Data-Driven Learning and Model Predictive Control for Heating Systems

Vorgelegt im Promotionsausschuss der
HafenCity Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

Dissertation

von
Björn Lautenschlager

aus
Henstedt-Ulzburg

2019

1. Gutachter: Univ.-Prof. Dr.-Ing. Harald Sternberg
(HafenCity Universität)
2. Gutachter: Prof. Dr.-Ing. habil. Gerwald Lichtenberg
(Hochschule für angewandte Wissenschaften)

Vorsitzende des Prüfungsausschusses: Prof. Dr.-Ing. Anette Eicker
(HafenCity Universität)
Zusätzlicher Professor im Prüfungsausschuss: Prof. Dr.-Ing Jochen Schiewe
(HafenCity Universität)

Tag der mündlichen Prüfung: 07.05.2019

Summary

Up to 40% of the total energy consumption in Germany is used in buildings. Up to 80 % of the total energy consumed by buildings is used for heating the rooms and supplying the buildings with warm water. This means that the heating system of a building consumes the larges part of the total energy demand of the building. The high energy consumption in the building sector results in a large potential for energy savings. This potential can be utilized by optimizing the building automation systems, e.g., using new control methods for the heating systems.

This thesis deals with the question how stored measurement data, which are often used for monitoring purposes only, can be used for optimizing the control strategies. The developed controllers will be applied to a heating system and the results will be evaluated. Model predictive methods are a suitable choice for the control of systems with delay times, such as heating systems. The model predictive controller (MPC) uses a model to calculate the future behavior of the plant, so that the controller can react early to changes in the environmental conditions. For heating systems, the weather forecast is used for the MPC. An MPC solves an optimization problem by minimizing a cost function with respect to the future behavior of the plant, represented by the model. One aspect of this thesis is the modeling of heating systems by using stored measurement data.

First, two control strategies for today's applications will be introduced and applied to a heating system example. On the one hand the classical proportional-integral (PI) controller and on the other hand a linear MPC. The PI controller is a standard controller for many applications and also for heating systems. The simulation results will be used in the following for the comparison with other control methods. Both controllers use a reference trajectory. Finding a suitable reference can be difficult and time consuming.

An economic model predictive controller (EMPC) will be introduced next. The EMPC uses also a model of the system and optimizes a cost function, but without using any reference signals. That means, that no reference has to be defined and the optimization is solved with constraints. The constraints are defined by the system requirements, such as the comfort conditions of the room temperature for heating systems. Additionally, time dependent constraints were introduced and the possibility to use discrete input signals. The discrete input signals change the optimization problem to a mixed-integer optimization problem, which leads to more complex and time consuming computations. The EMPC was applied to a heating system and the simulation results were compared with the results of the linear MPC. A real-time implementation of the EMPC for a heating system of an office building was stated and the results are evaluated.

A linear MPC uses a linear model of the system. If the system description by a linear model is not accurate anymore, the modeling of the system leads, in general, to a nonlinear model. A subclass of the general nonlinear model class are the multilinear time-invariant (MTI) models. The structure of the MPC optimization problem was investigated with the assumption that an MTI model is used, with the focus on the convexity analysis of the optimization problem. It could be proven that the optimization problem is convex for a subclass of MTI systems and a restricted prediction horizon.

An iterative learning controller (ILC) is another approach to use measurement data for control. An ILC uses the stored data for the calculation of the input signal of the next iteration. A data-driven ILC was introduced as well as an element selector for choosing a stored data set. For the element selector an optimization problem was formulated. A combined data-driven learning MPC was presented and tested by an application example. This combined control approach was implemented for a prototype heating system. The data-driven ILC collects and stores the data of all historic iterations. The historic data was stored in a tensor structure and the canonical-polyadic (CP) decomposition method was applied to reduce the storage demand and enlarge the possible implementation platforms to hardware with limited storage capacities. It could be shown that the calculation of the similarity criterion for the choice of a historic data set can be performed with the CP decomposed tensor, which leads to a significant reduction of the storage demand.

In this thesis, measurement data was used for the modeling process for predictive control methods to take the future behavior of the plant into account and for iterative learning control, which uses the measurement data to consider the past behavior of the system. The developed control methods were applied to a heating system and compared to the results with a standard controller.

Zusammenfassung

Im Gebäudebereich werden heutzutage schon rund 40 % der gesamten in Deutschland verbrauchten Energie, eingesetzt. Bis zu 80 % der verbrauchten Energie im Gebäudebereich wird für die Erwärmung des Gebäuden und die Warmwasserbereitung verwendet. Das bedeutet, dass der Heizungsanlage im Gebäude auf Grund des hohen Energieverbrauchs eine besondere Bedeutung zukommt. Durch den hohen Energieverbrauch ergibt sich auch ein entsprechendes Einsparpotential, welches unter anderem durch eine verbesserte Betriebsführung, wie z.B. durch die Verwendung neuartiger Regelungskonzepte, ausgenutzt werden kann.

Diese Arbeit beschäftigt sich mit der Frage, wie gesammelte und gespeicherte Messdaten, welche oftmals nur für Monitoring und Auswertungszwecke verwendet werden, für die Optimierung von Regelungen genutzt werden kann. Die entwickelten Regelungsstrategien werden am Beispiel von Heizungsanlagen getestet und die Ergebnisse ausgewertet. Modellprädiktive Verfahren haben sich für die Regelung von Systemen mit größeren Totzeiten als besonders geeignet erwiesen. Zu solchen System gehören auch Heizungsanlagen, wobei für die modellprädiktive Regelung (model predicitve control - MPC) die Wettervorhersagen und insbesondere die Vorhersage der Außentemperatur berücksichtigt wird. Durch die Nutzung eines Modells für die Regelung kann das zukünftige Anlagenverhalten vorausberechnet werden und so auf Änderungen in den Umgebungsbedingungen frühzeitig reagiert werden. Dabei löst der MPC ein Optimierungsproblem, welches eine Kostenfunktion minimiert unter Berücksichtigung des Anlagenverhaltens, welches durch das Model wiedergegeben wird. Aus diesem Grund ist ein Aspekt der Nutzung von Daten die Modellbildung für prädiktive Regelungsstrategien.

Anhand eines Beispiels einer Heizungsanlage werden zwei Regelungsstrategien vorgestellt, wie sie heutzutage in der praktischen Anwendung verwendet werden. Auf der einen Seite der klassische proportionale-integrale (PI) Regler und auf der anderen Seite ein linearer MPC. Dabei gilt der PI-Regler als ein Standardregler, der in vielen Anwendungsbereichen genutzt wird, auch im Bereich der Heizungssysteme. Die Ergebnisse werden für Vergleichszwecke mit den weiteren Regelungsstrategien genutzt. Sowohl der PI-Regler, als auch der lineare MPC verwenden für die Regelung eine Referenz. Eine geeignete Referenz zu finden kann mitunter aufwendig und schwierig sein.

Ein sogenannter EMPC (economic model predictive controller), welcher ebenfalls ein Modell verwendet und ein Optimierungsproblem löst, unterscheidet sich vom linearen MPC zum einen durch die Art der Kostenfunktion und zum anderen wird keine Referenz für die Regelung verwendet, wodurch die Wahl einer geeigneten Referenztrajektorie wegfällt. Stattdessen wird die Optimierung unter Randbedingungen ausgeführt welche durch die Anforderungen des Systems definiert werden. So können z.B. für Heizungsanlagen die Komfortbedingungen an die Raumtemperatur mit Hilfe der Randbedingungen definiert werden. Dieser Ansatz wird zum einen um zeitabhängige Randbedingungen erweitert und zum anderen um diskrete Stellsignale, wodurch die Anwendung eines EMPC auf Anlagen mit schaltenden Stellsignalen ermöglicht wird. Durch die Einführung von diskreten Stellsignalen wird das Optimierungsproblem zu einem gemischt-ganzzahligen Problem, wodurch die Lösung wesentlich komplexer und zeitaufwendiger wird. Der

EMPC wurde auf eine Heizungsanlage angewandt und die Simulationsergebnisse mit denen des linearen MPC verglichen. Anschließend wurde der EMPC in einer realen Heizungsanlage implementiert und das Echtzeitverhalten dieser Regelung untersucht.

Die vorgestellten prädiktiven Regelungen verwenden ein lineares Modell des Systems. Wenn die Beschreibung des Systemverhaltens durch ein lineares Modell nicht ausreichend genau ist, führt dies im Allgemeinen zu nichtlinearen Modellen. Eine Unterklasse der nichtlinearen Modelle bilden die multilinearen zeit-invarianten (multilinear time-invariant MTI) Modelle. Unter Verwendung von MTI Modellen wurde die Struktur des modellprädiktiven Optimierungsproblems untersucht. Dabei stand die Konvexitätsanalyse im Vordergrund und es konnte gezeigt werden, dass das Optimierungsproblem für Unterklassen der MTI Systeme und bestimmte Vorhersagehorizonte konvex ist.

Ein weiterer Ansatz, um Messdaten zu verwenden, ist die iterativ lernenden Regelung (iterative learning control - ILC). Dabei nutzt ein ILC die gespeicherten Daten, um das Stellsignal der nächsten Iteration zu berechnen. Es wurde ein datanbasierter ILC eingeführt und Auswahlkriterien für die Wahl eines gespeicherten Datensatzes definiert und in einem Optimierungsproblem zusammengefasst. Die Kombination eines datenbasierte ILC mit einem linearen MPC wurde vorgestellt und am Anwendungsbeispiel getestet. Der datenbasiert lernende MPC wurde in einem prototypischen Heizungssystem implementiert und erste Messdaten ausgewertet. Der datenbasierte ILC speichert die Daten von vergangen Iterationen. Um eine Anwendung eines solchen ILCs auch auf Plattformen mit wenig Speicherplatz zu ermöglichen, wurden die Messdaten in einer Tensorstruktur gespeichert und die kanonisch-polyadische (canonical polyadic - CP) Tensorzerlegung auf die Messdaten angewandt. Es konnte gezeigt werden, dass die Berechnungen des Ähnlichkeitskriteriums für die Auswahl eines historischen Datensatzes auf Basis des CP Tensors erfolgen kann und somit der benötigte Speicherbedarf um ein Vielfaches gesenkt werden konnte.

Das heißt, in dieser Arbeit wurden Messdaten zum einen für die Modellbildung für prädiktive Regelungen verwendet um das zukünftige Verhalten des Systems zu berücksichtigen und zum anderen für die iterativ lernende Regelung, welche aufgrund der verwendeten Messdaten das vergangene Verhalten des Systems berücksichtigt und in die Regelung mit einfließen lässt. Die entwickelten Regelungen wurden auf Heizungssysteme angewandt und die Ergebnisse mit denen der Standardregelungen verglichen.

Acknowledgment

At this point I want to say thank you to all the people who have supported me during the PhD and some people in particular.

First of all, I want to say thank you to you, Gerwald Lichtenberg, that you have given me the great opportunity doing my PhD and opening my mind for the research field of control theory and heating systems. Many thanks, for your great support during the whole time of the PhD, for all the helpful and very inspiring discussions and conversations about the research topic and also about many other things beside.

Thank you, Harald Sternberg for being my supervisor and for the opportunity to being part of the HafenCity University. Thank you, Annette Eicker and Jochen Schiewe for leading the examination board and for being part of the examination process.

Thank you, Kai Kruppa for all your help and discussions regarding the research. But also and not at least for your great company through out the project and beyond. That was a very enjoyable time. Thank you for all of that! Thank you, Dagmar Rokita and Carsten von Westarp for all the coffee rounds, conversations and for the relaxed atmosphere in Bergedorf. Thank you, I have had always enjoyed my time in Bergedorf!

Thank you, Michael Boeck and André Woflram for the great implementation cooperation and the user perspective. Thank you, Thorsten Müller and Erik Sewe for the successful way through the project and beside, it was a lot of fun for me and not just work.

I want to say thank you to all the people, my friends and colleagues who have supported me directly or indirectly during this time.

Finally, a very special thank you to my whole family and my girlfriend Sarah! Without your amazing support, your patience and your encouragement all of this would not have been possible. Thank you for always being there!

THANK YOU ALL, THIS WAS AN AMAZING JOURNEY!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An overview of the energy consumption of buildings related to the heat supply is given first. The basics of heating systems are stated next, before the research question and the state of research are introduced. The introduction ends with the presentation of the thesis structure.

1.1 Energy consumption of buildings

Energy efficiency and the associated potentials of energy savings have become an actual and omnipresent theme, due to the constantly increasing consumption of primary energy, the climate change, the limited resources of fossil energy carriers and the greenhouse gas emission. The German government has defined extensive goals for the reduction of the greenhouse gas emission. In 2050 the emission of greenhouse gases shall be reduced by 80 to 95 % compared to the emission in 1990 [1]. To reach the climat protection goal many different activities are defined. One is the reduction of the heat demand for buildings by optimizing the building envelop but also optimizing the building automation systems, such as control algorithms for heating systems. The energy consumption of buildings has increased over 40% of the total energy consumption in Germany (2011), which shows the large possible savings [1].

In the years 2013 to 2016, 55 % for commercial buildings and even 80 % for residential buildings of the total energy consumption for buildings was used for heating the rooms and supplying the buildings with warm water [79]. An energy waste of 5 % up to 30 % for the heat supply of buildings due to badly tuned controllers is stated in [61]. The presented numbers are related to the energy consumption in Germany. But also for other developed countries these numbers are in the same region [63].

The evaluation of the energy consumption for heating systems on the on hand and the potential for energy savings due to optimized controllers on the other hand, points out the advantage of advanced control methods for heating systems as contribution to reach the goal of a heat demand reduction of 80 % until the year 2050.

1.2 Heating systems

A heating system provides thermal power to satisfy the heat demand of a building, which means in general, there is a supply part and a consumer part. The heat demand depends on the environmental conditions and the defined comfort room temperature for the building users. The environmental conditions depend on the location of the building and can change over a year with the seasons. The comfort room temperature depends on the use of the building. For residential buildings there are other conditions as for a commercial building, which is only occupied at daytime. Also the architecture and the construction of a building influences the heat demand. This results in the fact, that each heating system has to be planned individually for a building to ensure that the room climate or temperature remains in the same comfort zone for different environmental conditions.

The supplier is a heat generation unit, e.g, a boiler with a burner. The heat generation unit can also be a solar thermal collector, an oven or an electric heat pump. Also the energy source can be different for the devices, such as solid fuel, liquids, gases, solar radiation or electricity. The investigated heating systems in this thesis uses a boiler with a burner as heating unit. All of these devices heat up a medium to transfer and distribute the heat into the building. Radiators can be supplied via pipes with the heated medium to transfer the heat to the surrounding air and heat up the rooms of a building. According to the length of the pipes and the mass of the building there is a time delay between the heat supply and the heat consumption in a room. If the heated medium is air, then the heat can be transferred into the rooms via a ventilation system without using a radiator. The heating systems, which are introduced in this thesis uses radiators for the heat distribution and it is assumed that the heat transfer medium is water. The radiators together with the building are the consumer part of the system, whereas the boiler is the supply part.

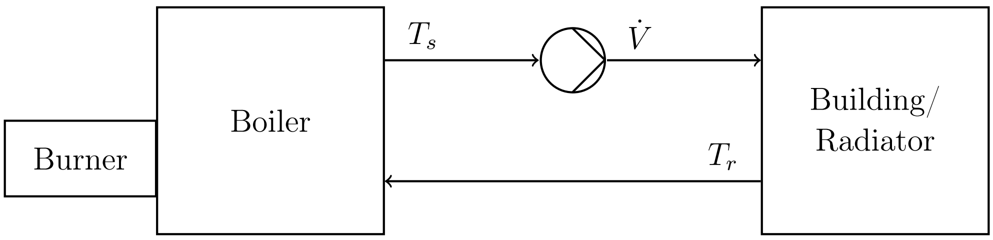


Figure 1.1: Scheme of a simple heating system

The boiler heats up the water, which leaves the boiler with a supply temperature T_s. A pump provides a volume flow $\dot{V}$ in the pipes to transfer the heated water with the supply temperature T_s to the radiators. The radiators release the heat to the rooms, which means that the water in the radiators gets colder. The water leaves the radiators with a return temperature T_r and returns to the boiler where the water is heated up again. The supply temperature of the boiler to satisfy the heat demand of the building is given by a heating curve which defines a reference supply temperature according to the ambient conditions or more specific the outside temperature T_{out}. The described heating system is shown in Figure 1.1.

1.3 Research question

The energy consumption for heating rooms and for supplying buildings with warm water is up to 80 % of the total energy consumption of the buildings [79]. Due to the climate protection goals of the government the energy consumption of buildings have to be optimized. One aspect is energy waste according to badly tuned controllers, which also includes the difficulties of well suited references for the individual buildings [61]. This points out the large potential of energy savings by using optimized and advanced control methods. Many processes are monitored today and the number of sensors increases. As a result the amount of available data increases too. This data is often used for monitoring and analysis purposes only, e.g., for the evaluation of the energy consumption of buildings. It is rarely used for control purposes for heating systems.

This leads to the research question, which is investigated in this thesis.

How can control methods for heating systems be improved by direct use of stored measurement data?

It is obvious that data can be used in many different ways, so that not all possibilities can be stated here. One way to use the data indirectly is the modeling process, for instance the whole model or only the unknown system parameters can be estimated from measurement data. This developed models can be used for model predictive control (MPC) algorithms [54]. The linear MPC problem is established and also applied to heating systems with a potential to decrease the energy demand [31, 53]. A model predictive controller uses the model to predict the future behavior of the system for the calculation of the control signals. Predictive controllers are useful for systems with delay times, such as heating systems. But if a linear model is not sufficient for the system dynamics the linear MPC changes into a nonlinear MPC. Heating system models are inherited in the class of multilinear models, which are nonlinear models, but there are no structural investigations about the model predictive control optimization problem if a multilinear model is used instead of a linear model.

A linear MPC uses a reference of the system. Finding a suitable reference can be difficult, which means that a model predictive controller without using a reference could be beneficial for applications. Due to the advantages of a linear MPC, a simple linear model can be used, the problem is well known and the results for heating systems are very promising [66, 73]. Besides the modeling part, the question arises if the measurement data can be used for control in other ways. Iterative learning control (ILC) uses stored measurement data to calculate the input signal of the next iteration, which means learning from the past [18, 76]. This works perfectly well for periodic processes. Heating systems show a periodicity in the ambient conditions or the disturbances, e.g., the outside temperature, which increases during the day and decreases during the night.

Finding a way to apply an ILC to a linear MPC for improving the MPC performance could be very useful for applications. Such approach would connect the use of data for the modeling and the prediction of the future and the use of historic data for learning from the past.

All of the investigated control algorithms will be applied to heating systems with the main goals to save energy and keeping the room temperatures in a comfort zone defined according to the German norm DIN-EN15251 [24].

1.4 State of research

For today's applications classical controllers, like a bang-bang controller, three-point controller and proportional-integral-derivative (PID) controller, are well known and widely-used, also for the control of heating systems [10]. Much research is going on in the field of model predictive control [46, 51]. An overview of predictive control methods is given in [55]. MPC is applied in various industrial fields, like automotive, food processing or chemicals industry [67]. But also the field of MPC for heating and cooling systems is of a strong scientific interest [10, 31, 53]. An applied MPC to a building heating system with respect to the weather forecast with energy savings from 15 % up to 28 % compared to the classical control approach is shown in [66, 73]. This shows the large potential of a linear MPC in contrast to the conventional implemented control strategies. Using a nonlinear model, e.g. an multilinear time-invariant (MTI) model, leads to a nonlinear MPC problem [26]. A few investigations of an NMPC for heating systems exist. An NMPC for a heating system using a bilinear model is investigated in [34]. An NMPC for a heating system using a nonlinear model for an optimized storage tank loading is introduced in [61].

Other predictive control approaches, like economic model predictive control (EMPC), are also applied to heating systems. An application for load shifting of an electrical heat pump or a chiller for buildings, due to varying electricity prices with constant constraints is given in [30, 56]. An application of an EMPC for energy minimization of a cooling system of a commercial building by calculating new setpoints for the underlying controller is give in [52]. All of these mentioned EMPC approaches consider varying electricity prices over time.

The field of iterative learning control shows various industrial applications where ILC algorithms are used. An overview of iterative learning control applied to batch proceses is given in [45]. For the wafer production iterative learning controllers are used for the temperature control as well as for the precise positioning of wafer scanners [58, 78]. But also in other application fields, like the control of a free-electron laser, an ILC is applied [64, 69]. First approaches of iterative learning control for heating systems and building temperature control are introduced in [57, 77] and a data-driven control approach which uses varying energy costs over time is given in [23].

The literature overview shows that the most common controllers are the bang-bang or three point controllers and the PID controllers or PI controllers. But also the linear

MPC with a focus on heating systems are investigated in theory with proposed energy savings up to 28 % in comparison to the classical control strategy. First experimental results are published where a linear MPC with respect to the weather forecast was used. Other advanced control approaches only show a few first investigations with a focus on heating or cooling systems and there is a lack of strategies combing iterative learning methods with predictive approaches for such systems. No structural investigations about the model predictive control optimization problem exists if a multinlinear time-invariant model is used instead of a linear model. In general, there is a lack of real-time applications of advanced control strategies.

1.5 Thesis structure

Chapter 2 introduces different model classes in state space representation, followed by the heating system modeling. The component-wise modeling of the different heating system parts, for instance the boiler, are shown. Chapter 3 gives an overview over different control methods which are relevant for this thesis. The overview starts with classical control, like bang-bang or proportional-integral-derivative (PID) control, followed by predictive control methods, e.g., linear model predictive control (MPC), and ends by the data-based control methods, such as iterative learning control (ILC). A heating system model of a test facility for an office of a non-residential building is introduced and a PI controller and an MPC are applied to this model. The simulation results of both controllers are compared. Chapter 4 deals with predictive control methods. First an economic model predictive controller (EMPC) is introduced for the control of continuous control signals and for discrete and continuous control signals. Structural investigations of the MPC optimization problem if a multilinear model is used instead of a linear one are presented. Chapter 5 shows a data-driven iterative learning control approach, which uses stored measurement data. The data-driven ILC is combined with an MPC to a data-driven learning MPC. For storage demand reduction, a tensor decomposition method is applied to the data-driven ILC with the result of a data-driven tensor ILC. These control approaches are applied to a heating system example and the simulation results are presented. Chapter 6 introduces the hardware components for real-time implementation. After the presentation of hardware-in-the-loop test, the real-time application of a data-driven learning MPC is shown. Followed by the application of an EMPC to a heating system of an office building. The thesis ends with a conclusion and an outlook for further investigations.

Chapter 2

Model classes and heating system modeling

This chapter describes the relevant model classes in state space representation, like nonlinear and linear state space models, as well as the modeling of the dynamical behavior of heating systems. Each building has different requirements to a heating system and the single components are combined individually. This suggests a component-based modeling of the system, so that the models of the single components can be reused for the models of different systems. The basic components of the heating system models are presented.

2.1 State space models

The dynamical behavior of a physical system can be described mathematically, which means the relation between the input signal $\mathbf{u}(t)$ and the output signal $\mathbf{y}(t)$ of the system. That is described by the system block in Figure 2.1.

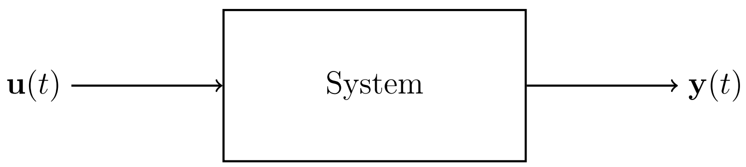


Figure 2.1: System block with inputs and outputs

A standard model representation for the dynamical behavior of the systems with time-independent parameters are time-invariant state space models, with the general form of time-invariant nonlinear state space models. Mathematical descriptions of the dynamical system behavior can be found, e.g. in [19] or [71].

2.1.1 Nonlinear model

The general form of time-invariant model classes are the nonlinear time-invariant models [35]. The dynamical behavior of the system can be described by the ordinary differential equation (ODE) of nth order for many applications. The differential equation describes the behavior between the input $\mathbf{u}(t)$ and the output $\mathbf{y}(t)$ of the system in the continuous-time domain. In general, the ODE of order n can be written as a system of first order ODEs and the right hand sides of the ordinary differential equations are nonlinear. The system of first order ODEs leads to the description of the dynamical system behavior as a nonlinear continuous-time state space model

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

where $\mathbf{f}_c$ is the state transition function, $\mathbf{g}_c$ is the output function, $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is the input vector, $\mathbf{y}(t) \in \mathbb{R}^p$ is the output vector, $\mathbf{x}_0$ is the initial state vector, t denotes the time and the subscribed c indicates that the model is a continuous-time model. The system has n states, m inputs and p outputs. The temporal development of the system states is given by the state equation (2.1). The system output is given by the output function (2.2).

So far, all signals were assumed to be known at any time t, which leads to a continuous-time description of the system. For the description in the discrete-time domain, the state, input and output signals are only known at fixed sample time steps, e.g., for an input $[u(0 \cdot t_s), u(1 \cdot t_s), ...]$, with the sample time t_s. For a simpler notation the sample time will be omitted, so that $u(k \cdot t_s) = u(k)$, where $k \in \mathbb{N}_0$ is the time step index. The system behavior in discrete-time is described by difference equations and not by differential equations such that the states of the system $\mathbf{x}(k+1)$ at time k depend on the actual and past states and inputs. The nonlinear discrete-time state space model is given by

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

Nonlinear state space models are a general model class. More restrictive model classes are the multilinear state space models or the linear state space models.

2.1.2 Multilinear model

The multilinear model class restricts the general class of nonlinear models. A definition of the multilinear time-invarinat (MTI) models can be found in [47]. The description of the dynamical behavior of a system with nonlinear state space models allows arbitrary functions as right hand sides, where as the right hand side of the ordinary differential equations of a multilinear state space model have to be multilinear. First the monomial vector is introduced to define the multilinear state space.

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

$$\mathbf{m}(\mathbf{x}(t),\mathbf{u}(t)) = \begin{pmatrix}1\\u_m\end{pmatrix} \otimes \cdots \otimes \begin{pmatrix}1\\u_1\end{pmatrix} \otimes \begin{pmatrix}1\\x_n\end{pmatrix} \otimes \cdots \otimes \begin{pmatrix}1\\x_1\end{pmatrix}$$

$$= \left(\bigotimes_{i=m}^{1} \begin{pmatrix}1\\u_i\end{pmatrix}\right) \otimes \left(\bigotimes_{i=m}^{1} \begin{pmatrix}1\\x_i\end{pmatrix}\right), \tag{2.7}$$

where $\mathbf{x} \in \mathbb{R}^n$ with the elements x_i, $i = 1, \ldots, n$ is the state vector and $\mathbf{u} \in \mathbb{R}^m$ with the elements u_j, $j = 1, \ldots, m$ is the input vector and $\otimes$ denotes the Kronecker product, as defined in Appendix B.1. For the sequence of Kronecker products $\bigotimes_{i=m}^{1}$ the index i is decremented in every step.

Example 2.1 *The monomial vector with two states, x_1 and x_2 and one input u_1 is given by*

$$\mathbf{m}(\mathbf{x},\mathbf{u}) = \begin{pmatrix}1\\u_1\end{pmatrix} \otimes \begin{pmatrix}1\\x_2\end{pmatrix} \otimes \begin{pmatrix}1\\x_1\end{pmatrix} = \begin{pmatrix}1\\x_1\\x_2\\x_1x_2\\u_1\\u_1x_1\\u_1x_2\\u_1x_1x_2\end{pmatrix}.$$

The state space model of a continuous-time multilinear time-invariant (MTI) system with n states, m inputs and p outputs in matrix representation is given by

$$\dot{\mathbf{x}}(t) = \mathbf{F}_c\mathbf{m}(\mathbf{x}(t),\mathbf{u}(t)) \tag{2.8}$$
$$\mathbf{y}(t) = \mathbf{G}_c\mathbf{m}(\mathbf{x}(t),\mathbf{u}(t)) \tag{2.9}$$

with the transition matrix $\mathbf{F}_c \in \mathbb{R}^{n\times 2^{n+m}}$ and the output matrix $\mathbf{G}_c \in \mathbb{R}^{p\times 2^{n+m}}$, in which the notation $\mathbb{R}^{\times 2^n}$ means $\mathbb{R}^{\overbrace{2\times\ldots\times 2}^{n\ times}}$.

The state space model of a discrete-time MTI system reads as follows

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{m}(\mathbf{x}(k),\mathbf{u}(k)) \tag{2.10}$$
$$\mathbf{y}(k) = \mathbf{G}\mathbf{m}(\mathbf{x}(k),\mathbf{u}(k)). \tag{2.11}$$

Example 2.2 *The state transition function of model with two states one input is given by*

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} f_{1,1} & f_{1,2} & f_{1,3} & f_{1,4} & f_{1,5} & f_{1,6} & f_{1,7} & f_{1,8} \\ f_{2,1} & f_{2,2} & f_{2,3} & f_{2,4} & f_{2,5} & f_{2,6} & f_{2,7} & f_{2,8} \end{pmatrix} \begin{pmatrix} 1 \\ x_1(k) \\ x_2(k) \\ x_1(k)x_2(k) \\ u(k) \\ u(k)x_1(k) \\ u(k)x_2(k) \\ u(k)x_1(k)x_2(k) \end{pmatrix}.$$

2.1.3 Linear model

A description of the restricted class of linear time-invariant models can be found in [49] or [70]. The dynamical behavior of linear systems is described by linear ODEs, which means that the right hand sides are linear functions. Thus, a linear continuous-time state space model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t), \tag{2.12}$$
$$\mathbf{y}(t) = \mathbf{C}_c\mathbf{x}(t) + \mathbf{D}_c\mathbf{u}(t) \tag{2.13}$$

where $\mathbf{A}_c \in \mathbb{R}^{n\times n}$ is the system matrix, $\mathbf{B}_c \in \mathbb{R}^{n\times m}$ is the input matrix, $\mathbf{C}_c \in \mathbb{R}^{p\times n}$ is the output matrix and $\mathbf{D}_c \in \mathbb{R}^{p\times m}$ is the feedthrough matrix.

The discrete-time linear state space model is given by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \tag{2.14}$$
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \tag{2.15}$$

Example 2.3 *The state equation of a linear state space model with two states and one input is given by*

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \begin{pmatrix} b_{1,1} \\ b_{2,1} \end{pmatrix} u(k).$$

For some applications input signals can be discrete and not only continuous, which means, that an input can only switch between different states, such as an on-off switching signal. That makes the introduction of the so-called hybrid state space models necessary, where discrete and continuous input signals can be used.

2.1.4 Hybrid model

The hybrid model that is used in this thesis is a linear state space model with a mixture of continuous and discrete input signals [14]. The description of standard linear state space model (2.12) and (2.13) for the continuous-time domain and (2.14) and (2.14) for the discrete-time domain, is extended by an additional input vector $\mathbf{u}_{dis}$, which includes the discrete input signals and the corresponding parameter matrix $\mathbf{B}_{dis}$. The values of $\mathbf{u}_{dis}$ have to be part of the natural number set $\mathbb{N}_0$. The hybrid linear continuous-time state space model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t) + \mathbf{B}_{c,dis}\mathbf{u}_{dis}(t), \tag{2.16}$$

$$\mathbf{y}(t) = \mathbf{C}_c\mathbf{x}(t) + \mathbf{D}_c\mathbf{u}(t) + \mathbf{D}_{c,dis}\mathbf{u}_{dis}(t) \tag{2.17}$$

with the input matrix $\mathbf{B}_{c,dis} \in \mathbb{R}^{n\times m_{dis}}$ and the input vector $\mathbf{u}_{dis} \in \mathbb{N}_0^{m_{dis}}$ of the discrete signals and the number of discrete inputs m_{dis}.

The hybrid linear discrete-time state space model is given by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_{dis}\mathbf{u}_{dis}(k), \tag{2.18}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{D}_{dis}\mathbf{u}_{dis}(k). \tag{2.19}$$

The developed models are part of one of the model classes introduced and it is stated at the corresponding part to which class the model belongs.

2.2 Modeling of heating systems

In general heating systems are planned and built individually for each building, due to different requirements based on the location and use of the building. Each building has its own combination of heating system components, but the single components are similar and every building needs a supply part, e.g., a boiler. This means, that the modeling has to be done individually for each heating system and building. Thus, a component-wise modeling is suitable for heating systems, because the physics of each component is similar for different heating systems. The single components can be reused for varying heating system setups and only the parameters of the components have to be adjusted, for instance the maximum heating power of a boiler. This procedure simplifies the modeling process for different heating systems.

The heating system components are modeled by thermal heat balances, based on the fact that energy can neither be created nor destroyed. Energy can only be transformed from one form into another. The law of conservation of energy means, that the sum of the supplied and removed thermal power of each component of the system has to be zero. The supplied thermal power is a positive and the removed thermal power a negative contribution creating a thermal power balance.

The thermal power $\dot{Q}$ is the time derivative of the thermal energy

$$Q(t) = c\rho V(t)T(t) \tag{2.20}$$

where c is the specific thermal capacity and ρ the density of a medium with the temperature T and the volume V. The thermal power is given by

$$\dot{Q}(t) = \frac{dQ(t)}{dt} = c\rho V(t)\frac{dT(t)}{dt} + c\rho T(t)\frac{dV(t)}{dt}. \tag{2.21}$$

Depending on the context where the thermal power balance is calculated, either the volume V or the temperature T is assumed to be constant. This modeling concept for a heating system is introduced in [38] and [61].

Figure 2.2 shows a heating system component with a thermal power input $\dot{Q}_{in}(t)$, a thermal power output $\dot{Q}_{out}(t)$ and thermal losses $\dot{Q}_{loss}(t)$ to the environment. Each of the heating system components have some in- and outflows of thermal power, e.g. a burner supplies a boiler with thermal power and heats up the water. The warm water from the boiler flows to the radiators, the radiators heat up a room or building and the cold water from the radiators returns to the boiler and the building has thermal losses to the environment.

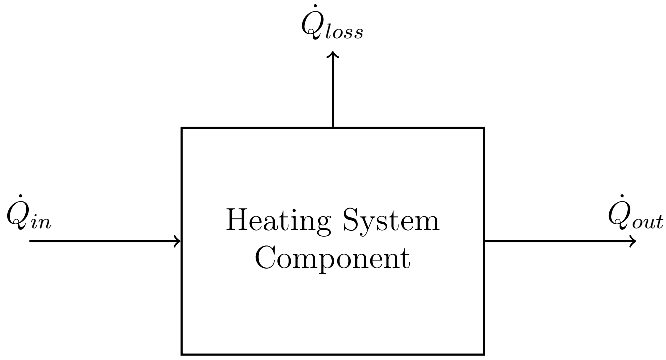


Figure 2.2: Heating system component

The thermal power $\dot{Q}_{com}(t)$ of the heating system component shown in Figure 2.2 is given by

$$\dot{Q}_{com}(t) = \dot{Q}_{in}(t) - \dot{Q}_{out}(t) - \dot{Q}_{loss}(t) \tag{2.22}$$

which leads to the differential equation

$$c\rho V_{com}\dot{T}_{com}(t) = c\rho\dot{V}_{in}(t)T_{in}(t) - c\rho\dot{V}_{out}(t)T_{o}(t) - k_{loss}\left(T_{com}(t) - T_{env}(t)\right) \tag{2.23}$$

of the component temperature T_{com} with the volume V_{com}. The medium flows in the component with the temperature T_{in} and the volume flow $\dot{V}_{in}$, and leafs the component with the temperature T_o and the volume flow $\dot{V}_{out}$. It is assumed that the medium of the component is mixed instantaneously so that T_{com} is the temperature for the entire volume V_{com} of the component, which means that $T_o(t) = T_{com}(t)$. The thermal losses $\dot{Q}_{loss}(t)$ are assumed to be proportional to the difference of the component temperature $T_{com}(t)$ and the temperature of the environment $T_{env}(t)$

$$\dot{Q}_{loss}(t) = k_{loss}\left(T_{com}(t) - T_{env}(t)\right) \tag{2.24}$$

with the proportional factor k_{loss} for the heat transfer.

The unknown parameters, such as the proportional factor k_{loss}, will be estimated from measurement data with the advantage that no pre-knowledge of the parameters is necessary but the measurement data has to be available. This modeling, where physical equations are used to approximatly capture the main dynamics of the systems and the parameters are estimated from measurement data, is called grey-box modeling, with the advantage that the internal signals have a physical meaning and the model complexity is reduced because of the neglection of minor dynamical effects. In contrast to the grey-box modeling, white-box modeling is also based on physical equations, but all parameters have to be known from data sheets with the advantage that no measurement data is used and the model can be derived even if the system does not exist at that moment. The drawback of the white-box modeling is that very detailed information has to be available to describe the dynamical effects mathematically, which leads to complex and accurate models. In contrast to the white-box modeling where every parameter and equation have to be known, the black-box modeling uses measurements of the input and output signals of the system to estimate a model which represents the input-output behavior of the system without any knowledge of the system physics. Consequently, the internal signals of the model have no physical meaning, hence being given the name black-box model. Further information about the different modeling methods can be found, e.g., in [48]. For all developed models it is assumed, that the estimated parameters are constant, which means time independent.

The models of heating systems, which are modelled on the basis of thermal heat balances belong to the class of multilinear models, introduced in Section 2.1.2, because of the multiplication of temperature and volume flows [61].

For a junction where n different volume flows $\dot{V}(t)$ with a temperature $T(t)$ are connected together, the resulting volume flow is given by

$$\dot{V}_{res}(t) = \dot{V}_1(t) + \dot{V}_2(t) + \cdots + \dot{V}_n(t). \tag{2.25}$$

The resulting temperature can be calculated by the heat balance for this junction

$$\dot{Q}_{res}(t) = \dot{Q}_1(t) + \dot{Q}_2(t) + \cdots + \dot{Q}_n(t) \tag{2.26}$$

and with equation (2.25) follows for the resulting temperature

$$c\rho\dot{V}_{res}(t)T_{res}(t) = c\rho\dot{V}_1(t)T_1(t) + c\rho\dot{V}_2(t)T_2(t) + \cdots + c\rho\dot{V}_n(t)T_n(t)$$

$$T_{res}(t) = \frac{\dot{V}_1(t)T_1(t) + \dot{V}_2(t)T_2(t) + \cdots + \dot{V}_n(t)T_n(t)}{\dot{V}_1(t) + \dot{V}_2(t) + \cdots + \dot{V}_n(t)}. \tag{2.27}$$

Some heating system components are modeled in Simulink and summarized in a Simulink library called HeatLib [38]. This library is used for the following component modeling.

2.2.1 Boiler

A boiler supplies a heating system with the thermal power to satisfy the heat demand of a building. A boiler heats the water and supplies the heating circuits with the warmed water. The heated water with the supply temperature $T_s(t)$ leaves the boiler with the volume flow $\dot{V}_s(t)$ and the thermal power $\dot{Q}_s(t)$. The returned water from the heating circuits with the return temperature $T_r(t)$ flows into the boiler with the volume flow $\dot{V}_{in}(t)$ and the thermal power $\dot{Q}_r(t)$. A burner heats the water inside the boiler with the volume V_{boiler}, which leads to the thermal power $\dot{Q}_P(t)$ and the boiler has thermal losses $\dot{Q}_{b,loss}(t)$ to the environment. Figure 2.3 shows a scheme of the boiler with the thermal power in- and outflows.

