{\underline{n}_a}$ of an automaton

$$\underline{\mathbf{x}}_a(k+1) = \underline{\mathbf{f}}_a(\underline{\mathbf{x}}_a(k), \underline{\mathbf{y}}(k)) \,, \tag{4.14}$$

$$\underline{\mathbf{u}}(k) = \underline{\mathbf{g}}_a(\underline{\mathbf{x}}_a(k), \underline{\mathbf{y}}(k)) \,, \tag{4.15}$$

$$\underline{\mathbf{x}}_a(0) = \underline{\mathbf{x}}_{a,0} \,. \tag{4.16}$$

such that the cost function (4.13) is minimized. For simplicity the output function $\underline{\mathbf{g}}_a$ is assumed to be fixed, as given in the application example.

The number of states $\underline{n}_a$ of the automaton can be chosen as a design parameter. The closed loop can be given by the state space model

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \underline{\mathbf{x}}_a(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\mathbf{x}(k), \begin{bmatrix} \widetilde{\mathbf{u}}(k) \\ \underline{\mathbf{g}}_a(\underline{\mathbf{x}}_a(k), \mathbf{g}(\mathbf{x}(k))) \end{bmatrix}\right) \\ \underline{\mathbf{f}}_a(\underline{\mathbf{x}}_a(k), \mathbf{g}(\mathbf{x}(k))) \end{bmatrix} . \tag{4.17}$$

Here the transition function $\underline{\mathbf{f}}_a$ and the output function $\underline{\mathbf{g}}_a$ are Boolean vector functions which could be represented by Boolean matrices $\underline{\mathbf{F}}_a \in \mathbb{B}^{\underline{n}_a \times 2^{\underline{l}+\underline{n}_a}}$ and $\underline{\mathbf{G}}_a \in \mathbb{B}^{\underline{m} \times 2^{\underline{l}+\underline{n}_a}}$. These matrices contain the truth vectors for the transition function and output function for each state and output respectively one per row.

Using the truth matrix $\underline{\mathbf{F}}_a$, the controller design problem can be formulated as a Boolean optimization problem

$$\min_{\underline{\mathbf{F}}_a \in \mathbb{B}^{\underline{n}_a \times 2^{\underline{l}+\underline{n}_a}}} J_x(\mathbf{X}, \mathbf{U}) \;\text{ with (4.17)} . \tag{4.18}$$

This optimization problem is hard to solve, since it is a discrete problem and therefore gradients do not exist because the decision variables are Boolean. With the number of inputs $\underline{l}$ and states $\underline{n}_a$ of the automaton the number of possible solutions is $2^{(\underline{n}_a 2^{\underline{l}+\underline{n}_a})}$, which means that the search space is growing with cardinality $2^{(\underline{n}_a 2^{\underline{l}+\underline{n}_a})}$. And since the hybrid dynamics of (4.6)-(4.8) lead to a cost function without any structure that is exploitable for optimization the computation of the cost function requires a simulation of the state trajectory. This is expensive in computation time.

Thus, in practice, approximations of the solution for high dimensional problems (4.18) by relaxations have to be found. In the following, we will give a relaxation which is based on the algebraic representation of Boolean functions by Zhegalkin Polynomials.

4.1.4 Relaxed algebraic problem

A requirement for this controller design approach is, that for the system given in (4.6)-(4.8) a relaxed transition function can be defined

$$\mathbf{x}(k+1) = \widetilde{\underline{\mathbf{f}}}(\mathbf{x}(k), \widetilde{\underline{\mathbf{u}}}(k)) \tag{4.19}$$

with an input vector $\widetilde{\underset{\sim}{\mathbf{u}}}(k) = \begin{bmatrix} \widetilde{\mathbf{u}}(k) \\ \underset{\sim}{\mathbf{u}}(k) \end{bmatrix} \in \mathbb{R}^{\widetilde{\mathbf{m}}} \times \mathbb{U}^{\underline{\mathbf{m}}}$ demanding

$$\begin{aligned} \widetilde{\underset{\sim}{\mathbf{f}}}(\mathbf{x}(k), \widetilde{\underset{\sim}{\mathbf{u}}}(k)) =& \mathbf{f}(\mathbf{x}(k), \underline{\mathbf{u}}(k)) \\ & \forall\, \mathbf{x}(k) \in \mathbb{R}^{\widetilde{n}} \times \mathbb{B}^{\underline{n}},\ \widetilde{\underset{\sim}{\mathbf{u}}}(k) \in \mathbb{R}^{\widetilde{m}} \times \mathbb{B}^{\underline{m}},\ k = 0, \ldots, T\ . \end{aligned} \tag{4.20}$$

This means, that the relaxed transition function describes the same dynamics as the original transition function, as long as the input is Boolean. Nevertheless the relaxed transition function will also compute the next state, if the input is continuous from the unit interval. It will be shown in the application example in the next section that for heating systems the definition of a relaxed transition function (4.19) is often straight forward. For example consider a system where the burners can be switched on or off and then produce their nominal power. The relaxed transition function models the burners as if they were able to produce a fraction of their nominal power. Formally, in the modelling, the Boolean functions are replaced by Zhegalkin polynomials. So in the modelling process there is practically no consequence.

The transition function (4.14) and the output function (4.15) of the controller can be written in algebraic normal form

$$\underline{\mathbf{x}}_a(k+1) = \underline{\mathbf{F}}_a \mathbf{L}(\underline{\mathbf{x}}_a(k), \underline{\mathbf{y}}(k))\ , \tag{4.21}$$

$$\underline{\mathbf{u}}(k) = \underline{\mathbf{G}}_a \mathbf{L}(\underline{\mathbf{x}}_a(k), \underline{\mathbf{y}}(k)). \tag{4.22}$$

If the Boolean matrix $\mathbf{F}_a$ is extended to the real unit hypercube resulting in $\underset{\sim}{\mathbf{F}}_a \in \mathbb{U}^{\underline{n}_a \times 2^{l+\underline{n}_a}}$ and $\underset{\sim}{\mathbf{x}}_a(k) \in \mathbb{U}^{\underline{n}_a}$ for $k = 1, 2, \ldots, T$, the transition and output function read

$$\underset{\sim}{\mathbf{x}}_a(k+1) = \underset{\sim}{\mathbf{F}}_a \mathbf{L}(\underset{\sim}{\mathbf{x}}_a(k), \underline{\mathbf{z}}(k))\ , \tag{4.23}$$

$$\underline{\mathbf{u}}(k) = \underline{\mathbf{G}}_a \mathbf{L}(\underset{\sim}{\mathbf{x}}_a(k), \underline{\mathbf{y}}(k))), \tag{4.24}$$

which still have the Boolean limits, i.e. if $\underset{\sim}{\mathbf{F}}_a \in \mathbb{B}^{\underline{n}_a \times 2^{l+\underline{n}_a}}$ then the state vector $\underset{\sim}{\mathbf{x}}_a(k) \in \mathbb{B}^{\underline{n}_a}$ is Boolean for all k if $\underset{\sim}{\mathbf{x}}_a(0) \in \mathbb{B}^{\underline{n}_a}$ and has the same values as $\underline{\mathbf{x}}_a(k)$ computed from (4.21). Note that the output $\underline{\mathbf{u}}(k)$ of the automaton in (4.24) is no longer restricted to be Boolean since the state of the automaton $\underset{\sim}{\mathbf{x}}_a(k)$ can take values from the unit interval.

The closed loop (4.17) with the extended transition equation (4.19) and the relaxed automaton (4.23)-(4.24) reads

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \underset{\sim}{\mathbf{x}}_a(k+1) \end{bmatrix} = \begin{bmatrix} \widetilde{\underset{\sim}{\mathbf{f}}} \left(\mathbf{x}(k), \begin{bmatrix} \widetilde{\mathbf{u}}(k) \\ \underline{\mathbf{G}}_a \mathbf{L}(\underset{\sim}{\mathbf{x}}_a(k), \mathbf{g}(\mathbf{x}(k))) \end{bmatrix} \right) \\ \underset{\sim}{\mathbf{F}}_a \mathbf{L}(\underset{\sim}{\mathbf{x}}_a(k), \mathbf{g}(\mathbf{x}(k))) \end{bmatrix}, \tag{4.25}$$

which generates for the subset $\underset{\sim}{\mathbf{F}}_a \in \mathbb{B}^{\underline{n}_a \times 2^{\underline{l}+\underline{n}_a}} \subset \mathbb{U}^{\underline{n}_a \times 2^{\underline{l}+\underline{n}_a}}$ and $\underset{\sim}{\mathbf{x}}_a(0) \in \mathbb{B}^{\underline{n}_a}$, the same trajectories as (4.17) . The cost function (4.13) leads to the relaxed problem

$$\min_{\underset{\sim}{\mathbf{F}}_a \in \mathbb{U}^{\underline{n}_a \times 2^{\underline{l}+\underline{n}_a}}} J_x(\mathbf{X}, \underset{\sim}{\mathbf{U}}) \quad \text{with (4.25)} \,, \tag{4.26}$$

where $\underset{\sim}{\mathbf{U}} = (\underset{\sim}{\mathbf{u}}(0), \ldots, \underset{\sim}{\mathbf{u}}(T))$ is the sequence of given continuous disturbance inputs.

This problem is a continuous optimization problem with constraints and a total number of $n_a 2^{r+n_a}$ decision variables in the unit interval, which can be solved by continuous solvers. Even if it is not possible or too extensive to analytically define gradients and Hessian matrices, they do exist. If nothing is known about the solution, the optimization problem can be initialized using a vector of 0.5, which will prevent the algorithm to be started in a certain direction. On the other hand, if one of the variables is most likely 1, this variable could be initialized using a value higher than 0.5. The result of continuous optimization will in general be a matrix $\underset{\sim}{\mathbf{F}}_a \in \mathbb{U}^{\underline{n}_a \times 2^{\underline{l}+\underline{n}_a}}$ showing no Boolean entries. Since a Boolean controller is to be designed, a penalty on non Boolean values is added to the cost function to enforce the optimization to end up at the Boolean boundary.

Given a weighting factor p_b, we define a penalty for non Boolean values of n decision variables by

$$J_b(\underset{\sim}{\mathbf{U}}) = p_b(1 - \frac{4}{n}\sum_{i=1}^{n}(u_i - 0.5)^2) \,. \tag{4.27}$$

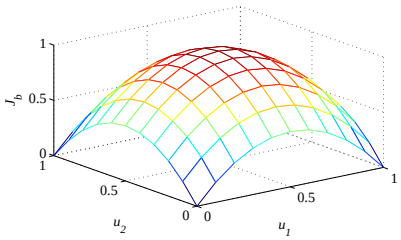


Figure 4.3: Penalty function for non-Boolean values (n=2)

Adding this term to the cost function (4.13) yields the penalized cost function

$$J_{x,b}(\mathbf{X}, \underset{\sim}{\mathbf{U}}) = J_x(\mathbf{X}, \underset{\sim}{\mathbf{U}}) + J_b(\underset{\sim}{\mathbf{U}}) \,. \tag{4.28}$$

Figure 4.3 shows the normalized penalty for two decision variables and $p_b = 1$.

4.1.5 Application example

In this section first the heating system introduced, then the controller structure as well as its output function is given. Finally results are discussed.

Heating System A MATLAB Simulink model of the heating system of the state office building was constructed using thermal balances, see section 3.1. The model simulates the supply and return temperatures, $T_s(k)$ and $T_r(k)$ for a given flow rate $\dot{V}(k)$ and heat demand $\dot{Q}_s(k)$. The assumption that the flow $\dot{V}(k)$ can be taken from measurement data is justified as long as the simulated temperatures are close to the measured ones. The assumption can be held since the main interest is to control which the heat generation units contributes heat to the pipeline network and the cumulative supply temperature is not changed. An additional input is the ambient temperature $T_a(k)$ which is used to adapt the reference supply temperature $T_{s,r}(k)$. The main complaint were the switching rates of the boilers, which were alarmingly high as shown in Figure 4.2. The pipes to the three boilers are equipped with controllable lids, such that the flow $\dot{V}$ is divided in up to three parts. This happens with a ratio of $1:2:2$, when all lids are opened, and with a ratio of $1:2:0$, when only the lids of boiler 1 and 2 are opened. Other admissible lid settings are that only the lid ob boiler 1 or 2 are opened in which case no distribution of the flow takes place. The lids can either be opened or closed, but not partly opened. Only in case a boilers lid is opened, the boiler can be switched on. A detailed description of the MATLAB Simulink Model is given in appendix C.

Important for the controller is, that the outputs of the system are quantized signals of the flow rate. The output function $\mathbf{g}$ of the system defines six signals $\underline{y}_i$

$$\underline{y}_i(k) = \begin{cases} 1 & \text{if } \dot{V}(k) > \dot{V}_i \\ 0 & \text{else} \end{cases} \qquad \text{for } i = 1, \ldots, 6 \,. \tag{4.29}$$

where the thresholds $\dot{V}_i$ are given in table 4.3.

$\dot{V}_i[\frac{m^3}{h}]$	8	11	19	23	29	34
i	1	2	3	4	5	6

Table 4.3: Thresholds for quantization

Controller The controller structure is given in form of an automaton, as given in (4.14) - (4.16) and will be used to control the steering of the lids. For computational reasons an automaton with one state is chosen. The hydraulics in the heating systems give some constraints, which are depicted in figure 4.4 using the thresholds from table 4.3. The constraints define which lid of which

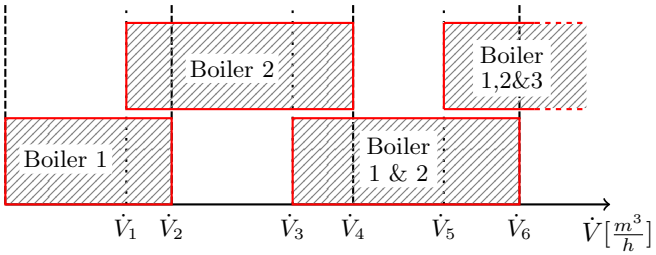


Figure 4.4: Operation range of the boilers

boiler has to be opened in dependence on the flow rate. Within the intervals where two different lid settings are possible the controller decides which configuration to choose.

As mentioned above the output equation $\underline{\mathbf{g}}_a$ of the automaton is fixed. The controller steers the lids of the boilers given in table 4.4 to be opened, depending on the state of the automaton $\underline{x}_a$ and the flow rate interval to which the flow rate currently belongs.

	$x_a(k) = 0$	$x_a(k) = 1$
$0 \leq \dot{V}(k) < \dot{V}_2$	boiler 1	boiler 2
$\dot{V}_2 \leq \dot{V}(k) < \dot{V}_4$	boiler 2	boiler 1 and 2
$\dot{V}_4 \leq \dot{V}(k) < \dot{V}_6$	boiler 1 and 2	boiler 1,2 and 3
$\dot{V}(k) \geq \dot{V}_6$	boiler 1,2 and 3	boiler 1,2 and 3

Table 4.4: Output Function

During the relaxation the state $\underline{x}_a(k)$ can take values from the unit interval.

In this case the ratio of the flow $\dot{V}(k)$ is changed accordingly. Even though partly opened lids are not possible in the system in static operation, they are possible in the model. If a lid is half opened, the flow rate is assumed to be reduced by half, as illustrated in the next example.

Example 4.2 *Let $\dot{V}(k) = 9\,\frac{m^3}{h}$ and let k denote the present time. For this flow, two configurations are possible. Either the lids of boiler 1 and 2 are opened or just the lid of boiler 1 is opened. The resulting flow rates through the boilers 1 and 2, $\dot{V}_{b1}(k)$ and $\dot{V}_{b2}(k)$ are given in Table 4.5. Now assume $\underline{x}_a(k) = 0.25$*

	ratio	$\dot{V}_{b1}(k)$	$\dot{V}_{b2}(k)$
$\underline{x}_a(k) = 1$	$1:2:0$	$\dot{V}_{b1,1}(k) = 3\,\frac{m^3}{h}$	$\dot{V}_{b2,1}(k) = 6\,\frac{m^3}{h}$
$\underline{x}_a(k) = 0$	$1:0:0$	$\dot{V}_{b1,0}(k) = 9\,\frac{m^3}{h}$	$\dot{V}_{b2,0}(k) = 0\,\frac{m^3}{h}$

Table 4.5: Flow rates through the boilers

then the flows can be calculated as

$$\dot{V}_{b1}(k) = (1-0.25)\dot{V}_{b1,0}(k) + 0.25\dot{V}_{b1,1}(k) = 7.5\,\frac{m^3}{h}$$

$$\dot{V}_{b2}(k) = (1-0.25)\dot{V}_{b2,0}(k) + 0.25\dot{V}_{b2,1}(k) = 1.5\,\frac{m^3}{h}\,.$$

The input signals $\mathbf{y}(k)$ to the automaton are preprocessed such that only three signals have to be taken into account for the state transition function of the automaton, to be still able to test all Boolean controllers. Define the intervals of the flow in $\left[\frac{m^3}{h}\right]$

$$I_1 = [0\ \ 11[,\ I_2 = [11\ \ 23[,\ I_3 = [23\ \ 34[,\ I_4 = [34\ \ \infty[$$
$$I_{b1} = [8\ \ 11],\ I_{b2} = [19\ \ 23],\ I_{b3} = [29\ \ 34]$$

then the three signals can be given as

$$\underline{y}_{p1}(k) = \begin{cases} 1 & \text{if } \dot{V}(k-1) \in I_i \wedge \dot{V}(k) \notin I_i, i = 1,2,3 \vee 4 \\ 0 & \text{else} \end{cases}$$

$$\underline{y}_{p2}(k) = \begin{cases} 1 & \text{if } \dot{V}(k-1) < \dot{V}(k) \\ 0 & \text{else} \end{cases}$$

$$\underline{y}_{p3}(k) = \begin{cases} 1 & \text{if } \dot{V}(k) \in I_{bi},\ \ i = 1,2 \vee 3 \\ 0 & \text{else} \end{cases}\,.$$

The signal $\underset{\sim}{y}_{p1}(k)$ identifies flow rate crossings of the thresholds $\dot{V}_2, \dot{V}_4$ or $\dot{V}_6$, the signal $\underset{\sim}{y}_{p2}(k)$ distinguishes between rising or falling flow rates and the signal $\underset{\sim}{y}_{p3}(k)$ detects flow rates with undetermined admissible configurations. A controller with three inputs and one state is given by the values of the next state for all $2^4 = 16$ combinations of inputs and state, i.e. a controller

$$\underset{\sim}{x}_a(k+1) = \underline{\mathbf{f}}_a \mathbf{L}(\underset{\sim}{x}_a(k), \underset{\sim}{y}_{p1}(k), \underset{\sim}{y}_{p2}(k), \underset{\sim}{y}_{p3}(k)) \tag{4.30}$$

is defined by a vector $\underline{\mathbf{f}}_a$ of length 16. For this application example, it is still possible to evaluate all possible, $2^{16} = 65536$ controllers.

