Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Model Predictive Control for Buildings with One-pipe Hydronic Heating

Bc. Jan Předota

Supervisor: Ing. Jiří Dostál

Field of study: Cybernetics and Robotics
May 2018

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Předota Jan** Personal ID number: **420211**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

II. Master's thesis details

Master's thesis title in English:

Model Predictive Control for Buildings with One-pipe Hydronic Heating

Master's thesis title in Czech:

Prediktivní řízení budov s jednopotrubním otopným systémem

Guidelines:

1. Study topics in optimization and model predictive control for buildings.
2. Create mathematical model of one-pipe hydronic heating network. Develop a static optimizer for various optimization problems over the network.
3. Develop a model predictive control for a building with the one-pipe heating system (introduces nonlinear constraints).
4. Validate control strategy using high-fidelity building model.

Bibliography / sources:

[1] Borrelli, F., Bemporad, A. & Morari, M. (2017). Predictive control for linear and hybrid systems. Cambridge, UK New York, NY: Cambridge University Press

[2] Andersson, J. (2013). A general-purpose software framework for dynamic optimization (PhD thesis, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium).

[3] Hauser, J. (2017) Distributed Predictive Control of Buildings. (Diploma thesis, Czech Technical University in Prague, Prague)

[4] Baumelt, T. (2016), Distributed building identification. (Diploma thesis, Czech Technical University in Prague, Prague)

Name and workplace of master's thesis supervisor:

Ing. Jiří Dostál, Department of Control Engineering, FEL

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **13.02.2018** Deadline for master's thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

Ing. Jiří Dostál
Supervisor's signature

prof. Ing. Michael Šebek, DrSc.
Head of department's signature

prof. Ing. Pavel Ripka, CSc.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

26.4.2018
Date of assignment receipt

Student's signature

Acknowledgements

First and foremost I want to thank my supervisor Ing. Jiří Dostál. I really appreciate all his contribution of time, innovative ideas, and helpful comments.

I also must not forget to express my gratitude to my family, who supported me during my studies and in my life in general.

Declaration/Prohlášení

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 20. May 2018

Abstract

The thesis discusses the problematics of hydronic heating in buildings in which one-pipe hydronic heating systems are used. It describes thermal relations in buildings and relations of heat transfer via heat exchangers. It presents current types of hot water distribution networks and provides a tool for design and analysis of a one-pipe network for a given building.

The main motive of this thesis is model predictive control of such a network. The development of an appropriate controller is described gradually. It starts with a relatively simple problem of calculation of required heat flow in each room and ends with a more difficult problem that takes into consideration structure of the one-pipe network and parameters of individual heat exchangers. Problems that make the controller's underlying optimization problem more difficult or ill-defined are presented. Their consequences are that software tools have problems (efficiently) solving the optimization problem. The controller is developed for different control scenarios. The functionality of the introduced controller is verified on a building model of higher fidelity.

Keywords: Static optimization, steady state, dynamic model, nonlinear system, MPC, automatic differentiation, heat exchanger, one-pipe hydronic heating networks

Supervisor: Ing. Jiří Dostál

Abstrakt

Tato práce se zabývá problematikou teplovodního vytápění v budovách, ve kterých jsou využity jednopotrubní otopné systémy. Popisuje tepelné vztahy v budovách a vztahy týkající se přenosu tepla skrze teplotní výměníky. Představuje současné typy teplovodních sítí a přináší nástroj pro návrh a analýzu jednopotrubní sítě pro zadanou budovu.

Hlavním motivem práce je centralizované prediktivní řízení takové sítě. Postupně popisuje vytvoření odpovídajícího regulátoru. Začíná u poměrně jednoduchého problému výpočtu potřebného dododaného tepla v každém pokoji a končí u složitějšího problému, který počítá se strukturou jednopotrubní sítě a s parametry jednotlivých teplotních výměníku v ní umístěných. Jsou představeny problémy, které mají často za důsledek, že optimalizační problém, na jehož základě je regulátor postaven, je moc složitý nebo špatně definováný, a vybrané softwarové nástroje ho nejsou schopny (efektivně) řešit. Regulátor je navržen pro různé scénáře řízení. Navržený regulátor je otestován na přesnějším modelu budovy.

Klíčová slova: Statická optimalizace, ustálený stav, dynamický model, nelineární systém, MPC, automatická diferenciace, teplotní výměník, jednopotrubní otopná síť

Překlad názvu: Prediktivní řízení budov s jednopotrubním otopným systémem

Contents

Figures

Tables

Chapter 1

Introducton

In the modern world, some kind of heat distribution systems is typically present in every inhabited building built in the last century or two. While the technology of heat exchangers that ensures the heat exchange between a heat-transfer fluid and a room is continuously evolving, the way hot water is distributed has been quite stagnant. In the vast majority of buildings, we can find themselves similar two-pipe heat distribution systems.

This thesis originates from a research project developed at the Department of Control engineering in collaboration with UCEEB[1]. The project's aim is to make heating of buildings more efficient and do so predominantly by utilizing a one-pipe hot water distribution network. However, doing so has huge impact on control of a building heating.

The goal of this thesis is to research the problematics of mathematical optimization and the problematics of predictive control of buildings, to create a mathematical model of a one-pipe hydronic heating network and to develop a static optimizer for various optimization problems over the network. Then it includes development of a model predictive controller for a building with the one-pipe hot water distribution system. The one-pipe structure introduces nonlinear constraints and the thesis' objective is a suggestion of a way how to effectively solve such optimal control problem and a validation of the controller on a high-fidelity building model.

1.1 Thesis outline

In the beginning, an overview of current technological solutions of water distribution systems and some reasons why the two-pipe variant is so widespread are described and the obstacles that prevent the one-pipe network from being used much more are identified. A comparison of the two along with their variants are listed.

In chapter 3 heat exchange is described mathematically, NTU analysis is explained and some optimization problems regarding the behavior of a one-pipe heat distribution system are introduced. In the following chapter, the results of these static optimizations are discussed.

In chapter 5 starts the essential part of this thesis - the model predictive control. A short description of it is given and necessary mathematical models are created. Then a first controller is constructed. Its inputs are temperature demands in all rooms over some

[1] University Centre for Energy Efficient Buildings of Technical University in Prague

time horizon and its outputs are heat flows that need to be transferred into each room. The rest of the chapter builds on this controller and gradually makes the problem larger and closer to reality. Predominantly by switching demanded heat flows with demanded rooms' temperatures and by chaining heat exchangers in a row which introduces nonlinear constraints.

The nonlinear constraints cause the optimization methods used so far to be insufficient or incapable of solving such problems. Therefore CasADi, a symbolic framework for algorithmic differentiation, is presented. With it, a fully working controller based on nonlinear optimization can be designed. Implementation details, conducted experiments and verification of the controller on a high-fidelity building model are contained in chapter 6.

Chapter 2

Hot water distribution systems

Nowadays mainly two-pipe systems are used. They allow good zone temperature control, and their hydronic balancing[1] is quite easy in simple distribution networks. Larger buildings require more advanced balancing techniques, but it is still pretty standard at present. Water flows into all radiators at the same temperature, so the temperature gradient is the same over each radiator and is not affected by other radiators current power. For two-pipe networks there are several different ways of control. By far the most widespread is the throttle control where the hydraulic resistance of a secondary circuit is controlled by a valve. Changing the resistance effectively controls the water flow. The simplest way is a static valve that can be opened or closed manually. An automated solution is a thermostatic valve that mechanically controls the opening based on the room temperature.

A different approach to the network's control is not to use throttling but directly control flows through each radiator using a decentralized pump placed into each secondary circuit. For further needs, let's call systems using throttle control as **passive** and those controlled by pumps as **active**.

The ideas discussed in this section regarding the classification of hydronic heating networks and their comparison are taken from the yet not published article by Ing. Ondřej Zlevor, Bc. Jan Předota and Ing. Jiří Dostál. [1]

2.1 Passive one-pipe

A passive one-pipe hydronic heating network consists of a boiler, a pump, and a single pipe that brings hot water from a boiler to the heat exchangers connected in series and also returns cooled water back to the boiler. Water flow through a heat exchanger is controlled, in possible cases, by a valve. In the fig. 2.1 various types of arrangements used in one-pipe passive hydronic heating networks are shown. The most basic one (a) is the flow-through arrangement, which makes it impossible to control individual heat exchangers gradient and limits the maximum number of radiators in a row. The other arrangements with the bridging pipe (b) or with the particular mixing valve (c) are not that restrictive, however, there are still problems with hydronic balancing and difficult zone control. One-pipe systems were widely introduced in the German Democratic Republic in the 1970s and 1980s mainly due to low acquisition costs. However, these

[1] https://en.wikipedia.org/wiki/Hydronic_balancing

systems have many disadvantages:

- Complex design caused by the cooling of the heating water in the heat exchanger series.
- Difficult extendability.
- Difficult thermal and hydronic balancing.