The thermal power balance of the boiler is given by

$$\dot{Q}_b(t) = \dot{Q}_P(t) + \dot{Q}_r(t) - \dot{Q}_s(t) - \dot{Q}_{b,loss}(t). \tag{2.28}$$

This leads to the differential equation for the supply temperature of the boiler

$$\begin{aligned} c\rho V_{boiler}\dot{T}_s(t) &= c\rho \dot{V}_s(t)T_r(t) - c\rho \dot{V}_s(t)T_s(t) - k_{b,loss}\left(T_s(t) - T_{b,env}\right) \\ \dot{T}_s(t) &= \frac{1}{V_{boiler}}\dot{V}_s(t)\left(T_r(t) - T_s(t)\right) - \frac{k_{b,loss}}{c\rho V_{boiler}}\left(T_s(t) - T_{b,env}\right) \end{aligned} \tag{2.29}$$

where ρ and c are the density and the specific heat capacity of water, $k_{b,loss}$ is the heat transfer coefficient from the boiler to the environment, $T_{b,env}$ is the temperature of the surrounding of the boiler and assuming that $\dot{V}_{in} = \dot{V}_s$. The thermal power $\dot{Q}_P = \alpha P_{max}$ is controlled by the modulation signal $\alpha \in [0, 1]$, where P_{max} is the maximum power of the boiler.

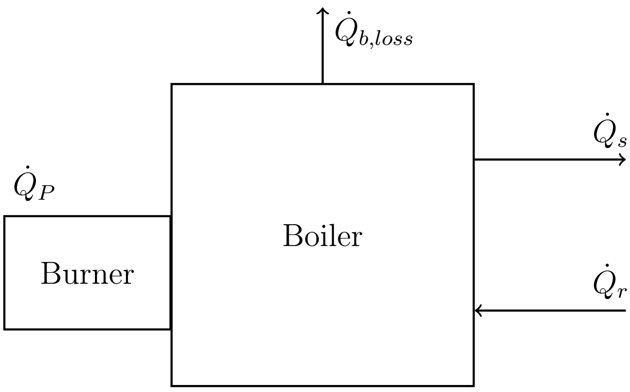


Figure 2.3: Scheme of a Boiler

2.2.2 Consumer

The consumer model consists of two parts, the radiators model and the building model. The supplier, e.g. a boiler, provides the radiators with the thermal power $\dot{Q}_s$ and the radiator transfers the thermal power to the building to satisfy the heat demand $\dot{Q}_d$. Warm water flows into the radiator with the volume flow $\dot{V}_r$ and the supply temperature T_s, and leaves the radiator with the same volume flow $\dot{V}_r$ and the return temperature T_r, which corresponds to the returning thermal power $\dot{Q}_r$. Figure 2.4 shows a scheme with the thermal in- and outflows of the radiator and the building.

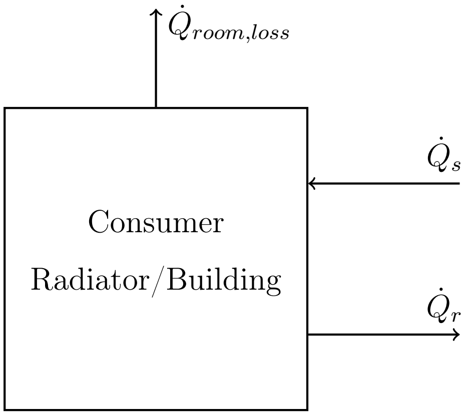


Figure 2.4: Scheme of a consumer

The heat balances of the radiator is given by

$$\dot{Q}_{radiator}(t) = \dot{Q}_s(t) - \dot{Q}_r(t) - \dot{Q}_d(t),$$

where the heat demand of the building

$$\dot{Q}_d(t) = k_{r,room}\left(T_r(t) - T_{room}(t)\right)$$

corresponds to the thermal power, which is transferred from the radiator to the building and is assumed to be proportional to the difference between the return temperature T_r and the room temperature T_{room}. The heat transfer proportional factor from the radiator to the room is denoted by $k_{r,room}$. The ODE of the return temperature is given by

$$\begin{aligned} c\rho V_r \dot{T}_r(t) &= c\rho \dot{V}_r(t) T_s(t) - c\rho \dot{V}_r(t) T_r(t) - k_{r,room}\left(T_r(t) - T_{room}(t)\right), \\ \dot{T}_r(t) &= \frac{1}{V_r}\dot{V}_r(t)\left(T_s(t) - T_r(t)\right) - \frac{k_{r,room}}{c\rho V_r}\left(T_r(t) - T_{room}(t)\right) \end{aligned} \tag{2.30}$$

with the overall volume of the radiators V_r.

The thermal heat balance of the building is given by

$$\dot{Q}_{room}(t) = \dot{Q}_d(t) - \dot{Q}_{room,loss}(t).$$

with the thermal losses $\dot{Q}_{room,loss} = k_{room,o}\left(T_{room} - T_{out}\right)$ from the building to the environment, where T_{out} is the outside temperature and $k_{room,o}$ the heat transfer coefficient from the building to the outside. The ODE of the room temperature is given by

$$\dot{T}_{room}(t) = \frac{k_{r,room}}{C_{room}}\left(T_r(t) - T_{room}(t)\right) - \frac{k_{room,o}}{C_{room}}\left(T_{room}(t) - T_{out}(t)\right). \tag{2.31}$$

with the thermal capacity of the building C_{room}.

From an application point of view, the model should be as simple as possible to reduce the modeling effort but with respect to the task which uses the model. In this thesis not every room is modeled explicitly but the entire building is represented by one room with the temperature T_{room} and every single radiator of the building is summarized to one radiator with the temperature T_r, or the rooms and radiators of each floor are modeled as a single zone. These models do not capture every single thermal transition from one room to another but the main thermal dynamics of a floor or the building are represented by those one zone models. Such models are sufficient to evaluate controller designs [61].

2.2.3 Valves

A three-way or four-way valve mixes together two volume flows $\dot{V}_1$ and $\dot{V}_2$ with the temperatures T_1 and T_2, e.g., to reduce the supply temperature of one heating circuit by mixing water from the return to the supply. Figure 2.5 shows the schemes of the three-way valve. If the mixing ratio

$$\phi_3 = \frac{\dot{V}_2}{\dot{V}_1} \tag{2.32}$$

is known, the resulting volume flow for a three-way valve can be calculated by

$$\dot{V}_m = \dot{V}_1 + \dot{V}_2 = (1 + \phi_3)\,\dot{V}_1 \tag{2.33}$$

and the resulting temperature is given by

$$T_m = \frac{T_1\dot{V}_1 + T_2\dot{V}_2}{\dot{V}_1 + \dot{V}_2} = \frac{1}{1 + \phi_3}\,(T_1 + \phi_3 T_2) \tag{2.34}$$

The mixing ratio is determined in the interval $[0, 1]$, which means that the volume flow $\dot{V}_m$ is maximum twice the volume flow $\dot{V}_1$. A mixing ratio of zero means that $\dot{V}_1 = \dot{V}_m$ and nothing is mixed to the volume flow $\dot{V}_1$.

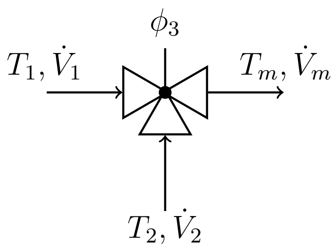


Figure 2.5: Scheme of a three-way valve

Figure 2.6 shows a scheme of the four-way valve.

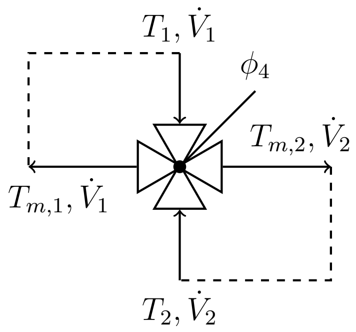


Figure 2.6: Scheme of a four-way valve

The resulting temperatures $T_{m,1}$ and $T_{m,2}$ of a four-way valve are given by

$$T_{m,1} = \frac{(1-\phi_4)\, T_1 \dot{V}_1 + \phi_4 T_2 \dot{V}_2}{(1-\phi_4)\, \dot{V}_1 + \phi_4 \dot{V}_2} \tag{2.35}$$

$$T_{m,2} = \frac{\phi_4 T_1 \dot{V}_1 + (1-\phi_4)\, T_2 \dot{V}_2}{\phi_4 \dot{V}_1 + (1-\phi_4)\, \dot{V}_2} \tag{2.36}$$

with the mixing ratio $\phi_4 \in [0, 1]$.

2.2.4 Pump

The pump represents the hydraulic part of the heating system, which is described as changes of the volume flow in dependency of the room temperature. It is assumed that the hydraulic is adjusted and works fine. The pump determines the volume flow $\dot{V}(t)$ in dependency of the difference between the room temperature $T_{room}(t)$ and a reference room temperature $T_{room,r}(t)$. For large differences between $T_{room}(t)$ and $T_{room,r}(t)$ the volume flow $\dot{V}(t)$ is in saturation. Such behavior is known from thermostatic valves of radiators. The position of a thermostatic valve defines the reference room temperature $T_{room,r}(t)$. The pump reproduces the impact of thermostatic valves on the volume flow of the heating system with respect to $T_{room,r}(t)$. The assumption that the difference between $T_{room}(t)$ and $T_{room,r}(t)$ is not too large, which means that the thermostatic valve is not completely open or closed, leads to the linear relation

$$\dot{V}(t) = \dot{V}_{mean} + b_{vol}\left(T_{room,r}(t) - T_{room}(t)\right) \tag{2.37}$$

with the slope b_{vol} and the mean volume flow $\dot{V}_{mean}$. If the behavior of the volume flow can not be assumed as constant or linear as described previously, then the pump is modeled by a Black-Box-Model by using measurement data. For the estimation the MATLAB function n4sid of the System Identification Toolbox was used [9].

2.2.5 Simple heating system model

A simple example of a heating system consists of a boiler, a radiator, a building and a pump. Figure 2.7 shows the setting of such a heating system. The model was introduced in [40] and the paper shows how such a model in state space representation can be used for different tasks like control design or fault detection. In this thesis the developed models are used for control design only.

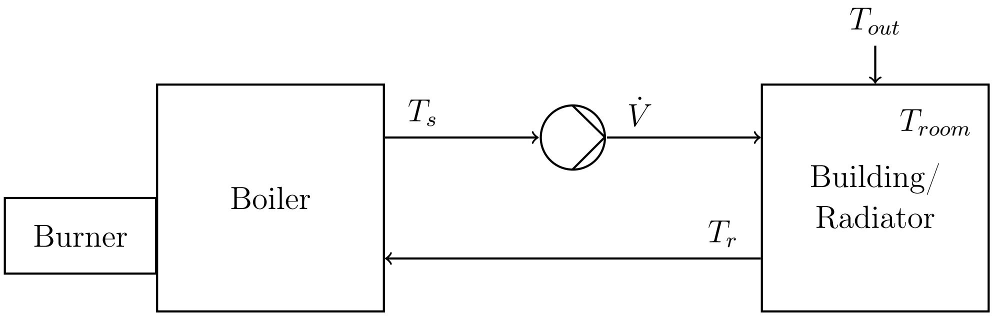


Figure 2.7: Scheme of a simple heating system

The heating system is modeled as a grey box model, where only the main dynamics are captured by the model. The supply part is represented by a boiler as introduced in Section 2.2.1. The consumer, introduced in Section 2.2.2, which includes the radiators and the building is modeled as a one zone model. The changes of the volume flow over time are represented by the pump, see Section 2.2.4.

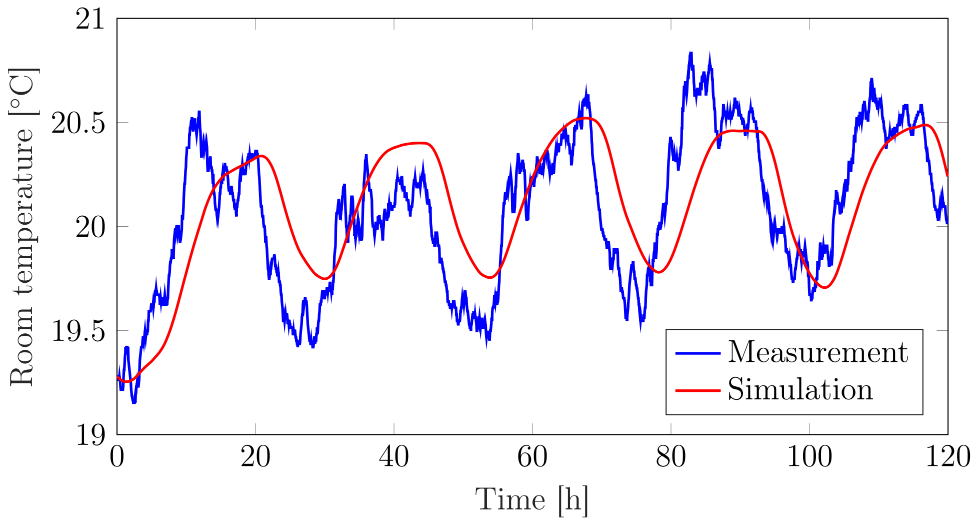


Figure 2.8: Comparison of the measured and simulated room temperature [40]

The dynamical behavior of the heating system is described by three differential equations (2.29), (2.30) and (2.31) and the equation for the volume flow (2.37). As an example, the ODE of the room temperature (2.31) includes the unknown parameters C_{room}, $k_{r,room}$ and $k_{room,o}$. These three unknown parameters are estimated from measurement values. The resulting model consists of the multilinear model class as introduced in Section 2.1.2. Figure 2.8 shows the comparison of the measured and simulated values of the room temperature after the parameter estimation.

The measurement values for the parameter estimation are taken from a school building. The results show that the simple one zone model represents the main dynamical behavior of the heating system. The estimated parameters of this model are summarized in the appendix section in Table A.1.

The next chapter gives an overview of different control methods for heating systems and compares simulation results of two control strategies. A proportional-integral controller and a model predictive controller is applied to the same heating system model.

Chapter 3

Control methods for heating systems

The first part of this chapter gives an overview of different control methods for heating systems or more generally heating, cooling and ventilation systems (HVAC) and refers to the corresponding literature. It gives the reader an impression about different control strategies, which are relevant for this thesis, with a focus on model predictive control and iterative learning control. A more detailed description, including formulas, is given for the controllers, which are used in this thesis. The second part introduces a heating system model and considers two control strategies in detail, the proportional-integral control and the model predictive control, which are used as reference control strategies in the following chapters of this thesis.

3.1 Standard control methods

Standard control methods, such as bang-bang and three point control with discrete control signals $\mathbf{u}_{dis}$ and proportional-integral-derivative (PID) control with continuous control signals $\mathbf{u}$ will be introduced next. These control methods are commonly used in many of today's applications and also for the control of heating systems [10].

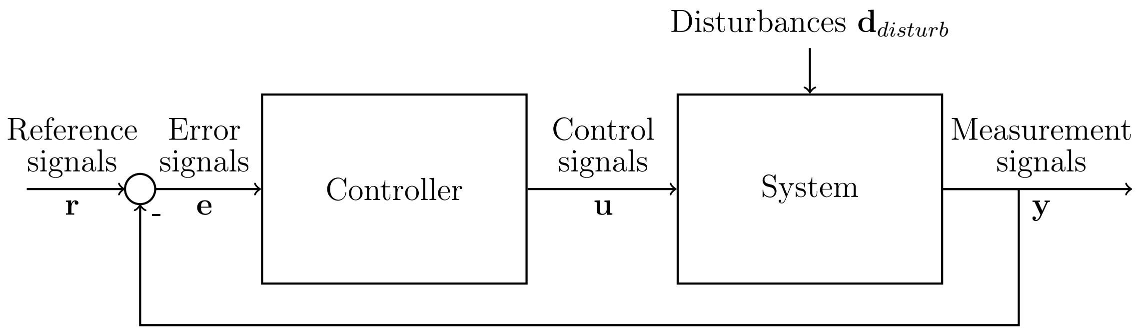


Figure 3.1: Scheme of a standard control loop

The control goal is the tracking of a given trajectory or setpoint. Figure 3.1 shows a scheme of a standard control loop, with the control signals $\mathbf{u}$, reference signals $\mathbf{r}$, error signals $\mathbf{e}$, disturbances $\mathbf{d}_{disturb}$ and the measured output signals $\mathbf{y}$. A classical reference signal for heating systems is the reference supply temperature, e.g., for a boiler, given by a heating curve with respect to the outside temperature.

3.1.1 Heating curve as reference signal for heating systems

A heating system, as introduced in Section 2.2, provides a building with the thermal power to satisfy the heat demand of the building with the goal to reach a desired room temperature. A boiler supplies the radiators with warm water with the supply temperature T_s. The reference supply temperature r_s is defined by a heating curve with respect to the outside temperature T_{out}. The given reference temperature is inversely related to the outside temperature, which means that the heat demand increases if the outside temperature decreases. A simple heating curve is a linear equation defined by the slope a and the intercept of the y-axis b and with an upper and lower limit where r_s is in saturation. The two saturation points are defined by the coordinates $(T_{out,1}, r_{s,1})$ as upper limit and $(T_{out,2}, r_{s,2})$ as lower limit for the reference supply temperature, which leads to the definition of the heating curve

$$r_s = \begin{cases} r_{s,1} & \text{if } T_{out} < T_{out,1} \\ a \cdot T_{out} + b & \text{if } T_{out,1} < T_{out} < T_{out,2} \\ r_{s,2} & \text{if } T_{out} > T_{out,2}. \end{cases} \tag{3.1}$$

Figure 3.2 presents a heating curve estimated from measurement values of a heating system test facility for an office building of a non-residential building and will be used in the following of this thesis.

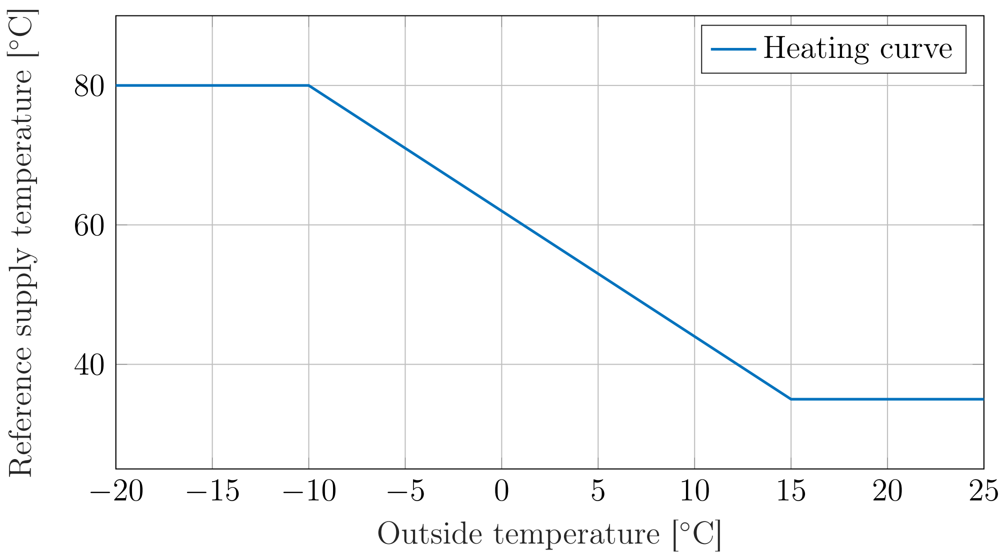


Figure 3.2: Heating curve estimated from measurement values

The heat demand differs from building to building due to the building size, the architecture, the materials, the location and so on. Also, the heating system of each building is individually planned and build. This means that every building needs its own heating curve or reference to satisfy the heat demand according to the outside temperature. Because of that, finding a suitable reference, which supplies the building with enough heat by wasting as less energy as possible, can be very difficult.

The control of the different components of the heating system, especially the thermal power of the boiler, and the tracking of the given references can be implemented with different control methods. An overview is given in the following.

3.1.2 Bang-bang and three-point control

A bang-bang or two step controller switches between two states. A three-point controller switches between three states. These controllers are used for systems which can not be controlled continuously, for instance, a system can be switched on and off or the power of a system can be controlled in three steps 0-1-2, e.g., zero means the system is switched off, one means 50 % of the maximum power is provided and two means the maximum power is provided by the system [28]. Because of the defined switching states these control signals are belong to the class of discrete control input signals $\mathbf{u}_{dis}$. The controller regulates the system, which is influenced by disturbances, according to a reference signal or setpoint, which means, that a measured output signal has to be fed back to the controller for comparison with the given reference signal. Figure 3.1 shows the scheme of a standard control loop.

To prevent the system from switching often, a hysteresis is used for such controllers. A hysteresis means that two thresholds are defined around the setpoint. The first one defines when the system switches on and the second one defines when the system switches off again. This also means, that the measured signal oscillates between the two thresholds. The distance between the two thresholds defines the switching frequency and also the amplitude of the oscillation. According to this, a three-point controller uses two setpoints with hystereses.

A simple example is an electric heater which switches on if the room temperature falls below the defined threshold and switches off again if the room temperature reaches the second threshold, with a hysteresis to prevent the heater from switching on and off again that often. Some boilers of today's heating systems are controlled by a three point controller, which uses the same principle, but with additional requirements like a second setpoint with hysteresis. The reference supply temperature r_s for the boiler is given by a heating curve, as introduced in Section 3.1.1. If the supply temperature T_s of the boiler is below the reference temperature the boiler switches from off to the first power level or from zero to one. According to the hysteresis the boiler switches off again if the certain threshold is reached. Otherwise, the boiler switches from the first power level to the maximum power level or from one to two, and again, if the given threshold is reached the boiler switches back from state two to one. Depending on the reference temperature the boiler switches off or back again to the second state and so on. If the control input is a continuous control signal a PID controller can be used.

3.1.3 Proportional-integral-derivative control

A proportional-integral-derivative controller is a standard controller. The controller consists of three parts: the proportional part, the integral part and the derivative part. A controller without the derivative part is called proportional-integral (PI) controller. The PID controller continuously calculates the control or input signal $u(t)$. The PID controller uses an error signal $e(t)$, like other standard single loop controllers, which is the deviation between a reference signal $r(t)$ and a measured output signal $y(t)$. The measured signal $y(t)$ has to be fed back to the PID controller to calculate the error signal. The PID controller adjusts the input signal $u(t)$ of the system with the goal to minimize the control error over time.

The error signal or control error is given by

$$e(t) = r(t) - y(t), \tag{3.2}$$

with the reference signal $r(t)$ and the output signal $y(t)$. The three parts of the PID controller, the proportional, integral and derivative part result in the computation of the control signal

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)\mathrm{d}\tau + K_D \frac{de(t)}{dt}, \tag{3.3}$$

with the three coefficients of the proportional part K_p, the integral part K_I and the derivative part K_D. The effect of the three different parts of the PID controller can be described as follows; the proportional term changes the control signal proportional to the control error, which means that a large positive control error leads to a proportional large positive control signal. The integral term changes the control signal as long as the error signal is unequal to zero. The derivative term only reacts to changes of the control error [49]. The control loop scheme is already shown in Figure 3.1.

A PID or PI controller is used for the control of the continuous signals, like the continuous control of the boiler power to minimize the difference between the reference supply temperature r_s and supply temperature T_s or to control a three-way valve for heating systems.

These classical controllers are well known and common in real-time applications, also for the control of heating systems [10]. These controllers react only on the actual output of the plant and disturbances, which means to the present. More advanced controllers, such as predictive controllers, consider the future behavior of the plant, whereas learning controllers learn from the past.

3.2 Predictive control methods

Predictive control methods use a model of the system to predict the dynamical behavior of the plant with the advantage to take the futur dynamics of the system and the changes of the disturbances into account for the calculation of the control signals by optimizing a given cost function, which defines the control goal. An overview of predictive control methods is given in [55], and with a focus on heating systems in [10].

3.2.1 Linear model predictive control

A linear model predictive controller (MPC) uses a linear model of the system for the calculation of the control inputs. An introduction to the MPC design by using a linear state discrete-time space model, as introduced in Section 2.1.3, is given in [54]. The MPC uses a linear model to predict the behavior of the system for future control inputs. The MPC calculates the optimal future input trajectory over a given prediction horizon H_p by using a model of the system and with respect to a cost function, which defines the control goal, e.g, minimizing the difference between the reference $\mathbf{r}_{mpc}(k)$ and the output signal $\mathbf{y}(k)$ with minimal control effort. The prediction horizon defines the number of time steps k, which the controller uses to calculate the future behavior of the plant. The controller optimizes the future input trajectory by minimizing a quadratic cost function. The measured actual states $\mathbf{x}(k)$ of the plant are fed back to the controller to initialize the model of the system for the optimization. An MPC is well suited for systems with large time delays because of the dynamical prediction of the system by using a model. The MPC optimization problem with a quadratic cost function and a linear state space model is convex and can be efficiently calculated by standard quadratic solvers, which is important for real-time applications, because the optimization results have to be available in one sample time step [17].

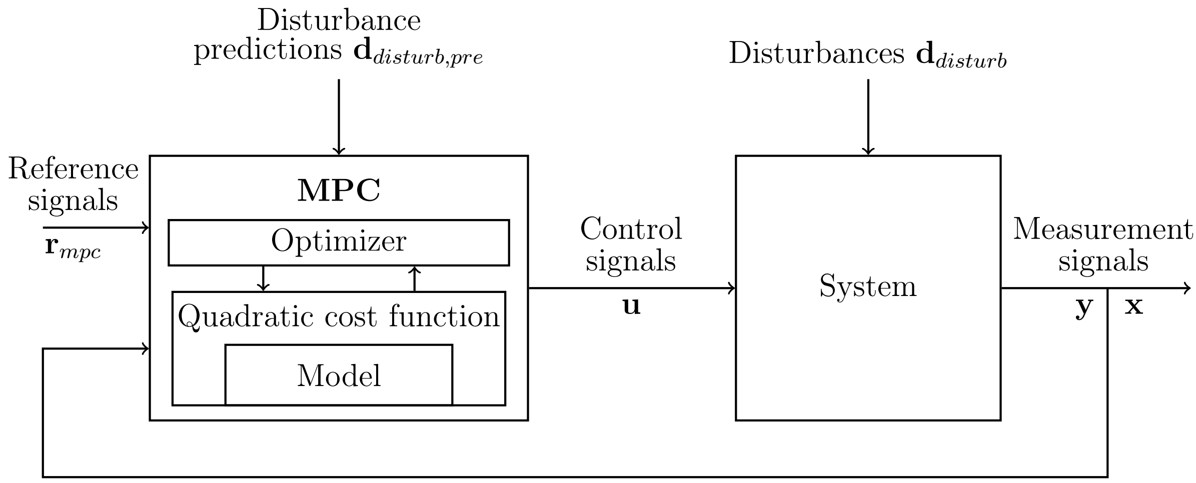


Figure 3.3: Scheme of a control loop with an MPC

The system is influenced by the control signals $\mathbf{u}(k)$ and the disturbances $\mathbf{d}_{disturb}(k)$.

The MPC calculates the optimal input trajectory of a given control horizon H_u, which means H_u changes allowed but H_p time steps, by minimizing a cost function with respect to the future behavior of the plant and the disturbance predictions $\mathbf{d}_{disturb,pre}(k)$. But that means also that a disturbance prediction as well as the future reference trajectory has to be available for the prediction horizon H_p. The future behavior of the plant is calculated for different control inputs over the prediction horizon H_p, which means for H_p time steps, and is evaluated according to the cost function

$$J(k) = \sum_{i=1}^{H_p} \|\mathbf{y}(k+i) - \mathbf{r}_{mpc}(k+i)\|^2_{\mathbf{Q}(i)} + \sum_{i=0}^{H_u-1} \|\Delta\mathbf{u}(k+i)\|^2_{\mathbf{R}(i)}, \tag{3.4}$$

with the input changes $\Delta\mathbf{u}(k+i) = \mathbf{u}(k+i) - \mathbf{u}(k+i-1)$ from one time step to another, where $i = 1, \ldots, H_u$ and the weighting matrices $\mathbf{Q} \in^{p\times p}$ and $\mathbf{R} \in^{m\times m}$. It is assumed that $H_u \leq H_p$ and that $\Delta\mathbf{u}(k+i) = 0$ if $i \geq H_u$, which means that $\mathbf{u}(k+i) = \mathbf{u}(k+H_u-1)$ if $i \geq H_u$, so that the input signal does not change anymore during the optimization process for $i \geq H_u$. The system response $\mathbf{y}(k+i)$, $i = 1, \ldots, H_p$, is predicted by the model of the plant. Exemplary for the first term of (3.4) the quadratic form is defined as follows

$$\sum_{i=1}^{H_p} \|\mathbf{y}(k+i) - \mathbf{r}_{mpc}(k+i)\|^2_{\mathbf{Q}(i)} = \sum_{i=1}^{H_p} (\mathbf{y}(k+i) - \mathbf{r}_{mpc}(k+i))^T \mathbf{Q}(i) (\mathbf{y}(k+i) - \mathbf{r}_{mpc}(k+i)).$$

The cost function (3.4) is minimized by solving the optimization problem

$$\min_{\mathbf{u}\in\mathcal{U}} J(k) \qquad \text{s.t. } \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max}, \tag{3.5}$$

where $\mathcal{U}$ is the set of optimization variables and with constraints on the input signals. The weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are used to adjust the reference tracking and the control effort. An increase of $\mathbf{R}$ penalizes the changes of the control signals more, which results generally in a slower reference tracking. Otherwise, an increase of $\mathbf{Q}$ weights the output reference tracking more, which in general leads to a larger control effort. This means there is always a trade off between the reference tracking and control effort. This is not only true for MPC, but also for other control strategies with reference tracking.

The optimization of the cost function is solved in every time step for the entire prediction horizon, but only the first element of the input trajectory is applied to the plant. This concept is called the moving horizon principle.

3.2.2 Economic model predictive control

An economic model predictive controller (EMPC) uses the same principle as the MPC. The input trajectory of a given prediction horizon is calculated by optimizing a cost function with respect to a linear model of the system. But, insteadt of a quadratic cost function a constrained linear cost function is used and every input variable can be individually weighted or rated by economic costs, which leads to an EMPC [12, 25].

This means, that economic aspects define the goals of advanced control strategies, like cost savings, and because of the linear cost function an economic optimization can be performed [68].

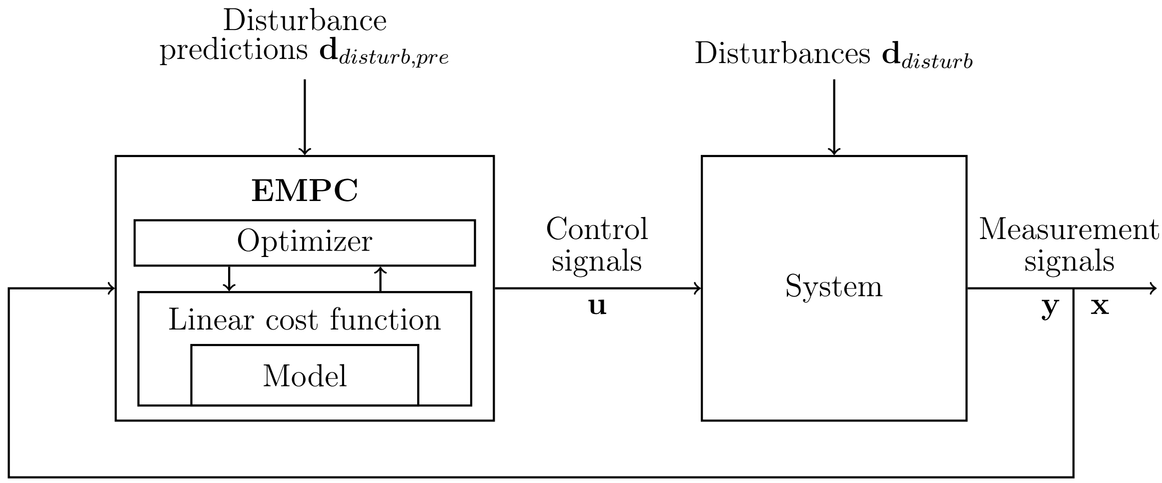


Figure 3.4: Scheme of a control loop with an EMPC

Figure 3.4 shows a scheme of the control loop with an EMPC. For an EMPC with continuous control signals the linear cost function is given by

$$J(\mathbf{u}) = \sum_{i=0}^{H_p-1} \mathbf{c}^T \mathbf{u}(k+i) = \sum_{i=0}^{H_p-1} (c_1 u_1(k+i) + \ldots + c_m u_m(k+i)) \tag{3.6}$$

with the weighting factors or economic costs c_j, the optimization variables u_j and with the index $j = 1, \ldots, m$, where m is the number of inputs, which also corresponds to the number of optimization variables. The cost function shows that no deviation from a reference is evaluated but the control signals are rated by individual costs. Minimizing the cost function $J(\mathbf{u})$ without constraints would lead to minus infinity, due to the linearity of the cost function and the fact, that the range of optimization variables are not resctricted. Because of that, the optimization is performed with constraints, which leads to the optimization problem [25, 30]

$$\min_{\mathbf{u}} J(\mathbf{u}) \tag{3.7}$$

$$\begin{aligned} s.t.\ \ \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \\ \mathbf{u}_{min} \leq &\mathbf{u}(k) \leq \mathbf{u}_{max} \\ \Delta\mathbf{u}_{min} \leq &\Delta\mathbf{u}(k) \leq \Delta\mathbf{u}_{max} \\ \mathbf{y}_{min} \leq &\mathbf{y}(k) \leq \mathbf{y}_{max}. \end{aligned} \tag{3.8}$$

The optimization is calculated with respect to the linear state space model of the system. The measured actual states $\mathbf{x}(k)$ of the plant are fed back to the controller to initialize the model of the system for the optimization. Additionally, the input signals $\mathbf{u}(k)$, the outputs signals $\mathbf{y}(k)$ and the changes of the input signals $\Delta\mathbf{u}(k)$ are restricted with lower and upper limits for the optimization. The result of (3.7) is the input signals with minimized costs for the entire prediction horizon. For the EMPC application also the moving horizon principle is used.

3.2.3 Nonlinear model predictive control

A nonlinear model predictive controller (NMPC) uses the introduced principles of model predictive control. If a linear model is not accurate enough to represent the dynamical behavior of the plant and nonlinear effects have to be taken into account, the resulting model is a nonlinear model. Using a nonlinear model for the MPC leads to a nonlinear MPC problem [26]. In general, the NMPC optimization problem is not convex, which can lead to a high computational effort. This causes the problem for applications, that on the one hand, the global optimum is not always found and on the other hand, the optimization time increases due to the complexity of the problem and can easily be longer than the sample time, which means that the optimization result is not solved in one time step [17]. But nevertheless, this can also happen using a linear MPC with a convex optimization problem. The sample time has to chosen in accordance to the complexity of the optimization problem.

3.3 Data-based control methods

Data-based control methods are a large field of control methods, which use collected data of a system for calculating the input signals of next iterations, adjusting model or control parameters or use the data to learn from the past. Data-based methods, like machine learning or artificial neuronal networks are not in focus of this thesis and will be not considered any further. A short summary on adaptive control will be given next, but is not in the focus of this thesis. A detailed description on iterative learning control will be stated next.

3.3.1 Adaptive control

Adaptive control is used to adapt the model parameters of a system or a controller by using the measurement data of the system, which are fed back to the controller. Processes can be changed or the environment and disturbances differ over time. If the system dynamics vary, the initial set of parameters are not suitable anymore [37]. For example, the mass of an airplane changes due to the fuel consumption during the flight. An adaptive controller takes the changes of a system or process into account. On the one hand, if a model is used for control purposes the adaptive controller adjusts the model parameter to the new environmental conditions. On the other hand, the adaptive controller can be used for tuning the parameters of ancillary controllers. One known field of adaptive control is aircrafts where the controllers have to deal with different masses and Mach numbers [15, 74]. In the field of heating systems for buildings adaptive control methods are also applied. For example, a self-tuning controller for the optimum start time of the heating system is investigated in [27] and an adaptive predictive controller for a floor heating system is introduced in [20].

3.3.2 Iterative learning control

Iterative learning control (ILC) is based on the idea that the performace of a system, which executes the same task again and again, can be improved by learning from previous trials or iterations [18, 76]. This is well known and established for industrial periodic processes where the same task is performed multiple times. Also for periodic references or periodic disturbances an ILC is a suitable choice. The ILC uses the measurement data of a past iteration to calculate the input trajectory of the next iteration. This means, that the performance increases from one iteration to the next iteration if the process has the same periodicity from one iteration to the next iteration.

A basic ILC algorithm calculates the input signal of the next iteration $\mathbf{u}_{d+1}(k)$ by using the input signal of the last iteration $\mathbf{u}_d(k)$ and the deviation of the output signal $\mathbf{y}_d(k)$ from the reference signal $\mathbf{r}_{ilc}(k)$, weighted by the learning gain $\gamma \in \mathbb{R}$, [11]

$$\mathbf{u}_{d+1}(k) = \mathbf{u}_d(k) + \gamma\left(\mathbf{r}_{ilc}(k) - \mathbf{y}_d(k)\right) = \mathbf{u}_d(k) + \gamma \mathbf{e}_d(k), \tag{3.9}$$

with the index of the iteration $d \in \mathbb{N}$ and the error signal $\mathbf{e}_d(k)$, which is deviation from the reference $\mathbf{e}_d(k) = \mathbf{r}_{ilc}(k) - \mathbf{y}_d(k)$. As before, k denotes the discrete time index. Figure 3.5 shows a scheme of the control loop with the system and the ILC.

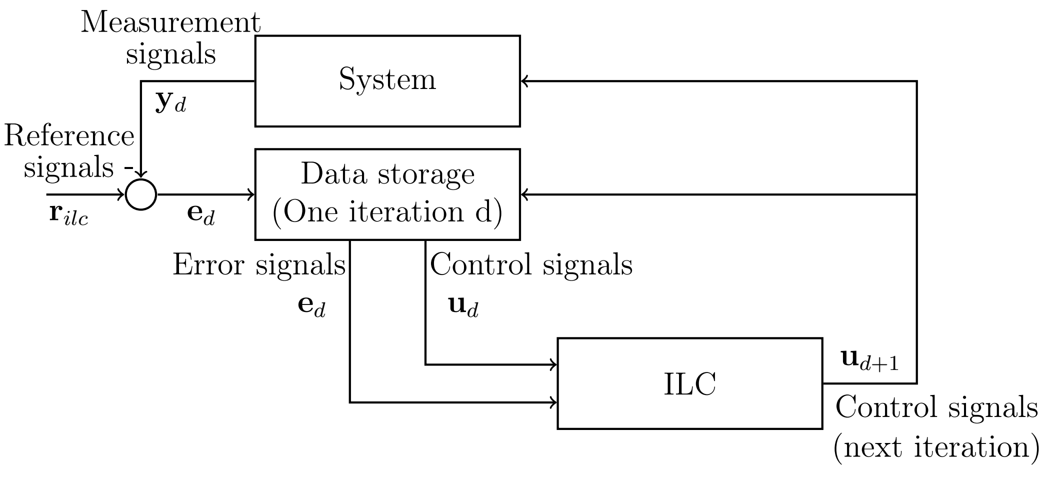


Figure 3.5: Scheme of a system with an ILC

The input and error data of the last iteration d have to be stored for the calculation of the ILC update (3.9) and the input signals are calculated for the entire next iteration. The number of stored elements for one signal is given by the quotient of the time length of one iteration t_t and the sample time t_s, which results in the number of samples per iteration N_m.

The focus in the following is on the iterative learning control for the adjustment of the control signals or the references by using measurement data and not on the adaptive control where the model parameters or controller parameters are adjusted. Also different predictive control approaches will be investigated. Real-time implementations of the presented control strategies will be applied to heating systems.

3.4 Control for a heating system example

The common control strategy for heating systems is a proportional-integral controller, where the reference is the supply temperature given by a heating curve with respect to the outside temperature. Another control strategy which is often discussed in literature is an MPC, which uses a linear model of the plant and takes the weather forecast into account. Because of this, a heating system example is introduced with the two well known control strategies, the PI controller and the MPC. The simulation results of these examples will be compared to each other and serve as basis for further discussions. For control systems the controllability and observability are important properties for the regulation of the system [50]. The used state space models should be controllable and observable. This is checked and true for all models used in this thesis. It is also assumed that all states are measurable, otherwise this is stated in the text.