Cost Function and Results The cost function has to take into account not only the number of switchings, as done in (4.1) but also the following criteria.

- Emergency shutdowns should be avoided,
- differences of reference and supply temperature should be small and
- flow rates that violate the constraints are forbidden.

All criteria have been quantified and added to the cost function (4.28).

Since the lids can only be opened or closed enitrely, non-Boolean values in the state $\underset{\sim}{x}_a$ have to be penalized (see equation (4.27)). To state the other terms of the cost function define the signal detecting switchings as

$$n_{s,i}(k) = \begin{cases} u_i(k) & \text{if} \quad \varphi_i(k-1) = 0 \text{ and } \varphi_i(k) \neq 0 \\ 0 & \text{else} \end{cases} \tag{4.31}$$

with the steering signal φ_1, φ_2 and φ_3 of the three boilers. Adding the three signals gives

$$n_s(k) = \sum_{i=1}^{3} n_{s,i}(k) \,. \tag{4.32}$$

To avoid violating flow rate constraints define

$$\dot{V}_v(k) = \begin{cases} \dot{V}_1 - \min(\dot{V}(k), \dot{V}_1) & & \dot{V}_{b1}(k) = 0 \\ \dot{V}_3 - \min(\dot{V}(k), \dot{V}_3) & \text{if} & \dot{V}_{b1}(k), \dot{V}_{b2}(k) \neq 0 \\ & & \text{and } \dot{V}_{b3}(k) = 0 \\ \dot{V}_5 - \min(\dot{V}(k), \dot{V}_5) & & \dot{V}_{b3}(k) \neq 0 \\ 0 & \text{else} & . \end{cases} \tag{4.33}$$

The emergency shutdown temperature of all three boilers is $T_e = 95\,^{\circ}\mathrm{C}$. The entire cost function can now be given as

$$\begin{aligned} J(\mathbf{X}, \mathbf{U}, \underset{\sim}{\mathbf{f}}_a) =& p_1 \sum_{k=2}^{T} \sum_{i=1}^{3} (n_s(k))^2 + p_2 \sum_{k=2}^{T} \sum_{i=1}^{3} (\max(T_e, T_{bi}(k)) - T_e) + \\ &+ p_3 \sum_{k=1}^{T} (T_s(k) - T_{ref})^2 + p_4 \sum_{k=1}^{T} (\dot{V}_v(k))^2 + \\ &+ p_b (1 - \frac{1}{4} \sum_{l=1}^{16} (\underset{\sim}{f}_a - 0.5)^2), \end{aligned} \tag{4.34}$$

where $\underset{\sim}{f}_a$ are the elements of the vector $\underset{\sim}{\mathbf{f}}_a$ and the parameters p_i, $i = 1, \ldots, 4, b$ are given in table 4.6. The parameters are chosen by looking at the terms of the cost function values for the Boolean possibilities individually.

p_1	p_2	p_3	p_4	p_b
$1.5 \cdot 10^{-3}$	2.14	$8.8 \cdot 10^{-5}$	0.86	1

Table 4.6: Parameter of the cost function

Evaluating the 65536 possible state transition functions gives the cost function values, depicted in blue in figure 4.5, where the horizontal axis gives the enumeration number of the controller. Using the approach of relaxing the Boolean problem to get a continuous optimization problem and the genetic algorithm MATLAB `ga` to solve the continuous optimization problem, controllers with cost function values depicted in red were found. The results of the continuous optimization have been rounded to Boolean values. Comparing the best Boolean controllers with the best controllers found with `ga` shows that the relaxed problem gives similar results as the original problem. A comparison of the number of switchings for the first three month after implementation of the controller is given in table 4.7. Measurement data of the supply temperature and the switching signal for July 5th, 2011 is displayed in figure 4.6. This figure shows measurement data of an entire day, where only 10 switchings occur. This is the same number as shown in figure 4.2 but there measurement data of just two hours is given.

4.1.6 Evaluation of the approach

This method was shown to give good results, for the case of a controller with one state, where the results could be compared to those generated by evalu-

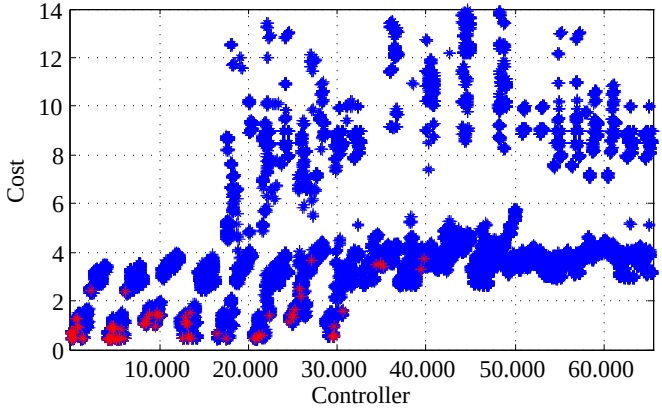


Figure 4.5: Cost function values of all evaluated controllers

	Number of switchings		
	burner 1	burner 2	burner 3
July 2010	1	2449	0
July 2011	228	291	9
August 2010	1	3103	2
August 2011	77	321	1
September 2010	8	3025	17
September 2011	130	224	2

Table 4.7: Number of switchings in 2010 and 2011

ating all Boolean controller. For a system with an additional input (resulting in $\approx 4.3 \cdot 10^9$ possibilities) or state (resulting in $\approx 18.4 \cdot 10^{18}$ possibilities), the simulation of all possible controllers could no longer be done. On the other hand the continuous optimization is still able to find at least a local minimum. A problem, that will occur is the selection of the cost function parameters. This was done here on the basis of the cost function values for the Boolean possibilities. The best controller was implemented in the real heating system in the summer of 2011, see [2, 42]. The plant operation has been improved

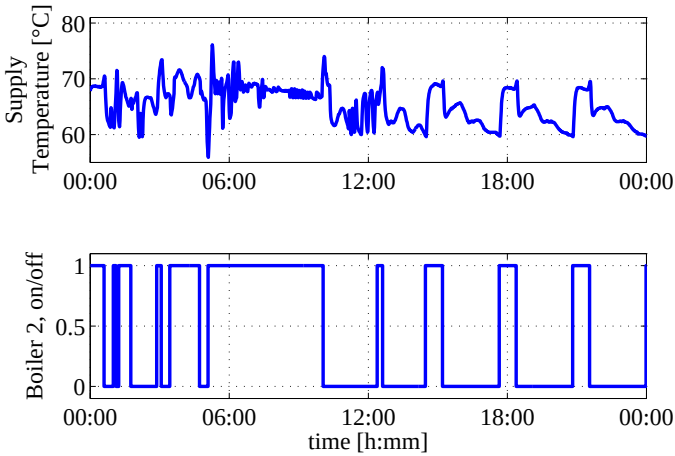


Figure 4.6: Supply Temperature and Boiler Switching Sequence

significantly and so far no problems occurred in the long term performance of the controller.

Interesting results are expected by combining this approach with the multilinearization method presented in section 3.3.6. It should be possible to extend the multilinearization method to be able to identify hybrid models, by replacing the Boolean functions by Zhegalkin polynomials and relaxing the discrete states to the unit interval. Since the model is assumed to be a Simulink model, it is obvious to assume, that this relaxation can be made.

4.2 Nonlinear model predictive control

Model predictive control is a control scheme with a structure as given in figure 4.7. The general idea will be explained in the following.

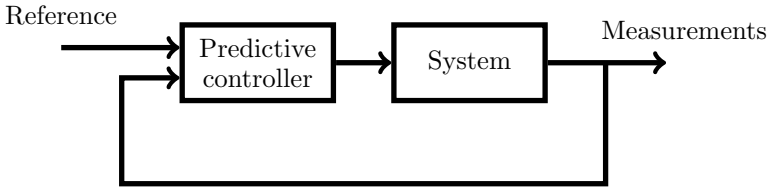


Figure 4.7: Structure of model predictive control

In each sampling instant the optimal input sequence is computed, which minimizes a cost function, using a model of the system in the controller. If the cost function is designed for good reference tracking and minimal control effort, the controller will look for the input sequence, that minimizes the tracking error while changing the input as little as possible. Here the control effort is defined as the change in the input signal. The control will just change within a time called control horizon (H_u) while the tracking error is computed for the time called prediction horizon (H_p), see figure 4.8. The prediction horizon will be larger or equal to the control horizon, $H_p \geq H_u$.

The hat symbol will be used to denote predicted signals, e.g. $\hat{y}(t)$ as the predicted output, which are generated in the predictive controller. Given the past output and input of the plant $y(k)$ and $u(k)$ respectively, at the sampling instances $k \leq 0$ as well as the reference $r(k)$, $k = 1, \ldots, H_p$, the task is to find the sequence of input changes $\Delta u(k)$, $k = 1, \ldots, H_u$ such that the sum $\sum_{k=1}^{H_p} (r(k) - \hat{y}(k))$ of the differences between reference $r(k)$ and predicted output $\hat{y}(k)$, $k = 1, \ldots, H_p$ within the prediction horizon and the control effort in the control horizon $\sum_{k=1}^{H_u} \Delta u(k)$ are minimal. These objectives are generally opposed such that a trade off is done by specifying weights in the cost function.

The computation of the future outputs can be done by matrix computations when the plant is linear and the cost function is quadratic, [43]. This is no longer possible when the plant is nonlinear, [44]. In the nonlinear case the time response of the plant needs to be simulated, e.g. by using a MATLAB Simulink model, to be able to compute the cost function value. This optimization is

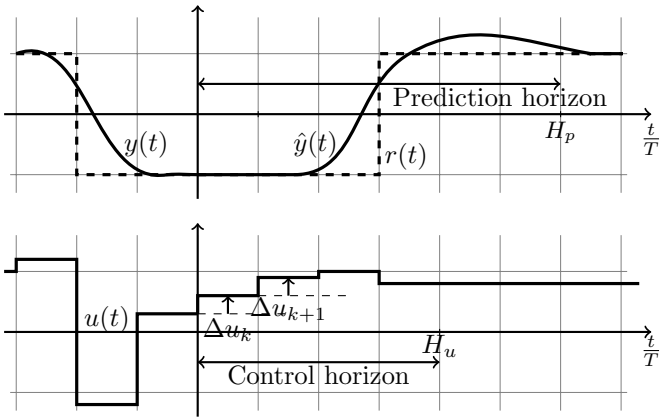


Figure 4.8: Prediction and control horizon

then no longer a quadratic program for which the convergence to a global optimum is guaranteed as in the linear case and therefore only local minima might be found, [45].

The nonlinear model predictive control scheme used in this approach can be categorized as economic model predictive control, see [46], because the cost function is used in the controller directly to evaluate the dynamic behaviour. This is in contrast to a two layer approach where a first layer determines reference trajectories, which are then tracked in the second layer by a model predictive controller. The advantage of the economic model predictive approach is, that the transients are also evaluated in terms of the cost function, whereas in the two layer approach only the steady state is optimized in terms of the cost function.

The nonlinear model predictive control is presented as follows. First the general optimization problem is given to compute the optimal input in a sampling instant for a given cost function. Then a nonlinear model of the heating system of a school building in Hamburg, Germany is derived, which will be used as application example. The goal is to operate the heating system in an energy efficient manner, while maintaining the comfort constraints. To achieve the desired plant operation a cost function function is defined in the following. Finally, simulation results are presented and conclusions are drawn.

4.2.1 Optimization problem

The search for an optimal control input can be formulated in terms of an optimization problem with constraints, [44]. The considered system is assumed to be represented by a nonlinear state space model, as given in (2.1) - (2.3)

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{4.35}$$
$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \tag{4.36}$$
$$\mathbf{x}(0) = \mathbf{x}_0. \tag{4.37}$$

The inputs and states underlie constraints given as

$$\mathbf{u}(t) \in \mathcal{U},\ \forall t \geq 0 \tag{4.38}$$
$$\mathbf{x}(t) \in \mathcal{X},\ \forall t \geq 0\,. \tag{4.39}$$

In case these constraints are constant intervals for each state and input, the sets can be stated as

$$\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^m \mid \mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}\}\,, \tag{4.40}$$
$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}\}, \tag{4.41}$$

where the inequalities are interpreted element wise. The optimal input is computed by solving the optimization problem

$$\min_{\hat{\mathbf{u}}(\cdot)}\ J(\mathbf{x}(t), \hat{\mathbf{u}}(\cdot)), \tag{4.42}$$

with

$$J(\mathbf{x}(t), \hat{\mathbf{u}}(\cdot)) = \int_t^{t+H_p} Y(\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau))d\tau \tag{4.43}$$

subject to the constraints

$$\dot{\hat{\mathbf{x}}}(\tau) = \mathbf{f}(\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau)), \quad \hat{\mathbf{x}}(t) = \mathbf{x}(t), \forall \tau \in [t, t + H_p], \tag{4.44}$$
$$\hat{\mathbf{u}}(\tau) \in \mathcal{U}, \quad \forall \tau \in [t, t + H_u], \tag{4.45}$$
$$\hat{\mathbf{u}}(\tau) = \hat{\mathbf{u}}(t + H_u), \quad \forall \tau \in [t + H_u, t + H_p], \tag{4.46}$$
$$\hat{\mathbf{x}}(\tau) \in \mathcal{X}, \quad \forall \tau \in [t, t + H_p]. \tag{4.47}$$

To convert the infinite search space to a finite one, the input sequence is quantized and assumed to be piecewise constant during the sampling instances. The cost function J is in general nonlinear and not necessarily convex, thus

no global optimum might be found. To find a minimum the MATLAB function `fmincon` is used. Also the computation time poses a problem because no guarantee on the upper limit of the computation time needed can be given. Online model predictive control is considered, since the offline computation for all possible system states and inputs might not be possible for systems with several states and inputs. Also, short horizons might be preferable for short computation times. On the other hand, short horizons are known to possibly cause problems in the plant operation even with no system model mismatch, [44]. For an implementation in the system, fall back control signals would be needed for the case, that no (local) optimal input is found within the sampling time.

4.2.2 Heating system model of the school building Mendelssohnstraße

The model of the school building's heating system is derived in [23] and will be described in this section. The heating system is build up of a heat generation unit, consisting of two boilers and a consumer, which has four circuits. Each circuit units all pipes and radiators. Furthermore, four hot water storage tanks are installed, which are connected in series. The inputs of the heating system are the produced heat in the boilers and flow rate generated by the pumps of the heat generation unit. The controller has access to all states. The outputs are the temperatures of the tank layers. The measurements used to identify the model parameters, are given with a sampling time of 60 s. Since some of the signals change up to 20 % between two sampling instances, it was challenging to identify good parameters. Figure 4.9 shows the simplified scheme of the heating system, where the pump of the consumer stands for the resulting flow rate of the four consumer circuits. The boilers are modeled as given in (3.10) but with the extension to include heat losses to the ambiance. These losses can be given in dependence of the boilers heat loss coefficient $k_{l,i}$ and the ambient temperature T_a as

$$\dot{Q}_{l,i} = k_{l,i}(T_{b,i} - T_a), \tag{4.48}$$

for $i = 1, 2$. The input power $P_{in,i}$ is given in percent of the nominal power as

$$P_{in,i}(t) = \varphi_i(t) P_{n,i}, \tag{4.49}$$

with $\varphi_i \in \{0, [\varphi_{min},\ \varphi_{max}]\}$. The boilers can be switched off but if they are running a minimal power, e.g. $\varphi = 0.2$, is produced. In the interval $[\varphi_{min},\ \varphi_{max}]$ a continuous control of the produced heat is possible. The boiler model parameters are given in table 4.8.

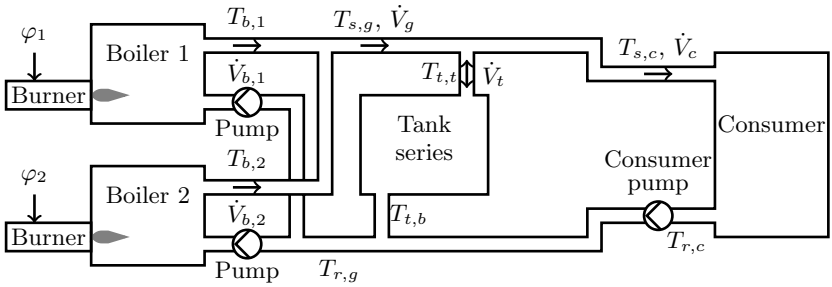


Figure 4.9: Scheme of the school building's heating system

Parameter		Value
Minimal power ratio	φ_{min}	25 %
Maximal power ratio	φ_{max}	100 %
Nominal power, boiler 1	$P_{n,1}$	275.287 kW
Nominal power, boiler 2	$P_{n,2}$	275.287 kW
Volume, boiler 1	$V_{b,1}$	0.605 m^3
Volume, boiler 2	$V_{b,2}$	0.605 m^3
Heat loss coefficient, boiler 1	$k_{l,1}$	0.352 $\frac{\mathrm{kW}}{\mathrm{K}}$
Heat loss coefficient, boiler 2	$k_{l,2}$	0.352 $\frac{\mathrm{kW}}{\mathrm{K}}$

Table 4.8: Boiler model parameters

Each pump of the boilers generates a flow rate $\dot{V}_{b,1}$ and $\dot{V}_{b,2}$ respectively, and is assumed to be controlled, such that arbitrary flow rates $\dot{V}_s = \dot{V}_{b,1} + \dot{V}_{b,2}$ can be reached. The flows coming out of the boilers have the temperature $T_{b,1}$ and $T_{b,2}$. Using equation (3.33), the supply temperature of the generation

$$T_{s,g} = \frac{T_{b,1}\dot{V}_{b,1} + T_{b,2}\dot{V}_{b,2}}{\dot{V}_g} \tag{4.50}$$

can be calculated.

Depending on the flow rate from the heat generation unit $\dot{V}_g$ and the flow rate $\dot{V}_c$ the consumer is demanding, the flow rate $\dot{V}_t$ through the tank can be

positive ($\dot{V}_g > \dot{V}_c$), then the tank is in charging mode or negative ($\dot{V}_g < \dot{V}_c$), which means the tank is in discharging mode. The 3-way valves, on the supply and the return side of the tank, change their role depending on whether the tank is charged or discharged. In charging mode, the supply side valve distributes the flow $\dot{V}_g$ and the flows $\dot{V}_t$ and $\dot{V}_c$ are flowing together in the return side valve. In discharging mode, in the supply side valve the flows $\dot{V}_t$ and $\dot{V}_g$ are merged and the return side valve distributes the flow rate $\dot{V}_c$. The temperatures of the flow rates coming out of the tank are determined by the temperatures in the tank. In charging mode, the flow $\dot{V}_t$ coming out of the tank has the temperature $T_{t,b}$ of the tank layer at the bottom connection, in discharging mode the flow rate coming out of the tank has temperature $T_{t,t}$ of the tanks layer at the top connection.