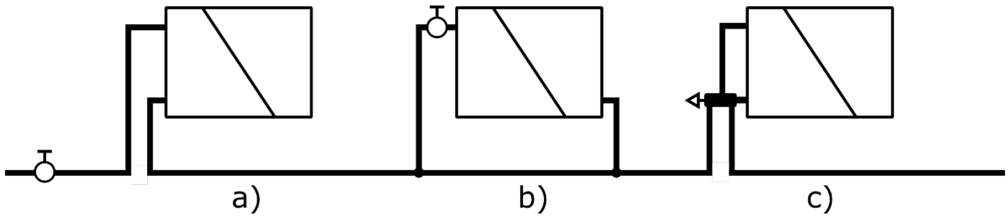


Figure 2.1: Various arrangements of a one-pipe system. [1]

On top of that, passive one-pipe networks in this form are energetically ineffective. For the last heat exchanger in the row to provide sufficient heat flow the inlet water flowing into it already cooled from preceding heat exchangers must be sufficiently hot. This can be achieved by increasing flow. But then, cooled water flowing into the boiler differs only by few degrees in temperature from the hot water. That decreases the efficiency of the boiler. By placing a valve, that opens and closes based on water temperature, in front of the boiler, the hydraulic resistance of the network and so the water flow can be changed. However, although that solves the problem of low boiler efficiency, the pump still rotates at maximum revolutions and its power is only taken by the added valve.

2.2 Passive two-pipe networks

Passive two-pipe networks are the most used hydronic heating systems. Compared to passive single-pipe systems, they have several advantages:

- The simplicity of design because of the identical temperature of the water coming into all radiators.
- Easy extendability.
- Allow equitherm control. For a given room it's possible to determine equitherm curves that describe dependence between the radiator's input water temperature and the outside temperature. Based on the desired room temperature the boiler heats water to the necessary level. It saves energy and gives nice non-oscillating thermal comfort.
- There are no thermal interactions between radiators.

However, there are still some disadvantages:

- The need for hydronic balancing.

- Loss of the pump's power on control actuators.
- There are pressure interactions between radiators.

For a valve to have good controllability it is necessary to be of high resistance, approximately half of the controlled heat exchanger.[2] By using control valves, the resistance of the system increases which increases demands on the pump. The figure 2.2 shows a scheme of a two-pipe system in direct return mode. It's disadvantages are the different lengths of the inlet and outlet pipes (from the pump to the radiators), which cause different differential pressures in the network and more difficult hydronic balancing. This problem is usually solved by reverse return (also called Tichelmann's, fig. 2.3) connection, where for all radiators the sum of inlet or outlet pipes is the same for all heat exchangers. That holds also for the different pressure drops, so the system is in balance. Also usage of hydraulic separators preventing hydraulic influences between boiler and radiators circuits is increasing.

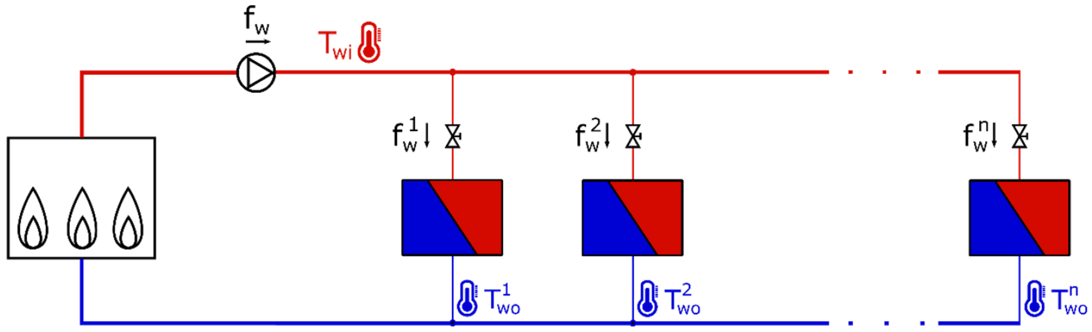


Figure 2.2: Scheme of a two-pipe passive system in the direct return mode. [1]

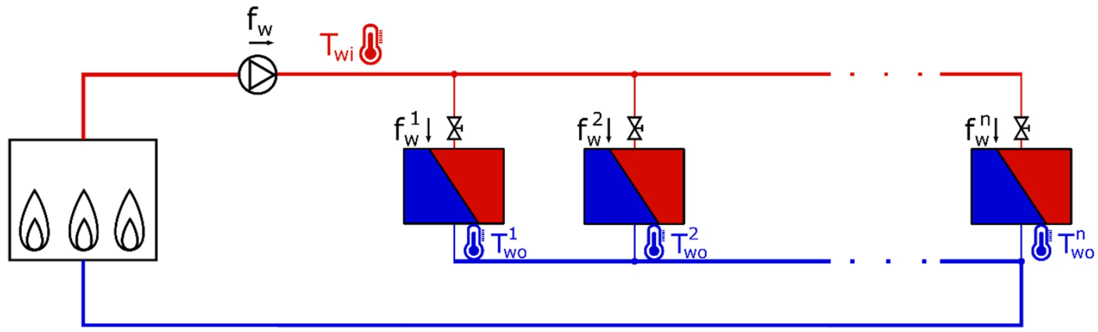


Figure 2.3: Scheme of a two-pipe system in Tichelmann's connection. [1]

2.3 Active two-pipe

The adjective *active* refers to a system in which each radiator has assigned a pump that can continuously control water flow. Such a system is shown in the fig. 2.4. In order to prevent counterflow for the situation when the pump is not running, it is necessary to put a check valve between the pump and the radiator. As opposed to the two-pipe passive systems, it has the advantages of:

- Not having any throttling elements means there is no loss of the pump's power.
- It's not necessary to perform hydronic balancing.
- Easier network design - one type of the pump can be used with a large variety of radiators of different sizes.

The disadvantages are:

- Higher investment costs.
- Pressure loss on one-way valves.

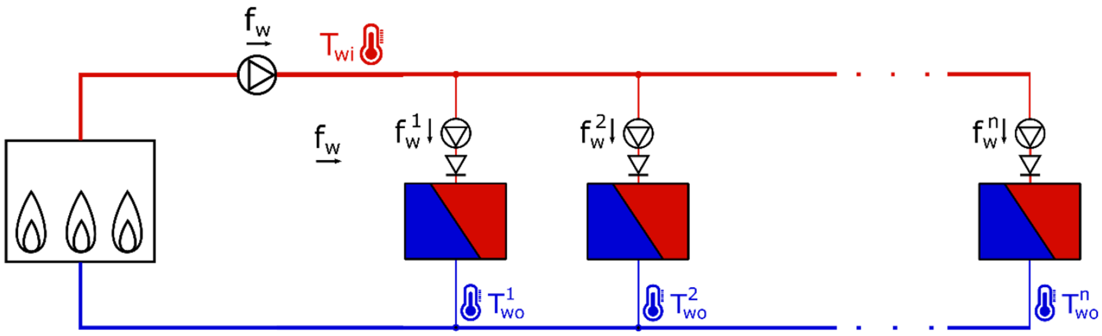


Figure 2.4: Scheme of an active two-pipe system. [1]

During the years 2001-2009 there was a couple of joint research projects between Wilo[2] and the Technical University in Dresden regarding hydronic heating in two-pipe networks controlled with decentralized pumps. The results show saving 20% of thermal energy and up to 70% of electrical energy when compared to a system with thermostatic heads.[3] However, these values indicate rather the advantages of zone control then specifically advantages of control with the use of pumps.

A piping design of an active two-pipe system doesn't differ from the design of its passive variant. Designed water flows are the same. Therefore the pipe dimensions are also the same. According to the desired temperature gradient, heat exchangers' power and network connection type it is only necessary to choose appropriate heat exchangers.

2.4 Active one-pipe

It is a system comprising a single-pipe primary circuit to which secondary circuits are connected by a double T-piece. An illustration is shown in the fig. 2.5. Outlet water from a heat exchanger is returned back to the primary circuit and cools the heating water coming into another heat exchanger. Water circulation through each secondary circuit is provided by the circuit-specific pump. Since the input and return pipes of the

[2] One of the world leaders in pumps manufacturing.

secondary circuits are close to each other, there is almost zero pressure drop between them. As a result, each secondary circuit is hydraulically separated from the primary circuit (change of water flow in the primary circuit doesn't affect flow in any secondary circuit in any way except it limits its maximum to primary flow current value). If a pump in a secondary circuit is off, there is now water flowing through a heat exchanger in the circuit. In the network, there are no pressure interactions, only thermal ones.

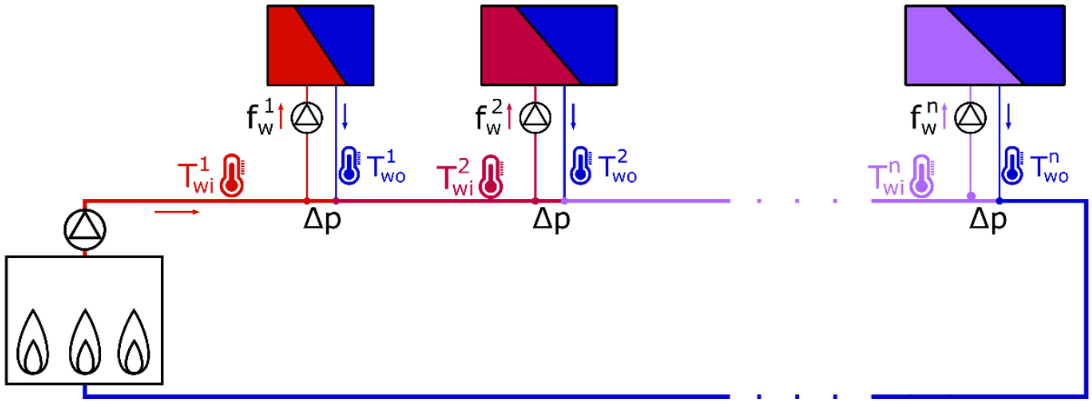


Figure 2.5: Scheme of an active one-pipe system. [1]

These active one pipe systems started being possible with the invention of wet electrical BLDC motors. They allow to continuously change the rotational speed in their range and thus to change the water flow, respectively power.