3.4.1 Heating system model

The heating system consists of a supplier and a consumer and the single components are introduced in Section 2.2. Figure 3.6 shows a scheme of the heating system. The supplier includes a boiler with the burner, which satisfies the heat demand of the consumer. The signal $\alpha \in [0,1]$ controls the thermal power of the burner. The consumer contains the building and the radiators. These components are modeled by heat balances according to the equations (2.29), (2.30), and (2.31). The pump provides a volume flow $\dot{V}$. The model of the four-way valve is a linear black-box model, which is estimated by measurement data and includes two states, three inputs and two outputs. The inputs of the black-box model are the supply temperature of the boiler $T_{s,b}$, the return temperature of the radiator $T_{r,r}$, and the control signal of the valve $\phi \in [0,1]$. The outputs of the four-way valve model are the supply temperature of the radiator $T_{s,r}$ and the return temperature of the boiler $T_{r,b}$. Depending on the valve position the valve mixes cold water of the radiator return to the supply of the radiator, and warm water from the boiler supply to the boiler return. The overall model of the heating system consists of two control signals α and ϕ as inputs and two additional inputs, the outside temperature T_{out} and the volume flow $\dot{V}$ which are interpreted as disturbance inputs. The output signals are the supply and return temperature of the radiator $T_{s,r}$ and $T_{r,r}$, the room temperature T_{room}, and the supply temperature of the boiler $T_{s,b}$.

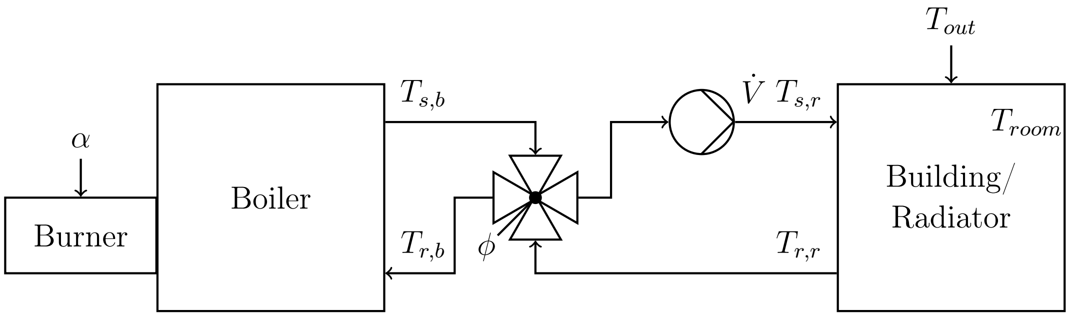


Figure 3.6: Scheme of the heating system

The entire model of the heating system is implemented in Simulink and the unknown parameters of the overall model are estimated and validated by measurement data using the Simulink parameter estimation tool. The Simulink model is linearized by using the MATLAB function *linmod*. The resulting continuous-time state space model is discretized applying the MATLAB function *c2d* to get a linear discrete-time state space model of the heating system, which is used for the MPC.

The heating system is a real prototype test facility for an office of a non-residential building. A Figure of the plant is shown in Section 6.3.1. PI control is a standard control strategy for heating systems. The reference is the supply temperature calculated by a heating curve with respect to the outside temperature. The results of the simulation with the PI controller is compared to the simulation results with an MPC.

3.4.2 Proportional-integral control for a heating system

PI control is one of the standard control strategies and a special form of the general PID controller [49]. The PID controller is introduced in Section 3.1.3 and Figure 3.1 shows a scheme of the control loop.

The control error (3.2) is used to calculate the input signals $\mathbf{u}(t)$. For the heating system example the control error is the difference between the reference supply temperature r_s and the supply temperature T_s and the disturbance $\mathbf{d}_{disturb}$ is the outside temperature T_{out}. The heating curve is introduced in Section 3.1.1 and shown in Figure 3.2. The PI controller uses only the proportional and integral part of equation (3.3) as follows

$$\mathbf{u}(t) = \mathbf{K}_p e(t) + \mathbf{K}_I \int_0^t e(\tau)\mathrm{d}\tau, \tag{3.10}$$

The PI controller reacts on the actual control error $e(t)$ without any knowledge about the future dynamical behavior of the plant or the disturbances. To consider the future dynamical behavior for the control of the heating system an MPC is used.

3.4.3 Model predictive control for a heating system

The linear MPC calculates the control inputs $\mathbf{u}(k)$ by using a discrete-time linear state space model of the plant. Basics of model predictive control are introduced in Section 3.2.1. The linearized model of the heating system, as introduced in Section 3.4.1, is used to predict the system behavior for future control inputs. The optimal future input trajectory is calculated by minimizing the quadratic cost function (3.4) with respect to the system model. Figure 3.3 shows the control loop of a system with an MPC.

The heating system is influenced by the two control signals α and ϕ. The disturbance signal $\mathbf{d}_{disturb}$ is the outside temperature T_{out} and the disturbance prediction $\mathbf{d}_{disturb,pre}$ is the outside temperature forecast $T_{out,for}$. The reference $\mathbf{r}_{mpc}$ for the entire prediction horizon H_p is calculated according to the heating curve shown in Figure 3.2 and the outside temperature forecast $T_{out,for}$.

The MPC uses the moving horizon principle, which means that the optimization problem (3.5) is solved in every time step k, but only the first element of resulting optimal input trajectory is applied to the plant. The three measured state signals $\begin{bmatrix} T_{s,b} & T_{r,r} & T_{room} \end{bmatrix}$ of the plant are fed back to the controller to initialize the model of the system in every time step. The MPC is implemented in Simulink using the MPC block of the MATLAB MPC Toolbox [2]. The two states of the four-way valve black-box model are not measurable. They are marked as unknown for the Simulink MPC block and the two states are estimated by the internal observer of the MPC block. A detailed description can be found in the documentation of the MPC Toolbox [32].

The MPC optimization problem (3.5) with a quadratic cost function and linear constraints by using a linear model is convex and can be effecitenly computed with standard quadratic solvers [17]. This is important for MPC applications because it has to be ensured that the optimization result is available at the next time step. The following Section shows a comparison of the simulation results of a PI controller and an MPC applied to the heating system model.

3.4.4 Simulation results of a heating system example

The heating system model, as shown in Section 3.4.1, is used for simulations with a PI controller, as introduced in Section 3.1.3 and 3.4.2, and an MPC, as described in Section 3.2.1 and 3.4.3. The simulation results are compared to each other. The main goal of the controllers is to keep the room temperature in a given comfort zone, which means the rooms of the building have to be warm enough for the users. Otherwise, the heating energy and in the end the money should not be wasted. For both controllers the same heating curve, as presented in Figure 3.2, is used as reference supply temperature for the simulation and the same outside temperatures T_{out} as disturbance signal. The outside temperature is also used as disturbance prediction for the MPC and for the calculation of the reference supply temperature for the prediction horizon. According to this reference each controller uses the two control signals α and ϕ to adjust the supply temperature of the radiator $T_{s,r}$. The MPC uses a discrete-time linearized model of the heating system and for the simulation a prediction horizon of $H_p = 5$ h, a control horizon of $H_u = 4$ h and a sample time of $t_s = 60$ s.

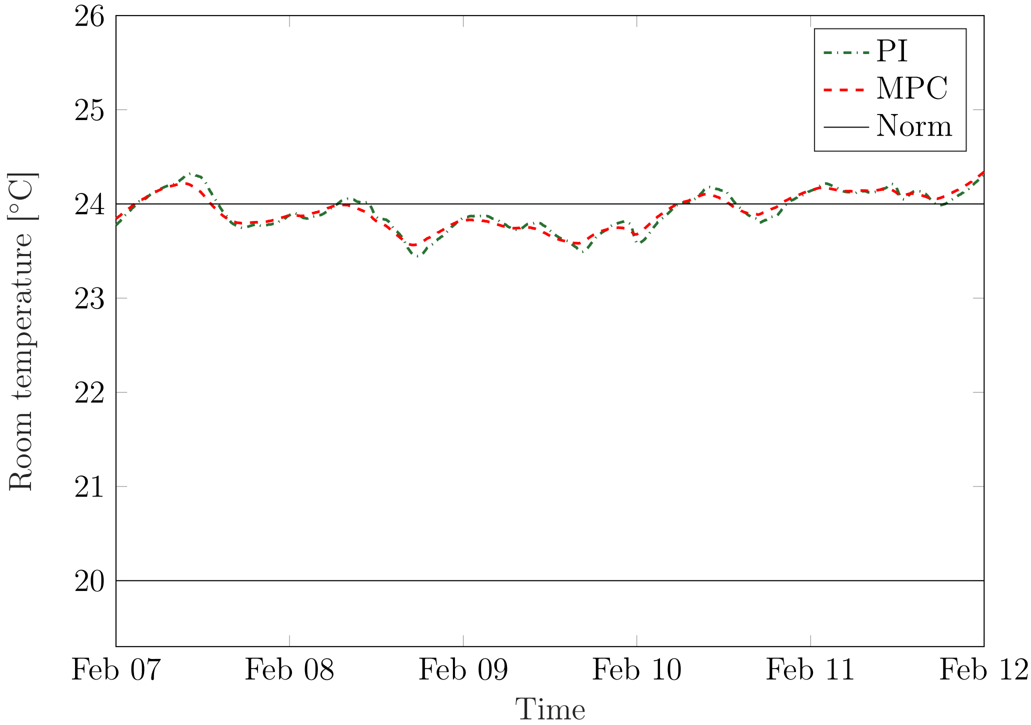


Figure 3.7: Simulated results of the PI controller compared to the MPC controller

The comparison of simulated room temperatures are exemplarily shown for five days in Figure 3.7. The desired comfort zone for the room temperature is defined according to the German norm DIN-EN15251 [24]. This norm defines a comfort room temperature of 22 ± 2 °C in winter and is indicated by the black solid lines in Figure 3.7.

The results show that the room temperatures are in the same range for both controllers and on the upper limit of the defined comfort zone. The progression of the room temperatures are nearly the same, but the room temperature of the MPC is a little smoother. The evaluation of the daily heating power over the mean values of the daily outside temperatures for two months is presented in Figure 3.8. For most of the days the MPC has decreased the heating power and thus the energy consumption. In total, the MPC reduces the heating power at about 4 % for the simulated time period of two months. This simulation shows that even for small heating systems, which includes one room only and thus a short delay time, the MPC is able to decrease the energy consumption.

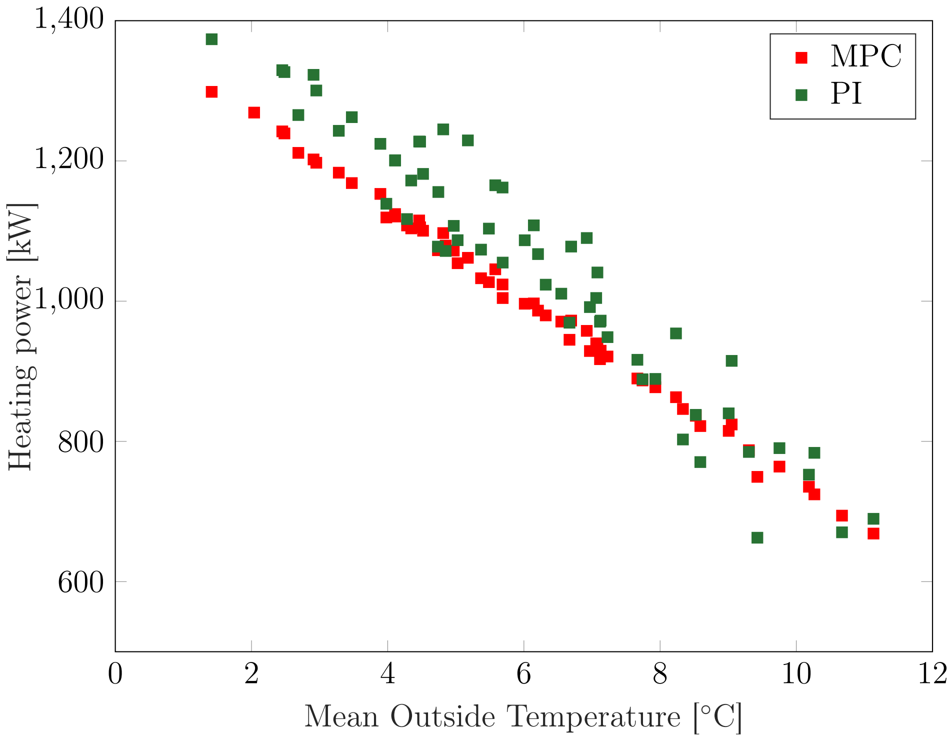


Figure 3.8: Comparison of the heating power of the PI controller and the MPC

The comparison of the control strategy with an MPC and a PI controller has shown, that the MPC decreases the heating power by keeping the room temperature in the same range as with the PI controller, which points out the advantage of model predictive control strategies with respect to the weather forecast for heating systems. Nevertheless, a linearized model of the heating system is used, which is inherited a multilinear model. Using a multilinear model would lead to a nonlinear MPC, which leads in general to a complex nonlinear optimization problem. This results in the question, how the structure of the MPC optimization problem is, if the multilinear model is used directly instead of a linearized model? Specifically, investigations about convexity properties of the optimization problem with a multilinear model are interesting for applications. Also the problem to find a suitable reference trajectory for the system still remains. This opens the question if there are predictive control methods which can keep the room temperature in the desired comfort zone without using any reference? Otherwise, if the well known linear MPC is used, are there possibilities to use measurement data for control, besides the modeling part, to improve the control results of a linear MPC?

The next chapter focuses in the investigation of a model predictive control approach without any references and the convexity properties using an MTI model for the MPC.

Chapter 4

Predictive control for linear and multilinear systems

The standard linear MPC with a quadratic cost function, which uses a linear state space model, was introduced in Section 3.4.3. The results of the linear MPC depends on the accuracy of the linear model as well as on the given reference trajectory. Due to the multiplication of the volume flow and the temperatures, heating system models are in the class of multilinear models. On the one hand, this leads to the question, how the structure of the MPC optimization problem changes related to the convexity of the problem if the multilinear model is used instead of a linearized model. On the other hand, there is the question how the reference trajectory can be optimized or how a model predictive control method can be applied without the use of a reference trajectory?

4.1 Economic model predictive control with continuous and discrete control signals

The basic pirinciple of an EMPC is already intrudoced in Section 3.2.2. An EMPC uses a model of the system for the prediction of the dynamical system behavior and a cost function is optimized by an optimization algorithm, which means that the basic principle of an EMPC is similar to the linear MPC, as introduced in Section 3.2.1. But the cost function of the EMPC differs from the quadratic cost function of the MPC. An EMPC uses a linear cost function (3.6). Due to the linearity of the cost function it is possible to use economic costs in e.g. euros of a process as weighting factors. This means, that the optimization of an EMPC can be an economic optimization and the real costs in euro of a specific process or system are minimized. The requirement of such an economic optimization is the knowledge about the specific process costs, otherwise it is a linear optimization with weighting or tuning factors. The EMPC does not use and evaluate a reference for the optimization. This is another difference to the standard MPC, where the deviation from a reference was evaluated for the optimization.

Figure 3.4 shows the control loop of a system with an EMPC. The comparison with the control loop of the standard MPC, in Figure 3.3, shows that the basic concept of these two controllers is the same. The differences are the linear cost function and the absence of any references. An EMPC for the control of continuous control signals is introduced and will be extended to the control of discrete and continuous control signals. Both controllers are applied to a heating system example for the control of the heating power of a boiler and the position of a valve. The results will be compared to the results of a linear MPC.

4.1.1 Economic model predictive control for continuous control signals

For an EMPC with continuous control signals the linear cost function (3.6) is already introduced in Section 3.2.2. The result of optimization problem (3.7) is the input signals with minimized costs for the entire prediction horizon. Assuming, that the constraints (3.8) are linear as well as the cost function, a linear optimizer can be used for solving the optimization problem (3.7), such as the *linprog* function of the MATLAB Optimization Toolbox [6].

The *linprog* function of MATLAB uses the standard expression for linear programming

$$\min_{\mathbf{x}_{opt}} \mathbf{c}^T \mathbf{x}_{opt}(k) \tag{4.1}$$

$$s.t.\ \ \mathbf{A}_{ineq}\mathbf{x}_{opt}(k) \leq \mathbf{b}_{ineq} \tag{4.2}$$

where $\mathbf{x}_{opt}(k)$ is the vector of optimization variables and the matrix $\mathbf{A}_{ineq}$ and the vector $\mathbf{b}_{ineq}$ define the linear inequality constraints. The constraints (3.8) have to be rewritten according to the general form of inequality constraints (4.2). The temporal evolution for the prediction horizon H_p of the state space model is written as a lifted system for the inequality constraints.

It may not always be possible to meet the hard constraints of the optimization problem, particularly if the system signals are close to the given constraints and the model is initialized by the measured values for each time step k. If one signal is outside of the defined constraints, this would lead to an infeasible optimization problem and the control algorithm stops, e.g., for heating systems, a measured temperature is outside of the desired range. The EMPC problem is relaxed by introducing some slack variables $\mathbf{s}(k)$ for the outputs $\mathbf{y}(k)$ with the associated weighting factors or economic costs $\boldsymbol{\eta}$. The weighting factors $\boldsymbol{\eta}$ of the additional variables should be set sufficiently large, such that the violation of the output constraints are penalized by high costs and the constraints are met when ever possible.

This leads to the new cost function

$$\begin{aligned} J(\mathbf{u}, \mathbf{s}) &= \sum_{i=0}^{H_p-1} \left(\mathbf{c}^T\mathbf{u}(k+i) + \boldsymbol{\eta}^T\mathbf{s}(k+i)\right) \\ &= \sum_{i=0}^{H_p-1} \left(c_1 u_1(k+i) + \ldots + c_m u_m(k+i) + \eta_1 s_1(k+i) + \ldots + \eta_p s_p(k+i)\right) \end{aligned} \tag{4.3}$$

and the optimization problem

$$\min_{\mathbf{u},\mathbf{s}} J(\mathbf{u},\mathbf{s}) \tag{4.4}$$

$$\begin{aligned} s.t.\ \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \\ \mathbf{u}_{min} &\leq \mathbf{u}(k) \leq \mathbf{u}_{max} \\ \Delta\mathbf{u}_{min} &\leq \Delta\mathbf{u}(k) \leq \Delta\mathbf{u}_{max} \\ \mathbf{y}_{min} &\leq \mathbf{y}(k) + \mathbf{s}(k) \\ \mathbf{y}_{max} &\geq \mathbf{y}(k) - \mathbf{s}(k) \\ \mathbf{s}(k) &\geq 0, \end{aligned} \tag{4.5}$$

with the added slack variable and the constraint that $\mathbf{s}(k)$ has to be greater or equal to zero. The resulting inequality constraint matrices $\mathbf{A}_{ineq}$ and $\mathbf{b}_{ineq}$ of the constraints (4.5) are introduced in the appendix Section C.1.

Constant minimum and maximum values are not suitable for the constraints of any signals, e.g., the room temperature or the supply temperature of a heating system of non-residential buildings should be reduced during the night. With respect to this fact the constraints (4.5) are enlarged to time dependent minimum and maximum values for the output signals

$$\begin{aligned} \mathbf{y}_{min}(k) &\leq \mathbf{y}(k) + \mathbf{s}(k) \\ \mathbf{y}_{max}(k) &\geq \mathbf{y}(k) - \mathbf{s}(k). \end{aligned} \tag{4.6}$$

For the application of an EMPC, the moving horizon principle is used as introduced in the MPC Section 3.4.3 and also a disturbance prediction has to be provided. The introduced EMPC with the cost function (4.3) and the optimization problem (4.4) is only applicable to systems with continuous control signals. Some applications have additional discrete control signals, like a simple signal for switching a system on and off. This leads to the question: How can this EMPC concept be adapted to systems with discrete and continuous control signals?

4.1.2 Economic model predictive control for discrete and continuous control signals

The extension of an EMPC with only continuous signals to an EMPC with discrete and continuous control signals changes the optimization problem from a linear optimization problem to a mixed-integer optimization problem and thus, a nonlinear optimization problem. A mixed-integer optimization problem can lead to high computational effort and a lengthy solving time depending on the complexity of the problem, for instance the number of discrete variables [17, 33]. For solving a mixed-integer optimization problem the *intlinprog* function of the MATLAB Optimization Toolbox was used. The benefit of that function is that the general structure of the EMPC optimization problem for continuous signals, as introduced in Section 4.1.1, is preserved. Nevertheless, the optimization problem (4.4) has to be adjusted. First of all, the linear state space model is replaced by a hybrid state space model (2.16) and (2.17), as introduced in Section 2.1.4, with the additional discrete control inputs $\mathbf{u}_{dis}(k) \in \mathbb{N}_0^{m_{dis}}$.

Two new variables are introduced for each discrete input $\mathbf{u}_{dis}(k)$. These new variables separate the up and down switching itself from the discrete input signals $\mathbf{u}_{dis}(k)$ and will be denoted by $\mathbf{u}_{up}(k) \in \mathbb{N}_0^{m_{dis}}$ and $\mathbf{u}_{down}(k) \in \mathbb{N}_0^{m_{dis}}$. The separation of the up and down switching process enlarges the opportunities to rate also up and the down switching with its own costs, besides from rating the absolute value of $\mathbf{u}_{dis}(k)$. The values of this three variables at time k result in the discrete input signal of the next time step

$$\mathbf{u}_{dis}(k+1) = \mathbf{u}_{dis}(k) + \mathbf{u}_{up}(k) - \mathbf{u}_{down}(k). \tag{4.7}$$

All of these additional variables $\mathbf{u}_{dis}$, $\mathbf{u}_{up}$ and $\mathbf{u}_{down}$ have their individual weighting vectors $\mathbf{c}_{dis}$, $\mathbf{c}_{up}$ and $\mathbf{c}_{down}$, which also means that the up and down switching can be associated with different costs. Furthermore, the variables $\mathbf{u}_{up}$ and $\mathbf{u}_{down}$ are used to restrict the step size within one time step to another by setting an upper and lower limit as follows

$$\begin{aligned}\mathbf{u}_{up,min} &\leq \mathbf{u}_{up}(k) \leq \mathbf{u}_{up,max}\\ \mathbf{u}_{down,min} &\leq \mathbf{u}_{down}(k) \leq \mathbf{u}_{down,max}.\end{aligned}$$

These considerations lead to the new cost function

$$\begin{aligned}J(\mathbf{u}, \mathbf{u}_{dis}, \mathbf{u}_{up}, \mathbf{u}_{down}, \mathbf{s}) = &\sum_{i=0}^{H_p-1} (\mathbf{c}^T\mathbf{u}(k+i) + \mathbf{c}_{dis}^T\mathbf{u}_{dis}(k+i) + \mathbf{c}_{up}^T\mathbf{u}_{up}(k+i)\\ &\quad + \mathbf{c}_{down}^T\mathbf{u}_{down}(k+i) + \boldsymbol{\eta}^T\mathbf{s}(k+i))\\ = &\sum_{i=0}^{H_p-1} (c_1u_1(k+i) + \ldots + c_mu_m(k+i)\\ &\quad + c_{dis,1}u_{dis,1}(k+i) + \ldots + c_{dis,m_{dis}}u_{dis,m_{dis}}(k+i)\\ &\quad + c_{up,1}u_{up,1}(k+i) + \ldots + c_{up,m_{dis}}u_{up,m_{dis}}(k+i)\\ &\quad + c_{down,1}u_{down,1}(k+i) + \ldots + c_{down,m_{dis}}u_{down,m_{dis}}(k+i)\\ &\quad + \eta_1s_1(k+i) + \ldots + \eta_ps_p(k+i))\end{aligned} \tag{4.8}$$

and the new optimization problem

$$\min_{\mathbf{u},\mathbf{u}_{dis},\mathbf{u}_{up},\mathbf{u}_{down},\mathbf{s}} J(\mathbf{u},\mathbf{u}_{dis},\mathbf{u}_{up},\mathbf{u}_{down},\mathbf{s}) \tag{4.9}$$

$$\begin{aligned}
s.t.\ \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_{dis}\mathbf{u}_{dis}(k)\\
\mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)\\
\mathbf{u}_{dis}(k+1) &= \mathbf{u}_{dis}(k) + \mathbf{u}_{up}(k) - \mathbf{u}_{down}(k)\\
\mathbf{u}_{min} &\leq \mathbf{u}(k) \leq \mathbf{u}_{max}\\
\Delta\mathbf{u}_{min} &\leq \Delta\mathbf{u}(k) \leq \Delta\mathbf{u}_{max}\\
\mathbf{u}_{dis,min} &\leq \mathbf{u}_{dis}(k) \leq \mathbf{u}_{dis,max} \qquad (4.10)\\
\mathbf{u}_{up,min} &\leq \mathbf{u}_{up}(k) \leq \mathbf{u}_{up,max}\\
\mathbf{u}_{down,min} &\leq \mathbf{u}_{down}(k) \leq \mathbf{u}_{down,max}\\
\mathbf{y}_{min}(k) &\leq \mathbf{y}(k) + \mathbf{s}(k)\\
\mathbf{y}_{max}(k) &\geq \mathbf{y}(k) - \mathbf{s}(k)\\
\mathbf{s}(k) &\geq 0.
\end{aligned}$$

The constraints (4.10) of the optimization problem have to rewritten according to the general form of the inequality constraints $\mathbf{A}_{ineq}\mathbf{x}_{opt}(k) \leq \mathbf{b}_{ineq}$. Both EMPC controllers will be applied to a heating system example.

4.1.3 Simulation results of a heating system example

First, the EMPC with continuous control signals is applied to a heating system example, with the goal to keep the room temperature in a defined comfort zone without using any references. Second, the EMPC with continuous and discrete control signals is applied to the same heating system, with the assumption, that one control signal is only controllable in discrete steps. The modeling of the heating system happens as introduced in Section 2.2. The heating system consists of a boiler, which satisfies the heat demand of the consumer. The consumer includes the radiator and the building with thermal losses to the environment. A pump provides a constant volume flow $\dot{V}$ and a four-way valve mixes cold water from the return of the radiator to the supply of the radiator and warm water from the supply of the boiler to the return of the boiler according to the equations (2.35) and (2.36). The estimated parameters are presented in the appendix Section A.2. The heating system is based on a real plant, which is a heating system test facility for an office of a non-residential building and previously introduced in Section 3.4.1. Figure 3.6 presents a scheme of the system.

The heating system has two continuous input signals α and ϕ. The signal α controls the thermal power of the boiler and the signal ϕ defines the mixing ratio of the four-way valve. The model has three outputs, the supply temperature of the boiler $T_{s,b}$, the return temperature of the radiator $T_{r,r}$ and the room temperature T_{room}. The three outputs are also the three states of the model. The return temperature of the boiler is denoted by $T_{r,b}$ and the supply temperature of the radiator by $T_{s,r}$. A linearized discrete-time state space model of the system is used for the EMPC.

For the simulation, a prediction horizon of $H_p = 2$ hours and a sample time of $t_s = 60$ seconds is used. As the forecast of the outside temperature, measurement values of the outside temperature were used. The EMPC optimization problem (3.7) for continuous control signals is solved by using the *linprog* optimizer of the MATLAB Optimization Toolbox. Table 4.1 presents the different constraints which are used for the EMPC with continuous control signals.

Table 4.1: List of the constraints for the EMPC with continuous control signals

Parameter	Lower limit daytime (nighttime)	Upper limit
T_{room}	22 °C (19 °C)	26 °C
$T_{s,b}$	15 °C	70 °C
$T_{r,r}$	15 °C	70 °C
ϕ	0	1
α	0	1

The results of the simulation with the EMPC are compared to the results when a linear MPC, as introduced in Section 3.4.3, controls the heating system. The simulation results of the room temperature T_{room} when an EMPC controls the heating system compared to the room temperature when an MPC controls the same heating system are shown in Figure 4.1. Both simulations use the same outside temperatures as disturbance signal and also the same outside temperature forecast signal, which is not an outside temperature forecast but real measured outside temperatures and the same as the disturbance signal itself.

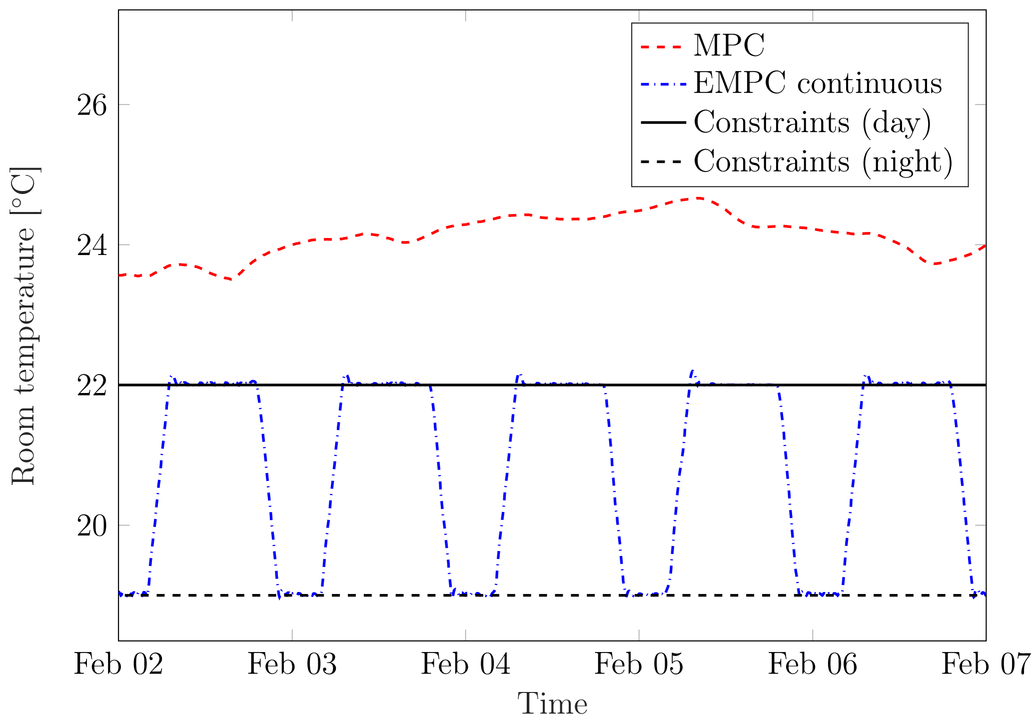


Figure 4.1: Comparison of the simulated room temperature with an MPC and an EMPC

The simulation results of the MPC are the same as shown for the comparison with the PI controller in Figure 3.7. The result with the EMPC shows that the room temperature can be kept within the defined limits for the room temperature and is moving on the lower constraint of 22 °C. Compared to the result with the MPC the room temperature is decreased significantly but still fulfills the comfort requirements of the room temperature. A reduction of the room temperature leads to reduced energy consumption because the supply temperature also decreases with the reduced room temperature. Figure 4.2 presents the evaluation of the daily heating power over the mean values of the daily outside temperatures for two months of the heating season (February and March). The comparison shows that the EMPC reduces the daily heating power significantly in contrast to the MPC with an over all reduction of about 15 %. Due to the introduced time depending constraints of the output signals (4.6), a night setback for the heating system is realized, which is also shown in Figure 4.1.

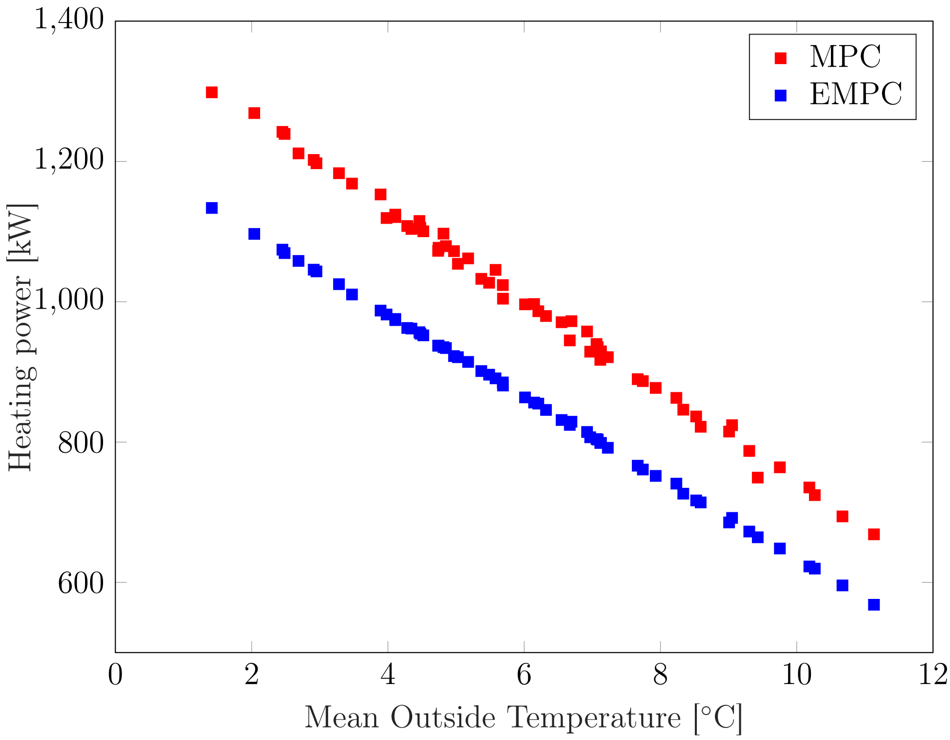


Figure 4.2: Evaluation of the heating power consumption

The EMPC with discrete and continuous control signals were applied to the same heating plant, with the assumption that the thermal power of the boiler is only adjustable by the discrete input signal α_{dis} witch the three levels 0-1-2. The first level means that the boiler is switched off, the second level means that 50 % of the maximum thermal power is provided and the third level means that the boiler provides the full thermal power. The constraints are summarized in Table 4.2. The constraint that $u_{up,\alpha} + u_{down,\alpha}$ is in the interval $[0, 1]$ ensures that the boiler is either switched up or down. The change to a discrete control signal of the boiler also affects the model class and is part of the hybrid state space model class as introduced in Section 2.1.4. All other parameters remain as before. The simulation results of the room temperature are shown in Figure 4.3. The room temperature can also be kept in the comfort zone in the same way, even if the boiler is only controllable by a discrete signal.

Table 4.2: List of the constraints for the EMPC with discrete and continuous control signals

Parameter	Lower limit daytime (nighttime)	Upper limit
T_{room}	22 °C (19 °C)	26 °C
$T_{s,b}$	15 °C	70 °C
$T_{r,r}$	15 °C	70 °C
ϕ	0	1
α_{dis}	0	2
$u_{up,\alpha}$	0	1
$u_{down,\alpha}$	0	1
$u_{up,\alpha} + u_{down,\alpha}$	0	1

Figure 4.4 shows the differences to the continuous controlled boiler in the supply temperatures of the boilers. The supply temperature of the boiler with the discrete control signal shows a characteristic oscillation due to the switching of the boiler.

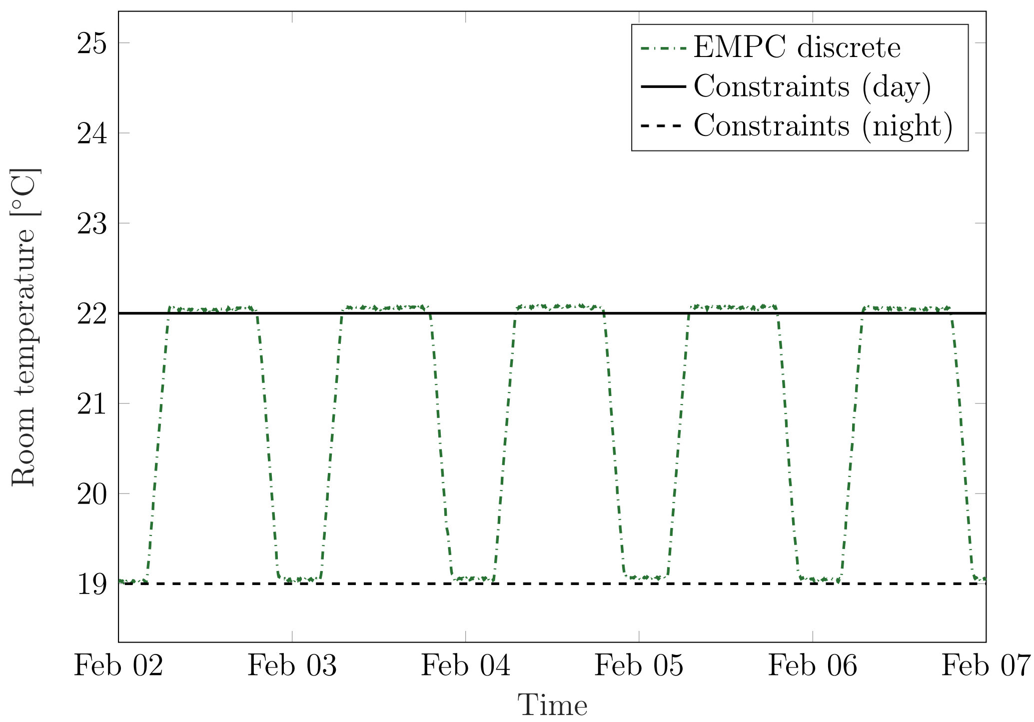


Figure 4.3: Simulated room temperature of an EMPC with a discrete control signal

The comparison of the time for solving the EMPC optimization problem of the two different controllers shows an increasing time for the EMPC from 0.07 seconds with the

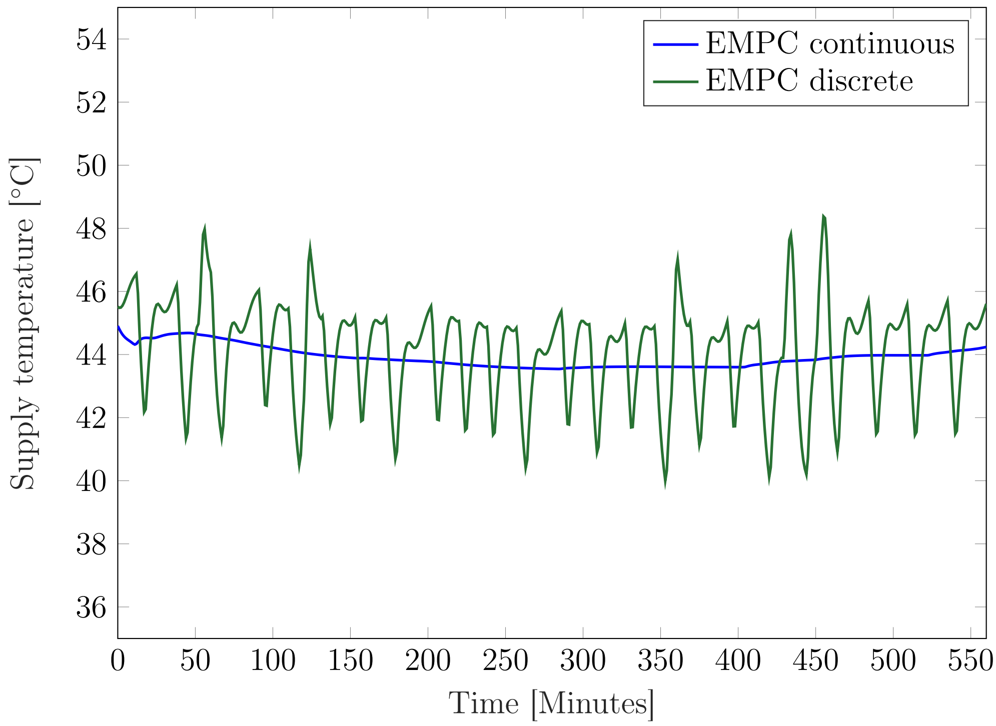


Figure 4.4: Supply temperature of an EMPC with a discrete control signal

continuous control signals to 0.11 seconds with the discrete control signal, which is equivalent to a factor of 1.5. For both tests, the prediction horizon was set to $H_p = 60$ and the sample time to $t_s = 60$ seconds. For the evaluation of the optimization time duration, the EMPC optimization problem was solved 2880 times for each controller. This corresponds to two days by using a sample time of $t_s = 60$ seconds. The duration is taken for every individual optimization process and the presented values are the mean values of the 2880 measured time durations. Changing the prediction horizon from $H_p = 60$ to $H_p = 70$ affects the computation time significantly. For a prediction horizon of $H_p = 70$, the computation time increases from 0.07 seconds to 0.09 seconds with the continuous control signals and from 0.11 seconds to 3.7 seconds with the discrete control signal. This means, that the optimization time with a prediction horizon of $H_p = 70$ increases by a factor of 38.9 for discrete control signals compared to the computation time with the continuous control signals. This investigation reveals, that the solving of a mixed-integer problem is much more complex than a linear one, which leads to an increased optimization time. The computation of the optimization problem was performed on a computer with an Intel Core i7-3540M processor (3.0 GHz, up to 3.7 GHz, 4 MB) and 16 GB Ram.