The tank model is given in section 3.1.3; the model parameters are given in table 4.9

Parameter		Value
Ambient temperature	T_a	$20\,°\mathrm{C}$
Number of layers	n	25
Number of tanks	k	4
Tank volume	V_t	$0.634\,\mathrm{m}^3$
Tank height	H_t	$1.697\,\mathrm{m}$
Position of upper connection	H_{in}	$1.418\,\mathrm{m}$
Position of lower connection	H_{out}	$0.494\,\mathrm{m}$
Heat transfer coefficient to ambiance	U	$4.388\,\frac{\mathrm{W}}{\mathrm{m}^2\,\mathrm{K}}$
Heat transfer coefficient between layers	λ_l	$10.710\,\frac{\mathrm{W}}{\mathrm{m}\,\mathrm{K}}$
Upper sensor position	h_1	$1.521\,\mathrm{m}$
Lower sensor position	h_2	$0.599\,\mathrm{m}$

Table 4.9: Tank model parameters

The consumer model uses the hydraulic structure of the heating system to avoid modeling the hydraulics in detail. The heat demand is assumed to be known for the future. The extension to include a building model, see section 3.1.4, in the consumer model is possible but omitted here for the sake of simplicity of the model. Also a mismatch between the predicted heat demand and the actual one due to inaccuracies in the ambient temperature are not considered. The flow rate $\dot{V}_c$ going into the consumer has temperature $T_{s,c}$ and is generated by the pumps in the consumer circuits. The heat is used to

supply the heat demand. Therefore the temperature is cooled down and the consumer returns the flow $\dot{V}_c$ with temperature $T_{r,c}$.

The four consumer circuits of the system are supplied with the same supply temperature. Each circuit has its own pump, bypass and 3-way valve. The 3-way valves are controlled, such that the supply temperatures in the circuits track their references, which are calculated in dependence of the ambient temperature. These four circuits are combined in the model leading to the overall flow rate of the consumer $\dot{V}_c$.

To avoid modeling the dependence of heat demand and ambient temperature, solar influences etc. the heat demand is taken from measurements. Since the consumer circuits are hydraulically decoupled, see figure 4.10, there is no need to know how the heat demand will be divided into temperature spread and flow rate in the consumer circuits.

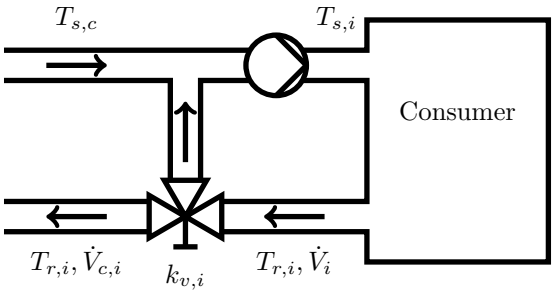


Figure 4.10: Scheme of the consumer circuits

The flow rate in the consumer circuits is taken from measurement. An I-controller is used to control the 3-way valve such that the supply temperatures $T_{s,i},\ i = 1, \ldots, 4$ of the circuits match their references $T_{s,i,r},\ i = 1, \ldots, 4$. This is achieved by mixing the colder circuits return temperatures $T_{r,i}$, which also are taken from measurements to the supply temperature $T_{s,c}$. Depending on the mixing ratio $k_{v,i} \in [0, 1]$ the flows $\dot{V}_{c,i}$ are determined. For the same heat demand sequence, different settings of the tank and heat generation can be simulated, which result in different supply temperatures $T_{s,c}$. The flows and temperatures in the consumer circuits will be still the same, as long as the supply temperature of the consumer $T_{s,c}$ is greater than the reference temperatures of the circuits. Another assumption is, that the 3-way valve

controller regulates the system quickly such that no significant tracking error occurs.

The supply temperatures of the circuits $T_{s,i}$ can be given, in dependence of the state of the valve k_v as

$$T_{s,i} = k_{v,i} T_{s,c} + (1 - k_{v,i}) T_{r,i} \tag{4.51}$$

$$\dot{V}_{c,i} = k_{v,i} \dot{V}_i. \tag{4.52}$$

Thus, for all $T_{s,c} \geq T_{r,i}$, the return temperatures $T_{r,i}$ and the flow rate $\dot{V}_i$ are valid and not depending on the heat generation. An increase of the supply temperature $T_{s,c}$ would lead to a decrease of the flow rate $\dot{V}_c$.

The assumption that the 3-way valve acts linear, meaning a half opened valve lets half of the flow rate pass, is probably not true, but since a controller is used to track the desired flow rate, the influence is minimal. The controller could compensate the tracking error even if the dependence would be nonlinear.

The combination of I-controller and valve position is modeled as a series connection of an integrator with saturation, [47] and first order system. The first order system is used to model the inertia of the valve. A time constant of $\tau = 120\,\mathrm{s}$ was identified. The integrator with saturation is given as

$$\dot{k}_i = \begin{cases} 0 & \text{if } \tilde{k}_{v,i} > 1 \text{ and } u > 0, \\ 0 & \text{if } \tilde{k}_{v,i} < 0 \text{ and } u < 0, \\ u & \text{else,} \end{cases} \tag{4.53}$$

with $u = K_i(T_{r,i} - T_{s,i})$ and $K_i = 10^{-4}$ for all circuits. The valve position is given as

$$k_{v,i} + \tau \dot{k}_{v,i} = k_i. \tag{4.54}$$

The return temperature of the consumer $T_{r,c}$ is then calculated in dependence of the return temperatures of the four circuits and the flow rate $\dot{V}_c$ as

$$T_{r,c} = \frac{\sum_{i=1}^{4} k_{v,i} \dot{V}_i T_{r,i}}{\dot{V}_c}, \tag{4.55}$$

where

$$\dot{V}_c = \sum_{i=1}^{4} k_{v,i} \dot{V}_i. \tag{4.56}$$

The input of the consumer model is the supply temperature $T_{s,c}$, disturbance inputs are the measurements of the reference temperatures of the circuits, the return temperatures $T_{r,i}$ and the flow rates $\dot{V}_i$, $i = 1, \ldots, 4$ of the circuits and the outputs are the flow rate $\dot{V}_c$ and the return temperature $T_{r,c}$.

4.2.3 Cost function

To define the optimization problem a cost function needs to be defined for the heating system example. A cumulative cost function will be used, see [23], which poses the problem how to choose the weightings. This is done here heuristically with the drawback of a quite high time consumption, since each try is associated with a not negligible simulation time of up to one minute for the search of the optimal input sequence in one time instance. The usual trade off between accuracy and control effort is not sufficient for this system, since e.g. switching of the boilers is undesirable and therefore should be penalized in the cost function.

The cost function can be given as

$$J(x(k), \hat{u}(k) \tag{4.57}$$

for a given prediction horizon H_p. Contrary to the cost function (4.43) in (4.42) this cost function is evaluated only at the sampling instances. Therefore the integral turns into a sum.

The model is evaluated at time t with the initial state $x(t)$ and the piece wise constant input sequence $\hat{u}$. The result is for each sampling instant a vector $\hat{\mathbf{y}} \geq 0$ containing the terms, which give the cost function, when they are summed up with the weights $\mathbf{w} \geq 0$. This is not a quadratic cost function but the summation of different terms corresponding to the different aspects which have to be considered in the plant operation. In the construction of the terms negative values are prevented. This cost function is then used in an economic model predictive control scheme. Thus the cost function is

$$J(x(k), \hat{u}(k) = \sum_{k=0}^{H_p-1} \mathbf{w}^T \hat{\mathbf{y}}. \tag{4.58}$$

All aspects of the cost function are:

$\hat{y}_1$	tracking of the reference temperatures in the consumer circuits,
$\hat{y}_{2,3}$	maximal temperature in the boilers,

$\hat{y}_4$	heat losses of the hot water storage tank series,
$\hat{y}_{5,6}$	efficiency of the boilers,
$\hat{y}_{7,8}$	switching of the boilers,
$\hat{y}_{9,\ldots,12}$	changes in the control signals of the pumps and the boilers.

The mathematical definition is given in appendix D.

4.2.4 Simulation results

Two nonlinear model predictive controllers were designed in [23]. The results will be presented in this section. The two controller were synthesized using different prediction horizons.

Predictive control with a prediction horizon of $\mathbf{H_p = 10}$

The first controller was designed with a prediction horizon of $H_p = 10\,\text{min}$ and a control horizon of $H_u = 10\,\text{min}$. Since the sampling time is 1 min this corresponds to horizons of 10 samples. The input signals (the steering signals of the boilers and the flow rate produced by the pumps of the heat generation) were constraint to be within the intervals given as

$$u_{min} = \begin{bmatrix} 0\,\% \\ 0\,\% \\ 0\,\frac{\text{m}^3}{\text{h}} \\ 0\,\frac{\text{m}^3}{\text{h}} \end{bmatrix} \quad \text{and} \quad u_{max} = \begin{bmatrix} 100\,\% \\ 100\,\% \\ 8\,\frac{\text{m}^3}{\text{h}} \\ 8\,\frac{\text{m}^3}{\text{h}} \end{bmatrix}. \tag{4.59}$$

The disturbance signals $[T_{s,i,r}\ T_{rl,i}\ \dot{V}_i]^T$, $i = 1, \ldots, 4$ used in the consumer model are generated using measurement data. Since the prediction horizon is relatively short, instead of using predicted disturbance signals, they are held constant. For the temperature signals this assumption yields no problem, since the temperature dynamics are slow, but for the flow rate this assumption will lead to inaccuracies, since the flow rate can change significantly within 10 min. Nevertheless, the reference tracking of the supply temperature is very good, see figure 4.11. Note that due to the initial input $u_0 = u_{min}$ a lot of computation time is needed to find a good steering signal in the first sampling instances.

The minimal and maximal temperatures are depicted in blue. The minimal temperature is chosen in dependence on the ambient temperature such that the comfort constraints in the rooms of the building can be held. The supply temperature of the existing controlscheme (black line) is often colder than the minimal temperature. This is caused by both boilers switched off, and a quick

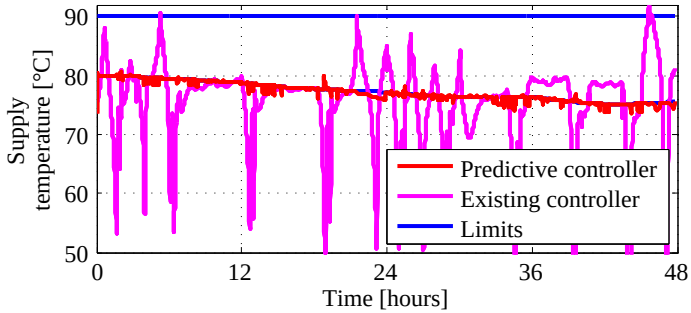


Figure 4.11: Supply temperatures of the NMPC and the existing controller with minimal and maximal temperatures starting at midnight.

discharge of the tank series. The nonlinear model predictive controller on the other hand holds the constraints of the supply temperature. To ensure a good efficiency of the boilers the predictive controller tries to reduce the supply temperature as far as possible. Therefore the supply temperature tracks the minimal temperature.

The eight sensor temperatures of the tank series - each tank has two sensors, one at the top and on at the bottom - are depicted in figure 4.12 for this run.

The model predictive controller starts discharging the tank at the beginning of the simulation. Since a charged tank has heat losses, it is preferable not to charge the tank. During the night, the tank is charged, since the cost caused by the heat looses is smaller than the cost that would be caused by switching off the boilers. At night, the boilers are running with minimal power but still produce more heat than is consumed by the system. To avoid switching off a boiler the excess heat is stored in the tank and used during the following day.

This predictive controller performs reasonably, but since the prediction horizon is short, the controller can not plan ahead very far. Also the total amount of heat losses in the tank during the night would justify switching off a boiler, but this is not true for the heat losses of the next ten minutes, therefore no switching occurs.

Predictive control with a prediction horizon of $H_p = 60$

The second model predictive controller was designed with a larger prediction

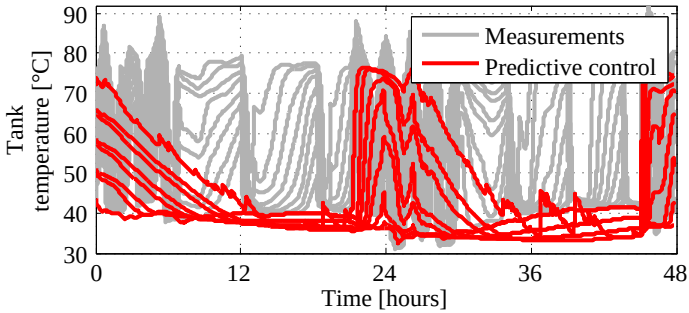


Figure 4.12: Tank temperatures of the NMPC and the existing control starting at midnight.

horizon of $H_p = 60\,\text{min}$ and a control horizon of $H_u = 30\,\text{min}$. The goal is to use only one of the boilers. In this case the tank series needs to be used to supply the system with heat in case of a high heat demand. Therefore, the tank needs to be charged, if a high heat demand is expected, but since the charged tank has heat losses it should just be charged, if the energy is needed. To reduce the computational effort, the control horizon was set to $H_u = 30\,\text{min}$ and the number of simulation executions was set to 5000.

The prediction horizon is now too long to hold the disturbance signals constant, such that predictions are necessary. To avoid modeling a consumer, which could predict the required signals, measurement data is used. This means that the model has access to the future signals.

The inputs are restricted to be within the intervals

$$u_{min} = \begin{bmatrix} 0\,\% \\ 0\,\% \\ 0\,\frac{\text{m}^3}{\text{h}} \\ 0\,\frac{\text{m}^3}{\text{h}} \end{bmatrix} \quad \text{and} \quad u_{max} = \begin{bmatrix} 24\,\% \\ 100\,\% \\ 8\,\frac{\text{m}^3}{\text{h}} \\ 8\,\frac{\text{m}^3}{\text{h}} \end{bmatrix}. \tag{4.60}$$

The upper limit on the first input will prevent the first boiler to switch on, see table 4.8. The initial input value is the maximal steering signal for the second boiler. Figure 4.13 shows the supply temperatures of the first two consumer circuits generated with the predictive controller with a prediction horizon of $H_p = 60\,\text{min}$ and their references. The consumer circuits three and

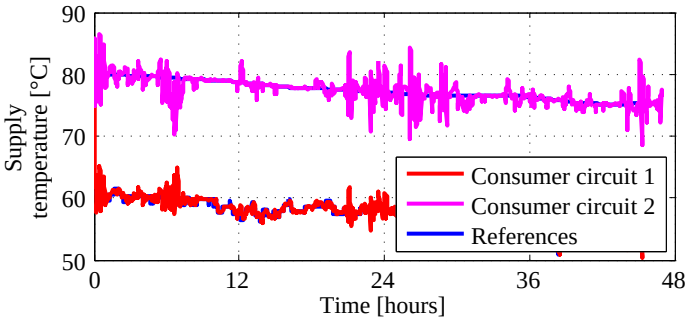


Figure 4.13: Reference and supply temperatures of the consumer circuits starting at midnight.

four show very similar dynamics as the second because they have the same reference and are therefore omitted.

To maintain the comfort constraints the reference temperatures need to track their references. As shown in figure 4.13 the control error is small. Compared to the predictive controller with shorter prediction horizon the valves in the consumer circuits need to act more often, since the supply temperature to the consumer is higher than required, when the tank is charged. In the time intervals with low heat demand, e.g. 22 h to 30 h, the supply temperature to the consumer oscillates between the upper and lower limit to supply the consumer with heat and charge the tank series. Since the valves of the consumer circuits act slow, these oscillations are forwarded to the supply temperatures of the consumer circuits. The control errors of the consumer circuits supply temperatures have a standard deviation of $\sigma < 1.5\,^{\circ}\mathrm{C}$ and a mean value of $\mu_H < 0.9\,^{\circ}\mathrm{C}$. The cumulative control error, corresponding to the difference in heat demand and produced heat over the two days of the simulation are nearly balanced. The deviation of the supply temperatures at $t \approx 6\,\mathrm{h}$ is due to a jump in the heat demand. The boiler is prevented to overheat by taking an increased control error into account. Note that this control error occurs also with the existing controller and is even worse.

The sensor temperatures of the tank series are depicted in figure 4.14.

The tank series is charged at the beginning of the simulation up to $t \approx 2\,\mathrm{h}$, also the charged tank generates heat losses, probably to avoid overheating of the boiler. During the simulation the tank is kept discharged most of the time.

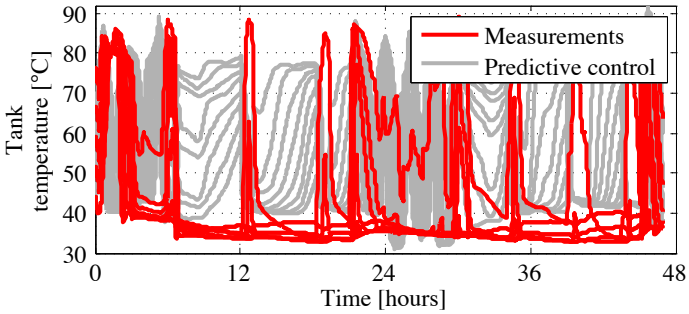


Figure 4.14: Tank temperatures of the nonlinear model predictive control ($H_p = 60\,\text{min}$) and the existing control starting at midnight.

The tank is charged to avoid switching and to be able to provide energy in time intervals with a high heat demand. Note that the predictive controller uses mainly the top of the tank. Leading to a mean temperature of the tank being low, i.e. heat losses are low, but the heat stored in the tank can be used to supply demand peaks, which have to be known in advance.

4.2.5 Evaluation of the approach

The nonlinear model predictive control shows very promising results. Opposed to the existing control, where charging and discharging of the tank series is steered indirectly by the plant operation the model predictive controller lets you choose how to operate the heating system. By adjusting the weights in the cost function, the use of only one boiler could be enforced. If the weights are chosen in a way given by the trade off between effectiveness and financial cost, one could easily adjust the controller to changes, e.g. the cost of gas. The problems for this controller to be implemented apart from the hardware implementation pose the steering signal of the pumps and the consumer model. This steering signal was assumed to be accessible and adjustable, while in reality this is not the case. To avoid loosing the warranty of the producer of the boilers, the control of the pumps which is included in the boilers, can not be changed. The use of a consumer model instead of using the measured heat demand brings in additional inaccuracies, which need further investigation. Another problem is, that the controller can not give a guarantee, that an

optimum is found within a sampling instant. Therefore a backup control, which acts if no steering signal was produced in a minute, is needed. This is a standard technique in NMPC and also an active field of research, see e.g. [48].

4.3 Controller design using the Multi - Parametric Toolbox

In this section the heating system of the state office building for nature, environment and consumerism in Düsseldorf, introduced in section 4.1 is again under investigation. For the controller, designed in section 4.1, no guarantees of stability or optimality can be given. This is also true for the predictive controller designed in section 4.2 for the heating system of the schools building. In this section, a predictive controller will be designed for the heating system of the state office building in Düsseldorf, guaranteeing optimality with respect to a cost function.