The advantages of a one-pipe active system are:

- The network contains pipes of only two different diameters (primary and secondary), so there is no need to design individual connectors concerning pressure drops.
- Secondary circuits are hydraulically separated from the primary circuit eliminating problems with hydronic balancing.
- Material and installation savings (less piping, fittings, valves and work).
- The secondary pump allows control of a wide range of heat exchangers.
- Compared to the active two-pipe there is no pressure loss on check valves, and the pump compensates losses only of the secondary circuit, therefore a weaker pump may be used.
- The total pipes' resistance that the pump must overcome is the smallest of all introduced network variants.
- The least amount of water in the system.

The disadvantages are:

- Generally, one-pipe systems have a bad reputation because of its passive variant.
- There is little experience with this technology because it was made possible with the introduction of the pumps with the so-called *wet rotor* and with technological progress in BLDC motors.

- Thermal interactions have to be considered during network design, and the system has to be computed iteratively.

Chapter 3

Static optimizations

Static optimizations serve as an introduction into control of a one pipe hydronic heating network. Based on demands on some of the network's parameters (e.g., transferred heat or sizes of heat exchangers) they allow calculation of values of other parameters (e.g., water temperature and water flows in the pipes) that are necessary to satisfy the demands. That is important so we can simulate what is a one-pipe hydronic heating network capable of and how one should design it and also how to design a controller for such network.

Before the steady state optimizations, the general physical principles of heat exchange are described. In the following section, they are applied to heat exchangers. The calculation of heat transfer via the exchangers using the NTU method is shown. After that, static optimizations are presented.

3.1 Physical principals of heat exchange

Thermal conduction is a heat transfer via direct contact based on the temperature difference between the two points and the conductivity of the material between them. The Fourier's law of heat conduction is generally described as:

$$\boldsymbol{q} = -k\nabla T \tag{3.1}$$

where $\boldsymbol{q}$ $[\mathrm{W \cdot m^{-2}}]$ is a local heat flux, k $[\mathrm{W \cdot m^{-1}K^{-1}}]$ is the material's conductivity, and ∇T $[\mathrm{K \cdot m^{-1}}]$ is the temperature gradient. The law can be rewritten for a homogeneous material into a 1-D variant:

$$\frac{Q}{\Delta t} = -kA\frac{\Delta T}{\Delta x} \tag{3.2}$$

where A $[\mathrm{m^2}]$ is the cross-sectional surface area that heat goes through, Δt [s] is the time change, $\Delta\ T[°\mathrm{C}]$ is the temperature difference between two ends and Δx [m] is the distance between the ends. If we write

$$U = \frac{k}{\Delta x}$$

where U $[W \cdot m^{-2} \cdot K^{-1}]$ is thermal conductance, the Fourier's law looks as:

$$\frac{Q}{\Delta t} = UA(-\Delta T) \tag{3.3}$$

The amount of heat Q [J] supplied to an object is computed as:

$$Q = C\Delta T \tag{3.4}$$

where C $[\mathrm{J} \cdot \mathrm{K}^{-1}]$ is the object's heat capacity and ΔT [K] is the temperature change.

We can expand that to:

$$Q = mc\Delta T \tag{3.5}$$

where m [kg] is the object weight and c $[\mathrm{J} \cdot \mathrm{kg}^{-1} \cdot \mathrm{K}^{-1}]$ object's specific heat capacity.

If we differentiate the equation above with respect to time and switch to the domain of liquids we obtain:

$$\dot{Q} = \dot{m}c\Delta T \tag{3.6}$$

where $\dot{Q}$ [W] is a heat flow rate, $\dot{m}$ $[\mathrm{kg} \cdot \mathrm{s}^{-1}]$ represents liquid's mass flow and c $[\mathrm{J} \cdot \mathrm{kg}^{-1} \cdot \mathrm{K}^{-1}]$ is liquid's specific heat capacity.

3.1.1 NTU Analysis

In the simplest heat exchangers, there are two streams of fluids and heat is exchanged from the hotter fluid to the colder one. There are two methods used to calculate the rate of heat transfer. The first is Log-Mean Temperature Difference method that is used when both the inlet and outlet temperatures are known.

The NTU (abbr. of Number of Transfer Units) method is usually used when the outlet temperatures of the fluids are not known although it can be used in almost any situations.

The equation 3.6 can be interpreted as how much heat is being transferred given the liquid's mass flow and temperature differences. So if there is a room that is heated by hot water (via heat exchanger) and where a heat loss is happening because of colder outside temperature, the equation can be used for the computation of water flow and water temperature necessary to compensate the loss.

However, the heat flow rate $\dot{Q}$ is the *theoretical maximum* that can be exchanged. The heat exchange occurs between water flowing in the pipes and air in the room via a heat exchanger. The exchange can be divided into two parts: heat is exchanged from water to the heat exchanger and then from heat exchanger to the air. Because the rate of heat exchange isn't the same between water and metal and between metal and air, the equation 3.6 is extended to:

$$\dot{Q} = \epsilon\dot{m}c\Delta T \tag{3.7}$$

where ϵ $[-]$, $\epsilon \in (0; 1)$ is the heat exchanger effectiveness.

Generally, the effectiveness is defined as:

$$\epsilon = \frac{\dot{Q}}{\dot{Q}_{max}} \tag{3.8}$$

where $\dot{Q}$ [W] is the real heat transfer rate and $\dot{Q}_{max}$ [W] is the maximum heat transfer rate possible.

The maximum temperature difference possible between the two fluids is:

$$\Delta T_{max} = T_{h,i} - T_{c,i} \tag{3.9}$$

where $T_{h,i}$ [°C], $T_{c,i}$ [°C] are the inlet temperature of the hotter fluid and of the colder fluid respectively. Let's also introduce two more temperatures $T_{h,o}$ and $T_{c,o}$ which are the outlet temperatures of both streams. If the fluids were stationary and didn't flow, final temperatures would be the same, but because of the nature of heat exchangers and the ongoing flows, we must distinguish between these two.

The theoretical maximum heat flow rate $\dot{Q}_{max}$ can be expressed as:

$$\dot{Q}_{max} = \dot{C}(T_{h,i} - T_{c,i}) \tag{3.10}$$

where

$$\dot{C} = \min(\dot{C}_h, \dot{C}_c)$$

The term $\dot{C}$ is referred to as the *heat capacity rate*. The mass capacity rate is defined as:

$$\dot{C} = c\dot{m} \;\; [\mathrm{W \cdot K^{-1}}]$$

while the volumetric capacity rate can be computed as:

$$\dot{C} = c_v\dot{V} \;\; [\mathrm{W \cdot K^{-1}}] \tag{3.11}$$

where $\dot{V}$ $[\mathrm{m^3\,s^{-1}}]$ is the volumetric flow rate and c_v $[\mathrm{J \cdot m^{-3} \cdot K^{-1}}]$ is specific heat capacity at constant volume.

If $\dot{C}_h$, $\dot{C}_h$ is defined as the capacity rate of the hot fluid, respectively cold fluid, the equation 3.8 can be expressed as:

$$\epsilon = \frac{\dot{C}_h(T_{h,i} - T_{h,o})}{\dot{C}_{min}(T_{h,i} - T_{c,i})} = \frac{\dot{C}_c(T_{c,i} - T_{c,o})}{\dot{C}_{min}(T_{h,i} - T_{c,i})} \tag{3.12}$$

but since the outlet temperatures are not known, the effectiveness must be computed in another way. In general, it is a function of the number of transfer units NTU and heat capacity ratio C_r:

$$\epsilon = f(NTU, \dot{C}_r) \tag{3.13}$$

where

$$\dot{C}_r = \frac{\dot{C}_{min}}{\dot{C}_{max}}, \;\; \dot{C}_{max} = \max(\dot{C}_h, \dot{C}_c)$$

and

$$NTU = \frac{UA}{C_{min}} \tag{3.14}$$

where A $[\mathrm{m^2}]$ is the heat transfer area, i.e., the area of a heat exchanger, and U $[W \cdot m^{-2} \cdot K^{-1}]$ is the heat transfer coefficient (also known as thermal transmittance or U-value). It has the same units as the thermal conductance used in 3.3, but that is primarily used for heat transfers between fluids while thermal transmittance is used to simplify an equation that has several different forms of thermal resistances. It is the reciprocal of the R-value known as thermal resistance.