Two economic model predictive controllers for continuous control signals as well as for discrete and continuous control signals were introduced and applied to a heating system example. The simulation results have shown that the room temperature can be kept in the given comfort zones by using a constraint optimization problem without any references, for instance a heating curve. The advantage of the EMPC is that no references are needed and the linear cost function opens the possibility to use real costs as weighting factors if the process costs of the system are known. Nevertheless, the EMPC also depends on the accuracy of the linear model. The introduced EMPC, as well as the MPC, use a linearized model of the system. The Simulink model is linearized by using the MATLAB function *linmod*. Due to the modeling of heating systems by using heat balances, the resulting models are part of the MTI model class, which leads to the question how are the properties of an MPC optimization problem if an MTI model is used instead of a linear model?

4.2 Properties of the model predictive control problem for multilinear systems

For model predictive control applications where a linear model is insufficient for the description of the physical behavior of the plant, nonlinearities have to be taken into account. For application areas where the system dynamics are modeled by energy, mass or heat balances, like heating systems or chemical processes, MTI systems are suitable to approximate the main dynamics of such systems [62]. The class of MTI models are introduced in Section 2.1.2 and extend the class of linear systems to squares-free multiplication of different states and inputs.

The linear discrete-time model predictive control problem was already introduced in Section 3.4.3. The use of a linear model in combination with a quadratic cost function leads to a convex optimization problem, which can be solved efficiently by standard quadratic programming solvers [54]. If a linear model is not accurate enough and nonlinear effects have to be considered, the resulting model is a nonlinear model. The use of a nonlinear model for the MPC results in a NMPC problem. In general, the NMPC optimization problem is non-convex, but special cases of nonlinear optimization can still be convex. The result of a non-convex NMPC optimization is not always the global optimum and the optimization process can lead to high computational effort [17]. Besides the modeling by energy, mass or heat balances where the models are inherited in the class of MTI models, a better approximation of the dynamical behavior of a nonlinear system can be reached with MTI models as compared to linear models [41]. But investigations of the structure of the model predictive control optimization problem using MTI models are nonexistent. This leads to the question: Is the MPC optimization problem for MTI models convex? If the optimization problem is convex, in general or for a subclass, then the MPC optimization problem can be computed quickly and efficiently with standard algorithms, e.g. interior-point, and results in the global optimum [17]. Some results of the following investigation are already published in [42]. The required theorems and lemmas are introduced and for the proofs references to the literature are given.

4.2.1 A subclass of multilinear systems

The general class of MTI models is introduced in Section 2.1.2. The investigation of the convexity properties focuses on a single-input MTI model. A subclass of single-input

MTI models, the so-called input linear MTI models, is investigated. In this class, the input contributes linearly to the state equation and the models are multilinear in the states only. This leads to the new state transition function

$$\mathbf{x}(k+1) = \hat{\mathbf{F}}\mathbf{m}(\mathbf{x}(k)) + \mathbf{B}u(k) \tag{4.11}$$

with the state matrix $\hat{\mathbf{F}} \in \mathbb{R}^{n \times 2^n}$ and the input vector $\mathbf{B} \in \mathbb{R}^{n \times 1}$.

Example 4.1 *The state transition function of an input linear MTI model with two states and one input is given by*

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} \hat{f}_{1,1} & \hat{f}_{1,2} & \hat{f}_{1,3} & \hat{f}_{1,4} \\ \hat{f}_{2,1} & \hat{f}_{2,2} & \hat{f}_{2,3} & \hat{f}_{2,4} \end{pmatrix} \begin{pmatrix} 1 \\ x_1(k) \\ x_2(k) \\ x_1(k)x_2(k) \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u(k)\,.$$

4.2.2 Convexity properties of the optimization problem for a subclass of multilinear systems

Solving a nonlinear optimization problem can lead to a high computational effort and it is not assured that the global optimum is found. But if the nonlinear optimization problem is convex, very efficient algorithms exist to solve the optimization problem and the global optimum is found [17]. For the investigation of the convexity properties the standard quadratic cost function

$$J(k) = \underbrace{\sum_{i=1}^{H_p} \|\mathbf{x}(k+i) - \mathbf{r}(k+i)\|^2_{\mathbf{Q}(i)}}_{J_x(\mathbf{u})} + \underbrace{\sum_{i=0}^{H_u-1} \|\Delta\mathbf{u}(k+i)\|^2_{\mathbf{R}(i)}}_{J_{\Delta u}(\mathbf{u})}, \tag{4.12}$$

as introduced in Section 3.4.3, was used, with the input changes $\Delta\mathbf{u}(k)$, the reference signal $\mathbf{r}(k)$ and the transition function $\mathbf{x}(k+1) = \mathbf{F}\mathbf{m}(\mathbf{x}(k), \mathbf{u}(k))$. The control horizon is denoted by H_u and the prediction horizon by H_p, assuming that $H_u \leq H_p$. It is assumed that the two weighting matrices $\mathbf{R}(i) \geq 0$ and $\mathbf{Q}(i) \geq 0$ are diagonal matrices with the diagonal elements $q_j(i)$ and $r_k(i)$, such that

$$\mathbf{Q}(i) = \underset{j=1,\dots,n}{\mathrm{diag}} \left(q_j(i)\right), \ \mathbf{R}(i) = \underset{k=1,\dots,m}{\mathrm{diag}} \left(r_k(i)\right).$$

The optimization problem

$$\min_{\mathbf{u}\in\mathcal{U}} J(k) = \min_{\mathbf{u}\in\mathcal{U}} \sum_{i=1}^{H_p} \|\mathbf{x}(k+i) - \mathbf{r}(k+i)\|^2_{\mathbf{Q}(i)} + \sum_{i=0}^{H_u-1} \|\Delta\mathbf{u}(k+i)\|^2_{\mathbf{R}(i)}, \tag{4.13}$$

is a minimization problem, where the vector $\mathbf{u} = (u(k), u(k+1), ..., u(k+H_p)) \in \mathcal{U}$ represents the optimization variables and $\mathcal{U}$ the set of possible inputs. The cost function (4.12) will be minimized by finding the best input values $\mathbf{u}$. If the set of optimization variables $\mathcal{U}$ and the cost function $J(k)$ is convex, then the optimization problem without additional constraints is convex. It is assumed that the set of optimization variables is convex for the following discussion.

The cost function is convex if the Hessian matrix $\mathbf{H}$ or the second derivative of $J(k)$, with respect to the optimization variables, is positive semi-definite

$$\mathbf{H} = \nabla^2 J(k) \geq 0. \tag{4.14}$$

To satisfy the condition $\mathbf{H} \geq 0$, the eigenvalues $\lambda_i,\ i = 1, \dots, H_p$ of the Hessian matrix $\mathbf{H}$ have to be greater than or equal to zero. A detailed description of convex optimization problems can be found in [17], as well as the methods to prove the convexity. The fact, that the sum of convex functions results in a convex function is used to simplify the investigation and evaluate the convexity of the two terms $J_x(k)$ and $J_{\Delta u}(k)$ of the cost function separately. Note that if one the two terms of the sum is convex and the other one is non-convex does not mean that the sum of the two functions is non-convex. The second term $J_{\Delta u}(k)$ is obviously convex because it is a sum of squared terms, independent of the choice of H_u. This reduces the following convexity analysis to the first term $J_x(k)$ of the cost function. The convexity of the reduced cost function

$$\begin{aligned} J_x(k) &= \sum_{i=1}^{H_p} \|\mathbf{x}(k+i) - \mathbf{r}(k+i)\|^2_{\mathbf{Q}(i)} \\ &= \sum_{i=1}^{H_p} (\mathbf{x}(k+i) - \mathbf{r}(k+i))^T \mathbf{Q}(i) (\mathbf{x}(k+i) - \mathbf{r}(k+i)) \end{aligned} \tag{4.15}$$

will be discussed for different prediction horizons H_p in the following.

First, a prediction horizon of one will be investigated, with the general definition of the MTI systems (2.8), introduced in Section 2.1.2, and the restriction of a single input, the dependence of $\mathbf{x}(k+1)$ on u can be written as

$$\mathbf{x}(k+1) = \mathbf{F}_{:,1:2^n}\mathbf{m}(\mathbf{x}) + \mathbf{F}_{:,2^n+1:2^{n+1}}\mathbf{m}(\mathbf{x})u(k) \tag{4.16}$$

with the two constant vectors $\mathbf{F}_{:,1:2^n}\mathbf{m}(\mathbf{x}) \in \mathbb{R}^n$ and $\mathbf{F}_{:,2^n+1:2^{n+1}}\mathbf{m}(\mathbf{x}) \in \mathbb{R}^n$. The vectors $\mathbf{F}_{:,1:2^n}\mathbf{m}(\mathbf{x})$ and $\mathbf{F}_{:,2^n+1:2^{n+1}}\mathbf{m}(\mathbf{x})$ are constant because they depend on the state $\mathbf{x}$ at time k only, which does not change during the optimization. A MATLAB wise notation is used and the colon means that the entire row or column of a matrix is chosen. The notation with a number before and after the colon means, that a subset of elements of a row or a column is chosen, e.g., for the notation $5:10$ the elements five to ten.

Lemma 4.1 *The optimization problem (4.13) of MTI systems (2.8) with one input and a one-step prediction horizon is convex [42].*

The proof of the Lemma 4.1 for the single-input MTI systems is given in [42].

The prediction horizon H_p will be extended to two. The input linear state transition function of two time steps is given by

$$x(k+2) = \hat{\mathbf{F}}\mathbf{m}(\mathbf{x}(k+1)) + \mathbf{B}u(k+1). \tag{4.17}$$

Inserting the state transition function (4.11) results in

$$\mathbf{x}(k+2) = \hat{\mathbf{F}} \left(\bigotimes_{i=n}^{1} \begin{pmatrix} 1 \\ \hat{\mathbf{F}}_{i:}\mathbf{m}(\mathbf{x}(k)) + \mathbf{B}u(k) \end{pmatrix} \right) + \mathbf{B}u(k+1), \tag{4.18}$$

and the states $\mathbf{x}(k+2)$ can be written in terms of $\mathbf{x}(k)$.

Lemma 4.2 *The optimization problem (4.13) with a two-step prediction horizon of input linear MTI systems (4.11) without additional structural constraints is not convex [42].*

The lemma 4.2 is proven by a counter example [42].

The class of input linear MTI systems is, in general, not convex for a prediction horizon of two, which means that this class has to be restricted for further investigations.

Definition 4.1 *A subclass of the input linear MTI systems (4.11) is defined with some structural restrictions to the matrices* $\hat{\mathbf{F}}$ *and* $\mathbf{B}$*. These constraints on* $\hat{\mathbf{F}}$ *and* $\mathbf{B}$ *depend on each other and the new matrix* $\hat{\mathbf{F}}$ *has to fulfill the relation*

$$P(\hat{\mathbf{F}}) \leq \mathbf{S} \tag{4.19}$$

elementwise and the required structure of $\hat{\mathbf{F}}$ *is defined by the structure of the comparison matrix* $\mathbf{S}$*. The operator* P *is applied elementwise to the matrix* $\hat{\mathbf{F}}$*, which gives a structure matrix of* $\hat{\mathbf{F}}$ *as follows*

$$P(\hat{f}_{ij}) = \begin{cases} 1 & \textit{if } \hat{f}_{ij} \neq 0, \\ 0 & \textit{if } \hat{f}_{ij} = 0, \end{cases} \tag{4.20}$$

filled with zeros and ones. The number of nonzero elements of $\mathbf{B}$ *is described by* l

$$b_i = \begin{cases} b_i & \textit{for } i = 1, ..., l, \\ 0 & \textit{otherwise.} \end{cases} \tag{4.21}$$

The construction of the comparison matrix $\mathbf{S}$ *is given by*

$$\mathbf{S} = \mathbf{e} \cdot \left(\bigotimes_{k=n-l}^{n} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \boldsymbol{\gamma} \right)^T . \tag{4.22}$$

where the vector $\mathbf{e} \in \mathbb{R}^n$ *is filled with ones and the vector* $\boldsymbol{\gamma} \in \mathbb{R}^{2^l}$ *is given by*

$$\gamma_i = \begin{cases} 1 & \textit{for } i = 1, \ldots, 2^{j-1} + 1 \qquad j = 1, ..., l, \\ 0 & \textit{otherwise,} \end{cases} \tag{4.23}$$

with $i = 1, ..., 2^l$.

With the new defined subclass of the input linear MTI systems the following theorem is defined.

Theorem 4.1 *The optimization problem (4.13) of an input linear MTI system (4.11) is convex if the structure condition (4.19) is fulfilled and the prediction horizon H_p is less or equal to 2 [42].*

The proof of the Theorem 4.1 is given in [42].

The convexity of the MPC optimization problem (4.13) is proven by checking the positive semi-definiteness of the Hessian matrix of the cost function. For the entire class of single-input MTI systems, the MPC optimization problem is convex if a prediction horizon of one is used. For a prediction horizon of two, a special subclass of input linear MTI systems is defined for which the convexity of the optimization problem was shown. This leads to the question, if there are applications which can use a prediction horizon of two for the optimization problem.

4.2.3 Use of convexity properties for small prediction horizons for applications

In general the optimization problem (4.13) is not convex and the optimization results are dependent on the choice of the initial values. An example with an input linear MTI system shows how the two step convex optimization problem can be used for the non-convex optimization problem with a prediction horizon of ten. An input linear MTI system with one input and two states is given with the state transition matrix and the input vector as follows

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The optimization problem (4.13) for this system is solved for a prediction horizon of ten, which means that the problem is not convex. The two weighting matrices $\mathbf{Q}(i)$ and $\mathbf{R}(i)$ are chosen as identity and the reference as $\mathbf{r}(k+i) = \begin{bmatrix} 2 & 2 \end{bmatrix}^T, \ i = 1, \ldots, 10$. The system input is limited to the interval $[-10, \ 10]$. The given system belongs to the subclass of input linear MTI systems defined by (4.1), which results in a convex optimization problem for a prediction horizon of two. Therefore, the two step optimization problem is computed first and the resulting global optimal solution of the inputs $u_{opt}(k)$ and $u_{opt}(k+1)$ are used as starting values for the optimization problem with a prediction horizon of ten instead of choosing the entire initial vector randomly $\mathbf{u}_0 \in \mathbb{R}^{10}$. The optimal solution $u_{opt}(k)$ and $u_{opt}(k+1)$ of the optimization with $H_p = 2$ is not the optimal solution for $\mathbf{u}(k)$ if $H_p = 10$ and is only used as first two elements of the initial vector $\tilde{\mathbf{u}}_0$. The new initial vector is then given by

$$\tilde{\mathbf{u}}_0 = \begin{bmatrix} u_{opt}(k) & u_{opt}(k+1) & \mathbf{u}_{0,3:10} \end{bmatrix}^T.$$

The optimization was solved 1000 times for both approaches with the initial vectors $\mathbf{u}_0$ and $\tilde{\mathbf{u}}_0$ and randomly different initial values. The results are compared to the solution of a global optimizer. The evaluation of the optimization with an interior point algorithm with the two different initial vectors $\mathbf{u}_0$ and $\tilde{\mathbf{u}}_0$ shows, that the algorithm failed 179 times with $\mathbf{u}_0$ and 48 times $\tilde{\mathbf{u}}_0$. A failed optimization means that the global optimum of the cost function was not found. The comparison shows that the rate of finding the global optimum is higher by a factor of 3.7 if the initial vector $\tilde{\mathbf{u}}_0$ is chosen.

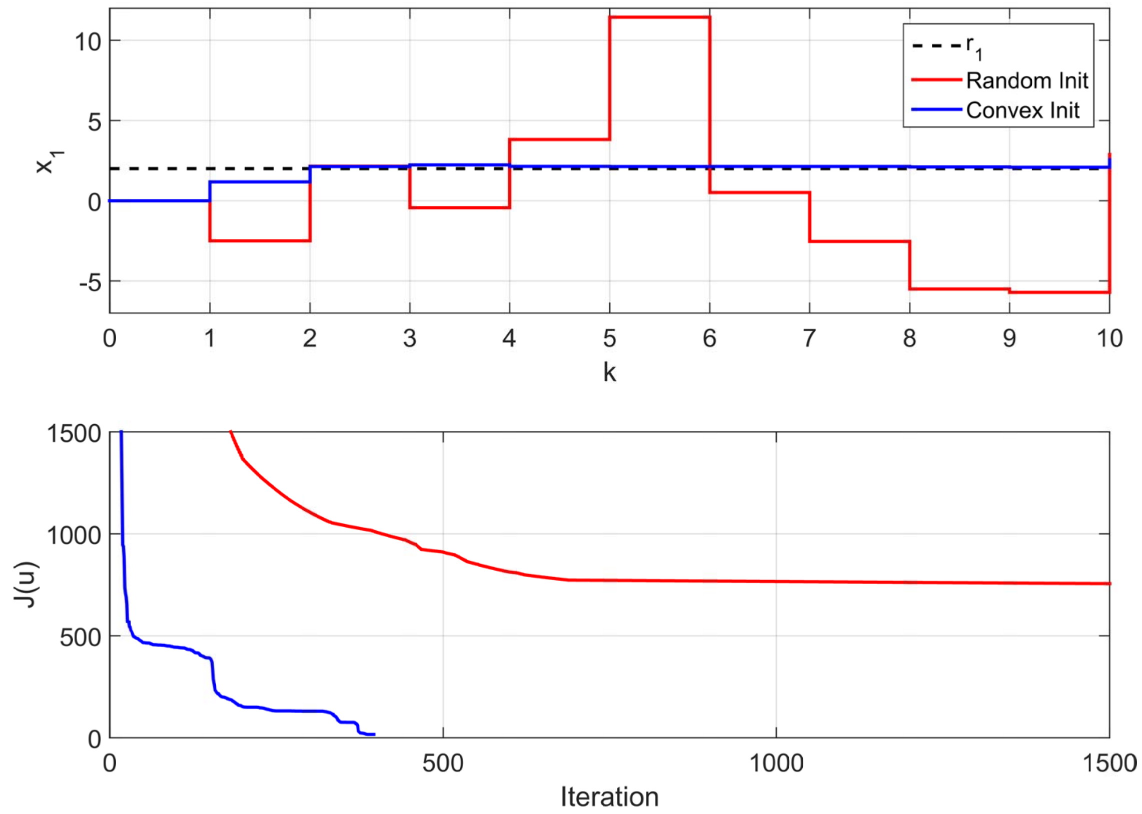


Figure 4.5: Comparison of optimization result with initial values $\mathbf{u}_0$ and $\tilde{\mathbf{u}}_0$ [42]

The simulated state trajectories of the first state x_1, with a prediction horizon of $H_p = 10$, are shown in the first plot of Figure 4.5. The blue line of top and bottom figure represents the result of one run if the initial vector $\tilde{\mathbf{u}}_0$ is used for the optimization and the red line the result of one run if $\mathbf{u}_0$ is used as the initial vector. Only the optimization with the initial vector $\tilde{\mathbf{u}}_0$ reached the global optimum. The second plot of Figure 4.5 shows the trajectory of the cost function over the number of iterations. The final value of the cost function is much smaller if $\tilde{\mathbf{u}}_0$ is used than with $\mathbf{u}_0$ and the optimization stops much faster, i.e., with less iterations.

4.2.4 Simulation results of a heating system example

The MPC optimization is applied to a heating system example where the model belongs to a class of input linear MTI systems. The model was introduced in [42] and only the general structure and the results will be stated here. The structure of the heating system is shown in Figure 4.6 and the modeling process was performed as introduced in Section 2.2.5.

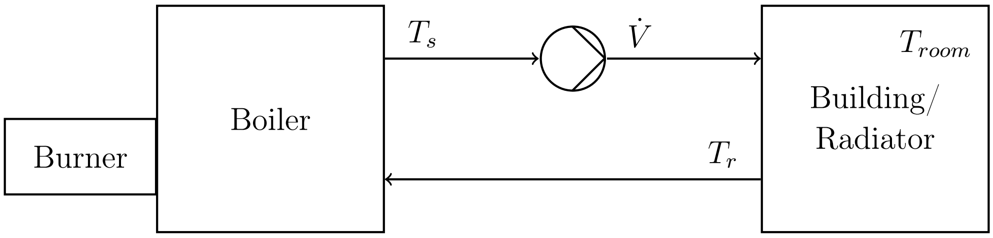


Figure 4.6: Scheme of a simple heating system

The discretized equations for the supply temperature T_s of the boiler (2.29), the return temperature T_r of the radiator (2.30), the room temperature T_{room} of the building (2.31) and the volume flow $\dot{V}$ provided by the pump (2.37) lead to a system of discrete-time difference equations. The model has three states, $\begin{bmatrix} T_s & T_r & T_{room} \end{bmatrix}$ and one input α, which controls the thermal power of the boiler. The output of the system are equal to the system states. The resulting heating system model can be described by an input linear MTI system with the state matrix and input vector given by

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 & \hat{f}_{1,2} & \hat{f}_{1,3} & 0 & 0 & \hat{f}_{1,6} & \hat{f}_{1,7} & 0 \\ 0 & \hat{f}_{2,2} & \hat{f}_{2,3} & 0 & \hat{f}_{2,5} & \hat{f}_{2,6} & \hat{f}_{2,7} & 0 \\ \hat{f}_{3,1} & 0 & \hat{f}_{3,3} & 0 & \hat{f}_{3,5} & 0 & 0 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} b_1 & 0 & 0 \end{bmatrix}^T.$$

According to the equations (4.21) and (4.23) the number of nonzero elements $l = 1$ and the vector $\boldsymbol{\gamma} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$. Equation (4.22) leads to the comparison matrix

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

The structure matrix of $\hat{\mathbf{F}}$ has to be checked by (4.19) which leads to

$$P(\hat{\mathbf{F}}) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \leq \mathbf{S}.$$

The model fulfills the constraints of Definition 4.1 and according to Theorem 4.1 the MPC optimization problem of such a system is convex for a prediction horizon up to two. The input is constrained to the interval $[0, 1]$ and thus, the set of optimization variables $\mathcal{U}$ is convex, which leads to the convex MPC optimization problem

$$\min_{u(k), u(k+1) \in \mathcal{U}} J(u(k), u(k+1)). \tag{4.24}$$

Any starting points $\mathbf{u}_0 = \begin{bmatrix} u_0(k) & u_0(k+1) \end{bmatrix}^T$ in $[0,1] \times [0,1]$ can be chosen, due to the fact that the optimization problem is convex. The references for the three model states are set to

$$\mathbf{r}(k+1) = \begin{bmatrix} 292K & 318K & 338K \end{bmatrix}^T, \ \mathbf{r}(k+2) = \begin{bmatrix} 292K & 318K & 338K \end{bmatrix}^T.$$

Solving the optimization problem with an interior point algorithm and different initial values always results in the global minimum

$$\mathbf{u}_{opt} = \begin{bmatrix} 0.2285 & 0.4624 \end{bmatrix}^T ,$$

and converges after a small number of iterations, which is the typical behavior for a convex optimization problem.

This shows that the MPC optimization for a special class of input linear MTI systems can be computed very efficiently and reliably for a prediction horizon of $H_p = 2$. If the model does not belong to the class of input linear MTI systems or a larger prediction horizon is used, then the optimization problem is not convex anymore. This leads to the question of whether there are other possibilities to improve the performance of an linear MPC with a convex optimization problem but using a simple linear model and the need of a suitable reference.

Chapter 5

Data-driven learning model predictive control

Model predictive control applied to a heating system with respect to the weather forecast is a promising approach for the optimization of energy consumption, [31, 60, 66] and was already introduced in Section 3.4.3. Finding a suitable model, which represents the dynamical behavior of the system can be difficult and often leads to a nonlinear model and thus to a nonlinear model predictive control problem. The nonlinear optimization often leads to a high computational effort due to the fact that the optimization problem is not convex. One option is to restrict the class of nonlinear models and find subclasses with a convex optimization problem, for example the subclass of MTI models, as introduced in Section 4.2. Such optimization problems can be computed effeciently with known algorithms, [17, 54]. Another option is the use of a simple linear model for a standard linear MPC and combine the MPC with another control strategy to improve the performance and compensate possible control errors due to the simplicity of the linear model and overcome the problem of finding a suitable reference trajectory.

Iterative learning control is well known in the field of periodic processes, for example the compensation of periodic disturbances or the tracking of a periodic reference. Common application fields are industrial processes, where the same task is performed again and again, like reference tracking of a robot arm, [18, 76]. But also for other applications an ILC was used, e.g. for the free electron laser (FLASH) at DESY (Deutsches Elektronen-Synchrotron) in Hamburg [64], or for HVAC systems [23, 57, 77]. All of these applications use only the last iteration for the calculation of the next input signal. That means that for applications with periodic disturbances or references an ILC is a natural choice.

For the application of heating systems the periodicity occurs in the disturbances, e.g., the outside temperature increases during daytime and decreases during nighttime. In comparison to industrial processes, where the same task is performed again and again, the weather conditions can change from day to day. Because of that, learning from the past trial is possibly not the best choice. But over a longer time period of a month or a year, there will be historic trials with similar ambient conditions as for the next trial. When all past trials are stored, the entire data set can be used for the choice which past trial best fits with the next trial on the basis of the ambient conditions. This leads to the question how the known ILC algorithms can be used for the improvement of an MPC? How can the stored measured data be used for control besides the modeling of the system? If all past trials are stored how can a historic data set for the ILC be chosen? Preliminary results to these questions are introduced in [43] and [44].

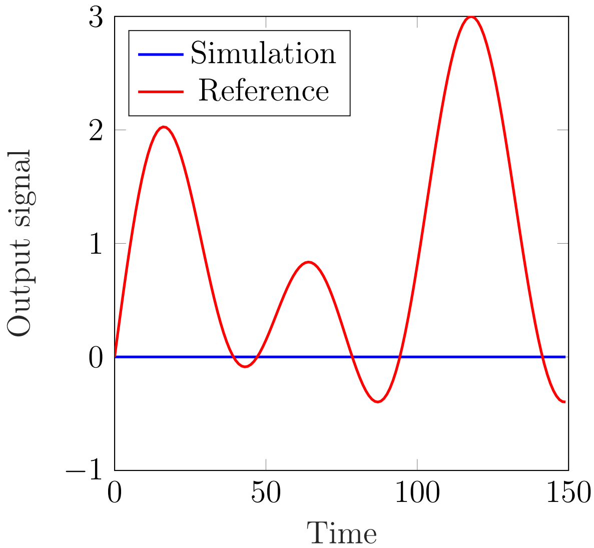
5.1 Data-driven iterative learning control for model predictive control

5.1.1 Iterative learning control

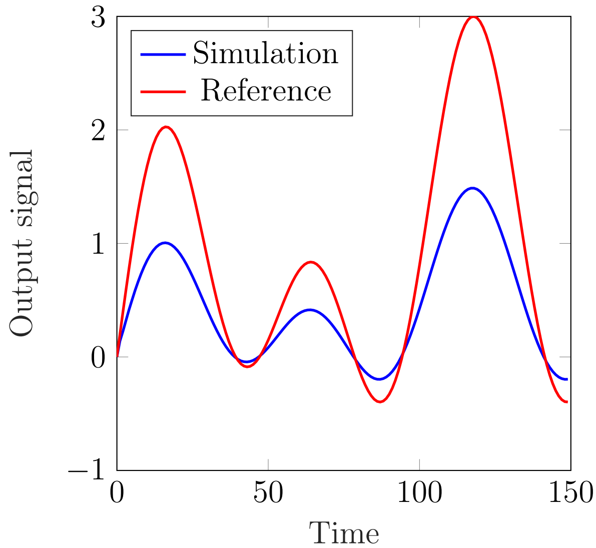
A natural choice of periodic processes with the same periodicity from one iteration d to the next iteration $d+1$ are iterative learning algorithms. The basics of an ILC are introduced in Section 3.3.2. Figure 3.5 shows a scheme of an ILC with the system. The input $\mathbf{u}_d(k)$ and error $\mathbf{e}_d(k)$ data of the last iteration d are stored and used for the calculation of the ILC update (3.9). The next example shows the control principle of an ILC.

Example 5.1 *Figure 5.1 (a)-(d) shows the simulation results of a linear model together with an ILC. This examples illustrates how the ILC algorithm (3.9) works from one iteration to another. The simulation shows how the output signal $\mathbf{y}_d$ gets closer to the reference $\mathbf{r}_{ilc}$ from iteration to iteration. The periodic reference is chosen arbitrarily for demonstration purposes only. The example model was a linear second order state space model with one input u and one output y. The sample time was set to $t_s = 0.1$ seconds, the learning gain to $\gamma = 1.2$, and the input signal of the first iteration to zero. Figure 5.1 (a) shows the simulation results of the first iteration. The other figures show the results after the second, the sixth, and the twelfth iteration. From iteration to iteration the output signal gets closer to the reference trajectory and tracks the reference after the twelfth iteration.*

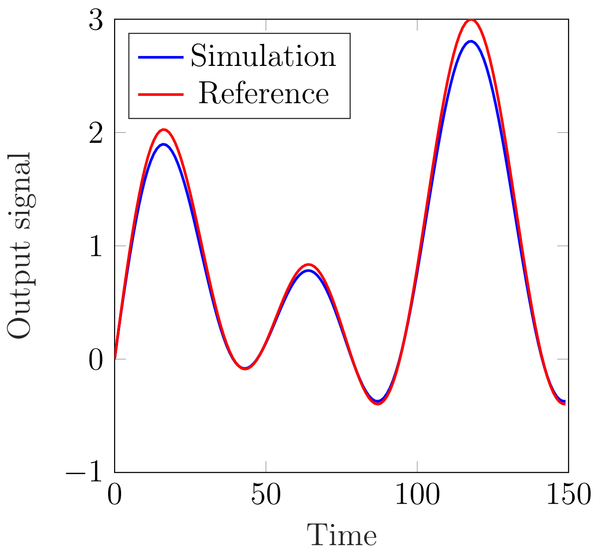
This simple example shows the simulation results, if the input signal $\mathbf{u}_{d+1}$ is calculated according to equation (3.9) and how the reference tracking works for a repetitive process. An ILC is the natural choice for systems operating repetitively in a fixed time interval with the advantage that a model of the system can be uncertain or unknown and there are no or less information about the system structure and nonlinearity [11].



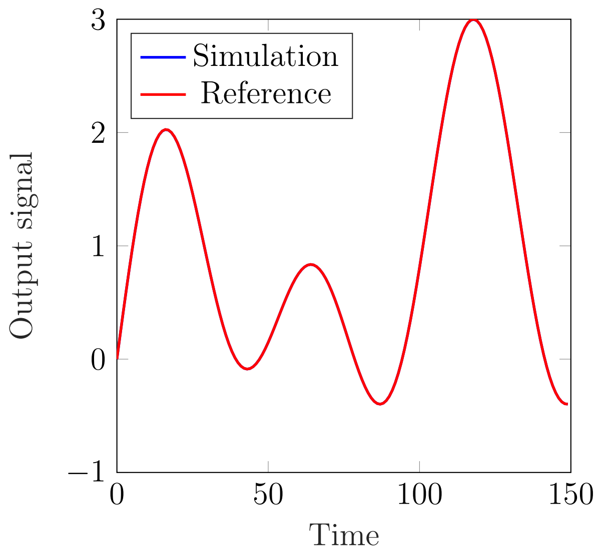
(a) Iteration 1



(b) Iteration 2



(c) Iteration 6



(d) Iteration 12

Figure 5.1: Simulation results of different iterations of an ILC

The ILC algorithm (3.9) uses the last iteration only, which means the periodicity has to occur from one iteration to the next iteration. But if the periodicity is not the same from one iteration to another but, e.g., every fifth iteration has the same periodicity the simple algorithm (3.9) cannot react on such periodic structure. In the next Section an ILC approach is introduced, which chooses one iteration from all stored and available data of past iterations to calculate the input signal of the next iteration to overcome the problem, that only simple periodic structures can be used.

5.1.2 Data-driven iterative learning control

The ambient conditions of some systems can change from trial to trial and thus the periodicity. But over a longer time period there are past trials with ambient conditions similar to the current trial. For example, for heating systems the increase and decrease of the outside temperature varies from day to day and setting one trial as one day is a useful choice. For such systems the input signal of the next trial cannot be calculated according to equation (3.9). For this reason, the learning update is not derived from the previous trial, but from a historic trial of a longer time period. A requirement is, that the historic data of past iterations is stored in a database or another storage device and available for the calculation of the next input signal. Thereby, the equation (3.9) changes to

$$\mathbf{u}_{d_{next}}(k) = \mathbf{u}_{d_{db}}(k) + \gamma \mathbf{e}_{d_{db}}(k). \tag{5.1}$$

The input signal $\mathbf{u}_{d_{next}}(k)$ of the next trial at time k is derived from a past trial stored in a database or local storage device indicated by the symbol extension *db*, i.e. its input $\mathbf{u}_{d_{db}}$ and error $\mathbf{e}_{d_{db}}$ signals. The criteria for the choice of the best stored trial is application dependent and will be done by the so-called element selector. One criterion for that decision is the comparison of the disturbance signals $\mathbf{a}_{d_{db}}$, e.g. the measurable ambient conditions, of the past and the previous trials. That presumes the availability of a prediction of the disturbance signal $\mathbf{a}_{pre}$ and the ambient conditions, respectively. Additionally to the input $\mathbf{u}_{d_{db}}$ and error $\mathbf{e}_{d_{db}}$ signals of every trial, the disturbance signals $\mathbf{a}_d$ have to be stored in a database as well. The ILC, which uses the stored data of the historic trials will be called data-driven ILC in the following. Figure 5.2 shows a scheme of the control loop with a system and a data-driven ILC.

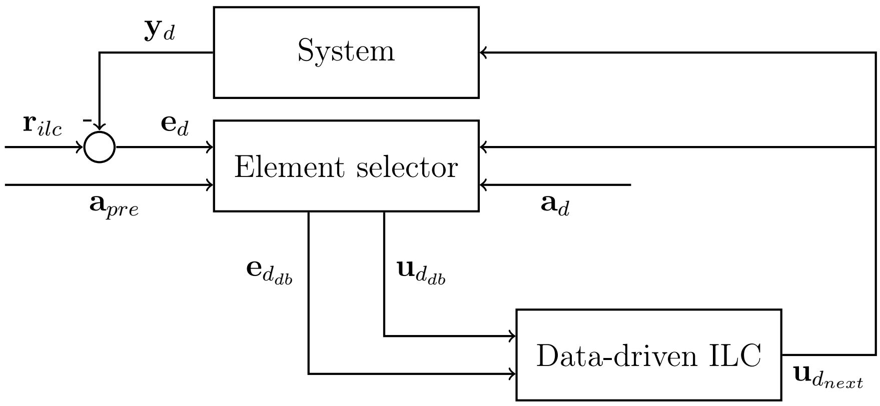


Figure 5.2: Scheme of a system with a data-driven ILC

Two decision criteria are defined for the element selector. The sum of squares of the difference at time k, between the stored disturbance signal $\mathbf{a}_{d_{db}}$ of the past trials and the prediction of the disturbance $\mathbf{a}_{pre}$ of the next trial can be used as a similarity measure [43]

$$\mathbf{E}_T(d_{db}) = \sum_k (\mathbf{a}_{d_{db}}(k) - \mathbf{a}_{pre}(k))^2. \tag{5.2}$$

Additionally, a standard squared error performance measure

$$\mathbf{E}_e(d_{db}) = \sum_k (\mathbf{r}_{ilc}(k) - \mathbf{y}_{d_{db}}(k))^2 = \sum_k \mathbf{e}^2_{d_{db}}(k) \tag{5.3}$$

is used. The performance measure evaluates the deviation between the output signal $\mathbf{y}_d(k)$ and the reference $\mathbf{r}_{ilc}(k)$ of each past iteration. This leads to an optimization problem

$$\min_{d_{db}\in D} \mathbf{E}_e(d_{db}), \qquad s.t.\, \mathbf{E}_T(d_{db}) < E_M, \tag{5.4}$$

with the set of the historic iterations D and a tuning parameter E_M giving a maximum allowed non-similarity of the ambient conditions, due to the fact that in many cases the minimum of equation (5.2) and equation (5.3) are not connected to the same iteration. The choice of the tuning parameter E_M influences the solvability of the optimization problem (5.4) and leads to an infeasable optimization problem if no value $\mathbf{E}_T(d_{db}) < E_M$ exists. Especially for the start-up period of the data-driven ILC with less historic data, the choice of E_M is critical. One option to relax the start-up problem is to collect data first and start the data-driven ILC after the data is taken, but with the disadvantage that the data-driven ILC cannot run from the beginning. Another option is to start with a very relaxed limit E_M and tighten the limit from time to time, with the disadvantage of starting with a higher allowed non-similarity but the advantage that the data-driven ILC can be used from the very beginning. An heuristic approach can be used to set E_M if historic data already exists by evaluating (5.2). The solution of the optimization problem (5.4) gives the trial d_{db}, which is used for the calculation of the ILC update (5.1).

For data-driven ILC, the error $\mathbf{e}_d$, input $\mathbf{u}_d$ and disturbance $\mathbf{a}_d$ signals of all historic trials have to be available and stored in a database or a local storage device. The storage demand is given by the product of the samples per iteration N_m and the available past iterations N_t.

5.1.3 A combined iterative learning model predictive controller

The data-driven ILC uses the historic data to calculate the input signal of the next iteration, but do not consider the future dynamical behavior of the system, e.g. by using a model. The model predictive controller uses a model to predict the dynamical behavior of the system, so that the controller can react early to changes in the environmental conditions, but do not learn from the past. A combined control strategy with a data-driven ILC and an MPC would learn from the past and take the future dynamical behavior into account.

The MPC part uses the standard linear discrete-time model predictive control as introduced in Section 3.4.3 with the definitions of the cost function (3.4) and optimization problem (3.5).

Finding a suitable reference trajectory $\mathbf{r}_{mpc}$ for the MPC can be difficult. Because of that, and due to the simplicity of the linear model, a data-driven ILC will be used to adjust the chosen reference $\mathbf{r}$ to a new reference for the MPC

$$\mathbf{r}_{mpc}(k) = \mathbf{r}(k) + \mathbf{u}_{d_{next}}(k). \tag{5.5}$$

The signal $\mathbf{u}_{d_{next}}(k)$ is calculated according to equation (5.1).