4.3.1 Constraint finite-time optimal control problem

The Multi-Parametric Toolbox ($\mathbb{MPT}$), [49] can be used to design model predictive controllers for piecewise affine systems, which were introduced in section 2.2.2. For each affine system given for a polyhedron in the state-input space, a state affine function is calculated by the toolbox which will control the system. The inputs can be calculated offline, i.e. before the system has to be controlled, in which case during the control, the state of the system defines the current active controller. This is called explicit model predictive control. The inputs can also be calculated online, i.e. during the control of the system, in which case between two sampling instances the optimal input needs to be found. The controller is designed by solving a multi-parametric program, which will lead to a linear or quadratic optimization problem depending on the chosen norm in the cost function.

The $\mathbb{MPT}$ computes the optimal explicit feedback controller for the system given in piecewise affine form as

$$\mathbf{x}(k+1) = \mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{u}(k) + \mathbf{f}_i \tag{4.61}$$

$$\mathbf{L}_i\mathbf{x}(k) + \mathbf{E}_i\mathbf{u}(k) \leq \mathbf{W}_i, \quad i \in \mathbf{I} \tag{4.62}$$

$$\text{if } \begin{bmatrix}\mathbf{x}^T(k) & \mathbf{u}^T(k)\end{bmatrix}^T \in \mathbb{D}_i, \tag{4.63}$$

see [50] for further details. This definition of a piecewise affine system includes the constraint (4.62). Depending on state and input located in domain $\mathbb{D}_i$, $i = 1, \ldots, d$ the active dynamics and constraints are selected. The dynamics are not required to be continuous valued. The controller is designed by solving the constrained finite-time optimal control problem, see [35, 51, 52],

given as

$$J_N^*(\mathbf{x}(0)) = \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{N-1}} ||\mathbf{Q}_f\mathbf{x}_N||_l + \sum_{k=0}^{N-1} (||\mathbf{R}\mathbf{u}_k||_l + ||\mathbf{Q}\mathbf{x}_k||_l) \tag{4.64}$$

$$\text{subject to } \mathbf{L}_i\mathbf{x}_k + \mathbf{E}_i\mathbf{u}_k \leq \mathbf{W}_i, \text{ if } \begin{bmatrix}\mathbf{x}_k^T & \mathbf{u}_k^T\end{bmatrix}^T \in \mathbb{D}_i, i \in \mathbf{I}, \forall k \in \{0,\ldots,N-1\}, \tag{4.65}$$

$$\mathbf{x}_N \in \mathbb{X}_{set}, \tag{4.66}$$

$$\mathbf{x}_{k+1} = \mathbf{A}_i\mathbf{x}_k + \mathbf{B}_i\mathbf{u}_k + \mathbf{f}_i, x_0 = x(0), \forall k \in \{0,\ldots,N-1\}, \tag{4.67}$$

$$\begin{cases} \mathbf{Q} = \mathbf{Q}^T \succ 0, \mathbf{Q}_f = \mathbf{Q}_f^T \succ 0, \mathbf{R} = \mathbf{R}^T \succ 0, & \text{if } \ l = 2, \\ \text{rank}(\mathbf{Q}) = n, \text{rank}(\mathbf{R}) = m, & \text{if } \ l \in \{1,\infty\} \end{cases} \tag{4.68}$$

where $\mathbb{X}_{set}$ can be used to guarantee stability and $\mathbf{Q}$, $\mathbf{Q}_f$ and $\mathbf{R}$ are weighting matrices.

4.3.2 Controller

The control loop used to control the heating system is depicted in figure 4.15. Recall that the lid setting is determined by an underlying controller designed by expert knowledge of the hydraulics. The steering signal φ for the boilers

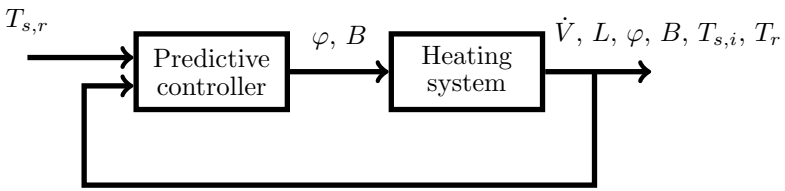


Figure 4.15: Control loop of the model predictive controller

as well as the boiler setting B are determined by the controller in a predictive fashion.

4.3.3 Application example

The results of the controller design problem described in this section were generated in [25]. The MPT will be used to generate model predictive controllers

for the model of the heating system of the state office building for nature, environment and consumerism in Düsseldorf, described in section 3.2.2, in an online procedure. Note that instead of the input power of the boilers $P = \varphi P_n$, the steering signal φ is used. Here P_n denotes the nominal power.

Since an online MPC is used to control the heating system, in each sampling instant a model of the system is used to find the optimal input. Recall, that e.g. the flow rate is assumed constant in the model. This constant can be updated in each sampling instant. To find the optimal control a cost function needs to be defined, leading to the minimization problem

$$\min_{\mathbf{u}} \left\{ \left\| \mathbf{P}_N \left(\mathbf{x}(N) - \mathbf{x}_r \right) \right\|_p + \sum_{k=0}^{N-1} \left\| \mathbf{Q} \left(\mathbf{x}(k) - \mathbf{x}_r \right) \right\|_p \right\}$$

$$= \min_{\mathbf{u}} \left\{ \sum_{k=0}^{N} \left\| \begin{bmatrix} q_1 v_1 & q_1 v_2 & q_1 v_3 & \mathbf{0} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{0} & q_2 & -q_2 \end{bmatrix} \left(\begin{bmatrix} T_{s,1}(k) \\ T_{s,2}(k) \\ T_{s,3}(k) \\ \vdots \\ u_1(k-1) \\ u_1(k-2) \end{bmatrix} - \begin{bmatrix} T_{s,r}(k) \\ T_{s,r}(k) \\ T_{s,r}(k) \\ \mathbf{0} \\ 0 \\ 0 \end{bmatrix} \right) \right\|_p \right\}$$

$$= \min_{\mathbf{u}} \left\{ \sum_{k=0}^{N} \left\| \begin{bmatrix} q_1 \left(T_s(k) - T_{s,r}(k) \right) \\ q_2 \left(u_1(k-1) - u_1(k-2) \right) \end{bmatrix} \right\|_p \right\},$$

with $\mathbf{P}_N = \mathbf{Q}$ and $\mathbf{0}$ being a vector with all entries 0 and appropriate dimension. The reference values $\mathbf{x}_r$ for the supply temperatures are calculated using a heating curve which is dependent on the ambient temperature, all other entries are zero. All reference values are kept constant. The factors q_1 and q_2 are the tuning knobs of the controller design. Increasing q_1 will lead to better tracking of the reference whereas increasing q_2 will lead to less boiler switching. The infinity norm ($p = \infty$) will be used, since the absolute control error in the supply temperature is weighted up against the cost of switching. Inserting the cost function into the MPT is done using the options `P_N`, `Q` and `xref`.

To reduce the computational effort the state space of the initial state $\mathbf{x}_0$

can be bounded in dependence of the actual state $\mathbf{x}$. Define

$$\underbrace{\begin{bmatrix} 1 & 0 & \cdots \\ -1 & 0 & \\ 0 & 1 & \cdots \\ 0 & -1 & \\ \vdots & & \ddots \end{bmatrix}}_{\mathbf{H}} \mathbf{x}_0 \leq \underbrace{\begin{bmatrix} x_1(k) + x_{tol} \\ -(x_1(k) - x_{tol}) \\ x_2(k) + x_{tol} \\ -(x_2(k) - x_{tol}) \\ \vdots \end{bmatrix}}_{\mathbf{K}}$$

for a given tolerance x_{tol} to look for feasible initial states in the given polytope, which can be done using the option `Pbnd` of the MPT.

The tuning knobs of the controller design are the horizon N, the norm p and the weightings q_1 and q_2. Using the MPT function `mpt_control` the controller evaluated in the next section was synthesized.

4.3.4 Simulation results

Three controllers generated with the MPT will be presented, [25]. First a controller was constructed using a time horizon of only one sampling $N = 1$, choosing the weightings as $\frac{q_1}{q_2} = 10$ and the maximum norm $p = \infty$. The computation time for the controller is quite short. All controllers constructed during the simulation were generated within one second. For the second controller, the weightings were chosen as $\frac{q_1}{q_2} = 7$. This increases the importance of a good reference tracking of the supply temperature. Finally a controller with a time horizon of $N = 2$ is synthesized. This leads to an increased computation time of up to 6 seconds.

Measurements taken on the same six days already used for the application example in section 4.1 were used to simulate the system. The supply temperature, the boiler setting and the steering signal for the boilers are shown in figure 4.16 for measurements taken on November 10, 2009. The simulation shows that the reference supply temperature is reached with a small exception after about 300 min. At that point the steering signal is at 100 % but the difference in the supply temperature and the reference

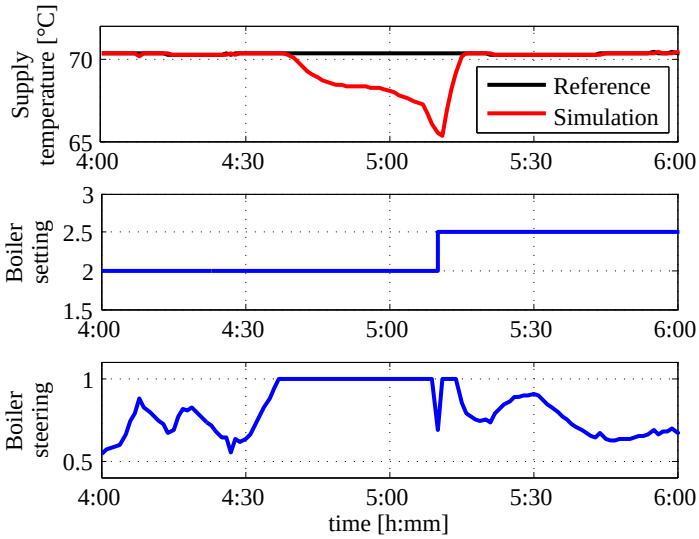


Figure 4.16: Simulation results using measurements from November 10, 2009

yields a smaller cost function value than would be caused by switching. Note that the simulation results are shown only for less than 2 h. During the rest of the simulation the reference is tracked satisfactory.

Using the measurements from July 1, 2010 the simulation results given in figure 4.17. The heat demand is very small in this time of the year such that even one boiler produces to much heat and needs to be switched on and off a lot. One can see that if the boiler is switched on, the steering signal of the boiler is increased up to 50 %. It would be better for this signal to stay at its minimal value, but for this the time horizon is not sufficiently large.

Decreasing the ratio of the weightings to $q_2/q_1 = 7$ leads to switching when smaller deviations of supply temperature and reference are present. In case of the simulation results with the measurements from November 11, 2009 (depicted in figure 4.16 for the ratio of $q_2/q_1 = 10$

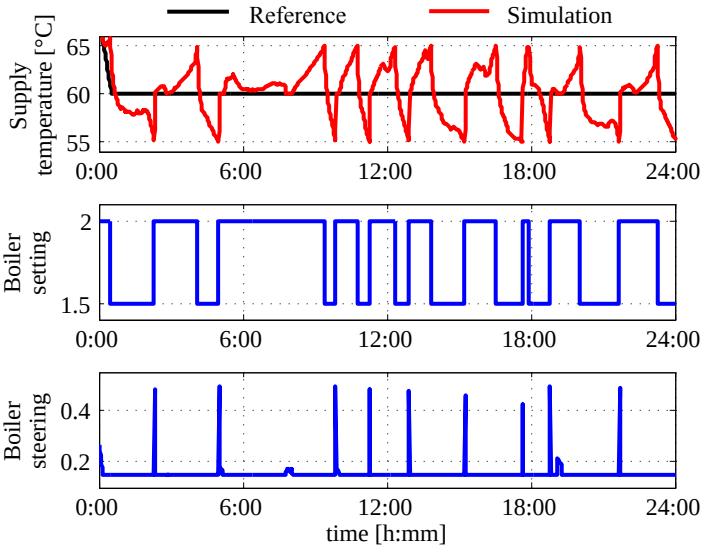


Figure 4.17: Simulation results using measurements from July 1, 2010

leading to a control error of 5 K) a difference of 3.5 K is sufficient for the controller to switch on the next boiler. This decrease in the control error is paid by an increase of the number of switchings of the boilers.

Finally the horizon is doubled to $N = 2$. Apart from some shifting of the importance of the weights, the interesting point is the computation time of up to 6 s. This does not pose a problem because between two samples 60 s are available. However, the advantage of two seconds over one second in the time horizon is very minimal.

4.3.5 Evaluation of the approach

The model of the heating system is constructed in a very abstract way. The basic equations are used, but special effects like the flushing of the boilers or the boilers emergency switching off are not modeled. This makes it questionable if the presented approach can be implemented in

a real system. Also the benefit of the approach is inexistent compared to general nonlinear model predictive control. Since the model, that is used here, is simplified, a guarantee of stability can not be provided. The optimal trade off between the number of switchings and the control error is found but this trade off reduces to a maximal tolerated deviation of the supply temperature from its reference. This is because of the very short time horizon. One switching of a boiler would lead to good tracking, but before this switching occurs first the reference tracking needs to create a cost function value that is bigger, than the one created be the switching. This lead to a deviation of the reference before each switching. The choice of the maximum norm is also responsible for this simple trade off, since just the higher weighed criterion is taken into account. The desired effect to switch the boiler on or off immediately, if switching can not be avoided and not at all, if a justifiable control error is present, would be possible with the 2-norm and a control horizon of at least 10 min. But a horizon of such a length is not possible for this system due to the curse of dimensionality, which all methods of multi-parametric programming suffer from, [50]. In conclusion, this approach is able to deal with systems which show hybrid dynamics, like heating systems, but the complexity of a heating system is too high, such that the implementation of a control using the MPT does not seem to be possible yet.

4.4 Controller design for multilinear input affine systems guaranteeing ellipsoidal domains of attraction

In section 3.3.5 the possibility to model a heating system as a tensor model was presented, see (2.60) - (2.61). This motivates the investigation of the model class. Here a method to design a controller guaranteeing the local stability of an operating point and its domain of attraction is given, where the multilinear properties of the tensor model are used. The model class under investigation in this section includes the continuous-valued part of a tensor model and does not include the discrete-valued part, i.e. the Boolean system.

An asymptotically stable operating point of a nonlinear system is characterized by being an equilibrium point and by having a domain of attraction, i.e. all state trajectories starting from any point in the region of attraction will end up in the equilibrium point, [53]. The equilibrium point is, without loss of generality, assumed to be the origin. The definitions for stability will be given in the next section.

The proposed controller design method was developed in [54] and works as follows. For a given system and its equilibrium point a desired region of attraction is defined, which is assumed to be known from the application at hand. The controller, if it can be found, then ensures the attractiveness within this region, i.e. the state trajectories will end up in the origin. The controller is designed using the Lyapunov theory, where a Lyapunov function is constructed, having the property of a negative time derivative in the domain of attraction. If such a function can be found the existence of the domain of attraction is guaranteed. The computation of the exact region of attraction is very difficult. There has been extensive research in this area, see e.g. [55]. To simplify this search, one can restrict the region of attraction to be ellipsoidal. In this case the scaling of certain directions in the state space can still be accounted for [56]. In this approach an ellipsoidal region of attraction is used, therefore the region defined from the application is assumed to be an ellipsoid as well. The controller consists of two parts. A linear one, which is designed using well known controller design methods for linear systems using a linear approximation of the system in the origin

ensuring local stability of the origin and a multilinear one, which will ensure the domain of attraction.

Estimating a subset of the region of attraction for the class of polynomial systems is done by [57], where the theorem of [58] and the extension of [59] are used to estimate the maximal value of the Lyapunov function derivative $\dot{L}(x)$[4] on the surface of an ellipsoid. The ellipsoid is defined by the Lyapunov function. Thus the closest approximation is achieved by maximizing the radius of the ellipsoid.

After the definition of stability and the definition of the controller design problem, the Lyapunov function estimate is given followed by the controller design procedure. As an application example, given a region of attraction, a controller is designed for the heating system (3.113), which is first adapted to the assumption to include only continuous-valued states.

4.4.1 Stability

First the definition of stability needs to be given, as it can be found in numerous publications, e.g. [53]. Assuming an autonomous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{4.69}$$

with $\mathbf{f} : \mathbb{D}^n \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ is Lipschitz over the domain $\mathbb{D}$ and has an equilibrium point at the origin, then the following definitions can be given.

Definition 4.3 *The equilibrium point of (4.69) is*

- locally stable *if for each* $\epsilon > 0$ *there is a* $\delta > 0$ *such that*

$$\|\mathbf{x}(0)\| < \delta \rightarrow \|\mathbf{x}(t)\| < \epsilon \forall t \geq 0,$$

- unstable *if it is not stable,*

[4] To avoid confusions with the flow rate, L is used instead of the usual V as Lyapunov function symbol.

- locally asymptotically stable *if it is locally stable and δ can be chosen such that*

$$\|\mathbf{x}(0)\| < \delta \rightarrow \lim_{t\to\infty} \mathbf{x}(t) = 0,$$

- globally asymptotically stable *if the equilibrium point is stable for all initial values of* $\mathbf{x}(0) \in \mathbb{R}^n$.

4.4.2 Controller design problem

For the discrete-time case, multilinear systems in matrix representation were introduced in (2.32)-(2.33). The systems under investigation in this section are continuous-time, input affine systems given as

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{X} + \mathbf{B}\mathbf{u}, \tag{4.70}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector with elements $x_i,\ i = 1,\dots,n$ and $\mathbf{u}(t) \in \mathbb{R}^m$ is the input vector with elements $u_i,\ i = 1,\dots,m$. Furthermore, the matrix $\mathbf{F} \in \mathbb{R}^{n\times 2^n-1}$ is called system matrix and the matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$ input matrix. The restriction that the system is input affine is used to ensure, that the closed loop still is multilinear, see section 2.3.5.