The function $f(NTU, \dot{C}_r)$ differs for different flow arrangements in heat exchangers and can be looked up at appropriate tables. For example for a parallel-flow concentric tubes heat exchanger (fig. 3.1) the formula is[5]:

$$\epsilon = \frac{\left[1 - \exp\left(-NTU(1 + \dot{C}_r)\right)\right]}{1 + \dot{C}_r} \tag{3.15}$$

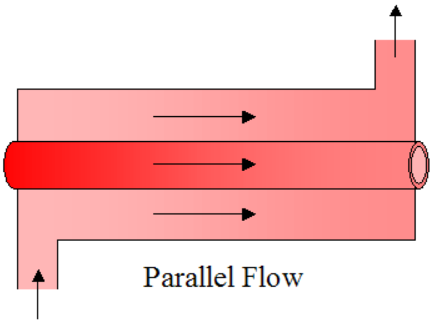


Figure 3.1: Scheme of fluids flows in a parallel heat exchanger. [4]

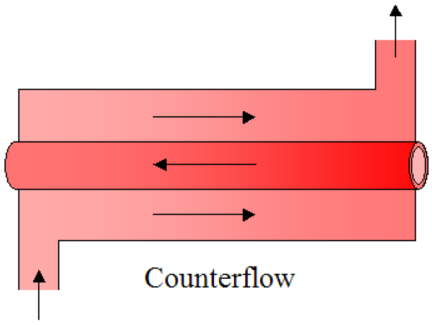


Figure 3.2: Scheme of fluids flows in a counterflow heat exchanger.[4]

or for a counter-current flow concentric tubes heat exchanger (fig. 3.2)[6]:

$$\epsilon = \frac{1 - \exp\left[-NTU\left(1 - \dot{C}_r\right)\right]}{1 - \dot{C}_r \exp\left[-NTU\left(1 - \dot{C}_r\right)\right]} \tag{3.16}$$

In this thesis a crossflow liquid-to-air heat exchanger (pic. 3.3) is used. Its efficiency formula is[7]:

$$\epsilon = 1 - \exp(-NTU). \tag{3.17}$$

which is the same as 3.15 or 3.16 for $\dot{C}_r = 0$.

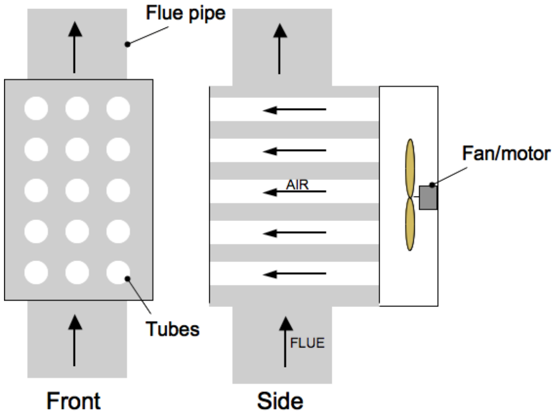


Figure 3.3: Scheme of a crossflow heat exchanger.[8]

The heat transfer coefficient U of a heat exchanger (used in the equation 3.14) depends on both the air volumetric flow $\dot{V}$ $[\mathrm{m^3 s^{-1}}]$ and water mass flow $\dot{m}$ $[\mathrm{kg \cdot s^{-1}}]$. The heat exchange is described by two steps. Firstly heat is transferred from water to the heat exchanger and then from the heat exchanger to the ambient air. Therefore, let's introduce thermal transmittance (U-value) for both these cases:

1. Water to body heat exchange. The U-value is approximated by the equation:

$$U_{wb}(\dot{m}_s) \doteq a \cdot \dot{m}_s^b \tag{3.18}$$

where $\dot{m}_s$ $[\mathrm{kg\,s^{-1}}]$ is the hot water flowing through the exchanger and a, b are heat exchanger dependent mapping coefficients.

2. Body to air heat exchange. The U-value is approximated by the equation:

$$U_{ba}(\dot{V}_a) \doteq r\dot{V}_a^2 - s\dot{V}_a + t \tag{3.19}$$

where $\dot{V}_a$ $[\mathrm{m^3\,h^{-1}}]$ is the cold air flowing across the exchanger and r, s, t are heat exchanger dependent mapping coefficients.

The coefficients in the two mapping equations were obtained by a series of measurements that were conducted by Ing. Jiri Dostal.

Because the meaning of U is thermal conductance their connection in a series is computed as:

$$\frac{1}{U} = \frac{1}{U_{wb}} + \frac{1}{U_{ba}} \quad \Rightarrow U(\dot{m}_s, \dot{V}_a) = \frac{U_{wb}(\dot{m}_s)U_{ba}(\dot{V}_a)}{U_{wb}(\dot{m}_s) + U_{ba}(\dot{V}_a)}. \tag{3.20}$$

3.1.2 Heat transfers in one zone

Let's show all heat transfers that are happening inside a zone (a room) with a heat exchanger (fig. 3.4). The heat rate being transferred from the water to the zone, in terms of using all the previously mentioned relations including the NTU analysis, is:

$$\begin{aligned}
&\dot{Q}_{HX}(\dot{m}, T_w, T_z, \dot{V}_{air}, A) = \epsilon \cdot \dot{C}_{min} \cdot (T_w - T_z) = f(NTU, \dot{C}_r) \cdot \dot{C}_{min} \cdot (T_w - T_z) = \\
&\left(1 - e^{-\frac{UA}{\dot{C}_{min}}}\right) \cdot \dot{C}_{min} \cdot (T_w - T_z) = \\
&\left[1 - \exp\left(-\frac{U(\dot{m}, \dot{V}_{air}) \cdot A}{\min(\dot{m} \cdot c_w, \dot{V}_{air} \cdot c_{air})}\right)\right] \cdot \min(\dot{m} \cdot c_w, \dot{V}_{air} \cdot c_{air}) \cdot (T_w - T_z)
\end{aligned} \tag{3.21}$$

3.2 Steady state optimization

Variables affecting transferred heat rate $\dot{Q}_{HX}$ can be altered in different combinations and still provide a same result. As an example let's put a zone that needs a constant supply of heat to keep itself at a specific temperature or to raise its temperature. Considering fans of the HX to be rotating at the same speed, therefore $\dot{V}_{air}$ being constant, A being constant by nature and T_z being given, there is still the mathematically infinite amount of combinations of inlet water temperature T_w and water flow $\dot{m}$ to deliver desired heat. To compute the combination that is most efficient (in a specified way) methods from the field of mathematical optimization are used.

3.2.1 Mathematical optimization

Mathematical optimization "is the selection of the best element (concerning some criterion) from some set of available alternatives"[9]. Usually, the goal is to find such combination

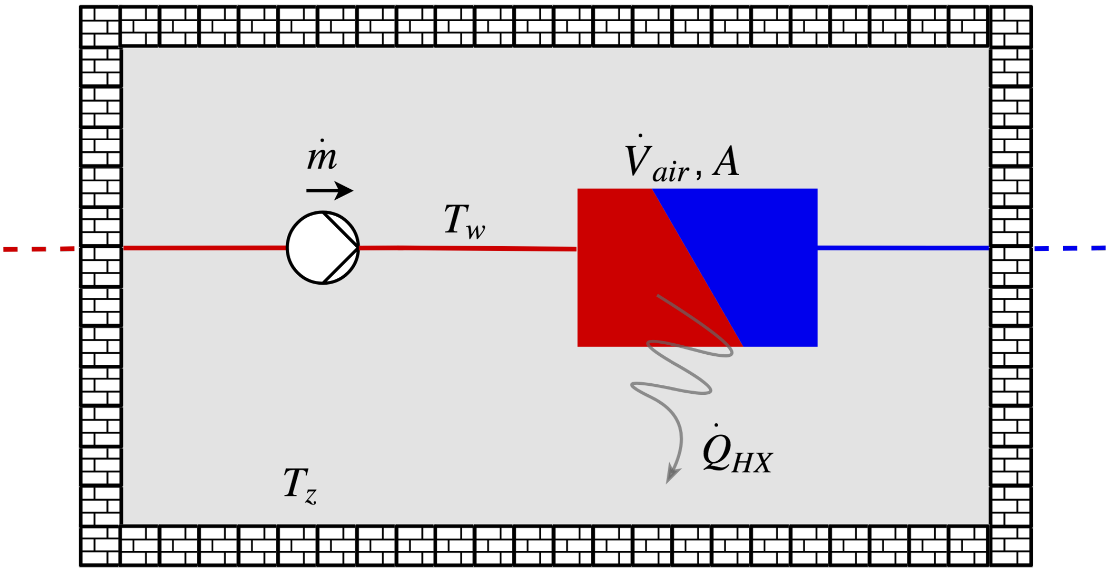


Figure 3.4: Scheme of a zone with a heat exchanger. The left pipe carries hot water with a temperature T_w and flow $\dot{m}$. The zone has temperature T_z. The heat exchanger has a heat transfer surface A and the air flows through it at the rate V_{air}. $\dot{Q}_{HX}$ is the total heat rate being transferred from water to the room.

of inputs to the objective function that the objective function's value is minimized while satisfying all present constraints. Mathematically expressed:

$$\begin{aligned} & \underset{\boldsymbol{x}}{\text{minimize}} && f_0(\boldsymbol{x}), \;\; \boldsymbol{x} \in \mathbb{R}^n \\ & \text{subject to} && g(\boldsymbol{x}) \leq b_i, \;\; i = 1, \ldots, m. \end{aligned} \tag{3.22}$$

There are many types of optimization problems that are characterized by different objective and constraint functions (linear, quadratic, combinatorial, non-linear, etc.) and each one is typically solved with a different method and having different computational complexity. Some methods use only function values, and some also use gradients or Hessians, which improves the rate of convergence. Mathematical optimization is a very complex topic, and its problematics isn't the main focus of this thesis. There are many great solvers for various problems already implemented, and the task is to choose the suitable one and properly formulate our optimization problem.