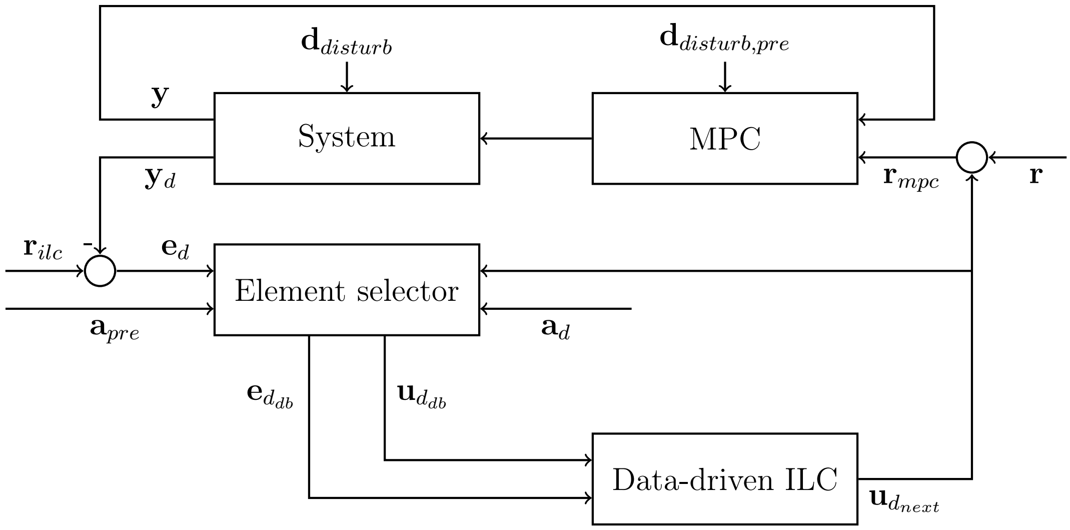


Figure 5.3: Scheme of a system with a MPC and a data-driven ILC

Figure 5.3 shows a scheme of the control loop with the system the MPC and the additional data-driven ILC. This data-driven learning MPC will be applied to a heating system example.

5.1.4 Simulation results for a heating system example

The model of the heating system used for the simulation is already introduced in Section 3.4.1 and is a prototype test facility for an office in a non-residential building. Figure 3.6 shows a scheme of the heating system.

The MPC uses a linearized state space model of the introduced heating system model, as introduced in Section 3.4.1, to calculate the two input signals α and ϕ. The reference of the MPC is given for the supply temperature of the radiator. The reference $\mathbf{r}$ is calculated by a simple heating curve with respect to the outside temperature forecast $T_{out,for}$. The heating curve was estimated from measurement values of the plant and was already shown in Figure 3.1.1. The data-driven ILC adjusts this reference $\mathbf{r}$ according to equation (5.5) which results in the new reference $\mathbf{r}_{mpc}$ for the MPC. As introduced before, the MPC uses the moving horizon principle with the prediction horizon H_p and the control horizon H_u.

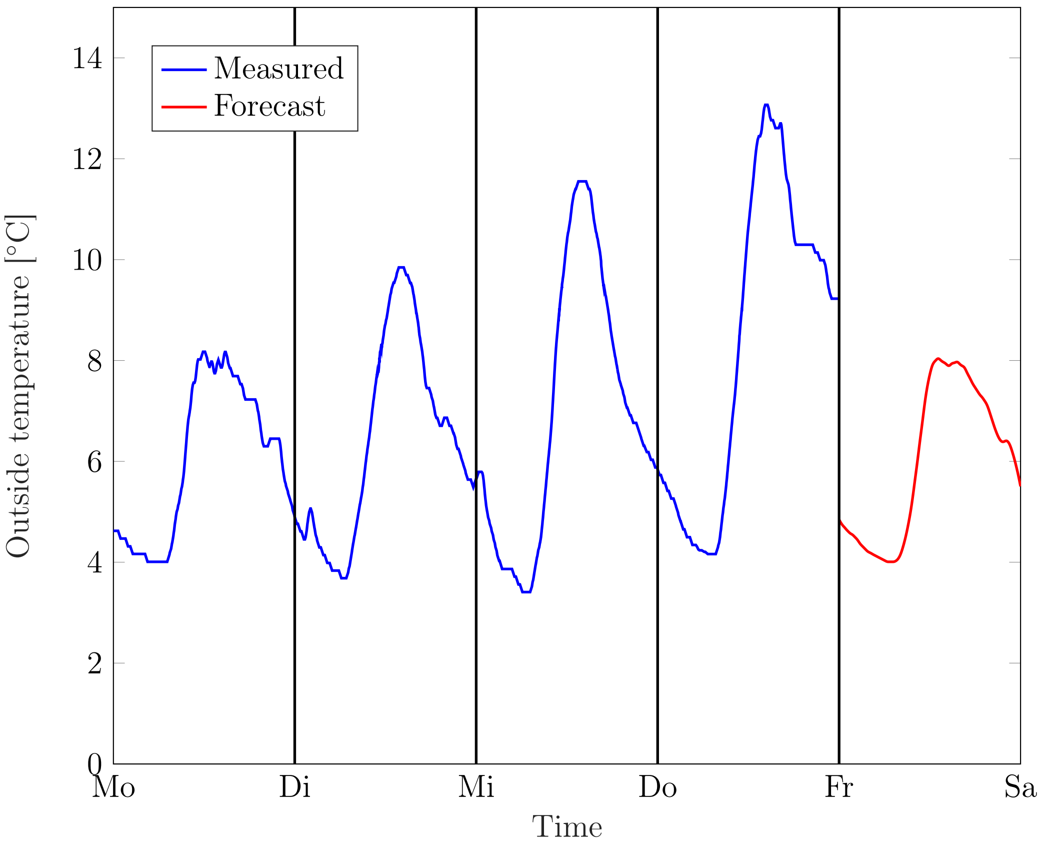


Figure 5.4: Different daily profiles of the outside temperature

The periodic disturbances of a heating system are the daily profiles of the outside temperature. Figure 5.4 shows a good example of the increasing and decreasing outside temperature during day and night, as well as, the changes from day to day, which suggests the use of a data-driven ILC. The outside temperature forecast, shown in red and also generated from measurement data, illustrates the differences, but also the similarities between the forecast and some of the past outside temperature profiles.

The comparison of the outside temperatures of different days with the generated outside temperature forecast shows, that the use of the data from Monday makes more sense for the calculation of the learning update (5.1), than the use of the data from Thursday, according to the outside temperature profiles. Therefore, a natural choice for the length of one iteration d is one day and will be used in the following. Nevertheless, choosing another time period, e.g. one week if the occupants follow a weekly rythm could also be a suitable choice but with the drawback that the update calculation is performed once a week only, which slows down the learning process. Choosing shorter time periods like one hour would lead to a faster learning process but the natural heat demand periodicity for buildings is one day due to the increasing and decreasing outside temperature and the day and night cycle of the occupants.

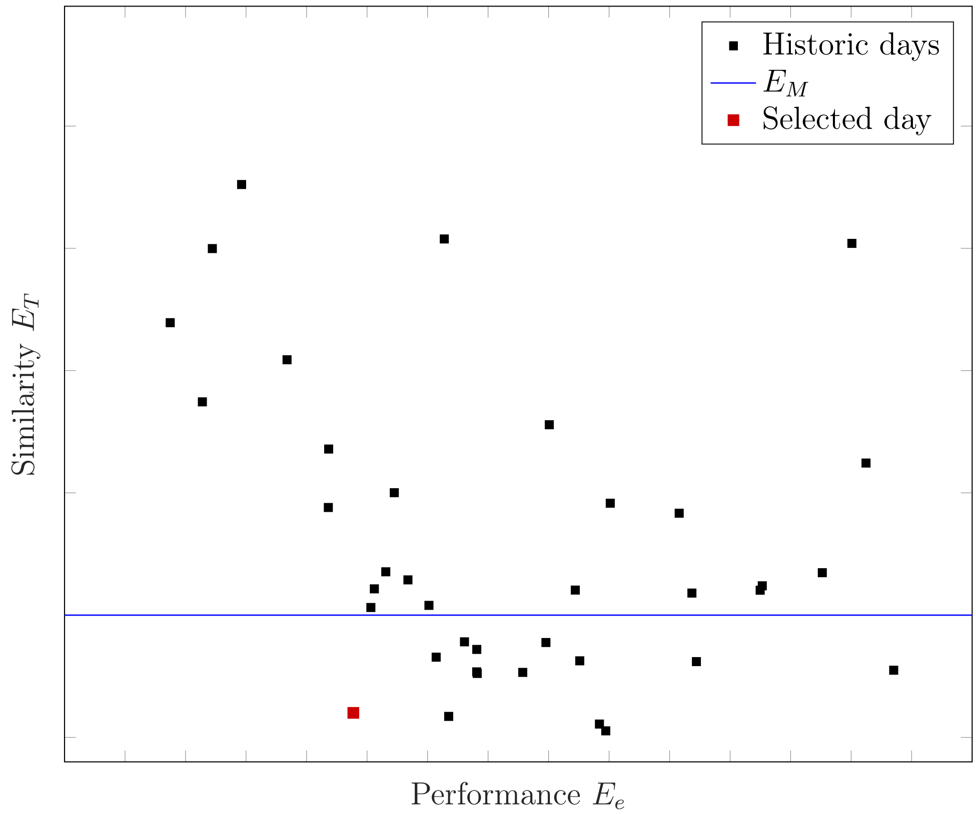
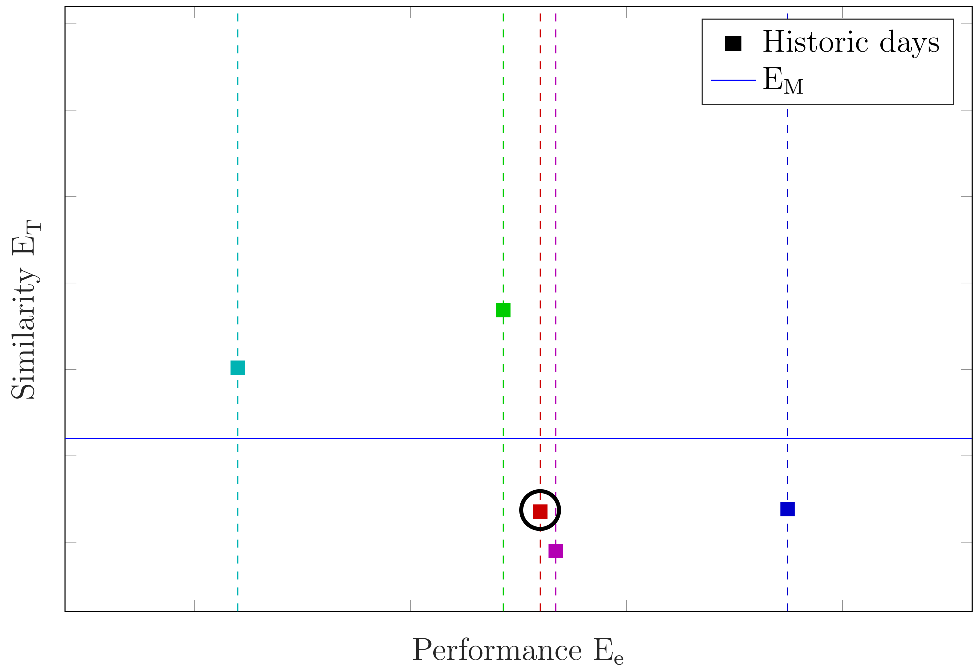


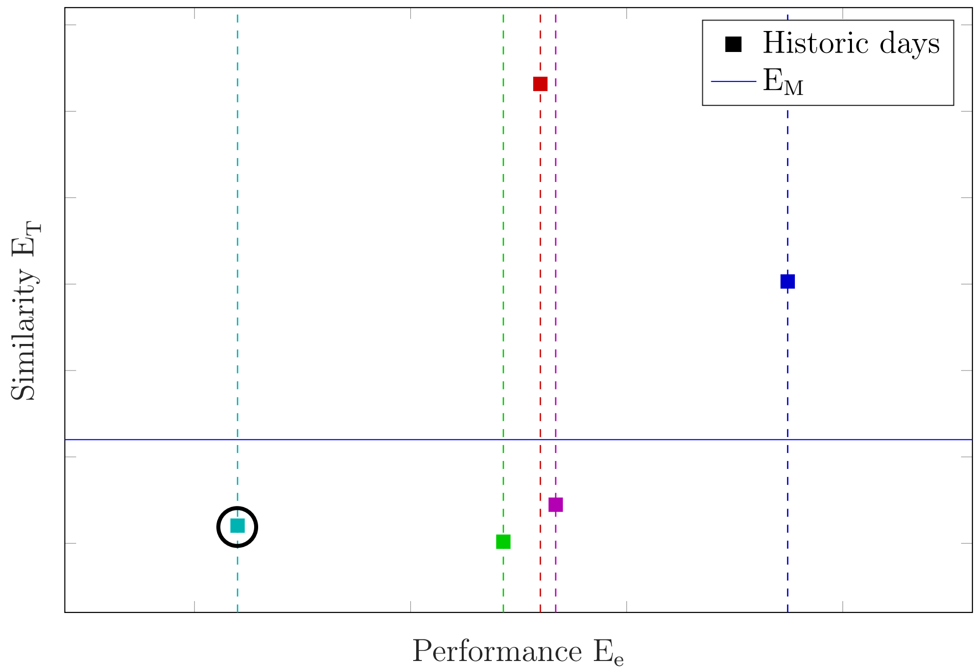
Figure 5.5: Results of the optimization problem (5.4)

The data-driven ILC optimizes the reference for the MPC, as presented in Figure 5.3, to compensate the periodic disturbances due to the outside temperature profile. The goal of the data-driven ILC is to maintain the room temperature in a specific comfort zone. Because of that, the reference of the ILC $\mathbf{r}_{ilc}$ is a profile of the desired room temperature, with a difference between day and night. At daytime, the reference room temperature is chosen according to the German norm DIN-EN15251 [24]. This norm prescribes the comfort room temperature to be 22±2 °C, during winter when the outside temperature is below 16 °C. The data-driven ILC uses the outside temperature as the ambient condition and the outside temperature forecast for the calculation of the similarity criterion (5.2). This means that the outside temperature forecast is used for both controllers, the MPC and the data-driven ILC.

Figure 5.5 shows the evaluation of the optimization problem (5.4) with the outside temperature forecast of one iteration as ambient conditions prediction $\mathbf{a}_{pre}$, which means the forecast of one day due to previous discussed fact that one iteration is set to one day for heating system applications. The x-axis represents the performance criterion (5.3) and the y-axis the similarity criterion (5.2). The smaller the values of the similarity and the performance criterion the better. Every single black square represents the evaluation of one past iteration (day) and the red square marks the chosen iteration (day),



(a) Prediction 1



(b) Prediction 2

Figure 5.6: Results of the optimization problem (5.4) for two different predictions $\mathbf{a}_{pre}$

which is the past iteration with the smallest performance value and a similarity value below the constant criterion E_M.

The Figures 5.6 (a) and (b) illustrate the changes of the calculation results of (5.4) for five historic days (iterations) and two different outside temperature forecasts $\mathbf{a}_{pre}$.

The comparison of the two Figures (a) and (b) shows that the calculated values of the performance criteria are fixed at the horizontal line. This follows directly from the fact that the stored error signals $\mathbf{e}_{d_{db}}$ do not change anymore. In contrast to that, the values of the similarity criterion vary at the vertical line for different forecasts $\mathbf{a}_{pre}$. This is illustrated by the dashed lines in Figure 5.6 (a) and (b). Due to the tuning parameter E_M, different days are chosen for different forecasts $\mathbf{a}_{pre}$. The chosen day for each outside temperature forecast is marked with a black circle. The optimization problem (5.4) is solved by full enumeration and the data of every new iteration is added to the end of the entire dataset. An improvement to solve the optimization problem could be to rearrange the historic data in increasing order of E_e, because the stored error signals $\mathbf{e}_{d_{db}}$ do not change anymore. Then starting from the first stored trial, which is the trial with the smallest value E_e, and check the criteria $E_T < E_M$ with increasing E_e until the condition $E_T < E_M$ is true. This procedure could speedup solving of the optimization problem because not the entire dataset has to be evaluated. But as a consequence the data of new iterations has to be inserted with respect to the value E_e.

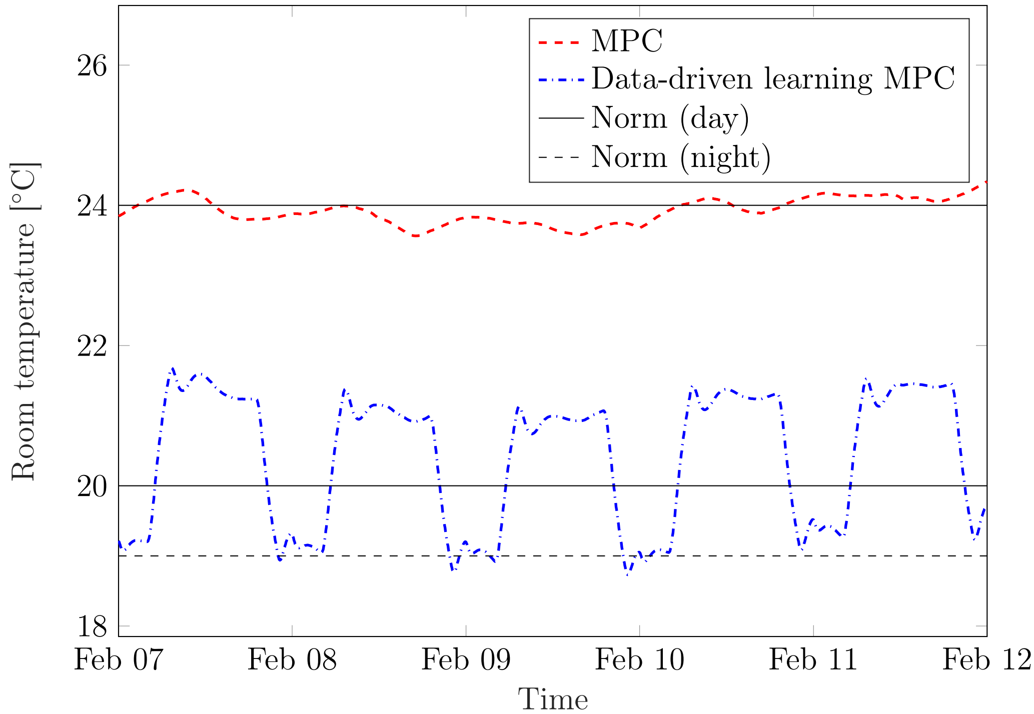


Figure 5.7: Comparison of the simulation results with the MPC and the additional data-driven ILC

For the simulation with the data-driven iterative learning model predictive controller the introduced model will be used, see Figure 3.6. The MPC block of the Model Predictive Control Toolbox of MathWorks will be used for the model predictive controller [2]. A prediction horizon of $H_p = 5$ hours, a control horizon of $H_u = 4$ hours, a learning gain of $\gamma = 2$, and a sample time of $t_s = 60$ seconds was used. First, simulation was performed with the MPC only. The second time, the additional ILC starts working, with the same simulation parameters, outside temperature profile and heating curve as before. The reference room temperature was set to 21 °C at daytime with a reduction of 2 °C during the night.

The stored historic data sets of $\mathbf{e}_{d_{db}}$ and $\mathbf{u}_{d_{db}}$ of the past days are generated by simulation. The associated stored outside temperature profiles belong to real measured data and were also used as disturbance signal for the simulation. The entire set of historic data includes 25 weeks, which is nearly half a year and corresponds to one heating season. The historic iteration is selected according to equation (5.4) for the calculation of the ILC update (5.1).

Figure 5.7 exemplary presents five days of a two month simulation and compares the control results of the MPC and the results of the MPC together with the additional data-driven ILC. The solid black lines define the upper and lower limit of the comfort zone for the room temperature T_{room} and the black dashed line shows the reference for the night. The red line represents the simulation where only the model predictive controller was used and the blue line is the result with the additional data-driven ILC.

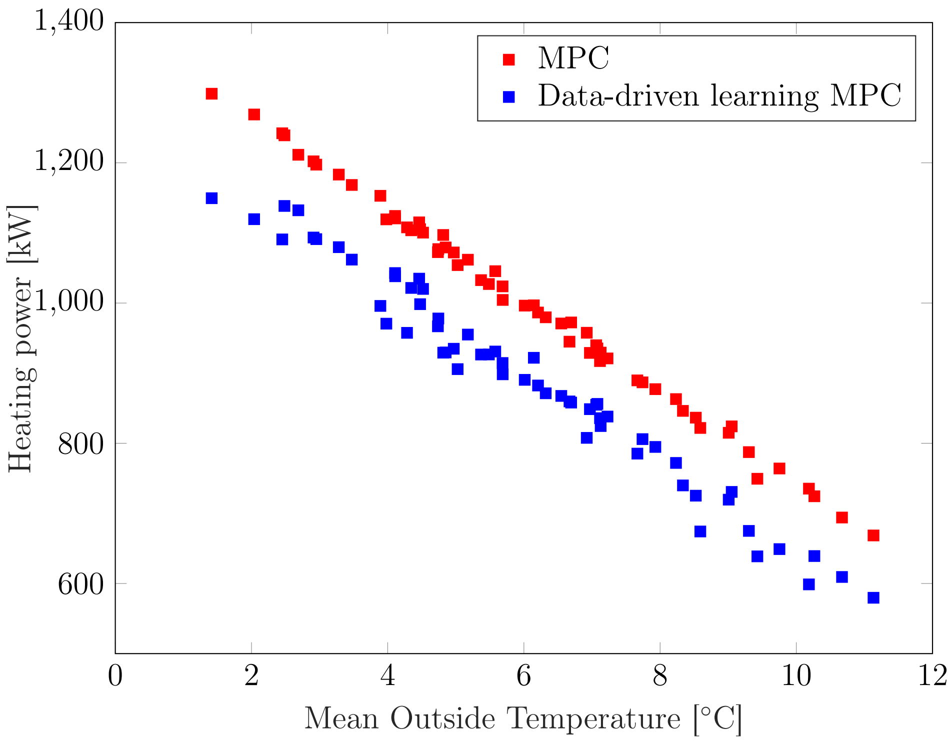


Figure 5.8: Heating power comparison with the MPC and the additional data-driven ILC

The comparison of the simulation results show that the room temperature T_{room} can be kept inside the desired comfort zone at daytime by using the additional data-driven ILC. The general reduction of the room temperature will lead to a reduction of the heating power of the building and thus to reduced energy consumption because of the fact that the heat demand increases and decreases with the room temperature. Additionally, the desired room temperature reduction during the night can be reached with the combined control strategy. The defined reference reduction of 2 °C leads to the room temperatures outside the comfort zone at nighttime, which is tolerable because it is a non-residential building and primarily used at daytime.

The room temperature reduction takes place for all simulated days with the combined control strategy, even if the reduction is not as strong for every day as shown in Figure 5.7, the heating power is reduced for every day. The evaluation of the heating power is presented in Figure 5.8 and shows the daily heating power of two months plotted over the daily mean values of the outside temperature.

The heating power is decreased in a significant manner for every day if the combined algorithm controls the heating system. The overall heating power reduction of these simulation is up to 11 %.

The storage of the data of every past iteration leads to an increasing storage demand. For devices and implementation hardware without access to a large storage device or a database, a storage demand reduction is necessary to make the algorithm applicable. This leads to the question how the storage demand can be reduced and how the storage demand reduction methods can be applied to the data-driven ILC.

5.2 Tensor based methods for data-driven iterative learning control

The control algorithm of a data-driven ILC, as introduced in section 5.1.2, collects and stores the data of all past trials. Not every automation system has access to a database or large local storage devices. Because of that, from an application point of view, it would be very useful to reduce the storage demand of the data to make this algorithm applicable on automation systems without network access and limited local storage devices. This leads to the questions: How can the storage demand be reduced? How can the calculation of the comparison criteria be performed with a compressed data set? And how does the data compression affect the results of the data-driven learning MPC?

Generally, data can be rearranged and stored in a tensor, which means allocation as multi-index array. For all kinds of data, this is possible in an arbitrary way, but many data sets have an internal structure, which can be used to adjust the dimensions of the tensor. In the following, a tensor based data-driven iterative learning controller will be introduced and a heating system example illustrates how natural data structures can be used to adjust the tensor dimensions. Preliminary results were introduced in [44].

5.2.1 Tensor algebra

A standard definition of a tensor and some mathematical operations, for instance the contracted product or the k-mode product, as well as the canonical polyadic (CP) decomposed tensor, will be used and can be found, e.g., in [21] or [36].

Definition 5.1 *A tensor of order n is a n-way array*

$$\mathsf{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}. \tag{5.6}$$

The elements $t_{i_1,i_2,\ldots,i_n}$ of the tensor are indexed by $i_j \in \{1, 2, \ldots, I_j\}$ for $j = 1, \ldots, n$.

Definition 5.2 *The outer product of the tensor X of the dimension $I_1 \times I_2 \times \cdots \times I_n$ and the tensor Y of the dimension $J_1 \times J_2 \times \cdots \times J_m$ is defined as follows*

$$\mathsf{Z} = \mathsf{X} \circ \mathsf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times J_1 \times J_2 \times \cdots \times J_m}, \tag{5.7}$$

with the elements of Z

$$z_{i_1,\ldots,i_n,j_1,\ldots,j_m} = x_{i_1,\ldots,i_n} y_{j_1,\ldots,j_m}. \tag{5.8}$$

Example 5.2 *The outer product of a vector $\mathbf{a} \in \mathbb{R}^I$ and a vector $\mathbf{b} \in \mathbb{R}^J$ results in a matrix*

$$\mathbf{C} = \mathbf{a} \circ \mathbf{b} = \begin{pmatrix} a_1 \\ \vdots \\ a_I \end{pmatrix} \begin{pmatrix} b_1 & \cdots & b_J \end{pmatrix} = \begin{pmatrix} a_1 b_1 & \cdots & a_1 b_J \\ a_2 b_1 & \cdots & a_2 b_J \\ \vdots & \vdots & \vdots \\ a_I b_1 & \cdots & a_I b_J \end{pmatrix} \in \mathbb{R}^{I \times J}.$$

Definition 5.3 *The hadamard product of two tensors X and Y of the same dimension $I_1 \times I_2 \times \cdots \times I_n$ is defined as the element wise product and results in a tensor*

$$\mathsf{Z} = \mathsf{X} \circledast \mathsf{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_n} \tag{5.9}$$

with the elements given by

$$z_{i_1,\ldots,i_n} = x_{i_1,\ldots,i_n} y_{i_1,\ldots,i_n}. \tag{5.10}$$

Example 5.3 *The hadamard product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{I \times J}$ results in a matrix*

$$\mathbf{C} = \mathbf{A} \circledast \mathbf{B} = \begin{pmatrix} a_{1,1} b_{1,1} & \cdots & a_{1,J} b_{1,J} \\ a_{2,1} b_{2,1} & \cdots & a_{2,J} b_{2,J} \\ \vdots & \vdots & \vdots \\ a_{I,1} b_{I,1} & \cdots & a_{I,J} b_{I,J} \end{pmatrix} \in \mathbb{R}^{I \times J}.$$

Definition 5.4 *The k-mode product of a tensor* X *of dimension* $I_1 \times I_2 \times \cdots \times I_n$ *and a vector* $\mathbf{a} \in \mathbb{R}^{I_k}$ *results in a tensor* $\mathsf{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_{k-1} \times I_{k+1} \times \cdots \times I_n}$

$$\mathsf{Y} = \mathsf{X} \times_k \mathbf{a}, \tag{5.11}$$

with the elements given by

$$y_{1,\ldots,i_{k-1},i_{k+1},\ldots,i_n} = \sum_{i_k=1}^{I_k} x_{i_1,\ldots,i_N} a_{i_k}. \tag{5.12}$$

A tensor with a large number of dimensions has a large storage demand because of the fact that the number of elements increases exponentially if the dimensionality increases. Common tools for memory and complexity reduction are tensor decomposition and factorization methods. In various application fields where a massive amount of multidimensional data is available, such as signal processing, machine learning or computational neuroscience, these decomposition and factorization methods are used [22, 29, 36]. They are also in the areas of fault detection [59, 72] or modeling and control [39, 47, 62, 65]. For many tensor structures decomposition algorithms exist today, such as Tucker (TU), Hierarchical Tucker (HT), Tensor Trains (TT) or Canonical Polyadic (CP) [29, 36].

5.2.2 Canonical polyadic decomposed tensors

A canonical polyadic (CP) decomposed tensor structure retains the dimensions of the original tensor T. Each dimension is represented by one factor matrix of the decomposed CP tensor $\tilde{\mathsf{T}}$, which is an approximate representation of T. The CP decomposition is well established and the structure helps developing new algorithms. Thus, this thesis focuses on the CP decomposed tensor structures. Detailed description about the CP decomposed tensors are given in [21] or [36]. The CP decomposition algorithm applied to a tensor $\mathsf{T} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$ leads to a decomposed tensor $\tilde{\mathsf{T}} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$. The CP tensor is represented by the factor matrices $\mathbf{T}_i \in \mathbb{R}^{I_i \times R}$ $(i = 1, \ldots, n)$ of rank R. The approximate CP tensor $\tilde{\mathsf{T}}$ is given by the factor matrices and the sum of outer products of the column vectors $\mathbf{t}_{i,r} \in \mathbb{R}^{I_i}$ of these factor matrices

$$\mathsf{T} \approx \tilde{\mathsf{T}} = \lambda[\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_n] = \sum_{r=1}^{R} \lambda_r \mathbf{t}_{1,r} \circ \mathbf{t}_{2,r} \circ \cdots \circ \mathbf{t}_{n,r}, \tag{5.13}$$

weighted by the elements of the weighting vector $\boldsymbol{\lambda} \in \mathbb{R}^{1 \times R}$. It is assumed that $\boldsymbol{\lambda}$ is a vector of ones, i.e. $\boldsymbol{\lambda} = (1 \quad 1 \quad \cdots \quad 1)^T$ if no weighting vector is given. The CP decomposition reduces the storage demand in a significant manner because only the factor matrices $\mathbf{T}_i$ have to be stored and not the full tensor $\tilde{\mathsf{T}}$. This means that the demand depends on the dimensionality of the tensor and the decomposition rank R, which results in a reduction from $I_1 \cdot I_2 \cdot \ldots \cdot I_n$ elements of the full representation to $(I_1 + \cdots + I_n)R$ elements of the factor matrices.

Example 5.4 *A three dimensional CP tensor of rank R is given by*

$$\tilde{\mathsf{T}} = \lambda[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] = \sum_{r=1}^{R} \lambda_r \mathbf{t}_{1,r} \circ \mathbf{t}_{2,r} \circ \mathbf{t}_{3,r}.$$

Figure 5.4 shows a graphical representation of such a three dimensional CP decomposed tensor.

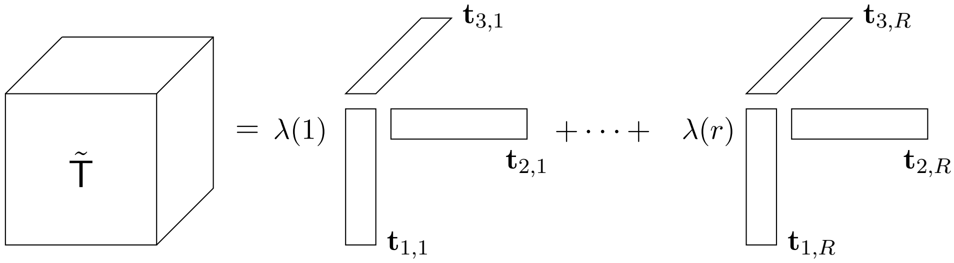


Figure 5.9: Graphical representation of a three dimensional CP tensor

5.2.3 Data-driven tensor iterative learning control

A data-driven ILC, introduced in Section 5.1.2, collects and stores the disturbance, error and input signals of all past trials in a database or a local storage device. As mentioned before, the number of stored elements of one signal, e.g. an input signal, depends directly on the product of the two parameters, the number of samples per trial N_m and the number of stored past trials N_t. For the applicability of such control approaches the storage demand reduction can be very useful for systems without any connection to large storage resources. Tensor decomposition methods can reduce the storage demand by an approximative representation of the original data and will be applied to the data-driven ILC.

As introduced in Section 5.1.2 the data is stored in time series depending on the past trials. These two-dimensional data sets can be rearranged in a multidimensional tensor structure with arbitrary dimensions, with the only restriction that the first dimension represents the samples of one trial d_{db}. For instance, a four dimensional tensor is a natural choice for heating systems, with the samples per day as one trial, the days in a week, the weeks in a year, and the number of years as dimensions. The four dimensional structure fits for a single signal application and is used in the following. But this four dimensional structure can easily extended to a five dimensional structure for multiple signal applications, where the fifth dimension represents the different signals. The disturbance signals or the measurable ambient conditions $\mathbf{a}_{d_{db}}$ of each trial can be stored, in general, in an n-dimesional tensor $\mathsf{T} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$. This tensor has the same number of elements as before when the data is stored as time series. The stored input and error signals can be rearranged in the same way. The number of elements remains for the tensorized structure with no reduction effects on the storage demand whereas the CP decomposed tensor reduces the number of stored elements in accordance to the decomposition rank. The following discussion focuses on the tensor of the ambient conditions and the calculation of the similarity criterion in CP decomposed tensor representation.

Similarity criterion with a decomposed tensor representation

Disturbances or ambient conditions play an important role, such as the outside temperature for building data and heating systems. The ambient conditions are used for the similarity criterion as in (5.2). A CP decomposition algorithm is applied to a data tensor $\mathsf{T} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$, which results in a rank R decomposed tensor $\tilde{\mathsf{T}} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$ with the factor matrices $\mathbf{T}_i \in \mathbb{R}^{I_i \times R}$ $(i = 1, \ldots, n)$, as shown in Section 5.2.2.

The main goal is the calculation of the similarity criterion (5.2) on the basis of the factor matrices $\mathbf{T}_i$ without recomputing the full tensor $\tilde{\mathsf{T}}$. The first dimension of the data tensor T represents the samples of one trial d_{db}. One trial d_{db} equals one day for the heating system application but is not fixed to a day and the iteration interval has to be chosen in accordance to the application. The data is stored in the factor matrices of the CP decomposed tensor and one trial d_{db} can be selected via the indices $i_2, \ldots, i_n$. The CP tensor representation of the similarity criterion is given by

$$\begin{aligned}\tilde{\mathsf{E}}_T(i_2, \ldots, i_n) &= \sum_{i_1} (\tilde{\mathsf{T}}(i_1, \ldots, i_n) - \mathbf{a}_{pre}(i_1))^2 \\ &= \sum_{i_1} (\tilde{\mathsf{T}}^2(i_1, \ldots, i_n) - 2\tilde{\mathsf{T}}(i_1, \ldots, i_n)\mathbf{a}_{pre}(i_1) + \mathbf{a}_{pre}^2(i_1)),\end{aligned} \tag{5.14}$$

with the data vector of the ambient condition prediction $\mathbf{a}_{pre} \in \mathbb{R}^{I_1}$ of the next trial. The CP tensor $\tilde{\mathsf{E}}_T$ with the dimension $I_2 \times \cdots \times I_n$ includes the results of the past trials d_{db}. Each term of (5.14) can be calculated separately.

The first term $\sum_{i_1} \tilde{\mathsf{T}}^2(i_1, \ldots, i_n)$ can be computed using only the factor matrices $\mathbf{T}_i$. The following shows how the square of a CP tensor $\tilde{\mathsf{T}}$ can be calculated by using the factor matrices only.

Lemma 5.1 *The square of a CP tensor $\tilde{\mathsf{T}}$ with the dimension $I_1 \times \cdots \times I_n$ and the factor matrices $\mathbf{T}_i \in \mathbb{R}^{I_i \times R}$ of rank R and $i = 1, \ldots, n$, is also a CP tensor $\tilde{\mathsf{S}}$ of the same dimension with the new factor matrices $\mathbf{S}_i \in \mathbb{R}^{I_i \times R_{max}}$ with a maximum rank of $R_{max} = R + \frac{(R^2 - R)}{2}$, introduced in [44].*

Proof 5.1 *The factor matrices of the new CP tensor $\tilde{\mathsf{S}}$ can be calculated as follows*

$$\begin{aligned}\tilde{\mathsf{S}}(i_1, \cdots, i_n) := \tilde{\mathsf{T}}^2(i_1, \cdots, i_n) &= \left(\sum_{r=1}^{R} \mathbf{t}_{1,r}(i_1) \cdots \mathbf{t}_{n,r}(i_n) \right)^2 \\ &= \sum_{r=1}^{R} \mathbf{t}_{1,r}^2(i_1) \cdots \mathbf{t}_{n,r}^2(i_n) + 2 \sum_{r<k}^{R} \mathbf{t}_{1,r}(i_1)\mathbf{t}_{1,k}(i_1) \cdots \mathbf{t}_{n,r}(i_n)\mathbf{t}_{n,k}(i_n)\end{aligned} \tag{5.15}$$

All the squared terms are represented in the first sum of equation (5.15). The squared terms are given by the outer product of the first R column vectors $\mathbf{s}_{i,r}$ of the factor matrices $\mathbf{S}_i$, with $i = 1, \ldots, n$ and $r = 1, \ldots, R$,

$$\sum_{r=1}^{R} \mathbf{s}_{1,r} \circ \cdots \circ \mathbf{s}_{n,r}. \tag{5.16}$$

The calculation of the column vectors $\mathbf{s}_{i,r}$ is given by

$$\begin{aligned}
\mathbf{s}_{1,r} &= \mathbf{t}_{1,r} \circledast \mathbf{t}_{1,r}, \qquad r = 1, \ldots, R \\
\mathbf{s}_{2,r} &= \mathbf{t}_{2,r} \circledast \mathbf{t}_{2,r}, \qquad r = 1, \ldots, R \\
\vdots \quad & \quad \vdots \qquad \vdots \\
\mathbf{s}_{n,r} &= \mathbf{t}_{n,r} \circledast \mathbf{t}_{n,r}, \qquad r = 1, \ldots, R.
\end{aligned}$$

All the cross coupling terms, i.e. the second element of the sum (5.15) can be written as an outer product of the last $\frac{R^2-R}{2}$ column vectors $\mathbf{s}_{i,r}$ of the factor matrices $\mathbf{S}_i$, with $i = 1, \ldots, n$ and $r = R+1, \ldots, R_{max}$,

$$\sum_{r=R+1}^{R_{max}} \mathbf{s}_{1,r} \circ \cdots \circ \mathbf{s}_{n,r}. \tag{5.17}$$

The column vectors $\mathbf{s}_{i,r}$ can be calculated as follows

$$\begin{aligned}
\mathbf{s}_{1,R+1} &= 2\mathbf{t}_{1,1} \circledast \mathbf{t}_{1,2}, \cdots, \mathbf{s}_{1,R_{max}} = 2\mathbf{t}_{1,R-1} \circledast \mathbf{t}_{1,R} \\
\mathbf{s}_{2,R+1} &= \mathbf{t}_{2,1} \circledast \mathbf{t}_{2,2}, \cdots, \mathbf{s}_{2,R_{max}} = \mathbf{t}_{2,R-1} \circledast \mathbf{t}_{2,R} \\
\vdots \quad & \quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
\mathbf{s}_{n,R+1} &= \mathbf{t}_{n,1} \circledast \mathbf{t}_{n,2}, \cdots, \mathbf{s}_{n,R_{max}} = \mathbf{t}_{n,R-1} \circledast \mathbf{t}_{n,R}.
\end{aligned}$$

The result is, that the CP tensor $\tilde{\mathsf{S}}$ is equal to the squared CP tensor $\tilde{\mathsf{T}}$

$$\tilde{\mathsf{T}}^2 = \left(\sum_{r=1}^{R} \mathbf{t}_{1,r} \circ \cdots \circ \mathbf{t}_{n,r} \right)^2 = \sum_{r=1}^{R} \mathbf{s}_{1,r} \circ \cdots \circ \mathbf{s}_{n,r} + \sum_{r=R+1}^{R_{max}} \mathbf{s}_{1,r} \circ \cdots \circ \mathbf{s}_{n,r} = \tilde{\mathsf{S}}. \tag{5.18}$$

The square of a CP tensor is also represented by a CP tensor with a new rank. The calculations are performed with the decomposed factors without recomputing the full tensor. An increasing CP decomposition rank leads to a higher rank of the resulting squared CP tensor, but the dimensionality remains as before. The higher rank leads to more internal memory demand but not to a higher local storage demand because every iteration needs a new calculation of the similarity criterion for the evaluation of the optimization (5.4) but no long term storage after the optimization is done.