The vector $\mathbf{X}(t) \in \mathbb{R}^{2^n-1}$ is defined as

$$\mathbf{X} = \begin{bmatrix}\mathbf{x}^T & \mathbf{X}_2^T & \mathbf{X}_3^T & \dots & \mathbf{X}_n^T\end{bmatrix}^T, \tag{4.71}$$

with the vectors $\mathbf{X}_i \in \mathbb{R}^{\sigma_i}$ containing all possible combinations of i different states from the set of n different state variables. The dimension of the vectors $\mathbf{X}_i$ is given as

$$\sigma_i = \frac{n!}{i!(n-i)!}. \tag{4.72}$$

To illustrate these definitions, the vectors $\mathbf{X}_i,\ i = 1,2,n$ are given

$$\begin{aligned}
\mathbf{X}_2 &= \begin{bmatrix}x_1x_2 & x_1x_3 & \dots & x_1x_n & x_2x_3 & \dots & x_{n-1}x_n\end{bmatrix}^T,\\
\mathbf{X}_3 &= \begin{bmatrix}x_1x_2x_3 & x_1x_2x_4 & \dots & x_2x_3x_4 & \dots & x_{n-2}x_{n-1}x_n\end{bmatrix}^T,\\
&\vdots\\
\mathbf{X}_n &= x_1x_2\cdots x_{n-1}x_n.
\end{aligned} \tag{4.73}$$

The controller which will be used, is a state feedback controller given as

$$\mathbf{u} = K(\mathbf{x}), \tag{4.74}$$

where $K : \mathbb{R}^n \to \mathbb{R}^m$ is a multilinear function. This controller will be used to ensure a given domain of attraction of the origin. The domain of attraction is given as an ellipsoid as

$$L(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x} \leq c, \quad c > 0 \tag{4.75}$$

by defining the symmetric, positive definite matrix $\mathbf{P}$ and the constant c. The function $L(\mathbf{x})$ will be used as the Lyapunov function. As long as the time derivative of this function

$$\dot{L}(\mathbf{x}) = 2\mathbf{x}^T\mathbf{P}\dot{\mathbf{x}} \tag{4.76}$$

is negative within the domain $\left\{\mathbf{x} \mid \mathbf{x}^T\mathbf{P}\mathbf{x} \leq c, \mathbf{x} \neq \mathbf{0}\right\}$, i.e. the domain of attraction, the origin is called asymptotically stable, [53]. In this equation the closed-loop dynamics are associated with $\dot{\mathbf{x}}$.

The controller design will be done in two steps. First a linear controller $\mathbf{K}_l$ is designed to ensure stability of the origin, then a multilinear controller $\mathbf{K}_m$ is designed to ensure the domain of attraction. The controller consisting of the linear and the multilinear part is a state feedback controller and the closed loop can be given as

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{X} + \mathbf{B}\mathbf{K}\mathbf{X} = (\mathbf{F} + \mathbf{B}\mathbf{K})\mathbf{X}, \tag{4.77}$$

where $\mathbf{K} = \begin{bmatrix}\mathbf{K}_l & \mathbf{K}_m\end{bmatrix} \in \mathbb{R}^{m \times 2^n - 1}$.

To design the linear controller $\mathbf{K}_l$ the system (4.70) is linearized around the origin, which gives the linear system matrix

$$\mathbf{A} = \left.\frac{\partial \mathbf{F}\mathbf{X}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{0}} \tag{4.78}$$

and the closed loop

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}_l\mathbf{x}. \tag{4.79}$$

To design this controller, standard techniques can be applied, e.g. optimal control by minimizing the cost function

$$J = \int_0^\infty \mathbf{x}^T\mathbf{Q}_l\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}\ dt, \quad \mathbf{R} > 0,\ \mathbf{Q}_l \geq 0. \tag{4.80}$$

The linear controller is used to ensure stability of the closed loop of the linearized system. Therefore, the origin of the closed loop of the multilinear system (4.77) will be stable.

The multilinear part of the controller $\mathbf{K}_m$ has to ensure the domain of attraction defined by (4.75). Formally this can by stated as an optimization problem given as

$$\min_{\mathbf{k}_m} \mathbf{k}_m^T \mathbf{Q} \mathbf{k}_m \quad \text{s.t.} \quad \mathbf{x}^T \mathbf{P} \mathbf{x} \leq c, \qquad 2\mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} < 0 \text{ for } \mathbf{x} \neq 0 \tag{4.81}$$

where the matrix $\mathbf{K}_m$ has been converted to a vector $\mathbf{k}_m \in \mathbb{R}^{m \cdot (2^n - 1 - n)}$ and the weighting matrix is $\mathbf{Q} \in \mathbb{R}^{m \cdot (2^n - 1 - m) \times m \cdot (2^n - 1 - m)} \geq 0$.

To simplify the constraints in (4.81) an upper bound on the temporal derivative of the Lyapunov function on the surface of the ellipsoid is used.

4.4.3 Upper and lower bounds on polynomials

For the estimation of a maximum on the temporal derivative of a Lyapunov function $\dot{L}(\mathbf{x})$ the following definitions and theorems will be used.

Definition 4.4 *A real algebraic polynomial of order m is given as*

$$p(x) = \sum_{i=0}^{m} a_i x^i \tag{4.82}$$

with the coefficients $a_i \in \mathbb{R}$ and $x \in \mathbb{R}$, [60].

Definition 4.5 *Let $\|\cdot\|_\infty$ denote the maximum norm and let $J = [a, b]$ be a nonempty compact interval, where $J \subset \mathbb{R}$. The set of Chebyshev points for an algebraic polynomial is given as*

$$x(N, J) := \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2\nu - 1)\pi}{2N}\right), \quad \nu = 1, \ldots, N \tag{4.83}$$

for $N > 0$, [60].

Theorem 4.1 *An upper bound on the maximum of a polynomial p in the interval J of one variable with degree $p \leq m$ is given with the inequality*

$$\|p\|_{\infty}^{J} \leq C \cdot \|p\|_{\infty}^{x(N,J)} \tag{4.84}$$

where $\|p\|_{\infty}^{x(N,J)}$ is the maximal value of the polynomial p evaluated in the $N > m$ Chebyshev points and

$$C := \frac{1}{\cos\left(\frac{m}{2N}\pi\right)}. \tag{4.85}$$

Proof: The proof is given in [58]. □

Definition 4.6 *A real trigonometric polynomial of order m is given as*

$$p(\theta) = \sum_{k=0}^{m} \sum_{i=0}^{k} a_{ki} \sin^{i}\theta \cos^{k-i}\theta = \sum_{k=0}^{m} \{a_k \cos(k\theta) + b_k \sin(k\theta)\} \tag{4.86}$$

with the coefficients $a_{ki},\ a_k,\ a_i \in \mathbb{R}$ and $\theta \in [0,\ 2\pi]$, [60].

Definition 4.7 *The set of Chebyshev points for a trigonometric polynomial within the interval J is given by [60]*

$$x(N,J) := a + \frac{\nu - 1}{N}(b-a), \quad \nu = 1, \ldots, N. \tag{4.87}$$

Theorem 4.2 *The upper bound on the maximum of the trigonometric polynomial is given with (4.84) where $N > 2m$ and*

$$C := \frac{1}{\cos\left(\frac{m}{N}\pi\right)}. \tag{4.88}$$

Proof: The proof is given in [58]. □

Corollary 4.1 *Using (4.84), the following upper and lower bounds could be established*

$$p_{min}^{J} = \frac{1}{2}\left((C+1)\, p_{min}^{x(N,J)} - (C-1)\, p_{max}^{x(N,J)}\right), \tag{4.89}$$

$$p_{max}^{J} = \frac{1}{2}\left((C+1)\, p_{max}^{x(N,J)} - (C-1)\, p_{min}^{x(N,J)}\right) \tag{4.90}$$

where p_{min}^{J} is the minimal and p_{max}^{J} is the maximal value of the polynomial p in the interval J, $p_{min}^{x(N,J)}$ is the minimal and $p_{max}^{x(N,J)}$ is the maximal value of the polynomial p evaluated at the N Chebyshev points in the interval J.

Proof: The proof is given in [59]. □

These inequalities are extended to hold true for several variables. The multivariable interval is constructed from the cartesian products of the individual intervals

$$\hat{J} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_n, b_n] \tag{4.91}$$

and the set of Chebyshev points is therefore

$$x(\hat{N}, \hat{J}) := x(N_1, J_1) \times x(N_2, J_2) \times \ldots \times x(N_n, J_n). \tag{4.92}$$

Corollary 4.2 *The upper and lower bounds for n variables are given as*

$$p_{min}^{\hat{J}} = \frac{1}{2}\left(\left(\hat{C}+1\right) p_{min}^{x(\hat{N},\hat{J})} - \left(\hat{C}-1\right) p_{max}^{x(\hat{N},\hat{J})}\right), \tag{4.93}$$

$$p_{max}^{\hat{J}} = \frac{1}{2}\left(\left(\hat{C}+1\right) p_{max}^{x(\hat{N},\hat{J})} - \left(\hat{C}-1\right) p_{min}^{x(\hat{N},\hat{J})}\right) \tag{4.94}$$

where

$$\hat{C} = \prod_{i=1}^{n} C_i\ . \tag{4.95}$$

Proof: The proof is given in [59]. □

The multipliers C_i are depending on the number of Chebyshev points N_i and the degree m_i of the variable x_i of the polynomial.

4.4.4 Replacing the constraint in the controller design problem

For preconditioning, two coordinate transformations will be performed, to eventually solve the optimization problem. First use the Cholesky decomposition $\mathbf{P} = \mathbf{C}^T\mathbf{C}$ to rewrite the Lyapunov function in the new coordinates $\hat{\mathbf{x}} = \mathbf{C}\mathbf{x}$ as

$$L(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x} = \hat{\mathbf{x}}^T\hat{\mathbf{x}} = \hat{L}(\hat{\mathbf{x}}), \tag{4.96}$$

see [57]. In the Cholesky decomposition the matrix $\mathbf{C}$ is an upper triangular matrix, which is unique since the matrix P is positive definite, see e.g. [60]. This transformation converts the ellipsoid into an n-sphere, graphically speaking. The second transformation converts the coordinates $\hat{\mathbf{x}}$ into n-dimensional polar coordinates given as

$$\begin{aligned}
\hat{x}_1 &= r\cos\varphi\sin\theta_1\sin\theta_2\cdots\sin\theta_{n-3}\sin\theta_{n-2},\\
\hat{x}_2 &= r\sin\varphi\sin\theta_1\sin\theta_2\cdots\sin\theta_{n-3}\sin\theta_{n-2},\\
\hat{x}_3 &= r\cos\theta_1\sin\theta_2\cdots\sin\theta_{n-3}\sin\theta_{n-2},\\
&\vdots\\
\hat{x}_{n-1} &= r\cos\theta_{n-3}\sin\theta_{n-2},\\
\hat{x}_n &= r\cos\theta_{n-2},
\end{aligned} \tag{4.97}$$

where the radius is $r \in [0, \infty)$ and the angles are $\varphi \in [0, 2\pi]$ as well as $\theta_i \in [0, \pi], i = 1, \ldots, n-2$. These coordinates allow a simple description of an ellipsoid.

Let $p(\mathbf{x}) = \dot{L}(\mathbf{x})$ be the temporal derivative of the Lyapunov function. The inverse transformation of $\hat{\mathbf{x}} = \mathbf{C}\mathbf{x}$ is used to obtain the original polynomial in polar coordinates depending on the radius and the angles $\hat{p}(r, \varphi, \theta_1, \ldots, \theta_{n-2})$. Using (4.94) and Chebyshev points for the angles φ and $\theta_i, i = 1, \ldots, n-2$ an upper bound on the maximum of $\dot{V}(\mathbf{x})$ on the boundary of the ellipsoid is found. The radius $r = \sqrt{c}$ is kept constant.

To reduce the computational effort, the following conjecture is used, [54].

Conjecture 4.1 *It is conjectured that for autonomous, multilinear systems*

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{X} \tag{4.98}$$

and a given matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ *with the properties* $\mathbf{P} = \mathbf{P}^T, \mathbf{P} > 0$*, the following statement holds true*

$$\forall c \geq c^* : \exists \mathbf{x} \in \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T\mathbf{P}\mathbf{x} = c, \dot{L}(\mathbf{x}) > 0 \right\} \tag{4.99}$$

where

$$c^* = \min_{\mathbf{x}\neq\mathbf{0}} \mathbf{x}^T\mathbf{P}\mathbf{x} \quad s.t. \quad \dot{L}(\mathbf{x}) > 0 \tag{4.100}$$

This assumption was not yet falsified by any counterexample, but tested for numerous multilinear systems. For polynomial systems this assumption is not true. A counter example is given in [54].

Using conjecture 4.1, the optimization problem (4.81) reduces to

$$\min_{\mathbf{k}_m} \mathbf{k}_m^T\mathbf{Q}\mathbf{k}_m \quad \text{s.t. } p_{max}^{\hat{J}} < 0, \tag{4.101}$$

where $p_{max}^{\hat{J}}$ given by (4.94). Stability guarantees, offered by a controller synthesized based on the above conjecture are analyzed in an a posteriori validation by additionally generating Chebyshev points for the radius. Therefore the entire content of the ellipsoid is checked.

The number of computations in each step is proportional to $\prod_{i=1}^{n-1} N_i$, where N_i is the number of Chebyshev points in each variable. The conjecture reduces the computational effort by a factor N_n. This means, if the conjecture turns out to be false, the optimization problem can still be solved, but an additional computational effort of factor N_n (i.e. the number of Chebyshev points for the radius) is needed.

4.4.5 Application examples

First a stabilizing controller for a fictitious multilinear system is designed, then a heating system is modeled as a multilinear system and a controller for this system is designed. Both examples are taken from [54].

Two states example

Given a fictitious multilinear system with two states and one input in matrix representation

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \begin{pmatrix}-1 & 1 & 11\\ 2 & -2 & 9\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_1x_2\end{pmatrix} + \begin{pmatrix}0\\ 1\end{pmatrix} u \tag{4.102}$$

and a fictitious desired region of attraction

$$\begin{pmatrix}x_1 & x_2\end{pmatrix}\begin{pmatrix}8 & -2\\ -2 & 1\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} \leq 9.3, \tag{4.103}$$

which is assumed to be known from the application, a stabilizing controller could be designed. The control law is given by

$$u = \begin{pmatrix}-3.1223 & -10.2574 & 1.5088\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_1x_2\end{pmatrix}, \tag{4.104}$$

where the first two entries are given by standard linear controller design techniques and the last entry is given by solving the minimization problem (4.101). Since the constraint in (4.101) is met, this controller guarantees the attractiveness of the region of attraction. It is therefore shown, that multilinear systems exist, for which the optimization problem (4.101) can be solved and stabilizing controllers can be synthesized.

Heating system example

For the heating system described in section 3.1.7, a controller is to be designed, which guarantees a domain of attraction. The model of the heating system is altered such that no discrete-valued signals and states are used. Therefor the model is not valid for the entire operating region of the plant. But by restricting the room temperature to be in a small interval the validity of the model for the restricted operating region is given.

Only the supply temperature T_s, the return temperature T_r and the building temperature T_b are used as states. The ambient temperature is assumed to be constant, $T_a = 15\,^{\circ}\mathrm{C}$.

The dependency of the flow rate on the building temperature is adapted and therefore only valid in the interval $T_b \in [T_{br} - \Delta T_b,\ T_{br} + \Delta T_b]$.

This assumption is required to avoid discrete signals arising from the saturation. Since the room temperature is assumed in an interval such that the flow rate is not in saturation, no inaccuracies arise here. The dependence of the flow rate on the building temperature is depicted in figure 4.18, where the dashed line is the actual flow rate, the solid line the modeled one and T_{br} is the building reference temperature.

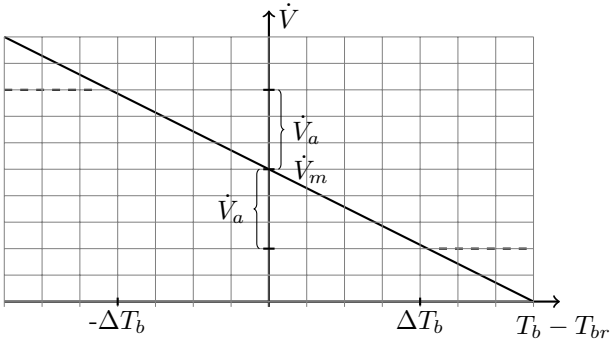


Figure 4.18: Real (dashed line) and modeled (solid line) dependence of the flow rate on the building temperature

The flow rate $\dot{V}$ can be given as

$$\dot{V} = \dot{V}_m + \frac{T_{br}\dot{V}_a}{\Delta T_b} - \frac{\dot{V}_a}{\Delta T_b} T_b. \tag{4.105}$$

By inserting the equation of the flow rate (4.105) into the heat power balances for the supply temperature (3.10), the return temperature (3.29)

and the building temperature (3.30), the multilinear state space model

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{T}_s \\ \dot{T}_b \\ \dot{T}_r \end{bmatrix} = \begin{bmatrix} f_{11} & 0 & f_{13} & f_{14} & 0 & f_{16} & 0 \\ 0 & f_{22} & f_{23} & 0 & 0 & 0 & 0 \\ f_{31} & f_{32} & f_{33} & f_{34} & 0 & f_{36} & 0 \end{bmatrix} \begin{bmatrix} T_s \\ T_b \\ T_r \\ T_s T_b \\ T_s T_r \\ T_b T_r \\ T_s T_b T_r \end{bmatrix} + \\ + \begin{bmatrix} \frac{P_n}{V_b \rho c} \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{k_{b,a}}{c_b} \\ 0 \end{bmatrix} T_a, \tag{4.106}$$

is obtained. The elements of $\mathbf{F}$ are given as

$$f_{11} = -f_{13} = -\frac{\Delta T_b \dot{V}_m + T_{br} \dot{V}_a}{\Delta T_b V_b}, \tag{4.107}$$

$$f_{14} = -f_{16} = \frac{\dot{V}_a}{V_b \Delta T_b}, \tag{4.108}$$

$$f_{22} = -\frac{k_{r,b} + k_{b,a}}{c_b}, \tag{4.109}$$

$$f_{23} = \frac{k_{r,b}}{c_b}, \tag{4.110}$$

$$f_{31} = \frac{\Delta T_b \dot{V}_m + T_{br} \dot{V}_a}{\Delta T_b V_c}, \tag{4.111}$$

$$f_{32} = \frac{k_{r,b}}{V_c \rho c}, \tag{4.112}$$

$$f_{33} = -f_{31} - f_{32}, \tag{4.113}$$

$$f_{34} = -f_{36} = -\frac{\dot{V}_a}{V_c \Delta T_b} \tag{4.114}$$

and the produced heat of the burner is $P_{in}(t) = P_n u(t)$, with $u(t) = [0, 1]$.

To design the controller, the model needs to be transformed such that the origin is an equilibrium point, which can be done using the coordinate transformation

$$\overline{\mathbf{x}} = \mathbf{x} - \mathbf{x}_{ss}, \quad \overline{u} = u - u_{ss} \tag{4.115}$$

where $\mathbf{x}_{ss}$, u_{ss} denotes an equilibrium point (see e.g. [53]).