3.2.2 Chaining heat exchangers

The one-pipe hot water distribution systems consist of a single distribution pipe. It carries hot water to all radiators which are connected in series, one after another. The hottest water from a boiler is always at the input of the first heat exchanger connected to the main pipe (in the direction from the boiler). Water at the input of another heat exchanger is always colder because it already passed some of its heat in the previous heat exchanger. For this reason, either size of the heat exchanger or water flow through it should be increased to provide the same heat output. There is a bridging pipe around each heat exchanger to make different water flows possible. A scheme of such a system is shown in the figure 3.5. Kirchhoff's circuit laws analogy can be used for theoretical analysis of the system. In any junction sum of all the input flows must equal the output flows (water flows or heat flow rates). Therefore:

$$\dot{Q}_p^{n+1} = \dot{Q}_b^n + \dot{Q}_{s,o}^n \tag{3.23}$$
$$\dot{Q}_{s,o}^n = \dot{Q}_{s,i}^n - \dot{Q}_{HX}^n \tag{3.24}$$
$$\dot{Q}_p^{n+1} = \dot{Q}_p^n - \dot{Q}_{HX}^n \tag{3.25}$$

where: upper indeces of the variables are connected to specific heat exchanger; bottom index p means *primary*, representing variables connected to the primary pipe, s means *secondary*, representing variables connected to the secondary circuits, b is connected to the bridging part of the main pipe and i, o mean input, respectively output, with respect to the heat exchanger. See fig. 3.5.

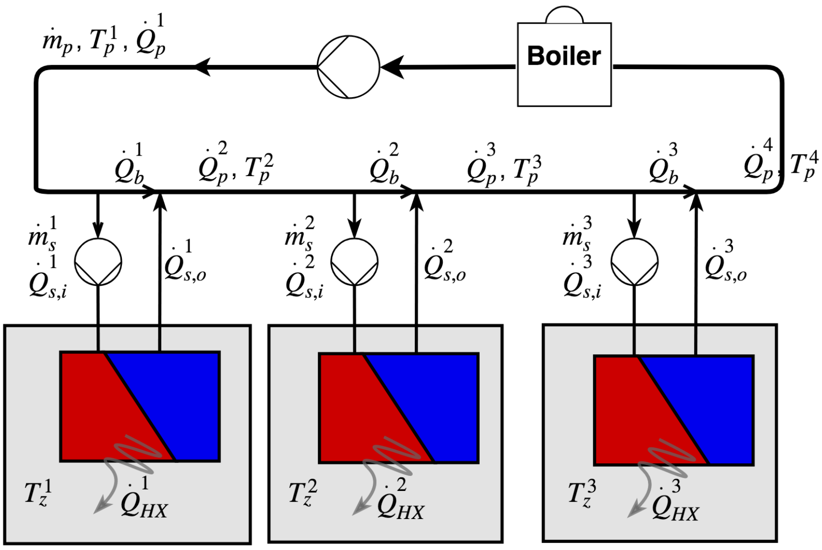


Figure 3.5: Scheme of a one pipe-network with three secondary circuits with heat exchangers. Each heat HX is in a different room with different temperature. There is a boiler heating water up and a main pump applying pressure in the main pipe. Every secondary circuit has also its own independent pump.

3.2.3 Water temperature in the primary pipe

In the following subsections, there are described three optimization problems. In all of them, temperature of water in the primary pipe (between any two heat exchangers) must be known. Whether because it is directly a part of an objective function or because it is needed for the computation of heat supplied by the following heat exchanger. The temperature can be expressed as:

$$T_p^{n+1} = T_p^n - \frac{\dot{Q}_{HX}^n\left(\dot{m}_s^n, T_p^n, T_z^n, \dot{V}_{air}^n, A^n\right)}{\dot{m}_p \cdot c_w} \tag{3.26}$$

From this equation, it's clear that for computation of the water temperature somewhere farther in the primary pipe, the heat transfer rate of all previous heat exchangers also has to be calculated, and because $\dot{Q}_{HX}$ contains an exponential term (see 3.21), that leads to a sum of exponential functions. Moreover, each exponential term depends on all the exponentials terms corresponding to the preceding HXs. As a result, calculation of water temperature after n-th heat exchanger means n iterations of the equation 3.26.

3.2.4 Constraints

During the optimization problems discussed later, several constraints related mainly to the physical restrictions of the system will be applied.

Maximal flow through the primary pipe

Maximal flow through the circular primary pipe is determined according to the used pipe's specifications. One should look for the inner diameter d[m] of a tube. That is specified by the ISO 7 norm[10] for all commonly used pipes and screw threads. Each pipe also has a recommended maximum flow rate for different fluids that should not be exceeded to minimize pipe's erosion wear and noise. Maximum flow is defined as:

$$\dot{m}_p = \pi \frac{d^2}{2} \rho v_{max}^2 \tag{3.27}$$

where ρ $[\mathrm{kg \cdot m^{-3}}]$ is the liquid's density and v_{max} $[\mathrm{m \cdot s^{-1}}]$ is the recommended maximum water flow rate. A good rule of thumb is to use $v_{max} = 1\ \mathrm{m \cdot s^{-1}}$.

Maximum flow through a secondary pipe

As a fluid flows through a tube, a pressure drop occurs due to friction (resistance of the pipes). Of course, that also happens in the main pipe. However, there is a more powerful pump that gives the maximum recommended water flow without problems and that won't be exceeded. For a pipe of diameter D [m] the pressure loss Δp_d $[\mathrm{Pa \cdot m^{-1}}]$ is characterized by Darcy-Weisbach equation [11]:

$$\frac{\Delta p_d}{L} = f_d \cdot \frac{\rho}{2} \cdot \frac{v^2}{d} \tag{3.28}$$

where L [m] is the pipe's length, ρ $[\mathrm{kg \cdot m^{-3}}]$ water density, v $[\mathrm{m \cdot s^{-1}}]$ means flow velocity and f_d $[-]$ the Darcy friction factor. The Darcy friction factor differs for different flow regimes. We will consider the two most used, for laminar flow:

$$f_d = \frac{64}{R_e} \tag{3.29}$$

and for turbulent flow (Colebrook equation) [12]:

$$\frac{1}{\sqrt{f_d}} = -2 \log \left(\frac{\epsilon / D}{3.7} + \frac{2.51}{Re \sqrt{f_d}} \right) \tag{3.30}$$

where R_e is Reynolds number, and $\epsilon[-]$ is pipe relative roughness.

Because of the difficulty of the last equation and the fact that the real system is usually in the turbulent flow mode (higher flows), an approximation to these equations by a second-order monomial (fig. 3.6) was used:

$$\frac{\Delta p_d}{L} = k_l \dot{m}^2 \tag{3.31}$$

with the coefficient k_l $[\mathrm{kg}^{-1} \cdot \mathrm{m}^{-2}]$.

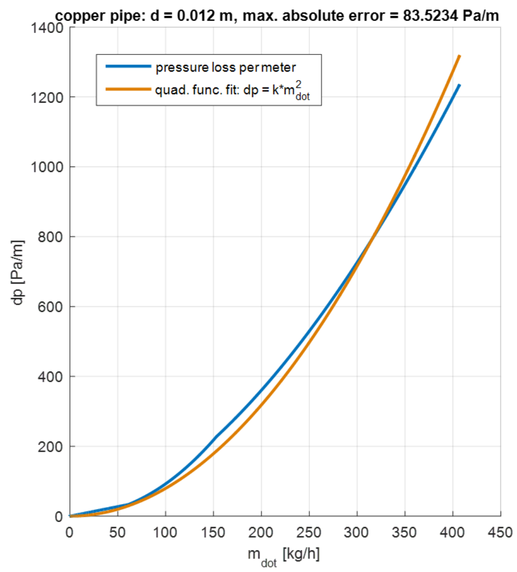


Figure 3.6: Experimental approximation of secondary pipe pressure drop per meter depending on water flow. [13]

Another resistance in the secondary circuit that must be taken into account is the resistance of a heat exchanger. That is also water flow dependent and is also approximated by a second-order monomial. Therefore the final approximation of the resistance in a secondary circuit becomes:

$$\frac{\Delta p_d}{L} = k_l \dot{m}^2 + k_{HX} \dot{m}^2 \tag{3.32}$$

In order to achieve desired water flow the pump must match the pressure loss of the circuit. For constant RPMs, the dependence of the pump pressure in the fluid on the resistance of the circuit is approximated by the following equation:

$$\Delta p_p = p_0 - b\dot{m} - a\dot{m}^2 \tag{3.33}$$

where p_0 is the theoretical maximum pressure the pump is able to apply when there is infinite resistance in the circuit (no water flow). With decreasing resistance the pressure also decreases and water flow $\dot{m}_s$ increases. With higher flows, the effect of flow friction will not be neglectable. Flow friction occurs in all the hydraulic components which the fluid flows and grows quadratically with the flow velocity and is represented by the $-a\dot{m}^2$ term. General pump losses are illustrated in picture 3.7.

To use the equation 3.33 for variable flows (shaft RPMs respectively), it has to be extended. That is done by using the affinity laws [14], which allow prediction of the pressure change characteristic of a pump from known characteristic measured at different speed. According to them, the pressure is proportional to the square of the shaft speed, mathematically $\frac{p_1}{p_2} = \left(\frac{s_1}{s_2}\right)^2$. Therefore the equation 3.33 can be modified to [15]:

$$\Delta p_p = a\dot{m}_s^2 + b\dot{m}_s s + p_0 s^2 \tag{3.34}$$

where s [rpm] is the pump speed. The sign of the two terms was changed so it can be manipulated better later on.