Example 5.5 *The squared three dimensional rank two CP tensor $\tilde{\mathsf{T}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ results* *in a CP tensor* $\tilde{\mathsf{S}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ *with the new rank* $R_{max} = 3$ *and the same dimensionality*

$$\begin{aligned}
\tilde{\mathsf{T}}^2(i_1, i_2, i_3) &= \left(\sum_{r=1}^{2} t_{1,r}(i_1) t_{2,r}(i_2) t_{3,r}(i_3) \right)^2 \\
&= (t_{1,1}^2(i_1) t_{2,1}^2(i_2) t_{3,1}^2(i_3) + 2 t_{1,1}(i_1) t_{1,2}(i_1) t_{2,1}(i_2) t_{2,2}(i_2) t_{3,1}(i_3) t_{3,2}(i_3) \\
&+ t_{1,2}^2(i_1) t_{2,2}^2(i_2) t_{3,2}^2(i_3)) = \tilde{\mathsf{S}}(i_1, i_2, i_3).
\end{aligned}$$

The factor matrices $\mathbf{S}_i \in \mathbb{R}^{I_i \times 3}$, $i = 1, 2, 3$ *are given by*

$$\begin{aligned}
\mathbf{S}_1 &= [\mathbf{t}_{1,1} \circledast \mathbf{t}_{1,1}, \mathbf{t}_{1,2} \circledast \mathbf{t}_{1,2}, 2\mathbf{t}_{1,1} \circledast \mathbf{t}_{1,2}] \\
\mathbf{S}_2 &= [\mathbf{t}_{2,1} \circledast \mathbf{t}_{2,1}, \mathbf{t}_{2,2} \circledast \mathbf{t}_{2,2}, \mathbf{t}_{2,1} \circledast \mathbf{t}_{2,2}] \\
\mathbf{S}_3 &= [\mathbf{t}_{3,1} \circledast \mathbf{t}_{3,1}, \mathbf{t}_{3,2} \circledast \mathbf{t}_{3,2}, \mathbf{t}_{3,1} \circledast \mathbf{t}_{3,2}]
\end{aligned}$$

The k-mode product $\times_k$ of the first mode with a vector $\mathbf{O} \in \mathbb{R}^{I_1}$ of ones is applied after computing $\tilde{\mathsf{T}}^2$ of the first term $\sum_{i_1} \tilde{\mathsf{T}}^2(i_1, \ldots, i_n)$, which results in

$$\tilde{\mathsf{Y}}_1(i_2, \ldots, i_n) = \sum_{i_1} \left(\sum_{r=1}^{R} t_{1,r}(i_1) t_{2,r}(i_2) \cdots t_{n,r}(i_n) \right)^2 \tag{5.19}$$

$$= \left(\sum_{r=1}^{R} t_{1,r}(i_1) t_{2,r}(i_2) \cdots t_{n,r}(i_n) \right)^2 \times_1 \mathbf{O}, \tag{5.20}$$

with the dimension $I_2 \times \cdots \times I_n$. The computation of the k-mode product in CP representation without recomputing the full tensor is already implemented as MATLAB function, e.g., in the tensor toolbox [13].

The second term $\sum_{i_1}(-2\tilde{\mathsf{T}}(i_1, \cdots, i_n)\mathbf{a}_{pre}(i_1))$ of equation (5.15)is calculated by the k-mode product of the first mode, which results in a $I_2 \times \cdots \times I_n$ dimensional CP tensor

$$\tilde{\mathsf{Y}}_2 = -2\tilde{\mathsf{T}} \times_1 \mathbf{a}_{pre}. \tag{5.21}$$

The third term $\sum_{i_1} \mathbf{a}_{pre}^2(i_1)$ is constant for every index i_l, $l = 2, \ldots, n$.

The similarity criterion (5.2) can be calculated with the factor matrices of the CP tensor without recomputing the full tensor representation retaining the benefits of the CP decomposition. The resulting values of the past trials d_{db} are given by the CP tensor $\tilde{\mathsf{E}}_T \in \mathbb{R}^{I_2 \times \cdots \times I_n}$.

This investigation has shown how tensors can be used for a data-driven ILC, which results in a data-driven tensor ILC. The input, error and disturbance signals are stored in a tensor structure with arbitrary dimensions, with the only assumption, that the first tensor dimension represents the samples per trial. The similarity criterion can be calculated by using the factor matrices of the decomposed CP tensor. The number of stored elements of the ambient conditions is equal to the product $N_m \cdot N_t$ in matrix representation. The sum of the dimensions multiplied by the rank of the factor matrices $\sum_i I_i \cdot R$, $i = 1, \ldots, n$ gives the number of elements of the CP tensor representation. Thus, the reduction of the storage demand depends on the dimensionality of the data tensor and the rank R of the CP decomposed tensor for the ambient conditions.

5.2.4 Decision process for seasonal structured data on the basis of a decomposed tensor

The ambient conditions and disturbances, such as the outside temperature for heating systems, play an important role. Due to social impacts, building data often shows a weekly structure, which means that the data of Fridays is similar to other Fridays and less to Saturdays, etc. For example an office building or school is more often occupied during the week than on weekends. Therefore, a four dimensional tensor structure for the data storage $\mathsf{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ is a natural choice if a single signal is used. The number of samples in a day is the first dimension I_1, the number of days in a week is the second dimension I_2, the number of weeks in a year is the third dimension I_3 and the fourth dimension I_4 is the number of years. The application of a CP decomposition algorithm to the data tensor T for storage demand reduction leads to the decomposed tensor $\tilde{\mathsf{T}}$ with the rank R of one disturbance signal and the four factor matrices $[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4]$. Section 5.2.3 shows the calculation of the similarity criterion with the factor matrices without recomputing the full tensor. The data is stored in a CP tensor structure and one trial equals one day indexed by i_2, i_3, i_4.

For the data-driven tensor ILC, the outside temperature is already used as the ambient conditions. The data of one year is stored in the four dimensional tensor $\mathsf{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$. The resulting data tensor T with the dimensions $1440 \times 7 \times 52 \times 1$ contains 524160 elements in total. The factor matrices of the CP decomposed tensor $\tilde{\mathsf{T}}$ represents an approximation of the original data tensor T, where the storage demand reduction depends on the decomposition rank R. A rank one ($R = 1$) CP decomposition of the data tensor T results in the four factor matrices

$$\mathbf{T}_1 \in \mathbb{R}^{1440 \times 1}, \mathbf{T}_2 \in \mathbb{R}^{7 \times 1}, \mathbf{T}_3 \in \mathbb{R}^{52 \times 1}, \mathbf{T}_4 \in \mathbb{R}^{1 \times 1}$$

with 1500 elements in total. This corresponds to a reduction of elements by a factor of ≈ 350 in comparison to the data tensor T and thus to the matrix representation of the data-driven ILC. A rank fifteen CP decomposition leads to the factor matrices

$$\mathbf{T}_1 \in \mathbb{R}^{1440 \times 15}, \mathbf{T}_2 \in \mathbb{R}^{7 \times 15}, \mathbf{T}_3 \in \mathbb{R}^{52 \times 15}, \mathbf{T}_4 \in \mathbb{R}^{1 \times 15}$$

with 22500 elements, which corresponds to a reduction by a factor of ≈ 23.

The Figure 5.10 shows the correlation between the storage demand reduction factor and the rank of the CP decomposition for a data set of one year and a data set of two years with the same dimensions as mentioned before. It is also shown that the storage demand reduction factor increases if the amount of elements of the original tensor increases from one year to two years of data.

These results show that the decomposition rank has to be chosen as low as possible for the storage demand reduction. But, there is a trade off between the relative error of the CP approximation, which depends on the difference between the CP tensor $\tilde{\mathsf{T}}$ and the original data tensor T. The dependency of the relative error of the CP decomposition on the rank R for a data set of one year is shown in Figure 5.11.

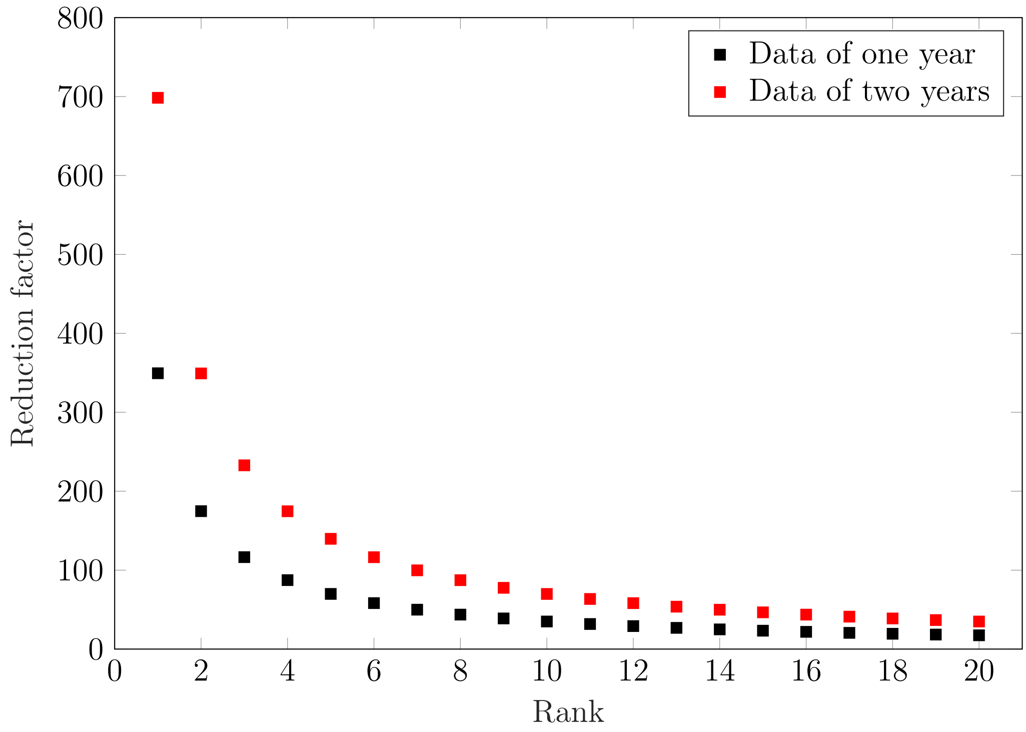


Figure 5.10: Correlation of the storage demand reduction factor and the rank

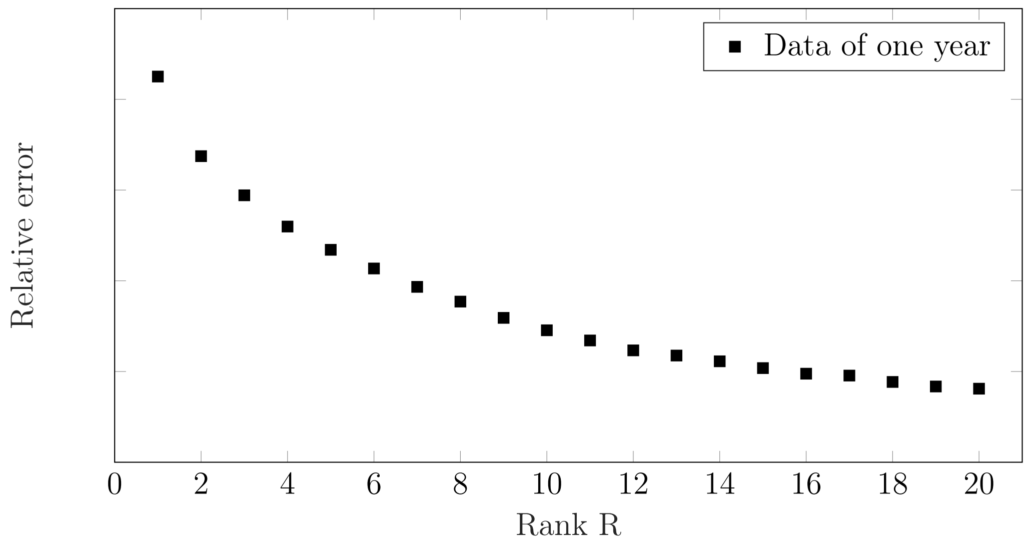


Figure 5.11: Relative error in dependency of the rank

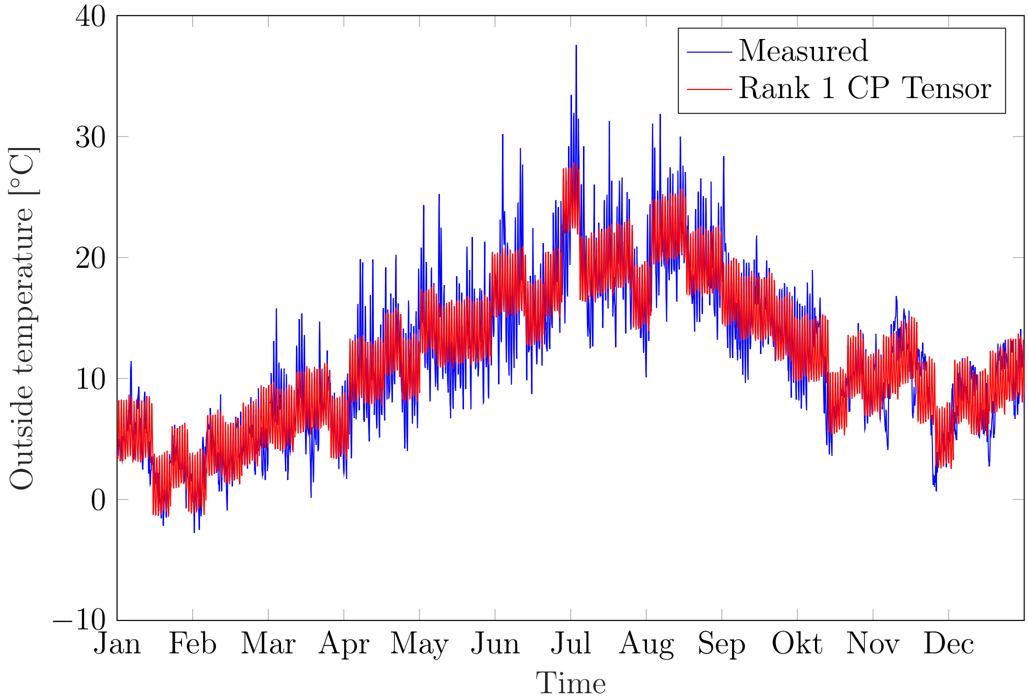


Figure 5.12: T_{out}: measured - CP decomposed, $R = 1$

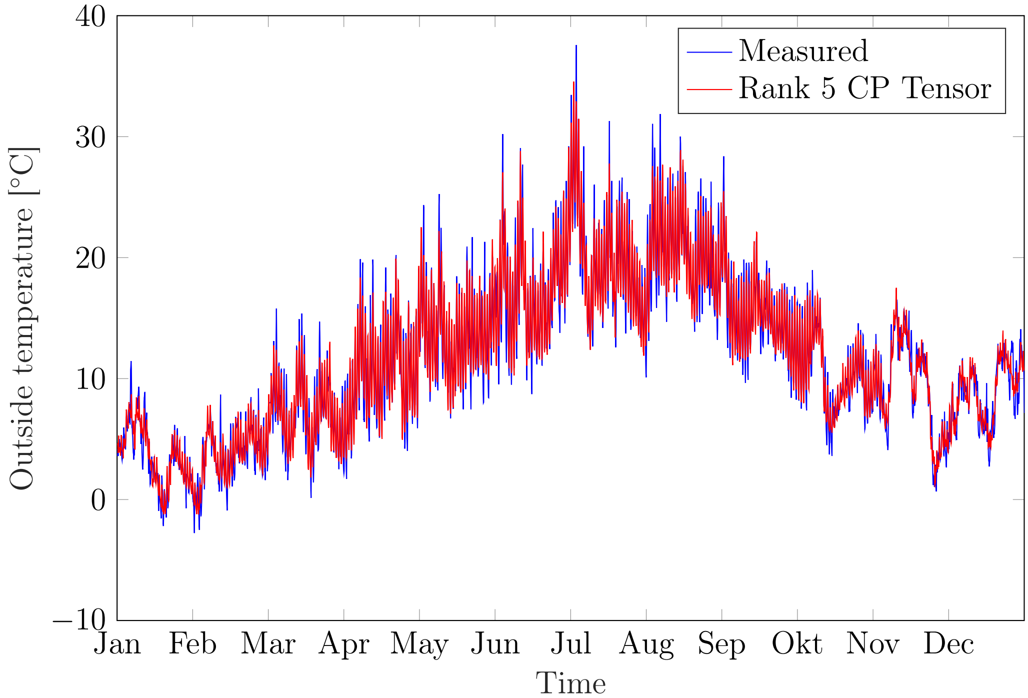


Figure 5.13: T_{out}: measured - CP decomposed, $R = 5$

The Figures 5.12 and 5.13 show the measured values of the outside temperature over one year compared to the results of the approximation of a rank one and rank five CP decomposition based on the same data sets of the outside temperature. The presented results illustrate how the approximation quality increases from a rank one to a rank five CP decomposition.

The outside temperature will be stored as decomposed CP tensor $\tilde{\mathsf{T}}$ with different ranks to calculate the similarity criterion (5.14). The results with the CP tensors will be compared to the result when the original data tensor is used for the calculation.

5.2.5 Simulation results for a heating system example

The simulation was performed with a full tensor representation used to store the data and calculate the update of the ILC. These results are compared to the simulation results where the rank five decomposed CP tensor for the outside temperature was used for the data storage and for the ILC update computation. For all simulations a prediction horizon of $H_p = 5$ h, a control horizon of $H_u = 4$ h, a learning gain of $\gamma = 2$ and a sampling time of $t_s = 60$ s was used, which means each simulation uses the same parameter set.

The stored values, $\mathbf{u}_{d_{db}}$ and $\mathbf{e}_{d_{db}}$, of the data-driven ILC are calculated from simulation results of past months. The set of available outside temperatures, error and input data includes 25 weeks or one heating season. The CP decomposed outside temperature tensor was generated with the tensorlab toolbox for Matlab [75].

First, the calculation of the decision criterion (5.4) with the original data tensor is compared with the results where the CP decomposed tensors $\tilde{\mathsf{T}}$ with different ranks were used ($R = 1, \ldots, 20$). The decision criterion is computed sixty three times with different outside temperature predictions. Figure 5.14 presents the percentage in dependency of the CP decomposition rank, where the calculation of (5.4) with the data tensor T leads to the same historic day d_{db} as the calculation with the CP decomposed tensor $\tilde{\mathsf{T}}$. This evaluation shows a strong connection between the rank of the CP tensor, which relates to the approximation accuracy, and the match of the computation results. A rank one CP tensor has an accordance of only ≈ 22 %, whereas a rank twenty CP tensor has an accordance of ≈ 99 %. For some CP decomposition ranks the accuracy decreases. The CP tensor is an approximation of the original data and the calculation of the similarity criterion (5.14) is calculated with the factor matrices of the CP tensor. This leads to the fact, that the result of (5.14) differs from the result of (5.2) with the original data and can also lead to the few drops of accuracy in Figure 5.14. But nevertheless, the over all trend follows the expected result of an increasing accordance with an increasing CP decomposition rank. For the first five ranks the number of matched calculations increases rather quickly up to ≈ 82 %. Because of these results, a rank five CP decomposition is used for the simulation to have a good accordance of ≈ 82 %, but also a storage demand reduction by a factor of ≈ 70 for one year of measured data, as Figure 5.10 shows.

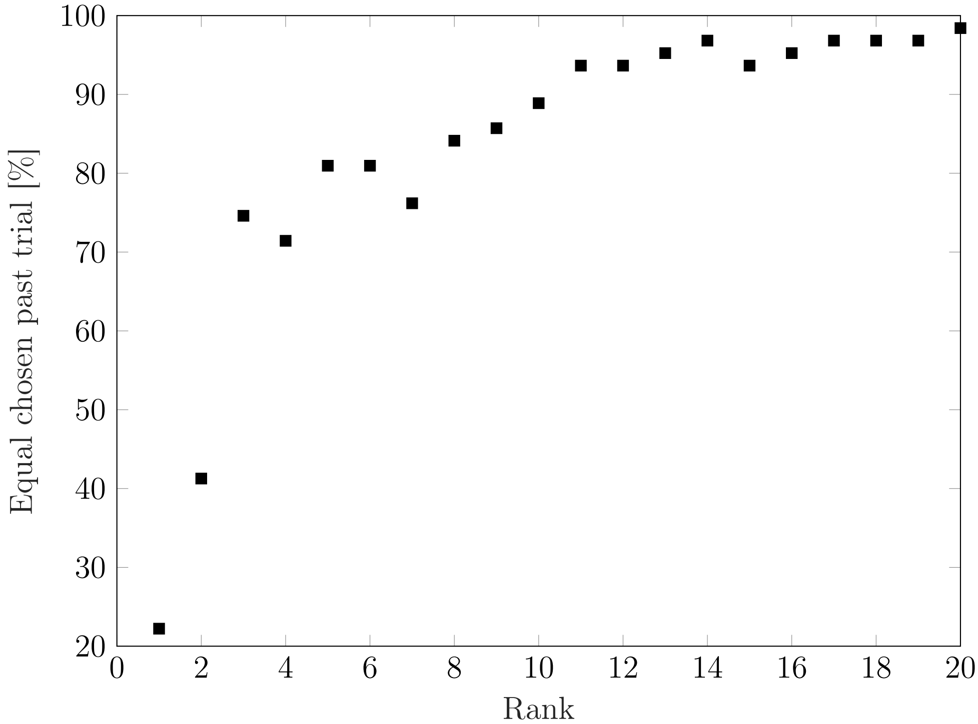


Figure 5.14: Equal chosen past trials in dependency of the rank

The comparison of the simulation results with the original data tensor and the rank five CP tensor leads to the same results and room temperatures for ≈ 82 % of the days, because of the fact that the same day d_{db} was chosen by the optimization process (5.4). Figure 5.15 shows the comparison of the simulation results of the room temperature for one day, if different days d_{db} were chosen for the ILC update calculation. The dashed line represents the simulation where the original data tensor of the outside temperature was used for the calculation and the dot-dashed line shows the result where all parameters are the same as before except that the tensor T of the outside temperature was replaced by the rank five CP decomposed tensor $\tilde{\mathsf{T}}$. The solid line shows the lower limit of the defined comfortable room temperature prescribed by the norm [24]. The simulation result shows that the deviation between the two curves remains under 0.5 K most of the time, so that the room temperature stays within the defined comfort zone.

The CP decomposition reduces the storage demand in dependency of the chosen decomposition rank in a significant manner, e.g. for two years of historic data, a full tensor with the dimensions $\mathsf{T} \in \mathbb{R}^{1440\times7\times52\times2}$ has 1048320 elements to store, whereas a rank five CP decomposed tensor with the four factor matrices $\mathbf{T}_1 \in \mathbb{R}^{1440\times5}$, $\mathbf{T}_2 \in \mathbb{R}^{7\times5}$, $\mathbf{T}_3 \in \mathbb{R}^{52\times5}$, and $\mathbf{T}_4 \in \mathbb{R}^{2\times5}$ has only 7505 elements to store, which leads to a storage demand reduction by a factor of ≈ 140.

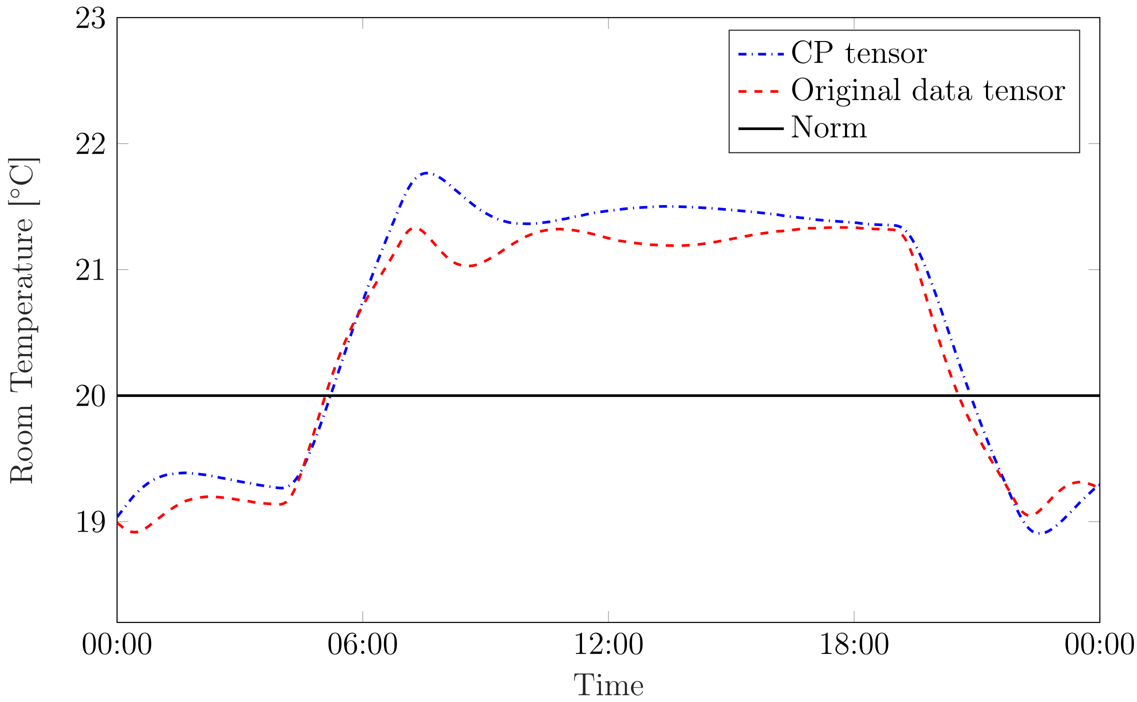


Figure 5.15: Comparison: room temperatures, different days d_{db} chosen

The investigations have shown that the similarity criterion can be calculated with the factor matrices of a CP decomposed tensor without recomputing the full tensor representation, which leads to a data-driven tensor ILC. The accordance of the simulation results with a CP decomposed tensor compared to the results with the original data tensor depends also on the chosen decomposition rank, e.g., ≈ 82 % for a rank five CP decomposed tensor. The storage demand reduction is a great benefit, particularly if the volume of stored data increases and large storage devices are unavailable.

Chapter 6

Real-time tests of predictive control algorithms for heating systems

The implementation and real-time tests of a control algorithm represent special challanges, like access to a real plant, a suitable communication interface between the plant and the hardware where the control algorithm should be implemented as well as the hardware itself. The complexity of a real-time test is also shown by the fact that the review paper, e.g. about MPC for HVAC systems cites many theory papers as opposed to few application papers where real measurement data is presented [10]. For a prototypical implementation of control algorithms, special attention should be given to the choice of the platform, the communication interfaces between the systems and the available programming language. All these factors influence the development of the control algorithms, the variety of available functions and possible restrictions to the algorithm, e.g., not every optimizer is supported by every hardware system or programming language. The developed control algorithms of a data-driven learning model predictive controller and an EMPC will be applied to a heating system. The real-time setups, the used hardware devices and the measured results are presented.

6.1 Implementation hardware for the control algorithms

Two different real-time systems were available for the implementation and first tests. One was a CompactRIO-9035 (cRIO-9035) controller from National Instruments with a dual-core CPU processor of 1.33 GHz, a random access memory of 1 GB, a local storage capacity of 4 GB and eight slots for I/O cards, such as I/O cards with digital and analog inputs and outputs. Also, two standard network ports (RJ45), one display port and two USB ports are provided by the system. This means that the real-time system can interact with other systems via the standard network ports or analog and digital signals of the I/O cards. Figure 6.1 shows a picture of the real-time system CompactRIO-9035.

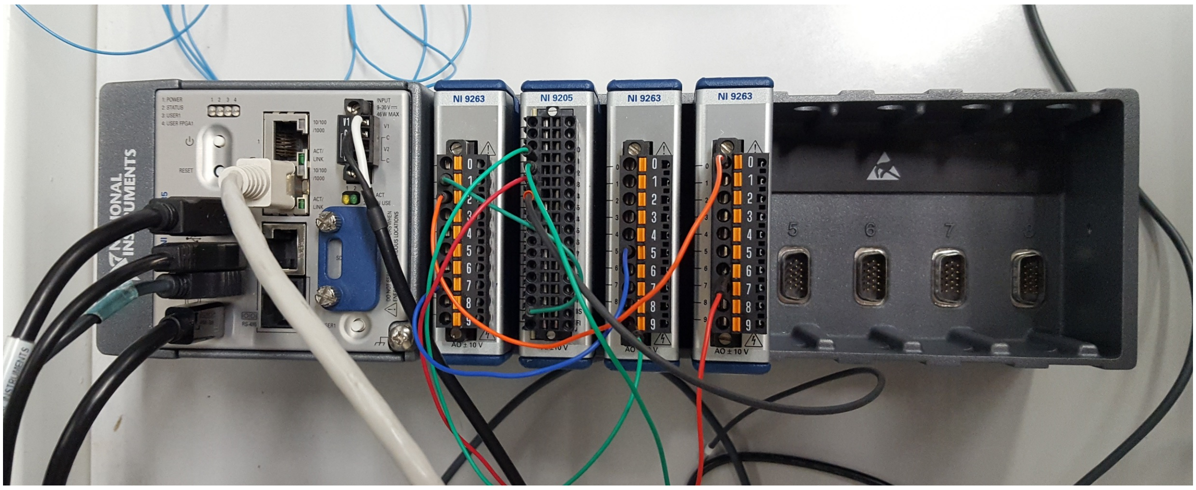


Figure 6.1: Real-time system cRIO

The cRIO system is programmed with the graphical programming language LabVIEW [3]. The programmed routines and functions in LabVIEW are called virtual instruments (VIs), which can be connected with wires to propagate variables. The control algorithms are developed in MATLAB and Simulink, which means, that these algorithms have to be transferred from MATLAB and Simulink into the LabVIEW environment. One possibility is to generate C code from the MATLAB source files with the MATLAB Coder from Mathworks [5]. But the generation of C code is only possible if the MATLAB functions are supported for code generation. Also Simulink models can be code generated by the Simulink Coder from Mathworks with the same limitation that the Simulink functions used are supported for code generation [7]. In this context, a Simulink model means not only the model of the system but also the control algorithm. The code generated Simulink model can be integrated in LabVIEW by using the software extension NI VariStand from National Instruments [4]. Thereby, the code generated Simulink model, which means the controller, is a LabVIEW VI with inputs and outputs and can be connected with other LabVIEW functions. The definition of the inputs and outputs of the Simulink model takes place in Simulink with the added NI VariStand Blocks library. The controller is developed in MATLAB and Simulink and the processing of the measurement data, control signals and program sequence are organized in LabVIEW. Another possibility is to reprogram the control algorithm in LabVIEW. This path requires that all functions, are also available in LabVIEW, especially the optimizer.

The second device was a performance real-time target machine from Speedgoat with a four-core CPU processor of 3.5 GHz, a random access memory of 4 GB, a local storage capacity of 60 GB and an I/O module with analog inputs and outputs, as well as, digital inputs and outputs. Also, two standard network ports, two display ports and five USB ports are provided by the system. Figure 6.2 shows a picture of the performance real-time target machine.

Figure 6.2: Real-time target machine

This target machine is programmed with MATLAB/Simulink. The developed Simulink models can be uploaded directly to the real-time target machine via a network connection. But the MATLAB and Simulink Coder have to be available, as well as, the Simulink Real-Time toolbox [8]. Therefore, the used MATLAB and Simulink functions have to be supported for code generation and the Simulink Real-Time Toolbox, which leads to the same limitations for the developed algorithms as for the real-time device of National Instruments. The connection from the Simulink model inputs and outputs to the analog or digital inputs and outputs of the target machine are defined in Simulink with the added Speedgoat I/O Driver Library.

Another implementation setup is required, if the developed control algorithm uses a MATLAB function which is not supported by the MATLAB and Simulink coder. Then, the implemented algorithm runs on a standard computer, e.g. a laptop. The program sequence and the communication with the plant is also implemented in LabVIEW. The difference is that the LabVIEW program calls MATLAB for solving the optimization problem instead of using a code generated Simulink model in the LabVIEW environment, with the benefit that all MATLAB functions are available for the control algorithm.

In general, a direct-digital-control device (DDC) is connected with the single components of a heating system. The control algorithm of a heating system is located at the DDC, as well as the collection of the measurement data for monitoring and control purposes. The building automation and control networks (BACnet) protocol is used for the network communication between the different components of the heating system, which is a standard network protocol for building automation systems. The implementation hardware of the control algorithm, like the CompactRIO system or the laptop, is connected to the direct-digital-control device (DDC) of the heating system and can access the control and measurement signals via the network. The benefit of this implementation setup is, that the given communication infrastructure of the HVAC system is retained. The DDC and the implementation hardware communicates via the BACnet protocol, which is realized in LabVIEW with the software tool BACnet I/P of the company OVAK Technologies.

The predictive control concepts in real-time applications need a disturbance prediction and the possibility to access this data, e.g., for heating systems the weather forecast and especially the outside temperature forecast. The weather forecast station WS-K ModBus RTU485 by the company HKW Elektronik GmbH, see Figure 6.3, provides weather forecast data for the European region, i.e., the outside temperature forecast on an hourly basis for the next four days. The company transmits the forecast data over a long wave transmitter and the weather station receives the data and provides the forecasts for further use.

Figure 6.3: Weather station

The communication of the weather station with other devices took place by the ModBus protocol. The field server EZ Gateway ModBus to BACnet by the company Sierra Monitor Corporation, see Figure 6.4, converts the ModBus protocol to the BACnet protocol, with the result that the entire communication between the different devices took place via the BACnet protocol.

Figure 6.4: Field server

The introduced components are connected through the network and communicate via the

BACnet protocol. The DDC is the interface from the external implementation hardware to the heating system. Figure 6.5 shows the implementation structure.

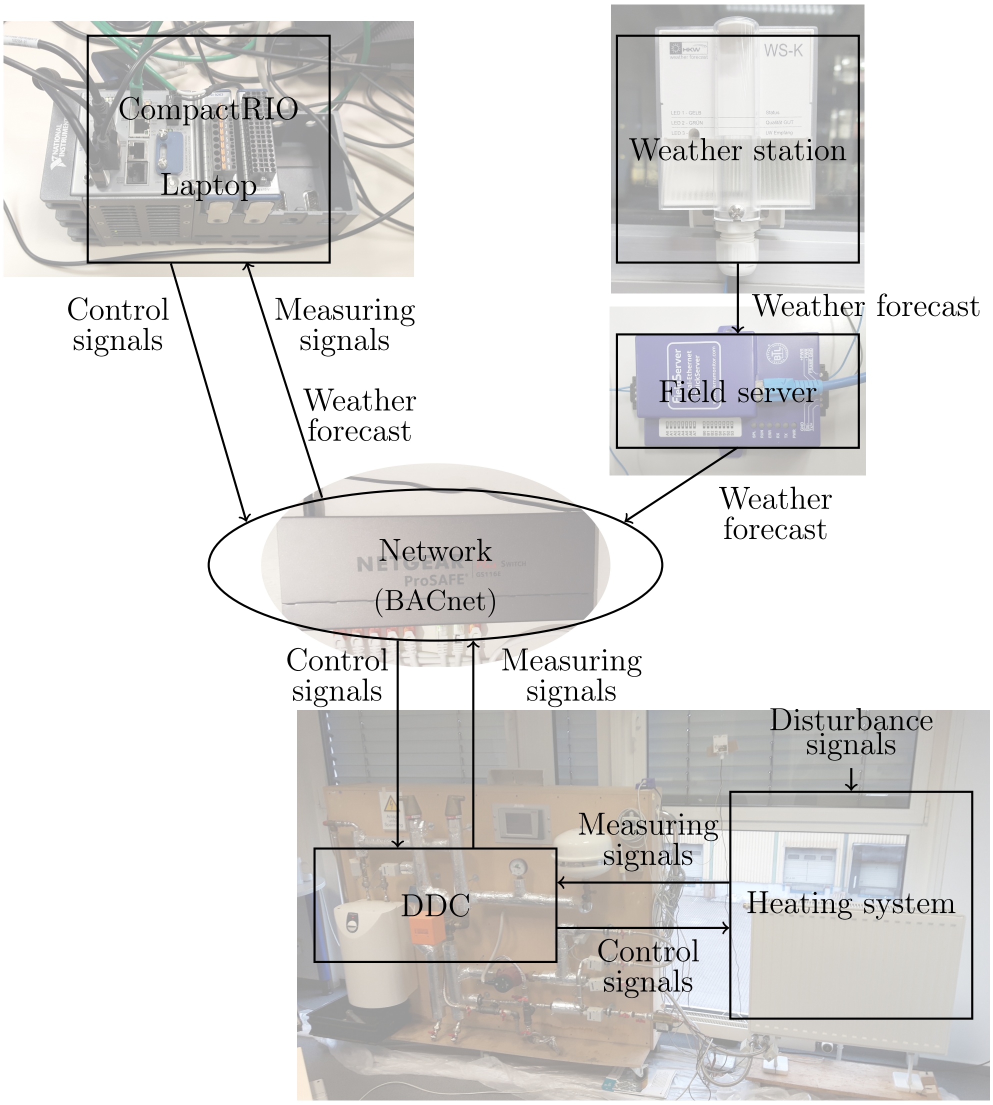


Figure 6.5: Scheme of the implementation structure

This setting was used for the implementation of a data-driven learning MPC and for an EMPC for a heating system. A hardware-in-the-loop (HIL) setup is used for testing the control algorithm in real-time on the cRIO system from National Instruments.

6.2 Real-time hardware-in-the-loop tests

Hardware-in-the-loop (HIL) simulation is used to test the real-time system and new algorithms, e.g., a control algorithm, before the algorithms were applied to the real plant. The benefit is that the system in operation is not interrupted if something were to go wrong during the first tests of the new algorithm. A scheme of the HIL structure is shown in Figure 6.6.

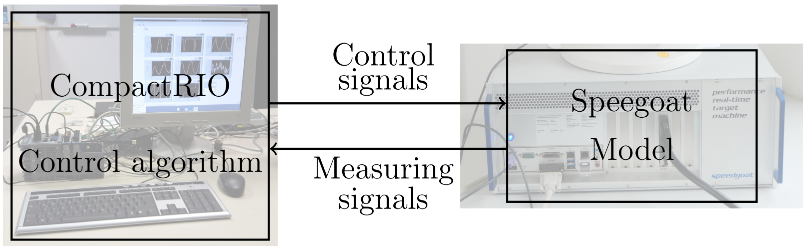


Figure 6.6: Scheme of the hardware-in-the-loop structure

Two real-time systems are connected directly via analog signals. One system is used for the control algorithm and the other one for the simulation of the dynamical behavior of the plant by using a model of the system. The control algorithm runs on the cRIO system and the corresponding model on the real-time target machine. Both system are described in the Section 6.1. The HIL setup was used for first real-time tests of the developed control algorithms.