For the example at hand, and in order to obtain an equilibrium point, the building temperature needs to equal its reference, i.e. $T_b = T_{br}$. This leads to a linear system of equations whose solution yields the transformation

$$\mathbf{x}_{ss} = \begin{bmatrix} \frac{k_{b,a}(T_{br}-T_a)}{\dot{V}_m \rho c} + \frac{(k_{b,a}+k_{r,b})T_{br}-k_{b,a}T_a}{k_{r,b}} \\ T_{br} \\ \frac{(k_{b,a}+k_{r,b})T_{br}-k_{b,a}T_a}{k_{r,b}} \end{bmatrix}, \tag{4.116}$$

$$u_{ss} = \frac{k_{b,a}}{P_n}(T_{br} - T_a). \tag{4.117}$$

The parameters are given in table 3.1 and table 3.2. To define the ellipsoid the values of the transformed variables are investigated. The temperature $\overline{T}_s$ goes up to 50 °C, where $\overline{T}_r$ reaches about 30 °C. The building temperature is not supposed to fall below $T_{br} - T_b = -2\,^{\circ}\mathrm{C}$. Thus, the ellipsoid is chosen as

$$L(\mathbf{x}) = \mathbf{x}^T \begin{bmatrix} 1 & 1 & -1.3 \\ 1 & 110 & 1 \\ -1.3 & 1 & 2.2 \end{bmatrix} \mathbf{x} \leq 700. \tag{4.118}$$

The poles of the linearized model are heuristically set by pole placement to be

$$p_1 = (-6.139 + 6.007\mathrm{i}) \times 10^{-4}, \tag{4.119}$$

$$p_2 = (-6.139 - 6.007\mathrm{i}) \times 10^{-4}, \tag{4.120}$$

$$p_3 = -1.29 \times 10^{-5}. \tag{4.121}$$

Using the above procedure the controller $\mathbf{K} = \begin{bmatrix} \mathbf{K}_l & \mathbf{K}_m \end{bmatrix}$ could be synthesized, where

$$\mathbf{K}_l = \begin{bmatrix} 21700 & -55500 & -29400 \end{bmatrix} \times 10^{-6}, \tag{4.122}$$

$$\mathbf{K}_m = \begin{bmatrix} -3764 & 6.426 & 3793 & -22.56 \end{bmatrix} \times 10^{-6}. \tag{4.123}$$

During the design of the multilinear part a small constraint violation occurred. The temporal derivative of the Lyapunov function $\dot{V}(\mathbf{x})$ is depicted in figure 4.19.

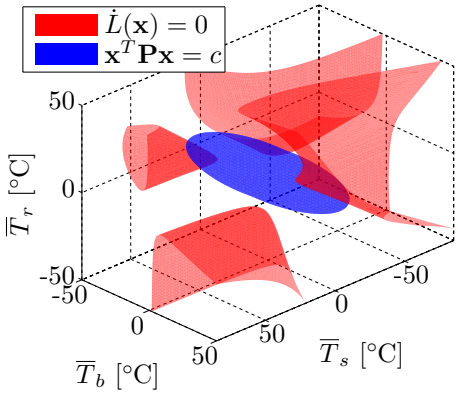


Figure 4.19: Ellipsoid $L(\mathbf{x}) \leq c$ and temporal derivative of the Lyapunov function $\dot{L}(\mathbf{x})$.

4.4.6 Evaluation of the approach

It was shown, that there exist multilinear systems for which the controller design method works. The separation of designing the linear part and multilinear part of the controller is beneficial, because the number of variables in $\mathbf{K}_m$ was reduced, such that less computation time is needed. For the heating system example, the constraint in the optimization problem was violated. Looking at the Lyapunov function and its derivative in figure 4.19 one can see, that the temporal derivative is indeed negative in the region of attraction. Further investigation yields, that in this area the building temperature is above its reference. Since the heating system - and the intervals on the supply and the return temperature - is not designed for cooling down the building, the controller can not ensure, that the building reference temperature is reached. In practice this will not be a concern, because the building temperature will equal the ambient temperature after the decay of the transient behaviour. The controller is not able the steer the building temperature back to its reference, but the temperature will be reduce with the time constant of the building.

4.5 Feedback linearization of discrete-time nonlinear state space systems using lattice theory

Linearizing nonlinear systems using nonlinear static state feedback in continuous time has been researched in the early 1980's. A differential geometric method is used to linearize continuous-time systems, see e.g. [61]. The discrete-time case did not get so much attention. Nevertheless some results were presented, e.g. [62, 63].

The discrete-time representation is used here because modeling discrete-valued signals in discrete time is much easier. In a new approach the feedback linearization problem was readdressed using algebra of functions, [64]. The advantage is, that in this approach also non-smooth functions can be considered.

The attempt was to adapt the approach to linearize tensor models. Unfortunately, up to now this could not be completed. Therefore this chapter presents work in progress.

First, algebra of functions and necessary operators are introduced, followed by the linearization procedure, proposed in [64]. A short application example is given, in which a multilinear model of a heating system is linearized. Eventually, the approach is evaluated.

4.5.1 Algebra of functions

Let $\mathbb{S}_X$ be the set of vector functions whose domain $\mathbb{X} \subseteq \mathbb{R}^n$ is the state space. The relation of partial preorder denoted by $\leq$ is given with the following definition, [64].

Definition 4.8 *For $\alpha,\ \beta \in \mathbb{S}_X$ and a function γ*

$$\alpha \leq \beta, \quad \textit{iff} \ \ \exists \gamma : \beta(\mathbf{x}) = \gamma(\alpha(\mathbf{x})), \forall \mathbf{x} \in \mathbb{X}. \tag{4.124}$$

Note that the case where $\alpha \not\leq \beta$ and $\beta \not\leq \alpha$ can occur. In this case α and β are incomparable. In case $\alpha \leq \beta$ and $\beta \leq \alpha$ the functions are called equivalent, which will be denoted by $\alpha \cong \beta$. Since equivalent functions can be in the set $\mathbb{S}_X$, the term partial preorder instead of partial order is used.

For the relation of partial preorder $\alpha \leq \beta$ in [65] a rank condition is given for smooth functions as, $\alpha \leq \beta$ iff

$$\operatorname{rank}\left(\frac{\partial \alpha}{\partial x}\right) = \operatorname{rank}\left(\left(\frac{\partial \alpha}{\partial x}\right)^T \quad \left(\frac{\partial \beta}{\partial x}\right)^T\right)^T. \tag{4.125}$$

A reduction in the rank can occur, if the rows or columns are functionally dependent. Rows being the same for a point in the state space, do not reduce the rank of the function.

The binary operations $\times$ and $\oplus$ are defined as, [64]

Definition 4.9

$$\alpha \times \beta = \inf(\alpha, \beta) \tag{4.126}$$
$$\alpha \oplus \beta = \sup(\alpha, \beta). \tag{4.127}$$

Example 4.3 *Assume a function δ has the properties*

$$\delta \leq \alpha, \tag{4.128}$$
$$\delta \leq \beta. \tag{4.129}$$

The function δ is equivalent to $\alpha \times \beta$ iff

$$\epsilon \leq \delta, \forall \epsilon \in \mathbb{S}_X : \epsilon \leq \alpha, \epsilon \leq \beta. \tag{4.130}$$

In other words $\alpha \times \beta = \delta$ is the function with the maximal number of functionally independent components which can be generated by the components of α and β. Here generated means, there exists a function, which applied to α is equivalent to δ and there exists a possibly different function, which applied to β also gives δ.

Example 4.4 *Assume a function δ has the properties*

$$\alpha \leq \delta, \tag{4.131}$$
$$\beta \leq \delta. \tag{4.132}$$

The function δ is equivalent to $\alpha \oplus \beta$ iff

$$\delta \leq \epsilon, \forall \epsilon \in \mathbb{S}_X : \alpha \leq \epsilon, \beta \leq \epsilon. \tag{4.133}$$

In other words $\alpha \otimes \beta = \delta$ is the function with the minimal number of functionally independent components which can generate the function α as well as β.

The rule to calculate $\alpha \times \beta$ is given as, [64]

$$(\alpha \times \beta)(x) = \begin{pmatrix} \alpha^T(x) & \beta^T(x) \end{pmatrix}^T, \tag{4.134}$$

where functionally dependent components need to be removed. The following example is taken from [65].

Example 4.5 *Let $X = \mathbb{R}^3$ and $\mathbb{S}_X$ be the set of functions having X as domain. Define*

$$\alpha = \begin{pmatrix} x_1 + x_2 & x_3 \end{pmatrix}^T \text{ and} \tag{4.135}$$

$$\beta = \begin{pmatrix} x_1 x_3 & x_2 x_3 \end{pmatrix}^T. \tag{4.136}$$

Calculating the function $(\alpha \times \beta)(x)$ gives

$$(\alpha \times \beta)(x) \cong \begin{pmatrix} x_1 + x_2 & x_3 & x_1 x_3 \end{pmatrix}^T. \tag{4.137}$$

Due to functional dependence the term $x_2 x_3$ does not occur. The calculation of the function $(\alpha \oplus \beta)(x)$ gives

$$(\alpha \oplus \beta)(x) \cong \begin{pmatrix} (x_1 + x_2)\, x_3 \end{pmatrix} \tag{4.138}$$

which can be generated by multiplying the first and second component of α and by adding the first and second component of β. No other functionally independent components can be generated from α and β.

For more complex examples, the calculation of the $\oplus$ operator is no longer that simple. Using the definition of $\oplus$ together with the rank condition of the relation of partial preorder leads to a partial differential equation. The calculation of $\oplus$ can be reformulated. Find all functionally independent γ_j such that

$$\frac{\partial \alpha_1}{\partial x_i} a_1 + \frac{\partial \alpha_2}{\partial x_i} a_2 + \ldots \frac{\partial \alpha_m}{\partial x_i} a_l = \frac{\partial \gamma_j}{\partial x_i} \tag{4.139}$$

$$\frac{\partial \beta_1}{\partial x_i} b_1 + \frac{\partial \beta_2}{\partial x_i} b_2 + \ldots \frac{\partial \beta_m}{\partial x_i} b_m = \frac{\partial \gamma_j}{\partial x_i} \tag{4.140}$$

for $i = 1 \ldots, n$ and some $a_1, \ldots, a_l$, $b_1, \ldots, b_m$. Since the rank needs to be the same $\frac{\partial \gamma_j}{\partial \mathbf{x}}$ needs to be a linear combination of $\frac{\partial \alpha}{\partial \mathbf{x}}$, $k = 1, \ldots, l$ and $\frac{\partial \beta}{\partial \mathbf{x}}$, $k = 1, \ldots, m$.

This reformulation is only valid for smooth functions, since the rank condition is only true for smooth functions. For an example with two components in α and three states, this set of equations can be rewritten by eliminating the a_i as

$$\begin{aligned}
&\left(\frac{\partial \alpha_1}{\partial x_3}\frac{\partial \alpha_2}{\partial x_2} - \frac{\partial \alpha_1}{\partial x_2}\frac{\partial \alpha_2}{\partial x_3}\right)\frac{\partial \gamma}{\partial x_1} + \\
&\left(\frac{\partial \alpha_1}{\partial x_1}\frac{\partial \alpha_2}{\partial x_3} - \frac{\partial \alpha_1}{\partial x_3}\frac{\partial \alpha_2}{\partial x_1}\right)\frac{\partial \gamma}{\partial x_2} + \\
&\left(\frac{\partial \alpha_1}{\partial x_2}\frac{\partial \alpha_2}{\partial x_1} - \frac{\partial \alpha_1}{\partial x_1}\frac{\partial \alpha_2}{\partial x_2}\right)\frac{\partial \gamma}{\partial x_3} = 0
\end{aligned} \tag{4.141}$$

To solve this kind of partial differential equations, i.e.

$$f(x_1, x_2, x_3)\frac{\partial \gamma}{\partial x_1} + g(x_1, x_2, x_3)\frac{\partial \gamma}{\partial x_2} + h(x_1, x_2, x_3)\frac{\partial \gamma}{\partial x_3} = 0 \tag{4.142}$$

the characteristic system

$$\frac{dx_1}{f(x_1, x_2, x_3)} = \frac{dx_2}{g(x_1, x_2, x_3)} = \frac{dx_3}{h(x_1, x_2, x_3)} \tag{4.143}$$

is used [66]. Let $u_1(x_1, x_2, x_3) = C_1$ and $u_2(x_1, x_2, x_3) = C_2$ be an integral basis, i.e. two different functionally independent integrals of the characteristic system (4.143), then the solution of (4.142) is given by

$$\gamma = \Phi(u_1, u_2), \tag{4.144}$$

where Φ is an arbitrary function.

In the case for two components in α and β and three states, two integral bases can be found. One for α and one for β. This poses nearly the same problem as the original one since we look for functions that can be constructed as well with the arguments found for α as those found for β.

In the following the binary relation Δ is defined, [64].

Definition 4.10 *Two functions $\alpha, \beta \in \mathbb{S}_X$ form an ordered pair, i.e. $\alpha, \beta \in \Delta$, if there exists a function f_* such that*

$$\beta(f(\mathbf{x}, u)) = f_*(\alpha(\mathbf{x}), u), \tag{4.145}$$

where $f(\mathbf{x}, u)$ is a next state function of a nonlinear state space system 2.4, $\mathbf{x} \in \mathbb{X}$ is the state and $u \in \mathbb{U} \subseteq \mathbb{R}$ is the input.

Next two operators m and M are defined, [64].

Definition 4.11 *Applying operator m to a function α gives a function $m(\alpha) \in \mathbb{S}_X$ which satisfies the following conditions*

$$(\alpha, m(\alpha)) \in \Delta \tag{4.146}$$

$$\text{if } (\alpha, \beta) \in \Delta, \text{then } m(\alpha) \leq \beta. \tag{4.147}$$

Definition 4.12 *Applying operator M to a function β gives a function $M(\beta) \in \mathbb{S}_X$ which satisfies the following conditions*

$$(M(\beta), \beta) \in \Delta \tag{4.148}$$

$$\text{if } (\alpha, \beta) \in \Delta, \text{then } \alpha \leq M(\beta). \tag{4.149}$$

The operator $m(\alpha)$ can be computed by calculating

$$m(\alpha) \cong ((\alpha \times u) \oplus f)^{-} \tag{4.150}$$

where $(\ldots)^{-}$ denotes a temporal back-shift, [64].

The operator $M(\beta)$ can be computed in the special case where $\beta(f(x, u))$ can be decomposed in

$$\beta(f(x, u)) = \sum_{i=1}^{d} a_i(x) b_i(u) \tag{4.151}$$

with $a_1(x), \ldots, a_d(x)$ being arbitrary functions and $b_1(u), \ldots, b_d(u)$ being linearly independent over $\mathbb{R}$ as

$$M(\beta) := a_1 \times a_2 \times \ldots \times a_d, \tag{4.152}$$

see [64].

Definition 4.13 *The function α is called invariant with respect to the system dynamics $\mathbf{x}(k+1) = f(\mathbf{x}, u)$ or f-invariant, if for arbitrary $u \in \mathbb{U}$*

$$\alpha(\mathbf{x}_1) = \alpha(\mathbf{x}_2) \Rightarrow \alpha(f(\mathbf{x}_1, u)) = \alpha(f(\mathbf{x}_2, u)) \tag{4.153}$$

for $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{X}$ holds true, [64].

To compute the minimal f-invariant function δ the following algorithm is introduced in [64].

Algorithm 4.1 *The sequence $\delta^1 \leq \delta^2 \leq \ldots \leq \delta^i \leq \ldots$ of non-decreasing functions can be computed for a given δ^1 by the recursive formula*

$$\delta^{i+1} = \delta^i \oplus m(\delta^i). \tag{4.154}$$

The algorithm is initialized by using $\delta^0 = \begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix}^T$ and choosing δ^1 as the minimal function, which contains the maximal number of functionally independent components, giving that the forward shift $\delta^1(f(\mathbf{x}, u))$ does not depent on the input u. The algorithm is terminated as soon as $\delta^j \cong \delta^j + l$, $l \geq 1$ holds true. The minimal f-invariant function is then $\delta = \delta^j$. In [67] it was shown, that there exist a finite $j \leq n$ such that the above conditions hold true and $\delta^i \ncong \delta^i + 1$ for $i < j$.

4.5.2 Linearization procedure

If a state transformation $\varphi : \mathbb{X} \to \mathbb{X}$ exists and in addition a static state feedback $u = \vartheta(\mathbf{x}, v)$, with $\vartheta : \mathbb{X} \times \mathbb{V} \to \mathbb{U}$ exists and has an inverse with respect to v, such that $z = \varphi(x)$ and

$$\Phi(z_i) = z_{i+1}, \text{ for } i = 1, \ldots, n-1 \tag{4.155}$$

$$\Phi(z_n) = v, \tag{4.156}$$

then, the system

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), u(k)) \tag{4.157}$$

is called static state feedback linearizable, see [64]. The condition for a system to be static state feedback linearizable is given in [64] with the theorems 24 and 25 and the proposition 26 therein. These theorems were combined to result in the following corollary.

Corollary 4.3 *The system (4.157) is static state feedback linearizable, iff*

$$\delta^i \neq \ const \text{ for } i = 1, \ldots, n-1 \text{ and } \delta^n = const. \tag{4.158}$$

Proof: The proof follows directly from the proofs for the theorems 24 and 25 and proposition 26 given in [64]. □

If a system is state feedback linearizable, the state coordinate transformation is given by

$$z_1 = \delta^{n-1} \tag{4.159}$$
$$z_2 = M(\delta^{n-1}) \tag{4.160}$$
$$z_3 = M(M(\delta^{n-1})) \tag{4.161}$$
$$\vdots \tag{4.162}$$

and the system in the new coordinates then reads

$$\Phi(z_1) = z_2 \tag{4.163}$$
$$\Phi(z_2) = z_3 \tag{4.164}$$
$$\vdots \tag{4.165}$$
$$\Phi(z_n) = \psi(\mathbf{z}, u), \tag{4.166}$$

with $\psi(\mathbf{z}, u) = \varphi_n(f(\mathbf{x}, u))$ being the static state feedback, [64].

4.5.3 Application example

Using the website [68] the feedback linearization for a heating system example could be obtained. The heating system under investigation was introduced in section 3.1.7. Here it was used in the simplified version

introduced in section 4.4.5. Since the website works with numerical values the parameters given in tabel 3.1 and table 3.2 are used. The state space system can be given as

$$\begin{aligned} x_1(k+1) &= 1.366u(k) + x_2(k)\left(13.92 - 0.04643x_3(k)\right) \\ &\quad + x_1(k)\left(-12.92 + 0.04643x_3(k)\right) && (4.167) \\ x_2(k+1) &= x_1(k)\left(2.923 - 0.00975x_3(k)\right) \\ &\quad + x_2(k)\left(-1.9947 + 0.00975x_3(k)\right) + 0.07174x_3(k) && (4.168) \\ x_3(k+1) &= 0.08095 + 0.00015x_2(k) + 0.9996x_3(k) && (4.169) \end{aligned}$$

where the input u is in MW and the states

$$\begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}^T = \begin{pmatrix} T_s & T_r & T_b \end{pmatrix}^T \tag{4.170}$$

are in K.