The bridging pipe across every heat exchanger is short and the pressure loss on that pipe is neglectable. Therefore, flow in the primary pipe doesn't influence flow in any secondary circuit. Because of this hydraulic separation, the algebraic sum of pressures in any circuit must be zero. This is illustrated in fig. 3.8. If the pump speed s is set to the maximal possible and the pump pressure equation 3.34 is put equal with pressure drop p_d from the equation 3.32, the maximal water flow in a secondary circuit can be expressed:

$$\begin{aligned}
\Delta p_p &= \Delta p_d \\
a\dot{m}_s^2 + b\dot{m}_s s + p_0 s^2 &= L\dot{m}_s^2(k_l + k_{HX}) \\
\dot{m}_{s_max} &= \pm\frac{\sqrt{s^2(-4ap_0 + b^2 + 4Lp_0\,(k_l + k_{HX}))} \mp bs}{2a - 2\,(k_l + k_{HX})}
\end{aligned} \tag{3.35}$$

Only the physically possible positive solution will be used.

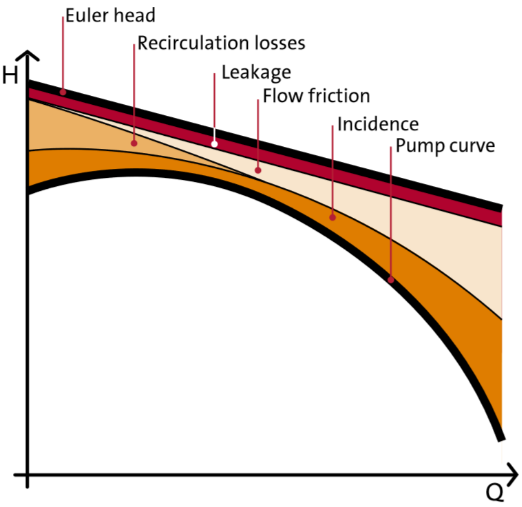


Figure 3.7: Illustration of various possible pressure losses in centrifugal pumps [16]. H [m] is the Euler head which says how high the pump is able to pump water and is converted into pressure with the relation $p = g\rho H$ where g [$\mathrm{m \cdot s^{-2}}$] is gravitational constant and ρ [$\mathrm{kg \cdot m^{-3}}$] water density. $\dot{Q}$ [$\mathrm{m^3 \cdot s^{-1}}$] is water flow. In this thesis only the Euler head loss and loss due to flow friction are considered because other losses are neglectable in the pump used.

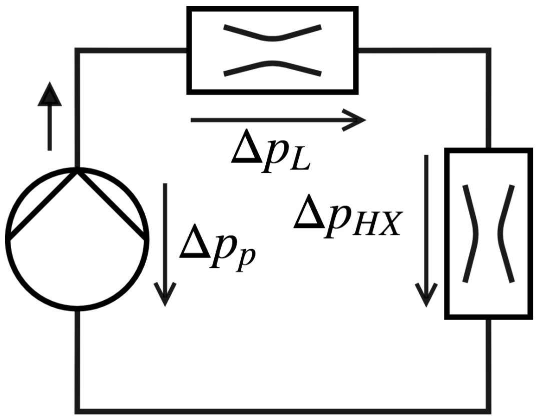


Figure 3.8: An illustration of pressures present in a secondary circuit. Pressure Δp_p applied by the pump must be equal to the sum of the pressure loss on the pipes Δp_L and the pressure loss on the heat exchanger Δp_{HX}.

No reverse flow

In this thesis the possibility of reverse water flow through the secondary circuits is not considered. Therefore, it must be specified in the optimization problem as a constraint. Water flow through a secondary circuit must not be less then zero and also not larger than flow in the primary pipe:

$$0 \leq \dot{m}_s^i \leq \dot{m}_p, \quad i = 1, \ldots, n$$

where n is the number of secondary circuits.

3.2.5 Optimization case: Heat source design

The first optimization task is to find the minimal possible temperature of the water coming out of the boiler while demanding specific fixed temperature at the end of the net (at boiler input). Individual heat exchangers have to satisfy the demand of transferred heat into the respective zones. Flows in pipes are constrained as described in the subsection 3.2.4. Mathematically:

$$\begin{aligned} \underset{\dot{m}_p,\ \dot{m}_s^i,\ T_p^1}{\text{minimize}} \quad & T_p^1 \\ \text{subject to} \quad & 0 \leq \dot{m}_p \leq m_{p|max}, \\ & 0 \leq \dot{m}_s^j \leq m_{s|max}, \quad j = 1, \ldots, n. \\ & \dot{m}_s^j \leq \dot{m}_p \\ & \dot{Q}_{HX}^j = \dot{Q}_{demanded}^j \\ & T_p^n = T_{p|demanded}^n \end{aligned} \tag{3.36}$$

This optimization problem is crucial because it allows the design of the boiler and helps to decide to choose an appropriate one. However, one of its results is that the water flow through the primary pipe is always at its maximum $\dot{m}_{p\_max}$ because the bigger the difference versus water flows though secondary circuits the smaller the temperature drops across heat exchangers.

Having the main pump operating around its maximum and the boiler heating water only by couple of degrees would be inefficient, therefore it is advised to constrain the primary flow $\dot{m}_p$ more, e.g. to its fourth of what was described in 3.2.4. Doing so will increase the sought temperature of the boiler outlet water and the designed water flow will be smaller.

3.2.6 Optimization case: Heat exchanger design

Another task worth examining is adding the possibility of having different HXs. That can be done by adding the variable A, which is the surface of an HX, to the optimization variables. The goal now is to find a compromise between minimizing the temperature at the end of the chain and the sum of the heat exchangers' surfaces while having input temperature fixed and satisfying the heat demands at each HX and other constraints already described above. Also the coefficient A must stay within some predefined values, e.g., $\langle \frac{1}{10}, 10 \rangle$. Mathematically expressed:

$$\begin{aligned}
\underset{\dot{m}_p,\ \dot{m}_s^i,\ A^i}{\text{minimize}} \quad & T_p^n + w\sum_i^n A^i \quad i = 1,\dots,n \\
\text{subject to} \quad & 0 \le \dot{m}_p \le m_{p|max}, \\
& 0 \le \dot{m}_s^j \le m_{s|max}, \quad j = 1,\dots,n. \\
& \dot{m}_s^j \le \dot{m}_p \\
& \dot{Q}_{HX}^j = \dot{Q}_{demanded}^j \\
& T_p^1 = T_{p|demanded}^1 \\
& A_{min}^j < A^j < A_{max}^j
\end{aligned} \tag{3.37}$$

where w is a weight that controls how much the surfaces' sum is penalized with respect to the output temperature.

This optimization case can find its use allowing one to design a one-pipe network in a building assuming at least rough estimates of typical heat flow rates demanded during operation are known.

3.2.7 Optimization case: Network operation

In the last optimization task, the goal is to minimize the temperature at the end of the net (between the last HX and the boiler) while having the temperature of the water coming out of the boiler fixed. How to calculate that temperature was shown in the eq. 3.26. Individual fan coil units have to satisfy the demand of transferred heat into the respective zones. That is implemented as an equality constraint. Flows in pipes are constrained similarly as in the previous subsections. Mathematically expressed:

$$\begin{aligned}
\underset{\dot{m}_p,\ \dot{m}_s^i}{\text{minimize}} \quad & T_p^n \\
\text{subject to} \quad & 0 \le \dot{m}_p \le m_{p|max}, \\
& 0 \le \dot{m}_s^j \le m_{s|max}, \quad j = 1,\dots,n. \\
& \dot{m}_s^j \le \dot{m}_p \\
& \dot{Q}_{HX}^j = \dot{Q}_{demanded}^j \\
& T_p^1 = T_{p|demanded}^1
\end{aligned} \tag{3.38}$$

This optimization task is useful for analysis of how given network and boiler will perform. The efficiency of a boiler (e.g., a condensing one) is the higher, the bigger the temperature difference. This objective function leads to the behavior that when there are reasonable heat rate demands the flow through the last secondary circuit is equal to the flow of the primary pipe and flows through all preceding HXs are smaller.

Chapter 4

Static optimization study for an example network

For all problems the optimization was performed over water flows $\dot{m}_p, \dot{m}_s^i$ and in 4.3 also over surfaces of the heat exchangers A^i. Air flow of fans $\dot{V}_{air}^i$ and zone temperatures T_z^i were kept constant. Two outcomes of each problem are discussed in the following paragraphs, however, there are certainly many more possible scenarios.

4.1 Implementation

All three optimization tasks mentioned in 3.2 were implemented and performed in Matlab®. For solving optimization problems `fmincon` function from Optimization toolbox was used. The problem, if feasible, was solved within few seconds for up to 30 secondary circuits. For simpler system behavior representation we set the number of zones with a heat exchanger to four.

All solving algorithms currently implemented in fmincon were tried and the *interior-point* method was found to perform the best by far. Description of this method can be found in [17]. Very shortly: Inequality constraints are added into objective function (usually via logarithmic barrier function) and have near-zero value when far from a respective boundary and analogically very high value when close to a boundary. Then the search towards the minimum of the objective function at each iteration tries to use *direct step* also called *Newton's*, which satisfies the KKT conditions[1]. If the approximate problem is not locally convex near the current iterate, it takes the step based on the *conjugate gradient*, using trust region [18].