Figure 6.7 shows the results of one real-time HIL test with a data-driven learning model predictive controller for a heating system, compared to the Simulink simulation results of the same system. The implemented data-driven learning model predictive control algorithm was introduced in Section 5.1. The real plant of the heating system is a test facility for one room of an office building. The model of that system was already introduced in Section 3.4.1. To summarize, the heating system has two control inputs. The thermal power of the boiler is controlled by the signal $\alpha \in [0, 1]$ and the position of the four way valve by the signal $\phi \in [0, 1]$. The two disturbance inputs are the outside temperature T_{out} and the volume flow $\dot{V}$, in which $\dot{V}$ is assumed to be constant for this test. The disturbance prediction of the outside temperature for the MPC is stored at the local memory of the real-time device and is the same for every day. The ambient conditions, like the outside temperature, are taken from measurement data and stored at the real-time device as well. The model of the heating system was implemented on the Speedgoat real-time target machine.

The data-driven learning model predictive controller was implemented on the National Instruments system CompactRIO-9030 and was transferred from MATLAB/Simulink to LabVIEW as described in the hardware section. The data-driven ILC is used in the same way as presented in the simulation Section 5.1.4 and adjusts the reference of the MPC with the main goal to keep the simulated room temperature in a defined range. The reference of the MPC was provided by a heating curve and the reference of the data-driven ILC $\mathbf{r}_{ilc}$, the room temperature, was set to 22 °C according to the German norm DIN-EN15251 [24], with a reduction of 2 °C during the night. A control horizon H_u of two hours and a prediction horizon H_p of three hours is used. The sampling time t_s is set to 60 seconds. All of these parameters, such as the sampling time, the ambient conditions, the prediction horizon and so on, are used for the HIL test and for the simulation in

MATLAB/Simulink. The three day test run was divided into two parts. On the first two days, the MPC was applied to the system without the data-driven ILC and on the third day the data-driven ILC was added, indicated by the vertical dot-dashed line.

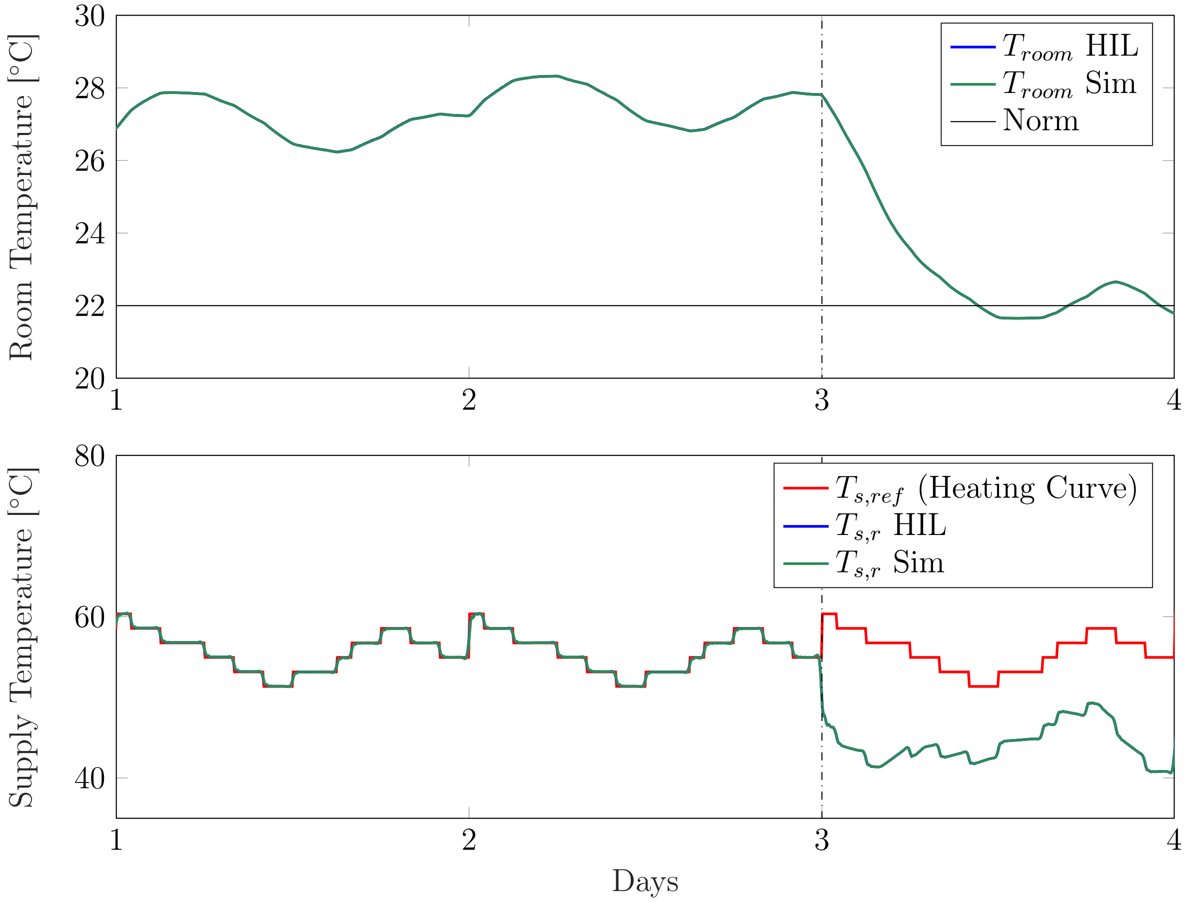


Figure 6.7: Comparison of the HIL and simulation results

The compared results of the HIL test and the simulation show no obvious differences in the room temperatures and the supply temperatures, which means that the transfer of the control algorithm to the real-time system was successful. The high room temperatures of the first two days are reduced significantly on the third day, which corresponds to the start of data-driven ILC with a reference room temperature of 22 °C. Also, the analysis of the supply temperatures show that the data-driven learning MPC works as expected. On the first two days, according to the MPC algorithm the supply temperature follows the reference $\mathbf{r}_{mpc}$ calculated by a heating curve and on the third day the supply temperature is reduced by the added data-driven ILC, which is in line with the high room temperatures of the first two days.

The real-time HIL test has shown that the data-driven learning model predictive control algorithm works on the real-time system in the same way as in the simulation. This means that the transfer from MATLAB/Simulink to the LabVIEW environment works as expected. Also a possible influence of the implemented communication between the controller and the heating system, which runs as a simulation on the second real-time device, could be investigated and did not occur. Thus, the real-time HIL test is a good way to investigate the new algorithm under real-time conditions and remove possible malfunction and errors from the algorithm without any impact to the real heating system and the building users in case of failures, which is a good preparation for the application of the algorithm to the real system.

6.3 Implementation of a data-driven learning model predictive controller

The data-driven learning model predictive controller, introduced in Section 5.1, was implemented to the heating system test facility for one room of an office building, introduced in 3.4.1 and was previously implemented and tested in the HIL setup. The result of the HIL test was that the data-driven learning model predictive controller runs stable on the National Instruments real-time system CompactRIO-9030. This system is used for the implementation of the control algorithm and the application to the real plant. The main goal of the data-driven learning MPC is to keep the room temperature in a desired comfort zone by adjusting the reference of the MPC, which is provided by a heating curve. First, the MPC is applied without the additional data-driven MPC to collect the learning data for the data-driven ILC. After three days the additional data-driven controller starts. Another aspect of the application to the test facility is the installation of a workaround and a hardware setup for the application of new control algorithms to real heating systems.

6.3.1 Plant - heating system prototype test facility

The heating system test facility for one room of an office building was previously mentioned a few times. The system consists of a boiler, a pump, a four way valve and a radiator.

Figure 6.8 shows a picture of that heating system. A model of this heating system was introduced in Section 3.4.1 and will also be used for the application to the real plant. In contrast to the HIL implementation, where the disturbance of the volume flow is assumed to be constant, the measured volume flow of each time step is used as prediction and is kept constant over the entire prediction horizon H_p. This means, that the volume flow is only constant during the optimization but changes from one time step to another. The disturbance forecast of the outside temperature $T_{out,for}$ was provided daily by the weather station introduced in the hardware section, segmented in hourly increments. The National Instrument system CompactRIO-9030 exchanges the control and measurement signals via the network with the DDC of the heating system. The complete implementation setup was shown in Figure 6.5.

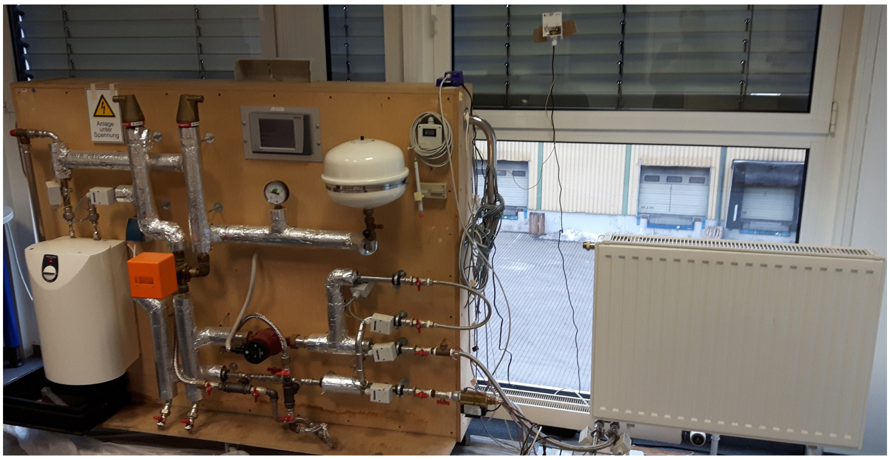


Figure 6.8: The test facility

6.3.2 Measured results - heating system prototype test facility

The measured results of the real-time implementation of the data-driven learning MPC for a heating system are presented next. For the implementation the reference supply temperature $\mathbf{r}$ for the MPC is calculated by a heating curve with respect to the outside temperature forecast. The reference $\mathbf{r}_{ilc}$ of the data-driven ILC is set to 22 °C according to the German norm DIN-EN15251 [24], with a reduction of 2 °C during the night, when the office is unoccupied. The sampling time t_s was set to 60 seconds, the control horizon H_u to one hour and the prediction horizon H_p to two hours. The outside temperature forecast was reloaded on a daily basis. According to this interval, also the data-driven ILC calculates the learning update (5.1) once a day with a learning gain $\gamma = 2$.

Figure 6.9 shows the first measured results of the data-driven learning MPC implementation of a heating system. Only the MPC is applied for the first three days, due to the fact that the data-driven ILC needs some historic data for the update calculation (5.1) and the adjustment of the reference supply temperature according to equation (5.5). The possibilities for the data-driven ILC start-up are already discussed in Section 5.1.2. Another benefit is that these initial days can be used for the comparison with the days where the data-driven ILC is added to the controller. The additional data-driven ILC on March 27th is indicated by the vertical dashed-dot line.

The measured room temperature T_{room} at daytime shows high values for all days, as Figure 6.9 shows. There are no differences between the days where the additional data-driven ILC is in operation and the days with the MPC only. But the high values of the room temperature correspond to the measured outside temperatures for these days, as Figure 6.9 shows. This means, that for these six days, the outside temperature and ambient conditions influence the room temperature in a significant way during daytime. But, the comparison between the first three days and the last three days of T_{room} at nighttime, shows that the room temperature can be reduced if the data-driven learning MPC runs. The analysis of the supply temperature $T_{s,r}$, as shown in the last subfigure of 6.9, compared to the supply reference $\mathbf{r}_{mpc}$, calculated by a standard heating curve, shows that the data-driven ILC works as expected and reduces $T_{s,r}$, if the room temperature is above the defined reference $\mathbf{r}_{ilc}$ of 22 °C during daytime and 20 °C at nighttime.

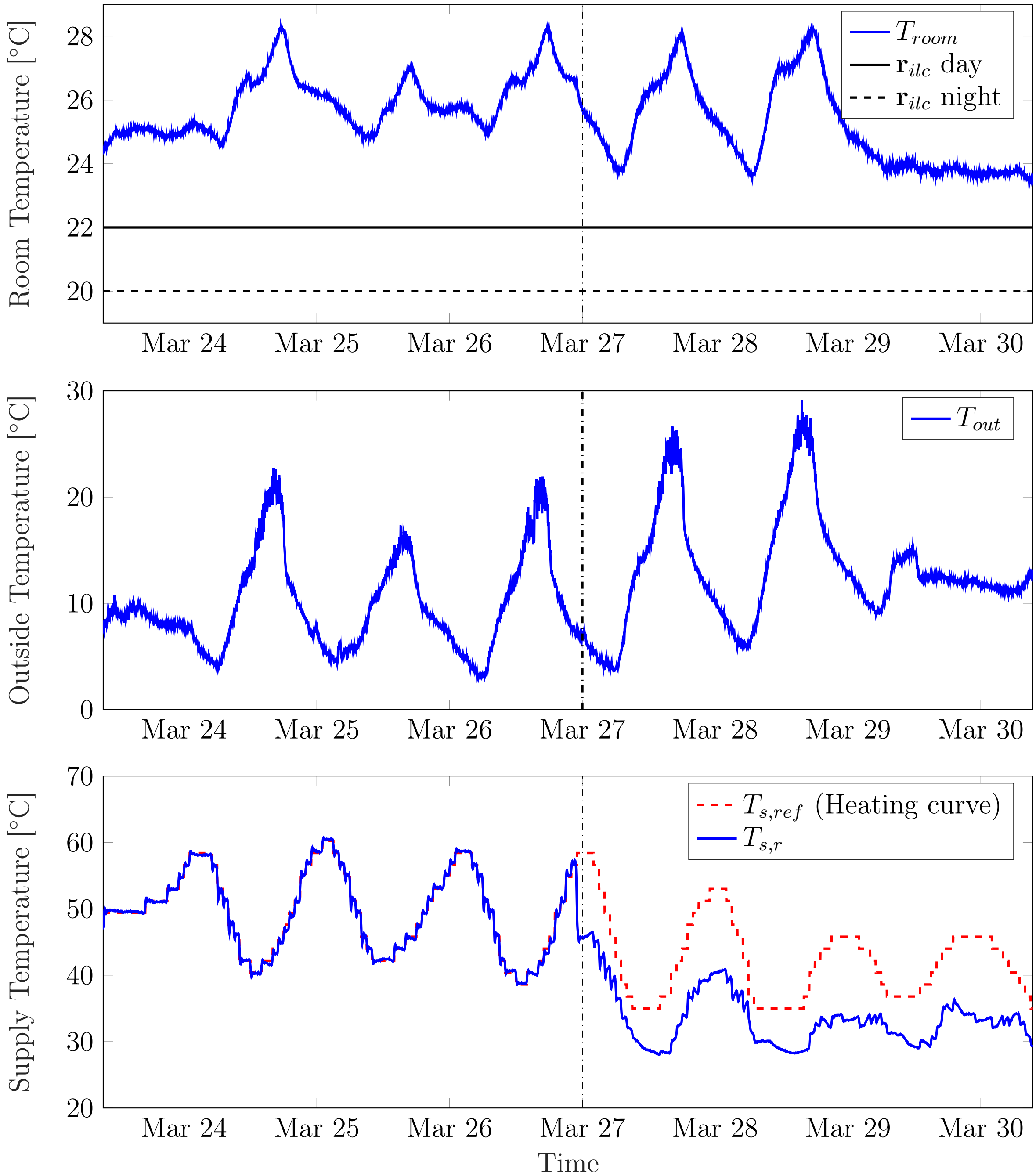


Figure 6.9: Measured results of the data-driven learning MPC real-time implementation

This application has shown, that the implemented data-driven learning MPC works as expected and reduces the supply temperature in accordance to the high room temperatures even if a small number of learning data is available. The reduced supply temperature leads to a reduction of the heating power and thus saves energy and in the end money. Nevertheless, a long run over several weeks would be helpful to acquire more experiences with this real-time implementation, specifically with outside temperatures below 20 °C. This long run was not done because of the end of the heating season and an increasing outside temperatures from day to day.

6.4 Implementation of an economic model predictive control with continuous and discrete control signals

The EMPC for continuous and discrete signals, introduced in Section 4.1.2, is applied to a heating system of an office building. This building includes seven floors with offices and each floor has its own heating circuit. The heat demand is satisfied by two boilers, which cannot be controlled continuously and the heat is transferred to the offices by radiators. The fifth floor is also supplied by an additional air ventilation system. The EMPC is applied to the system with the goal of maintaining the room temperatures in the defined corridors and comfort zones, without using any references to minimize the economical costs. Another aim is the reduction of the boiler switching frequency and energy consumption.

6.4.1 Plant - office building

The modeling of each component of the heating system would lead to a very complex model because of the high amount of floors and offices and thus to a high computational effort. For example, each floor includes more than thirty offices and room, respectively. For this reason, the model was kept as simple as possible, which means that every floor was modeled as one zone with one room temperature $T_{room,i}$ $(i = 0, \ldots, 6)$ and one radiator with the return temperature $T_{r,i}$. The heat demand of the heating circuits is satisfied by two boilers with the switching control signals α_1 and α_2, which provide the overall supply temperature $T_{s,o}$. The supply temperature $T_{s,o}$ is continuously adjustable for every heating circuit by a three-way valve with the control signal ϕ_i. The valve mixes cold water from the return to the supply of the boiler which leads to the supply temperatures $T_{s,i}$ of the floors. A pump provides a volume flow $\dot{V}_i$ for each floor separately. The modeling of the consumer and the supplier was done according to the equations introduced in Section 2.2 and identical to the process for the model of the test facility, introduced in Section 3.4.1. A scheme of one heating circuit is shown in Figure 6.10.

The fifth floor was provided with heat by an additional air handling unit (AHU), with the supply temperature $T_{s,AHU}$, the volume flow $\dot{V}_{AHU}$ and the return temperature $T_{r,AHU}$. Because of the absence of a three-way valve for the AHU the supply temperature $T_{s,AHU}$ is equal to $T_{s,o}$. The single return temperatures of the heating circuits are mixed together and water with the temperature $T_{r,o}$ returns to the boilers.

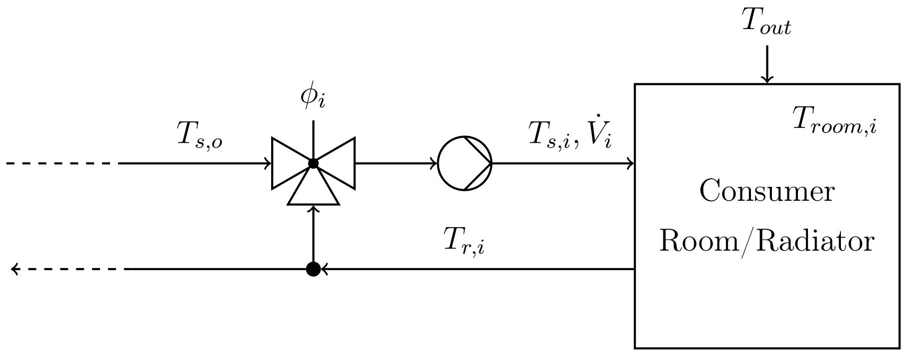


Figure 6.10: Scheme of a heating circuit

The supplier consists of two boilers with a heating power of 140 kW and 230 kW, which provides water for the heating system with the supply temperatures $T_{s,b1}$ and $T_{s,b2}$. The boilers can only be controlled in three steps and not continuously, which means that the control signals are discrete. Pumps for each boiler provide volume flows $\dot{V}_{b1}$ and $\dot{V}_{b2}$ with the assumption that the volume flows are constant and the pumps start if the corresponding boiler starts. Figure 6.11 shows the supply part of the heating system.

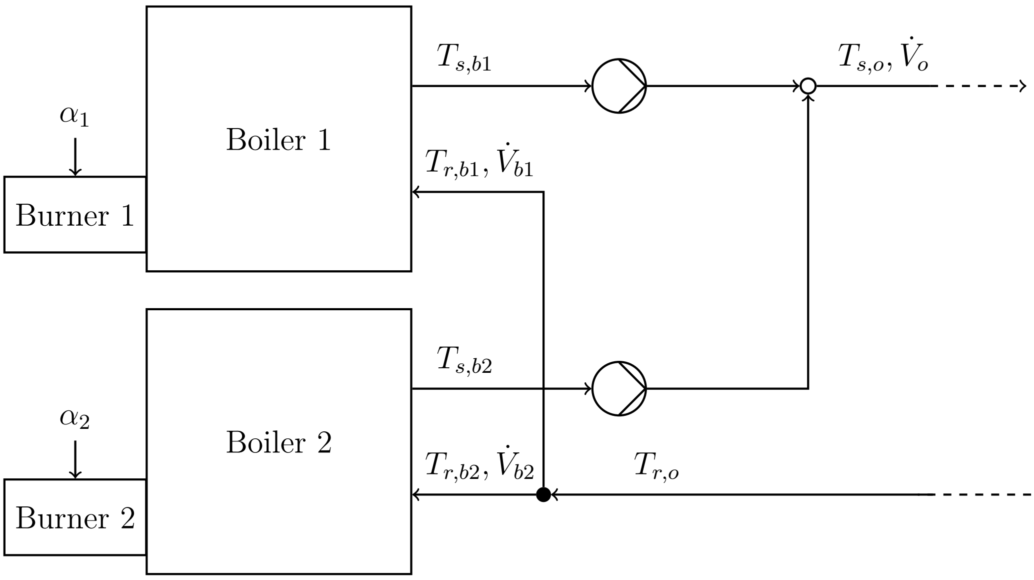


Figure 6.11: Scheme of the supplier

The overall supply temperature $T_{s,o}$ and the volume flow $\dot{V}_o$ are calculated according to the equations (2.27) and (2.25)

$$T_{s,o} = \frac{T_{s,b1}\dot{V}_{b1} + T_{s,b2}\dot{V}_{b2}}{\dot{V}_o}$$
$$\dot{V}_o = \dot{V}_{b1} + \dot{V}_{b2}.$$

A hydraulic separator separates the hydraulic of the supplier and the consumer. For simplification it is assumed that no water from return is mixed to the supply, so that the hydraulic separator can be neglected for the model.

That results in a model with 17 states, 18 inputs and 11 outputs. The volume flows of the heating circuits $T_{r,i}$ and $T_{r,AHU}$ are added as disturbance inputs to the model. The 18 inputs are separated into 9 control inputs and 9 disturbance inputs. Table 6.1 gives an overview of the different states, inputs and outputs. All state and output signals are measurable.

Table 6.1: List of the states, control signals, disturbances and outputs of the model

States	Control signals	Disturbances	Outputs
$T_{room,0}$ (Ground floor)	α_{b1}	T_{out}	$T_{room,0}$ (Ground floor)
$T_{room,1}$ (1st floor)	α_{b2}	$\dot{V}_0$ (Ground floor)	$T_{room,1}$ (1st floor)
$T_{room,2}$ (2nd floor)	ϕ_0 (Ground floor)	$\dot{V}_1$ (1st floor)	$T_{room,2}$ (2nd floor)
$T_{room,3}$ (3rd floor)	ϕ_1 (1st floor)	$\dot{V}_2$ (2nd floor)	$T_{room,3}$ (3rd floor)
$T_{room,4}$ (4th floor)	ϕ_2 (2nd floor)	$\dot{V}_3$ (3rd floor)	$T_{room,4}$ (4th floor)
$T_{room,5}$ (5th floor)	ϕ_3 (3rd floor)	$\dot{V}_4$ (4th floor)	$T_{room,5}$ (5th floor)
$T_{room,6}$ (6th floor)	ϕ_4 (4th floor)	$\dot{V}_5$ (5th floor)	$T_{room,6}$ (6th floor)
$T_{s,b1}$	ϕ_5 (5th floor)	$\dot{V}_6$ (6th floor)	$T_{s,o}$
$T_{s,b2}$	ϕ_6 (6th floor)	$\dot{V}_{AHU}$	$T_{r,o}$
$T_{r,0}$ (Ground floor)			$T_{s,b1}$
$T_{r,1}$ (1st floor)			$T_{s,b2}$
$T_{r,2}$ (2nd floor)			
$T_{r,3}$ (3rd floor)			
$T_{r,4}$ (4th floor)			
$T_{r,5}$ (5th floor)			
$T_{r,6}$ (6th floor)			
$T_{r,AHU}$			

The model parameters estimated from measurement values are summarized in the appendix in table A.3. The model parameters were estimated in two steps with the Simulink parameter estimation tool. Firstly, the parameters of the supply part and the consumer of each floor are estimated separately to determine the unknown parameters. Secondly, the entire model was used to repeat the parameter estimation starting with the predefined parameters of the first estimation step. A linearization of this model is used for the real-time implementation of the EMPC for continuous and discrete control signals. Because of the discrete control signals of the boilers and the continuous control signals of the three-way valves the linearized model is a linear hybrid model as introduced in Section 2.1.4, which results in a complex mixed-integer optimization problem for the EMPC.

6.4.2 Measured results - office building

The EMPC algorithm is implemented to the heating system of the office building introduced in 6.4.1. The algorithm runs on a laptop and communicates via the BACnet with the DDC of the system and the standard control algorithm was overridden.

The standard control algorithm uses a heating curve for every single heating circuit or floor, respectively, which results in seven heating curves and references $T_{s,i,ref}$. The corresponding supply temperatures of the particular floor are sent to the supply system and the highest supply temperature is used as reference $T_{s,ref}$ and will be provided by the boilers. According to the heating curves, the three-way valve of each heating circuit mixes cold water from the return to the supply according to the desired supply temperature $T_{s,i,ref}$. The boilers of the system are controlled by a discrete signal, which means that the boiler can be controlled in three steps 0-1-2, with two different heating power levels 1 and 2 corresponding to 50 % and 100 % of the maximum power, respectively. The first boiler has a maximum power of 175 kW and the second boiler a maximum power of 230 kW. The boiler is switched off if the control signal is zero. According to the reference supply temperature $T_{s,ref}$ the boiler is switched to the first level if the measured temperature $T_{s,o}$ is below the reference temperature $T_{s,ref}$, and switches to the second level to provide maximum heating power if the heating power of the first level is not sufficient. If the measured supply temperature $T_{s,o}$ has reached the reference supply temperature, the boiler is switched off and starts again if the measured temperature falls below the desired reference temperature. To prevent the boiler for switching with a high frequency, a hysteresis for the supply temperature is used.

In comparison to the standard control algorithm to reach the desired supply temperatures according to the seven heating curves, the goal of the EMPC algorithm is to maintain the room temperature $T_{room,i}$ of every floor in a defined corridor, together with a low switching frequency of the boilers and without any references like heating curves. The EMPC optimization problem (4.9) is solved with constraints. Table 6.2 gives an overview of the lower and upper limits of the output and the control signals.

The lower limits of the room temperatures will be decreased by 2 °C during the night. For each floor, one reference room exists where the room temperature is measured. Only the sixth floor has no reference room or measurement sensor and a mean value of all the other measured room temperatures is used as room temperature $T_{room,6}$. The measured room temperatures depend on the usage and the orientation of the reference offices. The orientation of the room defines the influence of the ambient conditions like the solar radiation to the room temperature. That means that the limits have to be chosen individually for every reference room and cannot be set necessarily to the same values for every reference room. In accordance to this, the limits for the first and second floor differ from the limits of the other floors, see Table 6.2.

Table 6.2: Constraints of the output and control signals

Outputs	Lower limit [°C]	Upper limit [°C]
$T_{room,0}$ (Ground floor)	20	22
$T_{room,1}$ (1st floor)	21	23
$T_{room,2}$ (2nd floor)	23	25
$T_{room,3}$ (3rd floor)	20	22
$T_{room,4}$ (4th floor)	20	22
$T_{room,5}$ (5th floor)	20	22
$T_{room,6}$ (6th floor)	20	22
$T_{s,o}$	40	80
$T_{r,o}$	30	80
$T_{s,b1}$	50	80
$T_{s,b2}$	50	80
α_{b1}	0	2
α_{b2}	0	2
ϕ_0 (Ground floor)	0	1
ϕ_1 (1st floor)	0	1
ϕ_2 (2nd floor)	0	1
ϕ_3 (3rd floor)	0	1
ϕ_4 (4th floor)	0	1
ϕ_5 (5th floor)	0	1
ϕ_6 (6th floor)	0	1

The costs or weighting parameters of the optimization variables depend on the individual system. The operational costs of the boiler with a heating power of 175 kW were calculated to 10.6 Euro per hour. These costs include the electrical power and heating oil consumption of the boiler. This value is adapted to the heating power of the particular boiler. Each boiler start provokes additional costs of 0.33 cents due to heating power losses to the environment. Changes of the valve position lead to minimal electricity costs of $6.25 \cdot 10^{-5}$ cent. Details about the economic cost estimation are given in [16]. These calculated costs are included directly in the cost function (4.8) of the EMPC optimization problem (4.9).

The application of the EMPC was performed while the offices were used as usual. For this reason, the installation of an automatic fallback mechanism to the standard control algorithm was necessary, to ensure that the offices were warm and the employees were able to work as usual in case of failures. One part was a keep-alive signal, which was sent from the EMPC algorithm and verified by the DDC. Another part of the security arrangements was the monitoring of the room and the return temperatures, with the goal to of preventing the building from cooling down. The fallback mechanism was located directly on the DDC and the programming was not part of this thesis. For further information see [16].

The implementation hardware and setup as introduced in Section 6.1 was used. The EMPC algorithm runs on a laptop because the MATLAB optimizer is not supported for code generation and the algorithm can not be easily transferred to the NI cRIO system and the LabVIEW environment. To ensure that the *intlinprog* function of MATLAB returns the result of the mixed-integer optimization in one sample time step, the maximal computation time for the optimization is set to two minutes. Beside from that, the default options of the *intlinprog* function are used. The prediction horizon was set to $H_p = 2$ hours and the sample time to $t_s = 2$ minutes. Figures 6.12 and 6.13 show the measured results of two days when the EMPC controlled the heating system. The supply temperatures $T_{s,b1}$ and $T_{s,b2}$ are shown in Figure 6.13 and the measured room temperatures $T_{room,i}$ and the outside temperature T_{out} in Figure 6.12.

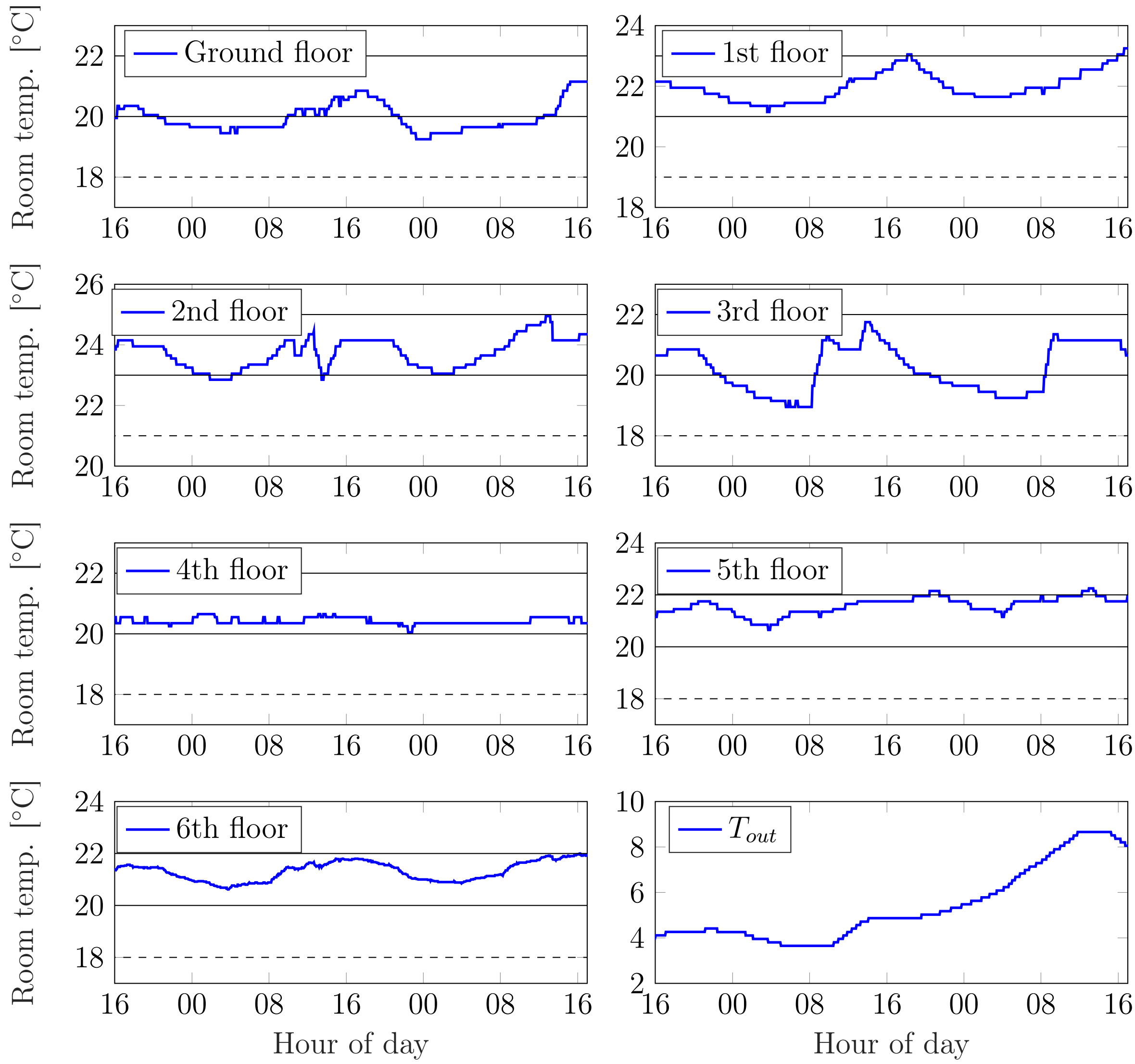


Figure 6.12: Measured room temperature of the EMPC real-time implementation starting from the 19th and ends on the 21st of December.

Drawing a general conclusion of the control behavior of the EMPC for this heating system is difficult with only two days of measured data. Nevertheless, these measured results show that the room temperatures remain in the defined comfort zones at daytime (black solid lines) and the room temperatures decrease during the night but staying inside the nighttime limits (black dashed line), as Figure 6.12 shows. Specifically the room temperatures of the ground and third floor show a significant reduction of the temperature during the night.

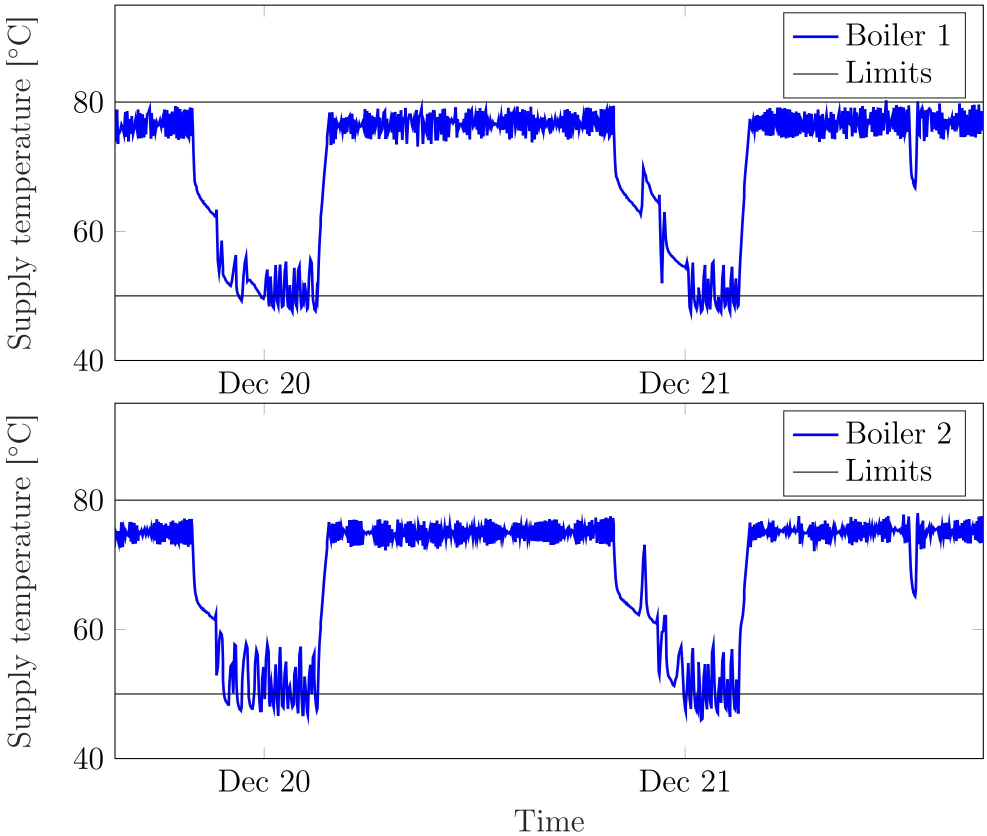


Figure 6.13: Measured supply temperature of the EMPC real-time implementation

The reduced room temperatures lead to a decreased supply temperature at nighttime, which is provided by the two boilers, see Figure 6.13. During the night, the supply temperatures of the two boilers are fixed to the lower limit of 50 °C (black solid line). Figure 6.14 shows the supply temperatures of the boilers when the standard algorithm controls the heating system in a similar situation two weeks earlier. The comparison of the supply temperatures of the two control strategies shows a significant supply temperature reduction during the night when the EMPC controls the heating system. The similarity criterion (5.2), which was introduced for the data-driven ILC for the comparison of the outside temperature forecast with the historic data sets of the outside temperature, is used to find two days when the standard control algorithm controls the heating system by comparing the outside temperature profile with the profile of the two days when the EMPC controls the system. That results in the choice of the two days which are presented in Figure 6.14.

These days are also used for the evaluation of the switching frequency by comparing the mean values of the boiler starts per day. Also, the boiler starts per day of one month are calculated to reduce the influence of daily effects. The results are summarized in Table 6.3.

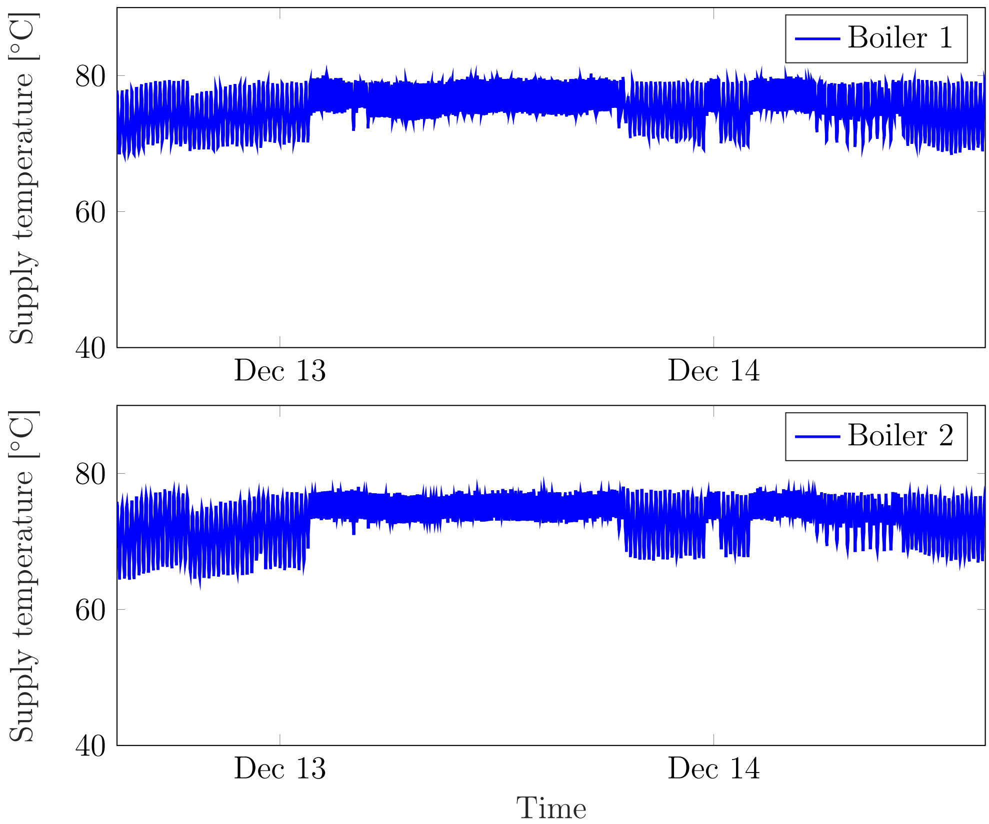


Figure 6.14: Measured supply temperature of the standard controller

The first boiler starts 49 (two days) and 30.07 (one month) times per day with the standard control strategy and 14.9 times per day with the EMPC, which corresponds to a reduction of ≈ 70 % and ≈ 50 % for the evaluated time periods. The second boiler starts 49 (two days) and 24.20 (one month) with the standard control strategy and 16.36 times per day with the EMPC, which corresponds to a reduction of ≈ 67 % and ≈ 33 % when the EMPC controls the heating system. These results show that the switching frequency can be reduced significantly when the EMPC controls the heating system. Especially for the two days with similar ambient conditions. But also if the mean values of an entire month with different ambient conditions are used for comparison.