Using the state transformation

$$\begin{aligned} z_1(k) =& x_3(k) && (4.171) \\ z_2(k) =& 0.08095 + 0.00015x_2(k) + 0.9996x_3(k) && (4.172) \\ z_3(k) =& 0.161868 + x_1(k)\left(0.00043845 - 1.4625 \cdot 10^{-6}x_3\right) && (4.173) \\ & + x_2(k)\left(-0.000149265 + 1.4625 \cdot 10^{-6}x_3\right) + 0.999211x_3(k) \end{aligned}$$

and the feedback

$$\begin{aligned} v(k) =& \frac{1}{-11692 + 39z_1(k)}\left(-944.635 + u(k)\left(-7.0026+\right.\right. \\ & +z_1(k)\left(0.023358 - 0.0000779132z_2(k)\right) + 0.023358z_2(k)\big) - \\ & - 348715z_2(k) + 1199.38z_2^2(k) + \\ & + z_1^2(k)\left(-617.681 + 2.06034z_2(k)\right) + 162695z_3(k) - \\ & - 617.877z_2(k)z_3(k) + z_1(k)\left(174337 - 4.25133z_2^2(k) - \right. \\ & \left.\left.\left. -581.688z_3(k) + z_2(k)\left(656.817 + 2.19102z_3(k)\right)\right)\right) \end{aligned} \tag{4.174}$$

the system (4.167)-(4.169) can be rewritten as the linear system

$$\begin{aligned} z_1(k+1) &= z_2(k) && (4.175) \\ z_2(k+1) &= z_3(k) && (4.176) \\ z_3(k+1) &= v(k). && (4.177) \end{aligned}$$

4.5.4 Evaluation of the approach

The investigation of this approach was motivated by a small example. For a multilinear model of a heating system a feedback could be found, such that the closed loop was linearized. The method succeeded in linearizing the model with a coordinate transformation and a static state feedback. The approach is appealing because the state update function does not need to be smooth, which would allow switching in the system. The importance of this property especially for heating systems was already established in this work. Within the procedure a partial differential equation was constructed, which can only be done for smooth functions, the extension on calculating the binary operation $\otimes$ to non-smooth functions is very poorly described in the literature. A solution might be contained in [69], which is in Russian.

So far the use of the multilinear property of the model was not exploited in the linearization procedure. Especially the temporal back-shift poses a problem, because the temporal back-shift of a multilinear function can contain stronger nonlinearities. This is similar to the discussion on connecting multilinear systems given in section 2.3.5. Future work can be done in the construction of the binary operators using the multilinear property.

4.6 Comparison of the controller design methods

In this chapter five different controller design methods were presented. The control objectives of the methods were different for all approaches.

The Boolean controller design by algebraic relaxation, see section 4.1, was used to bring down the amout of switchings in a heating system with three boilers. This approach was successfully tested in the real plant and the controller is still running. The method allows to find Boolean controllers by relaxing the optimization problem to a continuous one. The relaxation has the advantage, that one does not have to evaluate all Boolean controllers but can find the optimal by solving a continuous optimization problem. The requirement to use this method is a general nonlinear model of the plant and the possibility to evaluate the plant with non-Boolean input signals. Non-Boolean input signals pose a problem for Boolean controllers which can often be solved, e.g. if some input can be *on* or *off*, the model might still be able to simulate the time response for *half on* .

The nonlinear model predictive control presented in section 4.2 shows promising results. The control objective in this approach was to minimize the use of the second boiler by maintaining the comfort requirements. By adapting the weights in the cost function the objective could easily be changed to minimal energy consumption of to minimal economic costs. The requirements for this approach to work is a very efficient model of the plant. The optimal control sequence should be found in one time step (here one minute). For the real implementation also the target platform needs to be performant enough such that this calculation can still be done. Also a backup control needs to be provided because no guarantee is given that the optimal input sequence is found in one time step even though this might be true for most cases.

The model predictive control presented in section 4.3 uses the Multi-Parametric Toolbox to minimize the number of switchings of the boilers in a multi-boiler heating system and to guarantee the stability of the predictive control. This is done by using a switched affine model of the heating system. Therefore a switched affine model of the plant is the requirement of this approach. This method was shown to be inadequate to control a heating system in a predictive way. Not only was

the switched affine model of the heating system very abstract and could not capture special system dynamics like emergency shutdowns, also the time horizons for which solutions could be found were too short to be able to control this plant in a way where unnecessary switchings could be avoided.

The stabilizing controller design which was presented in section 4.4 was used to find stabilizing controllers for multilinear systems. The requirement for this approach is, that the system to be controlled can be modeled as a continuous-valued multilinear system. This approach uses the properties of multilinear systems for the controller design. Also this approach is very promising and was shown to work on a fictitious multilinear system, by applying the method to a multilinear model of a heating system a constraint violation occurred. This was identified to be related to the incapability of a heating system to cool a building.

The final controller design method, section 4.5 is still work in progress. The requirements for this approach should be a plant which can be modeled as a tensor model. The author has the opinion, that the properties of tensor systems could simplify the calculation of the supremum and infimum operators that are used in the recent approach to feedback linearization. Unfortunately so far this approach is not very extensively described in the literature.

All five approaches have different aspects of controlling heating systems in focus. The first approach is already shown to work in reality. The nonlinear model predictive control is very close to be able to be implemented. The basis for an implementation is given, but a lot of work will still be needed to be done in the implementation progress. Just to mention a few: problems in loosing warranty by changing the pump settings in the boilers, backup control, controller platform. The controller using the Multi-Parametric Toolbox will probably not be used in reality in the near future. The stabilizing controller on the other hand could easily be implemented yet, but probably not for a heating system. The feedback linearization still needs a lot of work until it will eventually be implemented. But in this approach so far no fatal obstacle has been met.

5 Conclusions

In this chapter a summary of the conducted research is given and conclusions are drawn. In addition a short outlook on possible future work is given.

5.1 Summary

Models of heating systems have been constructed for controller design. The models are generally hybrid, i.e. they have discrete-valued and continuous-valued signals. The continuous-valued subsystem was shown to have a multilinear structure, which is true for nearly all heating system components. The multilinearity, i.e. the multiplication of temperature difference and flow rate, arises from the heat power balances, which are the core of the models. General nonlinear state space models of heating system components were described. Also a piecewise affine model of a heating system was constructed. Finally tensor systems were shown to be capable to model heating systems. A method to get a multilinear model, which can easily be transformed into a tensor model from a nonlinear MATLAB Simulink file was proposed. Also tensor decomposition methods were shown to be applicable for rank reduction of tensor systems. All constructed models were shown to be beneficial for controller design.

With the help of a model it was possible to find the best Boolean controller, steering the switching of three boilers in a heating system supplying three buildings. This was done using a method where algebraic normal forms were used to model Boolean functions, which convert the discrete optimization problem by relaxation into a continuous one. The controller was implemented in the building and performs well.

Nonlinear model predictive control was used in a heating system with a hot water storage tank. The proposed controller uses general nonlinear model predictive control and is able to steer the charging and discharging of the tank with the benefit, that the building could be supplied with heat by only one of the two installed boilers. The underlying cost function also takes heat losses into account such that an energy efficient operation is possible.

The above methods do not guarantee stability of the closed loop system. A predictive controller which can guarantee stability was proposed for a heating system with three boilers. The model is given as a piecewise affine model and for the control the Multi-Parametric Toolbox is used. Unfortunately, this setting suffers from the curse of dimensionality, which is known for multi-parametric programs, leading to a very abstract model and very short time horizons.

A further approach to guarantee stability was done for multilinear models. Using ellipsoidal domains of attraction Lyapunov functions are used to guarantee the stability in this domain. As an application example a controller for a heating system is designed.

Finally for an approach of feedback linearization, an application example is given. The approach uses the algebra of functions, which is briefly introduced.

Each approach was shown to have its benefits, such that the effort of building the model was justified. In general, models of heating systems can be used to ensure the comfort in the building, reduce material fatigue of the heating system components or save energy.

5.2 Outlook

The most promising fields of future work are on the scientific level the feedback linearization and on a practical level the implementation of the nonlinear model predictive controller.

The feedback linearization application example shows that for a small example the existing approach can be used. For more sophisticated examples the steps in the approach get more complicated, such that it would be beneficial to adapt the operations to work with multilinear systems instead of the general nonlinear case.

The proposed nonlinear predictive controller was not implemented in the building. This is because on the one hand not all steering signals are accessible without loss of warranty of the manufacturers and on the other hand because of missing hardware on-site. The implementation would bring further insight into the plant operation.

Around the topic of heating systems a lot of research fields are active. One might think of model-based fault diagnosis, distributed controller design, etc.

References

[1] H. Ziesing. Anwendungsbilanzen für die Endenergiesektoren in Deutschland in den Jahren 2011 und 2012 mit Zeitreihen von 2008 bis 2012. www.bmwi.de, 2014. Date of download: 16.07.2014.

[2] E. Sewe, A. Harmsen, G. Pangalos, and G. Lichtenberg. Umsetzung eines neuen Konzepts zur Mehrkesselregelung mit Durchflusssensoren. *HLH Lüftung/Klima - Heizung/Sanitär - Gebäudetechnik*, 2012(1):37–42, 2012.

[3] T.S. Nouidui, K.P.W. Zuo, and M. Wetter. Validation and Application of the Room Model of the Modelica Buildings Library. In *9th International Modelica Conference*, Munich, 2012.

[4] M. Wetter. Multizone Building Model for Thermal Building Simulation in Modelica. *5th International Modelica Conference*, 2006.

[5] K. Kruppa, G. Pangalos, and G. Lichtenberg. Multilinear Approximation of Nonlinear State Space Models. In *IFAC World Congress*, Cape Town, 2014.

[6] M. Pcolka, E. Zacekova, S. Celikovsky, R. Robinett, and M. Sebek. From Linear to Nonlinear Model Predictive Control of a Building. In *IFAC World Congress*, Cape Town, 2014.

[7] A. Parisio, D. Varagnolo, M. Molinari, G. Pattarello, L. Fabietti, and K. H. Johansson. Implementation of a Scenario-Based MPC for HVAC Systems: An Experimental Case Study. In *IFAC World Congress*, Cape Town, 2014.

[8] D. Liberzon. *Switching in Systems and Control*. Systems & Control: Foundations & Applications. Birkhäuser, Basel, 2003.

[9] E. Sontag. Nonlinear Regulation: The Piecewise Linear Approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.

[10] B. Potočnik, G. Mušič, and B. Zupančič. A New Technique for Translating Discrete Hybrid Automata into Piecewise Affine Systems. *Mathematical and Computer Modelling of Dynamical Systems*, 10(1):41–57, 2004.

[11] F. Torrisi and A. Bemporad. HYSDEL - A Tool for Generating Computational Hybrid Models for Analysis and Synthesis Problems. *IEEE Transactions on Control Systems Technology*, 12(2), 2004.

[12] A. Bemporad and M. Morari. Control of Systems Integrating Logic, Dynamics and Constraints. *Automatica*, 35:407–427, 1999.

[13] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085 – 1091, 2001.

[14] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and Controllability of Piecewise Affine and Hybrid Systems. *IEEE Transactions on Automatic Control*, 45(10):1864–1876, 2000.

[15] G. Lichtenberg. Hybrid Tensor Systems. Habilitation, Hamburg University of Technology, 2011.

[16] T. Kolda and B. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009.

[17] A. Cichocki, R. Zdunek, A. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations*. Wiley, Chichester, 2009.

[18] G. Pangalos, A. Eichler, and G. Lichtenberg. Tensor Systems: Multilinear Modeling and Applications. In *3rd Int. Conference on Simulation and Modeling Methodologies, Technologies and Applications*, Reykjavik, 2013.

[19] I. Zhegalkin. Arithmetics of symbolic logic. *Mat. Sb.*, 35(3-4):311–377, 1928.

[20] M. Schmidt. Modellierung des Wärmetauschers des Bezirksregierungsgebäudes in Düsseldorf. Studienarbeit, Hamburg University of Technology, Institute of Control Systems, 2011.

[21] T. Spengler. Modellbildung einer Fernwärmeheizungsanlage und Identifikation der thermischen und hydraulischen Parameter. Bachelor thesis, Hamburg University of Technology, Institute of Control Systems, 2011.

[22] K. Kruppa. Dokumentation Matlab Heatlib. Technical report, Hamburg University of Applied Sciences, Faculty of Life Sciences, 2014. Manual.

[23] J. Seifert. Entwicklung einer nichtlinearen modellprädiktiven Heizungsregelung zur energetisch optimierten Nutzung eines Warmwasserspeichers. Master's thesis, Hamburg University of Technology, Institute of Control Systems, 2012.

[24] B.J. Newton. Modeling of solar storage tanks. Master's thesis, University of Wisconsin–Madison, 1995.

[25] F. Rößner. Prädiktiver Reglerentwurf mit einem stückweise affinen Modell einer hybriden Heizungsanlage mithilfe der Multi-Parametric Toolbox. Diplomarbeit, Hamburg University of Technology, Institute of Control Systems, 2012.

[26] G. Pangalos, A. Eichler, and G. Lichtenberg. Hybrid Multilinear Modeling and Applications. In M.S. Obaidat, S. Koziel, J. Kacprzyk, L. Leifsson, and T. Ören, editors, *Simulation and Modeling Methodologies, Technologies and Applications*. Springer Cham Heidelberg New York Dordrecht London, 2015.

[27] K. Kruppa. Multilineare Approximation nichtlinearer Zustandsraummodelle. Master's thesis, Hamburg University of Technology, Institute of Control Systems, 2013.

[28] R. Schaback and H. Werner. *Numerische Mathematik*. Springer-Verlag Berlin Heidelberg, 4. edition, 1992.

[29] D. Kincaid and E. Cheney. *Numerical analysis: Mathematics of scientific computing*. Brooks/Cole Publ., 2nd edition, 1996.

[30] D. Ackleh. *Classical and modern numerical analysis: theory, methods and practice*. CRC Press, 2010.

[31] G. Phillips and P. Taylor. *Theory and Applications of Numerical Analysis*. Academic Press, 2nd edition edition, 1996.

[32] B. Bader and T. Kolda. MATLAB Tensor Toolbox Version 2.5. Available online, 2012.

[33] G. Pangalos and G. Lichtenberg. Approach to Boolean Controller Design by Algebraic Relaxation for Heating Systems. In *4th IFAC Conference on Analysis and Design of Hybrid Systems*, Eindhoven, 2012.

[34] D. Delchamps. Stabilizing a linear systems with quantized state feedback. *IEEE Transactions on Automatic Control*, 35(8):916–924, 1990.

[35] A. Bemporad, F. Borrelli, and M. Morari. Optimal Controllers for Hybrid Systems: Stability and Piecewise Linear Explicit Form. In *Proceedings of the 39th IEEE Conference on Decision and Control*, 2000.

[36] P.P.H.H. Philips, W.P.M.H. Heemels, H.A. Preisig, and P.P.J. van den Bosch. Control of Quantised Systems Based on Discrete-Event Models. *International Journal of Control*, 76(3):277–294, 2003.

[37] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[38] S. Faisal, G. Lichtenberg, S. Trump, and S. Attinger. Structural Properties of Continuous Representations of Boolean Functions for Gene Network Modelling. *Automatica*, 46(12):2047–2052, 2010.

[39] D. Franke. Boolean Controller Design for Discrete Systems via Eigenvalue Assignment. In *8th International Symposium on System, Modelling, Control*, Zakopane, 1995.

[40] D. Bochmann and B. Steinbach. *Logikentwurf mit XBOOLE*. Verlag Technik, Berlin, 1991.

[41] G. Lichtenberg and A. Eichler. Multilinear Algebraic Boolean Modelling with Tensor Decompositions Techniques. In *18th IFAC World Congress*, page TuC01.2, Milano, 2011.

[42] U. Friedrich. Heizsysteme in Bürogebäuden optimal betreiben. *BINE Informationsdienst*, 5, 2013.

[43] J. Maciejowski. *Predictive Control with Constraints*. Pearson Education Limited 2002, 2002.

[44] R. Findeisen and F. Allgöwer. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, volume 11, pages 119–141, 2002.

[45] M. Morari and J.H. Lee. Model predictive control: Past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.

[46] J. B. Rawlings, D. Angeli, and C. N. Bates. Fundamentals of Economic Model Predictive Control. *51st IEEE Conference on Decision and Control*, 2012.

[47] H.M. Paiva and R.K.H. Galvão. Simulation of dynamic systems with output saturation. *Education, IEEE Transactions on*, 47(3):385–388, 2004.

[48] P. Mhaskar, A. Gani and P. D. Christofides. Fault-tolerant control of nonlinear processes: Performance-based reconfiguration and robustness. *Int. J. Robust Nonlinear Control*, 16:91–111, 2006.

[49] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. http://control.ee.ethz.ch/∼mpt/.

[50] M. Kvasnica, P. Grieder, M. Baotić, and F.J. Christophersen. Multi-Parametric Toolbox (MPT). Technical report, Institut für Automatik, ETH - Swiss Federal Institute of Technology, Zürich, 2006. Manual.

[51] M. Baotić, F.J. Christophersen, and M. Morari. A new Algorithm for Constrained Finite Time Optimal Control of Hybrid Systems with a Linear Performance Index. In *Proc. of the European Control Conference*, September 2003.

[52] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari. An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In *American Control Conference, 2003. Proceedings of the 2003*, volume 6, pages 4717–4722 vol.6, June 2003.

[53] H. Khalil. *Nonlinear Systems.* Prentice-Hall, Upper Saddle River, NJ, 3 edition, 2000.

[54] S. Harder. Methode zum Reglerentwurf für multilineare Systeme durch Einzugsbereichsanalyse mittels quadratischer Ljapunowfunktionen. Master's thesis, Hamburg University of Technology, Institute of Control Systems, 2013.

[55] A. Vannelli and M. Vidyasagar. Maximal lyapunov functions and domains of attraction for autonomous nonlinear systems. Automatica, Volume 21, Issue 1, January 1985, Pages 69-80.

[56] A. Chakraborty, P. Seilera and G. J. Balas. Nonlinear Region of Attraction Analysis for Flight Control Verification and Validation. Control Engineering Practice, Volume 19, Issue 4, April 2011, Pages 335–345.

[57] B. Tibken and K. Dilaver. Computation of subsets of the domain of attraction for polynomial systems. *IEEE Transactions on Nuclear Science*, 2(47):389–402, 2002.

[58] H. Ehlich and K. Zeller. Schwankung von Polynomen zwischen Gitterpunkten. *Mathematische Zeitschrift*, 86:41–44, 1964.

[59] U. Gärtel. *Fehlerabschätzungen für vektorwertige Randwertaufgaben zweiter Ordnung, insbesondere für Probleme aus der chemischen Reaktions - Diffusions - Theorie.* PhD thesis, Universität zu Köln, 1987.

[60] I.N. Bronshtein, editor. *Handbook of mathematics.* Springer, Berlin, 5. edition, 2007.

[61] A. Isidori. *Nonlinear Control Systems: An introduction.* Lecture Notes in Control and Information Sciences. Springer Verlag Berlin, 1995.

[62] J. W. Grizzle. *Feedback Linearization of Discrete-Time Systems.* Analysis and Optimization of Systems. Volume 83 of the series Lecture Notes in Control and Information Sciences pp 273-281, 1986.

[63] H. Nijmeijer and A. v. d. Schaft. *Nonlinear Dynamical Control Systems*. Springer Verlag New York, 3 edition, 1996.