After substitution for the values specific to this example, the two approximation equations regarding thermal transmittance in a heat exchanger 3.18 and 3.19 were:

$$U_{wb}(\dot{m}_s) \doteq 43.13 \cdot \dot{m}_s^{0.29} \tag{4.1}$$

$$U_{ba}(\dot{V}_a) \doteq \frac{21896}{3600^2}\dot{V}_a^2 - \frac{3132}{3600}\dot{V}_a + 261. \tag{4.2}$$

The coefficients present in the formula for the computation of the maximum secondary flow (subsubsec. 3.2.4):

$$\dot{m}_{s\_max} = \pm\frac{\sqrt{s^2(-4ap + b^2 + 4kLp)} \mp bs}{2a - 2kL}$$

[1] https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions

had the following values:

$$a = -0.048279, \quad b = -0.014058, \quad p_0 = 1.195428, \quad k_l = 0.8, \quad k_{HX} = 0.2, \quad L = 1$$

The maximum RPM of the used pump was $s = 160$.

4.1.1 Optimization variable scaling

Optimizing over variables with different orders of magnitude can cause numerical difficulties. Contributions of small magnitude variables to the objective function can get swamped by the large magnitude ones, leading to loss of information, especially in gradient estimation when finding the optimal step size. The best way to prevent this is to scale optimization variables to a similar order of magnitude before passing them to an optimizer and to scale them back when solved. For example if boiler outlet temperature T_p^1 were somewhere around 50 °C while a reasonable value of flows in secondary pipe is about 0.02 kg/s that would be a 3 orders of magnitude difference and should be scaled.

4.2 Heat source design

The optimizer's goal was to find the lowest possible water temperature at the boiler outlet while demanding temperature of 40 °C in the primary pipe after the last heat exchanger. In the first case, each heat exchanger should maintain the constant supply of 1200 W. Temperature of each zone was set to 20 °C. Water flow in the primary pipe was constrained by its recommended maximum and the optimization outcome hit this value. Temperature drops along primary pipe were small because the flow carrying hot water was by an order of magnitude larger than flows with cooled water in secondary pipes. The result is shown in fig. 4.1. The minimal temperature was found to be 42.23 °C.

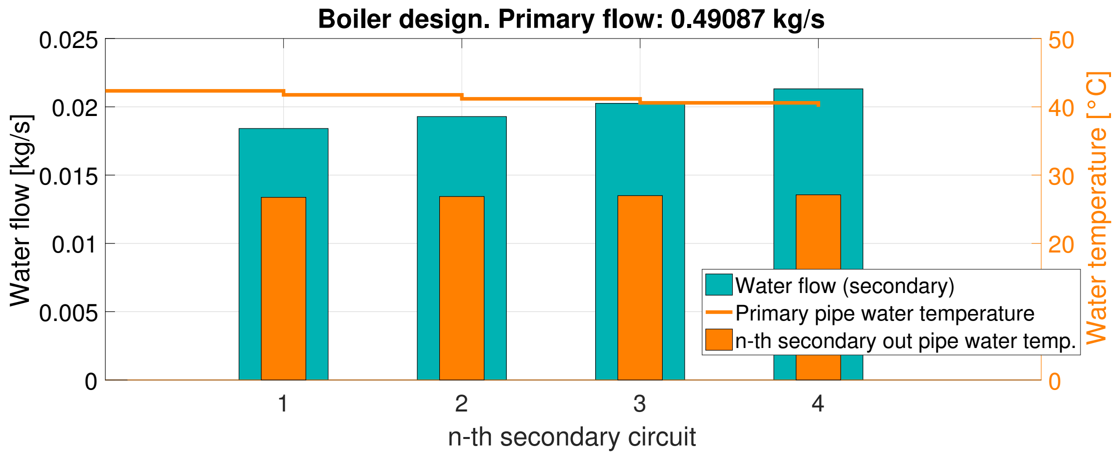


Figure 4.1: Result of the first optimization problem. Minimal boiler outlet water temperature found: 42.23 °C in order to maintain 40 °C at the boiler input. Demanded heat flow rates: 1200 W per heat exchanger. Water flow through the primary pipe is at its allowed maximum of 0.49 kg/s.

In the second case, the problem was made more real. Firstly, primary pipe water flow was constrained to the one fourth because when in operation it might not be efficient to have low-temperature difference at the boiler and large water flow thus having the main pump consume a lot of power. Demanded heat flows of each heat exchangers were (in order): 1600, 900, 800, 1200 W. Also temperature of the rooms were set to different values (in order): 20, 21, 23, 21 °C. The result is shown in fig. 4.2. The minimal temperature was found to be 48.96 °C.

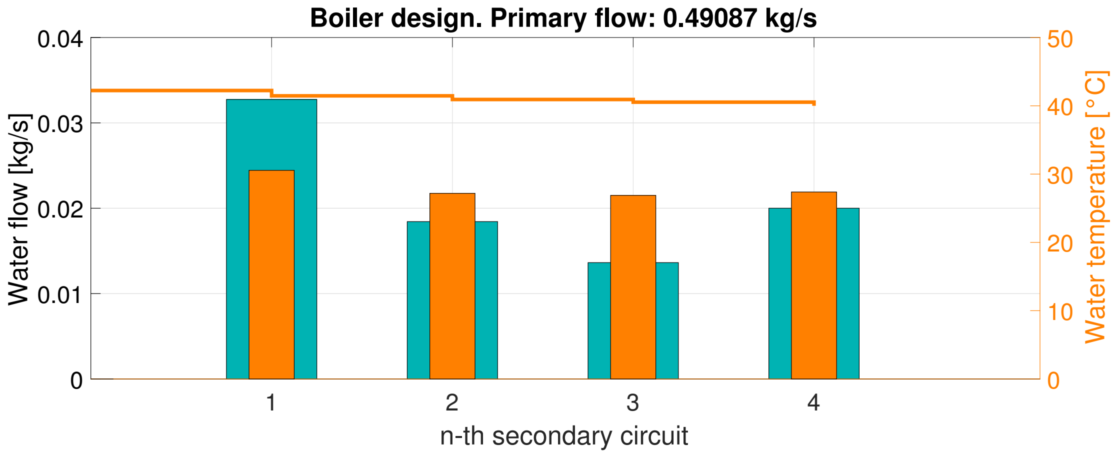


Figure 4.2: Result of the advanced first optimization problem. Minimal boiler outlet water temperature found: 48.96 °C in order to maintain 40 °C at the boiler input. Corresponding flows through secondary circuit needed for constant supply of demanded heat rates are shown. The water flow through primary pipe is at its allowed maximum of 0.12 kg/s. Legend in the figure 4.1.

4.3 Heat exchanger design

As opposed to the previous case, the problem was to finding minimal water temperature at the boiler *input* - extended with finding the optimal heat exchangers' sizes (heat transfer surfaces). Boiler output temperature was fixed to 60 °C, the temperature of each room to 20 °C and demanded heat flow rate to 1200 W per heat exchanger. Because it is a continuous optimization, the optimizer supposes any heat exchanger surface size within constraints is possible.

The result is shown in the fig. 4.3. Sizes of heat exchangers were designed such that water flowed only through the secondary circuits and nothing flowed through bridging pipes. Increasing weighting coefficient of the sum of surfaces in the optimization function (eq. 3.37) caused getting smaller HXs and increasing temperatures along the main tube, but otherwise, it didn't change the behavior (fig. 4.4).

4.4 Network operation

The boiler was able to heat water even under maximum primary flow to 60 °C. The goal was to find the lowest water temperature possible at the boiler inlet because that means maximum boiler effectiveness. Temperature of each room was set to 20 °C and demanded heat flow rate to 1200 W per heat exchanger.

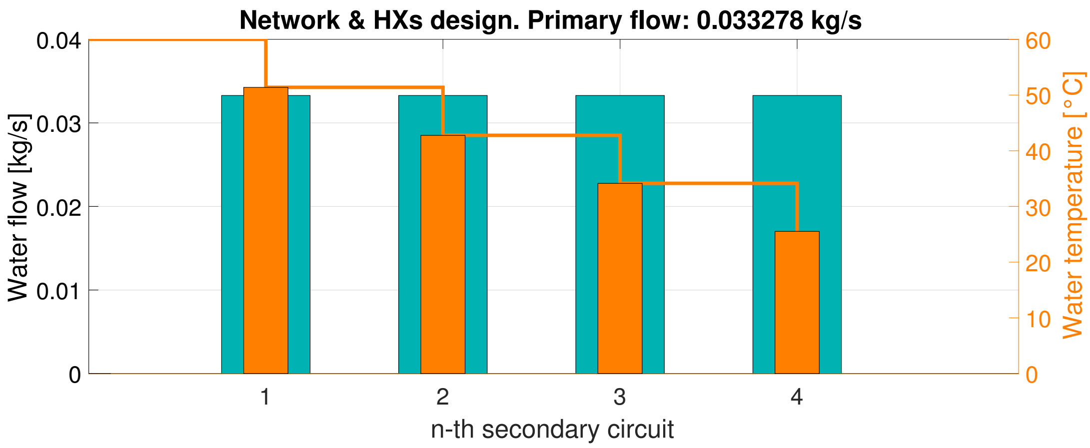


Figure 4.3: Result of the second optimization problem. Weight $w = 1$. Minimal boiler inlet water temperature found: 27.43 °C. HXs' surfaces: 0.16, 0.24, 0.43, 2.12 [m^2]. Legend in the figure 4.1.