Table 6.3: Mean values per day of the switching frequencies of the boilers

	EMPC	Standard Two Days	Standard Month
Boiler 1	14.90	49	30.07
Boiler 2	16.36	49	24.20

The investigation of the energy consumption with only two days of measurement data is difficult because of the fact that the effects, like the use of the building, the solar radiation and other ambient conditions are not considered for the energy consumption evaluation. Only the different outside temperatures are used in the form of the degree day number for the comparison of annually energy consumption. For a meaningful evaluation of the energy consumption a longer test period of measurement data is necessary, so that the effects on a daily basis are less weighted. Nevertheless, if the two days are compared to each other, a small amount of energy saving was found. But all of these values are inside the scattering pattern if the daily energy consumption is plotted against the daily mean values of the outside temperature, which means that a final conclusion of an energy consumption reduction is not possible with only two days of measurement data.

The test with the EMPC has shown that the room temperature can be kept in the defined corridors and the switching frequencies are reduced significantly. But also, that the choice of the reference rooms and suitable limits are important for the EMPC control strategy. On the one hand, if the corridor is chosen too high, the boiler will provide a high supply temperature in order to increase the room temperature in the given corridor, which results in an overheating of the building and a waste of energy. On the other hand, if the corridor for the room temperature is chosen too low, the boiler will provide a low supply temperature and the offices stay cold, which results in a higher number of user complaints. These experiences show the significance of choosing the right limits as constraints for the EMPC optimization problem. Furthermore, a long run of the EMPC control algorithm would be important for the evaluation of the energy consumption and the investigation of the control results under different ambient conditions.

Chapter 7

Conclusion and Outlook

A conclusion and an outlook for further investigations is given next.

7.1 Conclusion

The starting point of this thesis was the application of a PI controller and an MPC for a heating system, which are well known concepts. The comparison has shown, that a model predictive controller reduces the heating power in the range of 4 % but the room temperatures were nearly the same for both controllers. The application of a linear MPC was used as reference for further investigations.

The question if a model predictive control approach can be used without using a reference was stated in Section 4.1. An EMPC for continuous control signals was applied to a heating system with the advantage that no reference signal, like a heating curve, was needed. The desired room temperature was defined by the constraints of the optimization problem. Due to the added time dependent constraints a reduction of the room temperature during nighttime was realized. The simulation results have shown, that the room temperature can be kept in the defined constraints and stays at the lower limit of the comfort zone. The evaluation of the energy consumption compared to the results when a linear MPC was used shows heating power savings of approx. 15 %. Nevertheless, this approach still depends on the accuracy of the linear model. The extension of this approach to discrete and continuous control signals shows similar results in simulation. But, the comparison of the computation times of the optimization problem also clarifies that the solving of a mixed integer optimization problem is much more complex and needs more computation time. The complexity and the computation time increases with the number of discrete or integer variables. The real-time implementation results of the EMPC with discrete and continuous control signals for a heating system of an office building of two days have shown that the room temperature remains in the desired comfort zone and the switching of the boiler can be reduced significantly, which helps preserving the boilers. For the evaluation of the heating power for the two days no obvious results were shown and the heating power remains in the same area as when the conventional controller was used.

The question how the structure of the MPC optimization problem is, if a multilinear model is used was investigated in Section 4.2. This investigations of a multilinear MPC optimization problem points out that the optimization problem was convex for the class of input linear MTI systems and a prediction horizon of one. For a prediction horizon of two, the convexity could be proven for a special subclass of the input linear MTI systems. By using an example, it could be shown how the advantage of the convex optimization problem for an prediction horizon of two could be used for larger prediction horizons.

Besides the use of data for the modeling process, Section 5.1 introduces a data-driven ILC, which uses the stored historic data of the system. A data-driven ILC was applied to an MPC to improve the performance by using all stored historic iterations of the system. For the selection of one historic iteration an element selector was introduced, where a similarity and a performance criterion was stated, which leads to an optimization problem for the selection process. The simulation results have shown how a predictive controller could interact with a learning controller to improve the performance of the system by reducing the room temperature but staying inside the defined comfort zone and saves $\approx$ 11 % of heating power compared to the linear MPC. The real-time implementation of a data-driven learning MPC to a heating system test facility for an office has shown that the controller works as expected and reduces the supply temperature if the room temperature is above the defined reference. Storing the data of all historic iterations leads to an increasing storage demand. The CP tensor decomposition method is applied to the data-driven ILC, which results in a data-driven tensor ILC. The calculation of the similarity criterion was performed by only using the factor matrices of the CP decomposed tensor of the outside temperature without recomputing the full tensor representation. The investigations have shown that the storage demand reduction decreases with the CP decomposition rank and the accordance of the element selection compared to the tensor free element selection increases with the decomposition rank.

The presented investigations point out the advantages of advanced control methods for heating systems. With the results of this thesis the research question “How can control methods for heating systems be improved by using stored measurement data?” can be answered as follows. On the one hand, the data can be used for the modeling process. This developed models were used for different model predictive control approaches. On the other hand the data can be used for a data-driven controller to improve the performance of a linear MPC from iteration to iteration. The combined concept predicts the future behavior of the plant by using a model and learns from the past by using the stored historic data.

7.2 Outlook

The conclusion has summarized the advantage of model based and data-driven controllers applied to heating systems. Nevertheless, a few interesting research questions still remain.

A long run for the real-time implementation for the applied control methods would be beneficial to evaluate the controller behavior under different ambient conditions and large time periods.

The question how a data-driven ILC can be combined with an EMPC and does a combined controller improve the control results would be quite interesting.

Due to the fact that heating system models are inherited in the class of multilinear models further investigations about the convexity properties of the multilinear MPC optimization problem can be very useful. For example, finding other convex sets for larger prediction horizons. Also an extended investigation from single-input MTI systems to multi-input MTI systems would extend the field of applications.

For the data-driven tensor ILC the question how a data storage of the entire data set in a CP decomposed tensor structure effects the control results would be very interesting. Also the application of other decomposition methods to the data tensor could lead to new fascinating findings.

Bibliography

[1] *Forschung für eine umweltbezogene, zuverlässige und bezahlbare Energieversorgung, Das 6. Energieforschungsprogramm der Bundesregierung*. Bunderministerium für Wirtschaft u. Technologie. `www.bmwi.de`. Version: Juli 2011. – Download at 25.10.2018

[2] *Model Predictive Control Toolbox, Version R2015a*. Natick, Massachusetts : The MathWorks Inc., 2015

[3] *LabVIEW, Version 2016*. Austin, Texas : National Instruments, 2016

[4] *VeriStand, Version 2016*. Austin, Texas : National Instruments, 2016

[5] *MATLAB Coder, Version 2017b*. Natick, Massachusetts : The MathWorks Inc., 2017

[6] *Optimization Toolbox, Version R2017b*. The MathWorks Inc., 2017

[7] *Simulink Coder, Version 2017b*. Natick, Massachusetts : The MathWorks Inc., 2017

[8] *Simulink Real-Time, Version 2017b*. Natick, Massachusetts : The MathWorks Inc., 2017

[9] *System Identification Toolbox, Version 2017b*. Natick, Massachusetts : The MathWorks Inc., 2017

[10] Afram, A. ; Janabi-Sharifi, F.: Theory and applications of HVAC control systems–A review of model predictive control (MPC). In: *Building and Environment* 72 (2014), S. 343–355

[11] Ahn, Hyo-Sung ; Chen, YangQuan ; Moore, Kevin L.: Iterative Learning Control: Brief Survey and Categorization. In: *IEEE Transaction on Systems, Man, and Cybernetics - Part C: Application and Reviews* 37 (2007), Nr. 6

[12] Angeli, D. ; Amrit, R. ; Rawlings, J. B.: On Average Performance and Stability of Economic Model Predictive Control. In: *IEEE Transactions on Automatic Control* 57 (2012), July, Nr. 7, S. 1615–1626. `http://dx.doi.org/10.1109/TAC.2011.2179349`. – DOI 10.1109/TAC.2011.2179349. – ISSN 0018–9286

[13] Bader, B. ; Kolda, T.: *MATLAB Tensor Toolbox Version 2.5.* Available online. http://www.sandia.gov/~tgkolda/TensorToolbox/. Version: 2012

[14] Bemporad, Alberto ; Morari, Manfred: Control of systems integrating logic, dynamics, and constraints. In: *Automatica* 35 (1999), Nr. 3, 407 - 427. http://dx.doi.org/https://doi.org/10.1016/S0005-1098(98)00178-2. – DOI https://doi.org/10.1016/S0005–1098(98)00178–2. – ISSN 0005–1098

[15] Bodson, M. ; Groszkiewicz, J. E.: Multivariable adaptive algorithms for reconfigurable flight control. In: *IEEE Transactions on Control Systems Technology* 5 (1997), March, Nr. 2, S. 217–229. http://dx.doi.org/10.1109/87.556026. – DOI 10.1109/87.556026. – ISSN 1063–6536

[16] Boeck, M. ; Buchholz, J. ; Przybilla, H. ; Wolfram, A.: Implementierung und Evaluation in Gebäudeautomationssystemen. In: *Abschlussbericht - OBSERVE - Arbeitspaket A5* (2019)

[17] Boyd, Stephen ; Vandenberghe, Lieven: *Convex Optimization.* New York, NY, USA : Cambridge University Press, 2009. – ISBN 0521833787

[18] Bristow, D. A. ; Tharayil, M. ; Alleyne, A. G.: A survey of iterative learning control. In: *IEEE Control Systems* 26 (2006), Juni, Nr. 3, S. 96–114. http://dx.doi.org/10.1109/MCS.2006.1636313. – DOI 10.1109/MCS.2006.1636313. – ISSN 1066–033X

[19] Chen, C.T.: Linear System Theory and Design. In: *Oxford University Press, New York* (1998)

[20] Chen, T.Y.: Application of adaptive predictive control to a floor heating system with a large thermal lag. In: *Energy and Buildings* 34 (2002), Nr. 1, 45 - 51. http://dx.doi.org/https://doi.org/10.1016/S0378-7788(01)00076-7. – DOI https://doi.org/10.1016/S0378–7788(01)00076–7. – ISSN 0378–7788

[21] Cichocki, A. ; Zdunek, R. ; Phan, A. ; Amari, S.: *Nonnegative matrix and tensor factorizations.* Wiley, Chichester, 2009

[22] Cichocki, Andrzej ; Mandic, Danilo P. ; Phan, Anh H. ; Caiafa, Cesar F. ; Zhou, G. ; Zhao, Qibin ; Lathauwer, Lieven D.: Tensor Decompositions for Signal Processing Applications From Two-way to Multiway Component Analysis. In: *CoRR* abs/1403.4462 (2014). http://arxiv.org/abs/1403.4462

[23] Constanzo, G. T. ; Iacovella, S. ; Ruelens, F. ; Leurs, T. ; Claessens, B.: Experimental analysis of data-driven control for a building heating system. In: *CoRR* abs/1507.03638 (2015)

[24] DIN-EN-15251:2007 ; V., DIN Deutsches I. e. (Hrsg.): *Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics; German version EN 15251:2007.* Beuth Verlag GmbH, 2007

[25] Ellis, Matthew ; Durand, Helen ; Christofides, Panagiotis D.: A tutorial review of economic model predictive control methods. In: *Journal of Process Control* 24 (2014), Nr. 8, S. 1156 – 1178. `http://dx.doi.org/https://doi.org/10.1016/j.jprocont.2014.03.010`. – DOI https://doi.org/10.1016/j.jprocont.2014.03.010

[26] Findeisen, R. ; Allgöwer, F.: An introduction to nonlinear model predictive control. In: *21st Benelux Meeting on Systems and Control* Bd. 11, 2002, S. 119–141

[27] Florez, J. ; Barney, G.C.: Adaptive control of central heating systems: part 1: optimum start time control. In: *Applied Mathematical Modelling* 11 (1987), Nr. 2, 89 - 95. `http://dx.doi.org/https://doi.org/10.1016/0307-904X(87)90151-X`. – DOI https://doi.org/10.1016/0307–904X(87)90151–X. – ISSN 0307–904X

[28] Foellinger, O.: *Nichtlineare Regelung 1*. 8. Auflage. R. Oldenburg Verlag, 1998

[29] Grasedyk, L. ; Kressner, L. ; Tobler, C.: A literature survey of low-rank tensor approximation techniques. In: *GAMM-Mitteilungen* 36 (2013), S. 53–78

[30] Halvgaard, R. ; Poulsen, N. K. ; Madsen, H. ; Jørgensen, J. B.: Economic Model Predictive Control for building climate control in a Smart Grid. In: *Proc. IEEE PES Innovative Smart Grid Technologies (ISGT)*, 2012, S. 1–6

[31] Henze, G. P.: Model predictive control for buildings: a quantum leap? In: *Journal of Building Performance Simulation* 6 (2013), Nr. 3, S. 157–158

[32] Inc., The M.: *Model Predictive Control Toolbox, Online Documentation*, 03 2020. `https://de.mathworks.com/help/mpc/index.html?s_tid=CRUX_lftnav`

[33] Isidori, A.: *Nonlinear control systems*. Springer-Verlag London, 1995

[34] Kelman, Anthony ; Borrelli, Francesco: Bilinear Model Predictive Control of a HVAC System Using Sequential Quadratic Programming. In: *18th IFAC World Congress*, 2011, S. 9869–9874

[35] Khalil, H. K.: *Nonlinear Systems*. Bd. Third Edition. Prentice Hall, 2000

[36] Kolda, T. ; Bader, B.: Tensor Decompositions and Applications. In: *SIAM Review* 51 (2009), Nr. 3, S. 455–500

[37] Krstic, M. ; Kanellakopoulos, I. ; Kokotovic, P.: *Nonlinear and Adaptive Control Design*. John Wiley and Sons, Inc., 1995

[38] Kruppa, K.: Dokumentation Matlab HeatLib - Zur Vewendung mit Matlab / Institut für Regelungstechnik. TU Hamburg-Harburg, 2012. – Forschungsbericht

[39] Kruppa, K. ; Lichtenberg, G.: Decentralized State Feedback Design for Multilinear Time-Invariant Systems. In: *IFAC-PapersOnLine* 50 (2017), Nr. 1, 5616 - 5621. `http://dx.doi.org/https://doi.org/10.1016/j.ifacol.2017.08.1108`. – DOI https://doi.org/10.1016/j.ifacol.2017.08.1108. – ISSN 2405–8963. – 20th IFAC World Congress

[40] Kruppa, K. ; Mueller, T. ; Lautenschlager, B. ; Lichtenberg, G.: State space models as common tools for control design, optimization and fault detection in building systems. In: *Central European Symposium on Building Physics 2016 / BauSIM 2016*, 2016

[41] Kruppa, K. ; Pangalos, G. ; Lichtenberg, G.: Multilinear Approximation of Nonlinear State Space Models. In: *19th World Congress* IFAC, 2014, S. 9474–9479

[42] Lautenschlager, B. ; Kruppa, K. ; Lichtenberg, G.: Convexity Properties of the Model Predictive Control Problem for Subclasses of Multilinear Time-Invariant Systems. In: *IFAC-PapersOnLine* 48 (2015), Nr. 23, 148 - 153. `http://dx.doi.org/https://doi.org/10.1016/j.ifacol.2015.11.275`. – DOI https://doi.org/10.1016/j.ifacol.2015.11.275. – ISSN 2405–8963. – 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015

[43] Lautenschlager, B. ; Lichtenberg, G.: Data-driven Iterative Learning for Model Predictive Control of Heating Systems. In: *IFAC-PapersOnLine* 49 (2016), Nr. 13, S. 175 – 180. `http://dx.doi.org/http://dx.doi.org/10.1016/j.ifacol.2016.07.947`. – DOI http://dx.doi.org/10.1016/j.ifacol.2016.07.947. – ISSN 2405–8963. – 12th Workshop on Adaptation and Learning in Control and Signal Processing, 2016Eindhoven, The Netherlands, 29 June - 1 July 2016

[44] Lautenschlager, B. ; Schmidt, S. PfeS. P. ; Lichtenberg, G.: Real-Time Iterative Learning Control - Two applications with timescales between years and nanoseconds. In: *International Journal of Adaptive Control and Signal Processing: Special Issue: Learning-based Adaptive Control: Theory and Applications;* 33 (2018), Nr. 2, S. 424–444

[45] Lee, Jay H. ; Lee, Kwang S.: Iterative learning control applied to batch processes: An overview. In: *Control Engineering Practice* 15 (2007), Nr. 10, S. 1306–1318

[46] Lee, J.H.: Model predictive control: Review of the three decades of development. In: *Int. J. Control Autom. Syst.* (2011)

[47] Lichtenberg, G.: *Hybrid Tensor Systems*, Hamburg University of Technology, Habilitation, 2011

[48] Ljung, L.: *System identification: Theory for the User.* Englewood Cleffs, Prentice-Hall, 1987

[49] Lunze, J.: *Regelungstechnik 1.* 5. Springer, 2006. – ISBN 3–540–28326–9

[50] Lunze, J.: *Regelungstechnik 2.* 4. Springer, 2006. – ISBN 3–540–32335–X

[51] M., Manfred ; Lee, J. H.: Model predictive control: past, present and future. In: *Computers & Chemical Engineering* 23 (1999), Nr. 4-5, S. 667 – 682. – ISSN 0098–1354

[52] Ma, Jingran ; Qin, S J. ; Salsbury, Timothy: Application of economic MPC to the energy and demand minimization of a commercial building. In: *Journal of Process Control* 24 (2014), Nr. 8, S. 1282–1291

[53] Ma, Y. ; Borrelli, F. ; Hencey, B. ; Coffey, B. ; Bengea, S. ; Haves, P.: Model Predictive Control for the Operation of Building Cooling Systems. In: *IEEE Transactions on Control Systems Technology* 20 (2012), Mai, Nr. 3, S. 796–803. http://dx.doi.org/10.1109/TCST.2011.2124461. – DOI 10.1109/TCST.2011.2124461. – ISSN 1063–6536

[54] Maciejowski, J.: *Predictive Control with Constraints*. Pearson Education Limited 2002, 2002. – ISBN 0 201 39823 0

[55] Mayne, David Q.: Model predictive control: Recent developments and future promise. In: *Automatica* 50 (2014), Nr. 12, 2967 - 2986. http://dx.doi.org/https://doi.org/10.1016/j.automatica.2014.10.128. – DOI https://doi.org/10.1016/j.automatica.2014.10.128. – ISSN 0005–1098

[56] Mendoza-Serrano, David I. ; Chmielewski, Donald J.: Smart grid coordination in building HVAC systems: EMPC and the impact of forecasting. In: *Journal of Process Control* 24 (2014), Nr. 8, 1301 - 1310. http://dx.doi.org/https://doi.org/10.1016/j.jprocont.2014.06.005. – DOI https://doi.org/10.1016/j.jprocont.2014.06.005. – ISSN 0959–1524. – Economic nonlinear model predictive control

[57] Minakais, M. ; Mishra, S. ; Wen, J.T.: Groundhog Day: Iterative learning for building temperature control. In: *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, 2014, S. 948–953

[58] Mishra, S. ; Coaplen, J. ; Tomizuka, M.: Precision Positioning of Wafer Scanners Segmented Iterative Learning Control for Nonrepetitive Disturbances [Applications of Control]. In: *IEEE Control Systems* 27 (2007), August, Nr. 4, S. 20–25. http://dx.doi.org/10.1109/MCS.2007.384130. – DOI 10.1109/MCS.2007.384130. – ISSN 1066–033X

[59] Müller, Thorsten ; Kruppa, Kai ; Lichtenberg, Gerwald ; Réhault, Nicolas: Fault Detection with Qualitative Models reduced by Tensor Decomposition methods. In: *IFAC-PapersOnLine* 48 (2015), Nr. 21, 416 - 421. http://dx.doi.org/https://doi.org/10.1016/j.ifacol.2015.09.562. – DOI https://doi.org/10.1016/j.ifacol.2015.09.562. – ISSN 2405–8963. – 9th IFAC Symposium on Fault Detection, Supervision andSafety for Technical Processes SAFEPROCESS 2015

[60] Oldewurtel, Frauke ; Parisio, Alessandra ; Jones, Colin N. ; Gyalistras, Dimitrios ; Gwerder, Markus ; Stauch, Vanessa ; Lehmann, Beat ; Morari, Manfred: Use of model predictive control and weather forecasts for energy efficient building climate control. In: *Energy and Buildings* 45 (2012), S. 15 – 27. – ISSN 0378–7788

[61] Pangalos, G.: *Model-based controller design methods for heating systems*, TUHH, Diss., 2016. http://hdl.handle.net/11420/1327

[62] Pangalos, G. ; Eichler, A. ; Lichtenberg, G.: Tensor Systems: Multilinear Modeling and Applications. In: *3rd Int. Conference on Simulation and Modeling Methodologies, Technologies and Applications*, 2013. – submitted

[63] Pérez-Lombard, Luis ; Ortiz, José ; Pout, Christine: A review on buildings energy consumption information. In: *Energy and buildings* 40 (2008), Nr. 3, S. 394–398

[64] Pfeiffer, S. ; Lichtenberg, G.: Iterative Learning Control exploiting SO(2) Symmetry for a Free-Electron Laser. In: *IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*, 2013

[65] Pfeiffer, S. ; Lichtenberg, G. ; Schmidt, C. ; Schlarb, H.: Tensor techniques for iterative learning control of a free-electron laser. In: *2012 IEEE International Conference on Control Applications*, 2012. – ISSN 1085–1992, S. 160–165

[66] Privara, Samuel ; Siroky, Jan ; Ferkl, Lukas ; Cigler, Jiri: Model predictive control of a building heating system: The first experience. In: *Energy and Buildings* 43 (2011), S. 564 – 572. – ISSN 0378–7788

[67] Qin, S.Joe ; Badgwell, Thomas A.: A survey of industrial model predictive control technology. In: *Control Engineering Practice* 11 (2003), Nr. 7, 733 - 764. http://dx.doi.org/https://doi.org/10.1016/S0967-0661(02)00186-7. – DOI https://doi.org/10.1016/S0967–0661(02)00186–7. – ISSN 0967–0661

[68] Rawlings, J. B. ; Angeli, D. ; Bates, C. N.: Fundamentals of economic model predictive control. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012. – ISSN 0191–2216, S. 3851–3861

[69] Rogers, E. ; Owens, D. H. ; Werner, H. ; Freeman, C. T. ; Lewin, P. L. ; Kichhoff, S. ; Schmidt, C. ; Lichtenberg, G.: Norm-Optimal Iterative Learning Control with Application to Problems in Accelerator-Based Free Electron Lasers and Rehabilitation Robotics. In: *European Journal of Control* 16 (2010), Nr. 5, S. 497–522

[70] Rugh, W. J.: Linear System Theory. In: *Prentice Hall, Englewood Cliffs (NJ)* (1993)

[71] RUGH, Wilson J.: *Nonlinear System Theory.* The John Hopkins University Press, 1981. – ISBN 0–8018–2549–0

[72] SEWE, Erik ; PANGALOS, Georg ; LICHTENBERG, Gerwald: Fault Detection for Heating Systems using Tensor Decompositions of Multi-Linear Models. In: *7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (2017)

[73] ŠIROKÝ, Jan ; OLDEWURTEL, Frauke ; CIGLER, Jiří ; PRÍVARA, Samuel: Experimental analysis of model predictive control for an energy efficient building heating system. In: *Applied energy* 88 (2011), Nr. 9, S. 3079–3087

[74] STEINBERG, M: Historical Overview of Research in Reconfigurable Flight Control. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 219 (2005), Nr. 4, 263-275. `http://dx.doi.org/10.1243/095441005X30379`. – DOI 10.1243/095441005X30379

[75] VERVLIET, N. ; DEBALS, O. ; SORBER, L. ; VAN BAREL, M. ; DE LATHAUWER, L.: *Tensorlab 3.0.* `http://www.tensorlab.net`. Version: Mar. 2016. – Available online.

[76] WANG, Y. ; GAO, F. ; III, F. J. D.: Survey on iterative learning control, repetitive control, and run-to-run control. In: *Journal of Process Control* 19 (2009), Nr. 10, S. 1589 – 1600. – ISSN 0959–1524

[77] YAN, Xiuying ; REN, Qingchang ; MENG, Qinglong: Iterative learning control in large scale HVAC system. In: *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, 2010, S. 5063–5066

[78] YANG, Dae R. ; LEE, Kwang S. ; AHN, Hyo J. ; LEE, J. H.: Experimental application of a quadratic optimal iterative learning control method for control of wafer temperature uniformity in rapid thermal processing. In: *IEEE Transactions on Semiconductor Manufacturing* 16 (2003), Februar, Nr. 1, S. 36–44. `http://dx.doi.org/10.1109/TSM.2002.807740`. – DOI 10.1109/TSM.2002.807740. – ISSN 0894–6507

[79] ZIESING, H.: Anwendungsbilanzen für die Endenergiesektoren in Deutschland in den Jahren 2013 und 2016. In: *www.bmwi.de* (2017, Download at 25.10.2018)

Appendix A

Models

This appendix chapter corresponds to the models, which are used in this thesis.

A.1 Model parameter

Table A.1: Estimated parameters of the model in Section 2.2.5

Parameter	Symbol	Value
Boiler volume	V_{boiler}	$0.2275\,\mathrm{m}^3$
Transfer coefficient boiler - environment	$k_{b,loss}$	$23.83\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Maximal boiler power boiler	$\dot{Q}_P$	$249.984\,\mathrm{kW}$
Boiler environment temperature	$T_{b,env}$	$292\,\mathrm{K}$
Radiator volume	V_r	$6.6386\,\mathrm{m}^3$
Transfer coefficient building - environment	$k_{room,o}$	$9264.56\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building	$k_{r,room}$	$6739.15\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building	C_{room}	$1546510000\,\frac{\mathrm{J}}{\mathrm{K}}$
Reference room temperature	$T_{room,r}$	$293.5\,\mathrm{K}$
Mean volume flow	$\dot{V}_{mean}$	$0.0009542\,\frac{\mathrm{m}^3}{\mathrm{s}}$
Slope	b_{vol}	$0.000839\,\frac{\mathrm{m}^3\mathrm{K}}{\mathrm{s}}$

Table A.2: Estimated parameters of the test facility model in Section 3.4.1

Parameter	**Symbol**	**Value**
Boiler volume	V_{boiler}	$0.01\,\mathrm{m^3}$
Transfer coefficient boiler - environment	$k_{b,loss}$	$1.10815\,\frac{\mathrm{kW}}{\mathrm{m^2k}}$
Maximal boiler power boiler	$\dot{Q}_P$	$2\,\mathrm{kW}$
Boiler environment temperature	$T_{b,env}$	$296\,\mathrm{K}$
Radiator volume	V_r	$0.005\,\mathrm{m^3}$
Transfer coefficient building - environment	$k_{room,o}$	$37.6797\,\frac{\mathrm{kW}}{\mathrm{m^2K}}$
Transfer coefficient radiator - building	$k_{r,room}$	$33.7018\,\frac{\mathrm{kW}}{\mathrm{m^2K}}$
Thermal capacity of the building	C_{room}	$405436\,\frac{\mathrm{J}}{\mathrm{K}}$

Table A.3: Estimated parameters of the office building model in Section 6.4

Parameter	**Symbol**	**Value**
Boiler volume 1	$V_{boiler,1}$	$0.75\,\mathrm{m}^3$
Transfer coefficient boiler - environment 1	$k_{b,loss,1}$	$50\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Maximal boiler power boiler 1	$\dot{Q}_{P,1}$	$125\,\mathrm{kW}$
Boiler environment temperature 1	$T_{b,env,1}$	$294\,\mathrm{K}$
Boiler volume 2	$V_{boiler,2}$	$0.8\,\mathrm{m}^3$
Transfer coefficient boiler - environment 2	$k_{b,loss,2}$	$70\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Maximal boiler power boiler 2	$\dot{Q}_{P,2}$	$200\,\mathrm{kW}$
Boiler environment temperature 2	$T_{b,env,2}$	$294\,\mathrm{K}$
Radiator volume 0	$V_{r,0}$	$633\,\mathrm{m}^3$
Transfer coefficient building - environment 0	$k_{room,o,0}$	$2381\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building 0	$k_{r,room,0}$	$1375\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building 0	$C_{room,0}$	$1057129084\,\frac{\mathrm{J}}{\mathrm{K}}$
Radiator volume 1	$V_{r,1}$	$13\,\mathrm{m}^3$
Transfer coefficient building - environment 1	$k_{room,o,1}$	$1086\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building 1	$k_{r,room,1}$	$583\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building 1	$C_{room,1}$	$642228201\,\frac{\mathrm{J}}{\mathrm{K}}$
Radiator volume 2	$V_{r,2}$	$2.6\,\mathrm{m}^3$
Transfer coefficient building - environment 2	$k_{room,o,2}$	$774\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building 2	$k_{r,room,2}$	$447\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building 2	$C_{room,2}$	$520043646\,\frac{\mathrm{J}}{\mathrm{K}}$
Radiator volume 3	$V_{r,3}$	$0.6\,\mathrm{m}^3$
Transfer coefficient building - environment 3	$k_{room,o,3}$	$493\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building 3	$k_{r,room,3}$	$215\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building 3	$C_{room,3}$	$316256884\,\frac{\mathrm{J}}{\mathrm{K}}$
Radiator volume 4	$V_{r,4}$	$4.5\,\mathrm{m}^3$
Transfer coefficient building - environment 4	$k_{room,o,4}$	$923\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building 4	$k_{r,room,4}$	$490\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building 4	$C_{room,4}$	$162549112\,\frac{\mathrm{J}}{\mathrm{K}}$
Radiator volume 5	$V_{r,5}$	$3.3\,\mathrm{m}^3$
Transfer coefficient building - environment 5	$k_{room,o,5}$	$2788\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building 5	$k_{r,room,5}$	$469\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building 5	$C_{room,5}$	$943639500\,\frac{\mathrm{J}}{\mathrm{K}}$
Radiator volume AHU	$V_{r,AHU}$	$13\,\mathrm{m}^3$
Transfer coefficient radiator - building AHU	$k_{r,room,AHU}$	$638\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Radiator volume 6	$V_{r,6}$	$1.6\,\mathrm{m}^3$
Transfer coefficient building - environment 6	$k_{room,o,6}$	$913\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Transfer coefficient radiator - building 6	$k_{r,room,6}$	$1122\,\frac{\mathrm{kW}}{\mathrm{m}^2\mathrm{K}}$
Thermal capacity of the building 6	$C_{room,6}$	$584735473\,\frac{\mathrm{J}}{\mathrm{K}}$

Appendix B

Math

This appendix shows mathematical operations, which are used in this thesis.

B.1 Kronecker product

The Kronecker product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is defined as follows [21]

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \vdots \\ a_{I1}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{pmatrix} \in \mathbb{R}^{IK \times JL}.$$

The result is a matrix $\mathbf{C}$ with the new dimension $IK \times JL$. Similarly, the Kronecker product of two vectors $\mathbf{a} \in \mathbb{R}^{I}$ and $\mathbf{b} \in \mathbb{R}^{J}$ is given by

$$\mathbf{c} = \mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_1\mathbf{b} \\ a_2\mathbf{b} \\ \vdots \\ a_I\mathbf{b} \end{pmatrix} \in \mathbb{R}^{IJ}.$$

Appendix C

Economic model predictive control inequality constraints

This appendix shows the matrices of the EMPC inequality constraints.

C.1 Economic model predictive control inequality constraints for continuous control signals

The linearized state space model of the system

$$\Delta\mathbf{x}(k+1) = A\Delta\mathbf{x}(k) + B\Delta\mathbf{u}(k) + B_d\Delta\mathbf{u}_d(k)$$

with $\Delta\mathbf{x}(k+1) = \mathbf{x}(k+1) - \bar{\mathbf{x}}$, $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \bar{\mathbf{u}}$ and $\Delta\mathbf{u}_d(k) = \mathbf{u}_d(k) - \bar{\mathbf{u}}_d$, where $\bar{\mathbf{x}}$, $\bar{\mathbf{u}}$ and $\bar{\mathbf{u_d}}$ are the corresponding operating points and d denotes the disturbances.

The constraints (4.5) of the optimization problem (4.4) will be rewritten as inequality constraints in the general form $\mathbf{A}_{ineq}\mathbf{x}(k)_{opt} \leq \mathbf{b}_{ineq}$ for the optimization problem (4.1). An example is given next with a prediction horizon of $H_p = 3$ and the assumption that the states are equal to the outputs.

$$\mathbf{x}_{opt} = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \mathbf{u}(k+2) \\ s(k+1) \\ s(k+2) \\ s(k+3) \end{bmatrix}$$

$$
\mathbf{A}_{ineq} = \begin{bmatrix}
\mathbf{B} & 0 & 0 & -\mathbf{e} & 0 & 0 \\
\mathbf{AB} & \mathbf{B} & 0 & 0 & -\mathbf{e} & 0 \\
\mathbf{A}^2\mathbf{B} & \mathbf{AB} & \mathbf{B} & 0 & 0 & -\mathbf{e} \\
-\mathbf{B} & 0 & 0 & -\mathbf{e} & 0 & 0 \\
-\mathbf{AB} & -\mathbf{B} & 0 & 0 & -\mathbf{e} & 0 \\
-\mathbf{A}^2\mathbf{B} & -\mathbf{AB} & -\mathbf{B} & 0 & 0 & -\mathbf{e} \\
\mathbf{I} & 0 & 0 & 0 & 0 & 0 \\
-\mathbf{I} & 0 & 0 & 0 & 0 & 0 \\
-\mathbf{I} & \mathbf{I} & 0 & 0 & 0 & 0 \\
\mathbf{I} & -\mathbf{I} & 0 & 0 & 0 & 0 \\
0 & -\mathbf{I} & \mathbf{I} & 0 & 0 & 0 \\
0 & \mathbf{I} & -\mathbf{I} & 0 & 0 & 0 \\
\mathbf{I} & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{I} & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{I} & 0 & 0 & 0 \\
-\mathbf{I} & 0 & 0 & 0 & 0 & 0 \\
0 & -\mathbf{I} & 0 & 0 & 0 & 0 \\
0 & 0 & -\mathbf{I} & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}
$$

The vector $\mathbf{e}$ is a vector of the length n, (n = number of states) and all entries are one

$$
\mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.
$$

The matrix $\mathbf{I}$ is the identity matrix of the dimension $\mathbb{R}^{m \times m}$, (m = number of inputs)

$$
\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.
$$

$$
\mathbf{b}_{ineq} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \\ \mathbf{z}_5 \\ \mathbf{z}_6 \\ \Delta\mathbf{u}_{max} + \mathbf{u}(k-1) \\ -\Delta\mathbf{u}_{min} - \mathbf{u}(k-1) \\ \Delta\mathbf{u}_{max} \\ -\Delta\mathbf{u}_{min} \\ \Delta\mathbf{u}_{max} \\ -\Delta\mathbf{u}_{min} \\ \mathbf{u}_{max} \\ \mathbf{u}_{max} \\ \mathbf{u}_{max} \\ -\mathbf{u}_{min} \\ -\mathbf{u}_{min} \\ -\mathbf{u}_{min} \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$$
\begin{aligned}
\mathbf{z}_1 &= -\mathbf{A}\mathbf{x}(k)+\mathbf{A}\bar{\mathbf{x}}-\mathbf{B}_d\mathbf{u}_d(k)+\mathbf{B}_d\bar{\mathbf{u}}_d+\mathbf{B}\bar{\mathbf{u}}+\mathbf{x}_{max}(k+1) \\
\mathbf{z}_2 &= -\mathbf{A}^2\mathbf{x}(k)+\mathbf{A}^2\bar{\mathbf{x}}-\mathbf{A}\mathbf{B}_d\mathbf{u}_d(k)-\mathbf{B}_d\mathbf{u}_d(k+1)+\mathbf{A}\mathbf{B}_d\bar{\mathbf{u}}_d+\mathbf{B}_d\bar{\mathbf{u}}_d+\mathbf{A}\mathbf{B}\bar{\mathbf{u}}+\mathbf{B}\bar{\mathbf{u}}+\mathbf{x}_{max}(k+2) \\
\mathbf{z}_3 &= -\mathbf{A}^3\mathbf{x}(k)+\mathbf{A}^3\bar{\mathbf{x}}-\mathbf{A}^2\mathbf{B}_d\mathbf{u}_d(k)-\mathbf{A}\mathbf{B}_d\mathbf{u}_d(k+1)-\mathbf{B}_d\mathbf{u}_d(k+2)+\mathbf{A}^2\mathbf{B}_d\bar{\mathbf{u}}_d+\mathbf{A}\mathbf{B}_d\bar{\mathbf{u}}_d+\mathbf{B}_d\bar{\mathbf{u}}_d \\
&\quad +\mathbf{A}^2\mathbf{B}\bar{\mathbf{u}}+\mathbf{A}\mathbf{B}\bar{\mathbf{u}}+\mathbf{B}\bar{\mathbf{u}}+\mathbf{x}_{max}(k+3)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{z}_4 &= +\mathbf{A}\mathbf{x}(k)-\mathbf{A}\bar{\mathbf{x}}+\mathbf{B}_d\mathbf{u}_d(k)-\mathbf{B}_d\bar{\mathbf{u}}_d-\mathbf{B}\bar{\mathbf{u}}-\mathbf{x}_{min}(k+1) \\
\mathbf{z}_5 &= +\mathbf{A}^2\mathbf{x}(k)-\mathbf{A}^2\bar{\mathbf{x}}+\mathbf{A}\mathbf{B}_d\mathbf{u}_d(k)+\mathbf{B}_d\mathbf{u}_d(k+1)-\mathbf{A}\mathbf{B}_d\bar{\mathbf{u}}_d-\mathbf{B}_d\bar{\mathbf{u}}_d-\mathbf{A}\mathbf{B}\bar{\mathbf{u}}-\mathbf{B}\bar{\mathbf{u}}-\mathbf{x}_{min}(k+2) \\
\mathbf{z}_6 &= +\mathbf{A}^3\mathbf{x}(k)-\mathbf{A}^3\bar{\mathbf{x}}+\mathbf{A}^2\mathbf{B}_d\mathbf{u}_d(k)+\mathbf{A}\mathbf{B}_d\mathbf{u}_d(k+1)+\mathbf{B}_d\mathbf{u}_d(k+2)-\mathbf{A}^2\mathbf{B}_d\bar{\mathbf{u}}_d-\mathbf{A}\mathbf{B}_d\bar{\mathbf{u}}_d-\mathbf{B}_d\bar{\mathbf{u}}_d \\
&\quad -\mathbf{A}^2\mathbf{B}\bar{\mathbf{u}}-\mathbf{A}\mathbf{B}\bar{\mathbf{u}}-\mathbf{B}\bar{\mathbf{u}}-\mathbf{x}_{min}(k+3)
\end{aligned}
$$