[64] Ü. Kotta, M. Tõnso, A.Y. Shumsky, and A.N. Zhirabok. Feedback linearization and lattice theory. *Systems & Control Letters*, 62(3):248 – 255, 2013.

[65] V. Kaparin, Ü. Kotta, A.Y. Shumsky, M. Tõnso, and A.N. Zhirabok. Implementation of the Tools of Functions' Algebra: First Steps. In *7th Vienna International Conference on Mathematical Modelling*, pages 1231–1236, 2012.

[66] A.D. Polyanin, V.F. Zaitsev, and A. Moussiaux. *Handbook of first order partial differential equations*. Taylor & Francis, 2002.

[67] A. Shumsky and A. Zhirabok. Unified Approach to the Problem of Full Decoupling via Output Feedback. *European Journal of Control*, 16(4):313 – 325, 2010.

[68] J. Belikov, V. Kaparin, Ü. Kotta, and M. Tõnso. Nlcontrol website, http://www.nlcontrol.ioc.ee/webmathematica/nlcontrol/funcalg/discrete/linearization.html. Date of usage: 16.7.2014.

[69] A. Zhirabok and A. Shumsky. *The Algebraic Methods for Analysis of Nonlinear Dynamic Systems*. Dalnauka, 2008. In Russian.

[70] G. Pangalos. Reglerentwurf für hybride Heizungssysteme mittels algebraischer Relaxierung Boole'scher Optimierungsprobleme. Diplomarbeit, Hamburg University of Technology, Institute of Control Systems, 2011.

A Mathematical operations

A.1 Outer product

The definition of the outer product of two tensors is taken from [17].

Definition A.1 *Let* $\mathsf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ *and* $\mathsf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_m}$ *be two tensors with the given dimensions. The outer product of these tensors is given as*

$$\mathsf{Z} = \mathsf{X} \circ \mathsf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times J_1 \times J_2 \times \cdots \times J_m} \tag{A.1}$$

where Z *is a tensor of order* $n + m$*. The elements of* Z *are given as*

$$z_{i_1,i_2,\ldots,i_n,j_1,j_2,\ldots,j_m} = x_{i_1,i_2,\ldots,i_n} y_{j_1,j_2,\ldots,j_m}. \tag{A.2}$$

A.2 Kronecker product

Definition A.2 *The Kronecker product of two matrices* $\mathbf{X} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{Y} \in \mathbb{R}^{p \times q}$ *with the elements* $x_{i,j}$*,* $i = 1, \ldots, m$ *and* $j = 1, \ldots, n$ *and* $y_{k,l}$*,* $k = 1, \ldots, p$ *and* $j = 1, \ldots, q$ *respectively is*

$$\mathbf{X} \otimes \mathbf{Y} = \begin{pmatrix} x_{1,1}\mathbf{Y} & x_{1,2}\mathbf{Y} & \ldots & x_{1,n}\mathbf{Y} \\ x_{2,1}\mathbf{Y} & x_{2,2}\mathbf{Y} & \ldots & x_{2,n}\mathbf{Y} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1}\mathbf{Y} & x_{m,2}\mathbf{Y} & \ldots & x_{m,n}\mathbf{Y} \end{pmatrix} \in \mathbb{R}^{mp \times nq}. \tag{A.3}$$

A.3 Hadamard product

Definition A.3 *The Hadamard product of two matrices* $\mathbf{X} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{Y} \in \mathbb{R}^{m \times n}$ *with the elements* $x_{i,j}$ *and* $y_{i,j}$*,* $i = 1, \ldots, m$ *and* $j = 1, \ldots, n$*, respectively is*

$$\mathbf{X} \circledast \mathbf{Y} = \begin{pmatrix} x_{1,1}y_{1,1} & x_{1,2}y_{1,2} & \ldots & x_{1,n}y_{1,n} \\ x_{2,1}y_{2,1} & x_{2,2}y_{2,2} & \ldots & x_{2,n}y_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1}y_{m,1} & x_{m,2}y_{m,2} & \ldots & x_{m,n}y_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n}. \tag{A.4}$$

B Tensor model

The factor matrices for the supply temperature, the return temperature, the building temperature, the ambient temperature (day oscillation, derivative of the day oscillation, year oscillation and derivative of the year oscillation), the lower bound on the room temperature, the upper bound on the room temperature, the upper threshold on the supply temperature, the lower threshold on the supply temperature and the thermal power input in this order of the heating system are

$$\mathbf{F}_{T_s}=\begin{pmatrix} 0&1&0&1&0&1&0&1&0&1&0&1&1&0&1&1&0&1&0&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&1&1&1&1\\ 1&0&1&0&1&0&1&0&1&0&1&0&0&1&0&0&1&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{T_r}=\begin{pmatrix} 1&0&1&0&1&0&1&0&1&0&1&1&0&1&1&0&1&0&1&0&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&1&0&1&0&1&0&1&0&1&0&0&1&0&0&1&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{T_b}=\begin{pmatrix} 1&1&1&1&0&0&1&1&0&0&1&1&1&0&1&1&0&0&1&1&0&0&0&1&1&1&1&1&1&1&1&1&1&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&1&1&0&0&1&1&0&0&0&1&0&0&1&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{T_{a,1}}=\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&0&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{T_{a,2}}=\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{T_{a,3}}=\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{T_{a,4}}=\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{X_l}=\begin{pmatrix} 1&1&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{X_u}=\begin{pmatrix} 1&1&1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{F}_{H_l}=\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \end{pmatrix},$$

$$\mathbf{F}_{H_u}=\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{pmatrix},$$

$$\mathbf{F}_{B}=\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \end{pmatrix}.$$

The factor matrix F_Φ is

$$F_\Phi = \begin{pmatrix}
1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1
\end{pmatrix} .$$

And the parameter vector is

$$\begin{aligned}
\boldsymbol{\lambda}_f = \Bigg[& 1 - \frac{t_s(\dot V_a + \dot V_m)}{V_b},\ \frac{t_s(\dot V_a + \dot V_m)}{V_b}\frac{T_1}{T_2},\ \frac{t_s \dot V_a(\Delta T_b - T_{br})}{\Delta T_b V_b}\frac{1}{T_8},\ -\frac{t_s \dot V_a(\Delta T_b - T_{br})}{\Delta T_b V_b}\frac{T_1}{T_2 T_8},\ \frac{t_s \dot V_a}{\Delta T_b V_b}\frac{1}{T_3 T_8},\ -\frac{t_s \dot V_a}{\Delta T_b V_b}\frac{T_1}{T_2 T_3 T_8}, \\
& \frac{t_s \dot V_a(\Delta T_b + T_{br})}{\Delta T_b V_b}\frac{1}{T_9},\ -\frac{t_s \dot V_a(\Delta T_b + T_{br})}{\Delta T_b V_b}\frac{T_1}{T_2 T_9},\ -\frac{(t_s \dot V_a}{\Delta T_b V_b}\frac{1}{T_3 T_9},\ \frac{t_s \dot V_a}{\Delta T_b V_b}\frac{T_1}{T_2 T_3 T_9},\ \frac{t_s P_n}{V_b c\rho}\frac{T_1}{T_{12}},\ \frac{t_s(\dot V_a + \dot V_m)}{V_c}\frac{T_2}{T_1}, \\
& 1 - \frac{t_s(k_{r,b} + \dot V_a c\rho + \dot V_m c\rho)}{V_c c\rho},\ \frac{t_s k_{r,b}}{V_c c\rho}\frac{T_2}{T_3},\ -\frac{t_s \dot V_a(\Delta T_b - T_{br})}{\Delta T_b V_c}\frac{T_2}{T_1 T_8},\ \frac{t_s \dot V_a(\Delta T_b - T_{br})}{\Delta T_b V_c}\frac{1}{T_8},\ -\frac{t_s \dot V_a}{\Delta T_b V_c}\frac{T_2}{T_1 T_3 T_8},\ \frac{t_s \dot V_a}{\Delta T_b V_c}\frac{1}{T_3 T_8}, \\
& -\frac{t_s \dot V_a(\Delta T_b + T_{br})}{\Delta T_b V_c}\frac{T_2}{T_1 T_9},\ \frac{t_s \dot V_a(\Delta T_b + T_{br})}{\Delta T_b V_c}\frac{1}{T_9},\ \frac{t_s \dot V_a}{\Delta T_b V_c}\frac{T_2}{T_1 T_3 T_9},\ -\frac{t_s \dot V_a}{\Delta T_b V_c}\frac{1}{T_3 T_9},\ 1 - \frac{t_s k_{r,b}}{c_b} - t_s\frac{k_{b,a}}{c_b},\ t_s\frac{k_{r,b}}{c_b}\frac{T_3}{T_2},\ t_s\frac{k_{b,a}}{c_b}\frac{T_3}{T_4}, \\
& t_s\frac{k_{b,a}}{c_b}\frac{T_3}{T_6},\ t_s\frac{k_{b,a}}{c_b}T_{a,m}T_3,\ 1,\ \frac{T_4}{T_5},\ -5.2885\ 10^{-9}\frac{T_5}{T_4},\ 1,\ 1,\ \frac{T_6}{T_7},\ -3.9696\ 10^{-14}\frac{T_7}{T_8},\ 1,\ s\frac{T_8}{T_3},\ (-T_{br}s + \Delta T_b s + 0.5)T_8,\ s\frac{T_9}{T_3}, \\
& (-T_{br}s - \Delta T_b s + 0.5)T_9,\ -s\frac{T_{10}}{T_1},\ (H_{lb}s + 0.5)T_{10},\ s\frac{T_{11}}{T_1},\ (-H_{ub}s + 0.5)T_{11},\ 1,\ \frac{T_{12}}{T_{10}},\ -\frac{T_{12}}{T_{11}}\Bigg]^T
\end{aligned}$$

where $T = [0.0014\ \ 0.0015\ \ 0.0017\ \ 0.25\ \ 2500\ \ 0.04\ \ 150e6\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1]$ is a scaling vector with elements T_i, $i = 1, \ldots, 12$, $H_{ub} = 95\,°\mathrm{C}$ and $H_{lb} = 75\,°\mathrm{C}$ are the thresholds of the bang bang controller and $s = 0.01$ is a scaling factor for the quantizer. The other parameters can be found in table 3.1 and 3.2. The next state tensor is

$$\mathsf{F} = [\mathbf{F}_B, \mathbf{F}_{H_u}, \mathbf{F}_{H_l}, \mathbf{F}_{X_u}, \mathbf{F}_{X_l}, \mathbf{F}_{a,4}, \mathbf{F}_{a,3}, \mathbf{F}_{a,2}, \mathbf{F}_{a,1}, \mathbf{F}_{T_b}, \mathbf{F}_{T_r}, \mathbf{F}_{T_s}, \mathbf{F}_\Phi] \cdot \lambda_f \ . \qquad \text{(B.1)}$$

C Model of the heating system of the state office building in Düsseldorf

The MATLAB Simulink model of the heating system of the state office building for nature, environment and consumerism in Düsseldorf is described in this section, see [70]. The inputs of the model are the steering signals for the boilers, the outputs are the quantized measurements of the flow rate. The disturbance inputs are the heat demand, the ambient temperature and the flow rate. Figure C.1 shows the Simulink model of the heating system

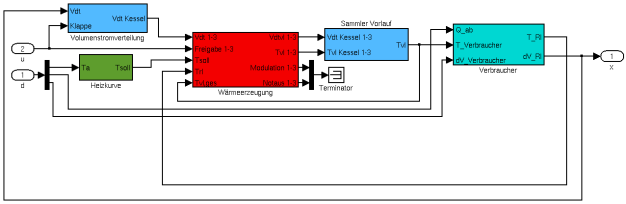


Figure C.1: Heating system of the state office building.

The heat generation unit consists of the three boilers with burners and a PI controller, used to calculate the steering signal for the power to be produced by the boilers, see figure C.2.

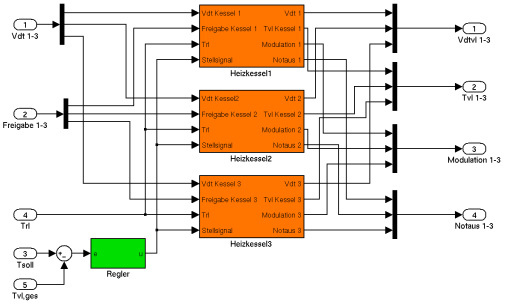


Figure C.2: Heat generation unit.

The PI controller is given by its transfer function as

$$G_{PI}(s) = 0.1\left(\frac{1}{10s} + 1\right) . \tag{C.1}$$

Additionally the integrator *windup* is prevented by a standard saturation.

The controller uses the overall supply temperature to control the power signal for all boiler simultaneously by tracking the reference. Each boiler is modeled by the thermal balances given in 3.1.1. In addition a flushing of the boiler is modeled by preventing the boiler to be switched on within the first 5 minutes after the last switching. The first and second boiler are identically modeled but with different volumes, nominal power and minimal powers. The third boiler is modeled as a two step device. It can be switched off, running with half its nominal power or running with its full power. This additional constraint is modeled with the device depicted in figure C.4. The third boiler is switched on with half its power, if the boilers steering signals are above $\varphi = 0.8$. If the signal reaches $\varphi = 0.95$ the full power is demanded. The lower limits of the steering signals are $\varphi = 0.6$ for the boiler to go back to half its power and $\varphi = 0.25$ to switch off entirely.

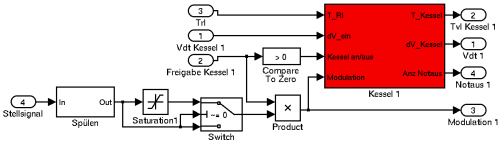


Figure C.3: Boiler 1

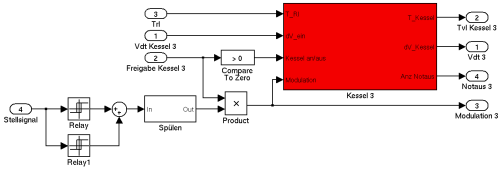


Figure C.4: Boiler 3

The supply temperature T_s is assembled by the three supply temperatures T_{b1}, T_{b2}, and T_{b3} as given in equation The supply temperature is used to control the steering signals of the boilers and to calculate the return temperature in the consumer. The consumer determines the return temperature T_r in dependence of the supply temperature, the heat demand and the flow rate, see (3.26).

After the consumer, the flow rate is divided in up to three parts as illustrated in example 4.2

The heating curve defines the reference supply temperature in dependence of the ambient temperature. In this model the curve is defined as in given in figure 3.18.

D Cost function for the nonlinear model predictive control example

The tracking error is given as

$$\hat{y}_1(k) = \begin{cases} (T_{s,c}(k) - T_{max})^2 & \text{if } T_{s,c}(k) > T_{max} \text{ and } \dot{V}_c > \dot{V}_0 \\ (T_{s,max,r}(k) - T_{s,c}(k))^2 & \text{if } T_{s,c}(k) < T_{\mathrm{s,max,r}}(k) \text{ and } \dot{V}_c > \dot{V}_0 \\ 0 & \text{else,} \end{cases} \tag{D.1}$$

where $T_{s,max,r}$ is the maximal reference temperature of the consumer circuits, and T_{max} is the maximal temperature of the system. The cost term is neglected for very small flow rates $\dot{V}_0$, the water is then assumed to have stopped moving. The maximal temperatures of the boilers are penalized with

$$\hat{y}_{1+i}(k) = \begin{cases} (T_{b,i}(k) - T_{\max})^2 & \text{if } T_{\mathrm{b,i}}(k) > T_{\max} \\ 0 & \text{sonst} \end{cases} \tag{D.2}$$

for $i = 1, 2$, with T_{max} being again the maximal temperature of the system. The heat losses of the tank series are depending on the temperature

of the tank and the ambiance. In detail all losses are given as

$$\dot{Q}_l = U\pi\frac{D^2}{4}((T_1 - T_a) + (T_n - T_a)) + \sum_{i=1}^{n} U\pi D h_i (T_i - T_a), \quad \text{(D.3)}$$

where $T_i, i = 1, \ldots, n$ are the temperatures of the layer in the tank, U is the heat transfer coefficient, D the diameter of the tank and h_i the height of layer i. The first two terms take the heat losses to the top and bottom into account and the third term the heat losses to the side. Rearranging and dropping the constant terms gives

$$\hat{y}_4 = \frac{D}{4H}(T_1 + T_n) - \left(1 + \frac{D}{2H}\right) T_a + \frac{1}{H}\sum_{i=1}^{n} h_i T_i \quad \text{(D.4)}$$

for the fourth cost function term. Operating points can be optimal, because the boilers are condensing boilers and therefore the efficiency is depending on the mean temperature $T_{m,i} = \frac{T_{b,i}+T_{r,g}}{2}$ and the steering signal of the boiler φ_i and can be given as

$$\eta_i(T_{m,i}, \varphi_i) = a_0 - a_1 T_{m,i} - a_2 \varphi_i. \quad \text{(D.5)}$$

The coefficients a_0,a_1, and a_2 are given by

$$\eta_{max} = \eta(T_{m,min}, \varphi_{min}) \quad \text{(D.6)}$$
$$\eta_{min} = \eta(T_{m,max}, \varphi_{max}) \quad \text{(D.7)}$$
$$\Delta\eta = \eta_{max} - \eta(T_{m,max}, \varphi_{min})\,. \quad \text{(D.8)}$$

with $T_{m,min} = 30\,^{\circ}\text{C}$, $T_{m,min} = 70\,^{\circ}\text{C}$, $\varphi_{min} = 0.25$, $\varphi_{max} = 1$ and $\Delta\eta = 0.05$, for both boilers. The minimal and maximal efficiency is given by the interval $[\eta_{min}\ \eta_{max}] = [0.98\ 1.1]$. The terms is the cost function are given by

$$\hat{y}_{4+i}(k) = \begin{cases} \frac{\eta_{max} - \eta(T_{m,i}, \varphi_i)}{\eta_{max} - \eta_{min}} & \text{if } \varphi_i \geq \varphi_{min} \\ 0 & \text{else,} \end{cases} \quad \text{(D.9)}$$

for $i = 1, 2$. Switching of the boilers is penalized by the term

$$y_{6+i}(k) = \begin{cases} 1 & \text{if } \varphi_i(k) < \varphi_{min} \text{ and } \varphi_i(k-1) \geq \varphi_{min} \\ 1 & \text{if } \varphi_i(k) \geq \varphi_{min} \text{ and } \varphi_i(k-1) < \varphi_{min} \\ 0 & \text{else,} \end{cases} \quad \text{(D.10)}$$

for $i = 1, 2$. And finally the changes in the steering signals of the boilers and the pumps are avoided due to the terms

$$y_{8+i}(k) = (\varphi_i(k) - \varphi_i(k-1))^2 \tag{D.11}$$

for $i = 1, 2$ and

$$y_{10+i}(k) = 3600^2 \cdot (\dot{V}_{b,i}(k) - \dot{V}_{b,i}(k-1))^2, \tag{D.12}$$

for $i = 1, 2$.