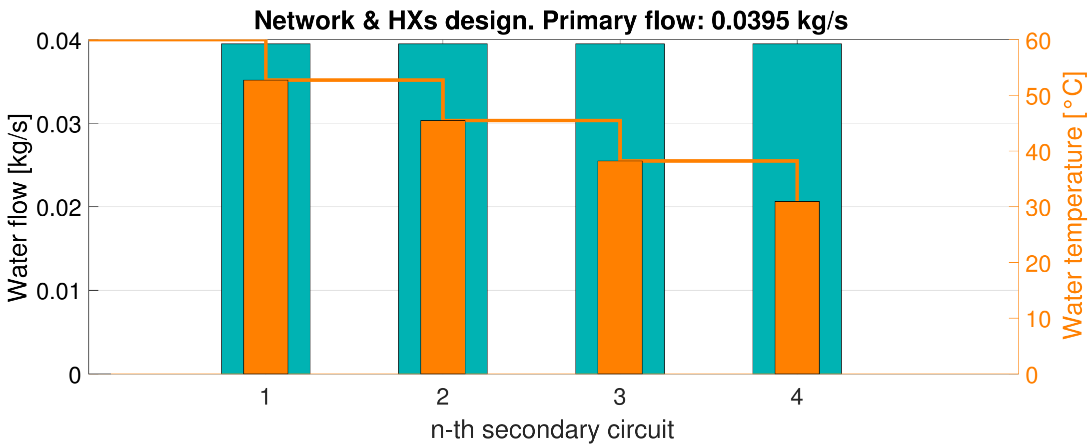


Figure 4.4: Result of the second optimization problem with more weighted heat exchanger surfaces. Weight $w = 10$. Minimal boiler inlet water temperature found: 30.96 °C. HXs' surfaces: 0.16, 0.22, 0.32, 0.63 [m^2]. Legend in the figure 4.1.

Primary water flow was found to be considerably smaller than in the previous optimization problem, which caused temperature drops along the main pipe to be much larger (units of degrees). Water flow through each secondary circuit rose to maintain constant heat flow rate. Because all water present in the primary tube flowed through the last heat exchanger and none went through the bridging pipe, the temperature at the last HX output was the same as in the primary pipe (fig. 4.5).

In the second case, each zone had different temperature and heat rate demand. Specifically, temperatures: 20, 21, 23, 20 °C and heat flow rates: 2200, 1100, 800, 1100 W. Because of the nature of one-pipe system networks, water flow through the last heat exchanger was double the flow through the first one even though the temperatures of the two rooms were the same and the demanded heat flow rate of the first HX was double the demanded rate of the fourth one (fig. 4.6).

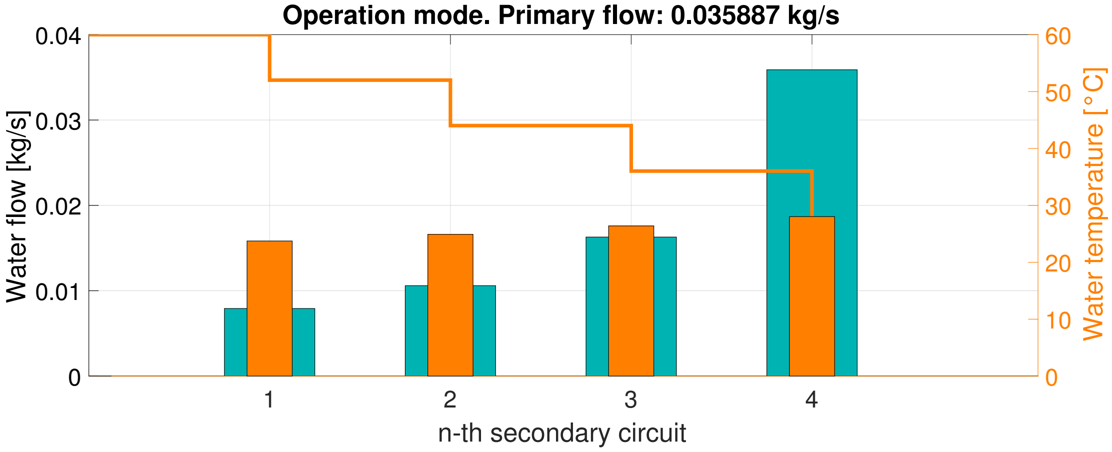


Figure 4.5: Result of the last optimization problem. Minimal boiler inlet water temperature found: 28.04 °C. Water flows through the heat exchangers rise with lower input temperature. All water from the primary pipe flows through the last HX. Legend in the figure 4.1.

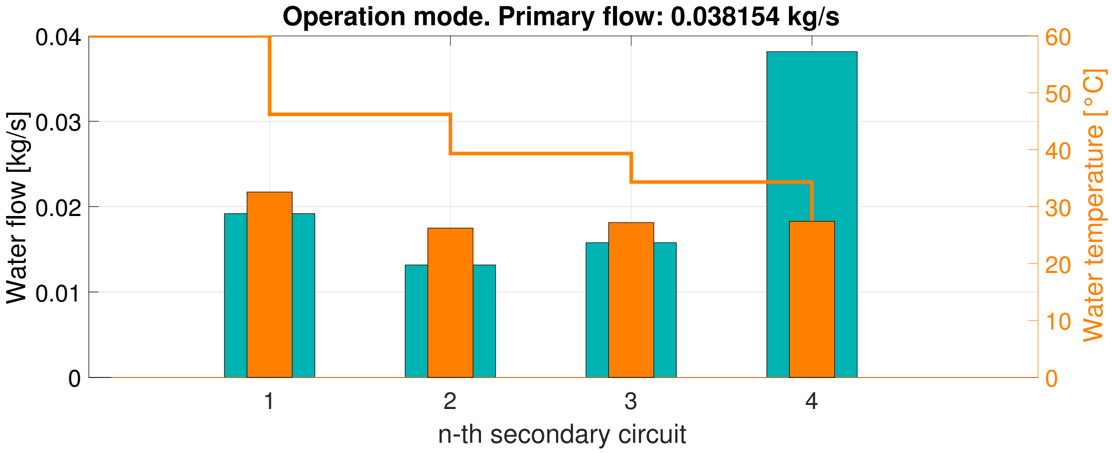


Figure 4.6: Result of the advanced last optimization problem. Minimal boiler inlet water temperature found: 27.43 °C. Water flow through the last heat exchanger is double the flow through the first one and it matches the flow in the primary pipe again. Legend in the figure 4.1.

Chapter 5

Model predictive control

In the chapter Static Optimizations the system of a one-pipe hydronic heating network was described, and its behavior was analyzed through series of experiments. In this chapter, how to control such system using the technique called *Model predictive control* is discussed. MPC is typically used to minimize the error between real and reference state trajectories.

An MPC controller predicts future system states using the current inputs and states and future controller actions.A necessary part of the controller is a system model, which describes system dynamics. Another important part is an objective function (also referred to as cost function) that contains the terms representing various control requirements and can be linear or nonlinear. Fig. 5.1 shows a scheme of an MPC controller.

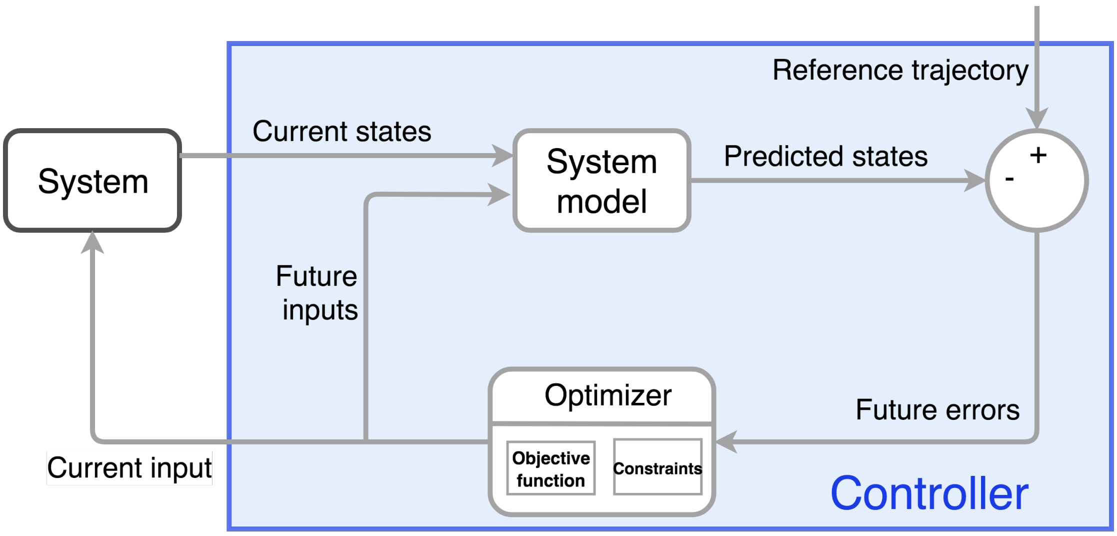


Figure 5.1: Scheme of an MPC controller.

The state at time t is known (measured or estimated) and based on that the sequence of control inputs over some time horizon N is computed such that the state evolution will be in some sense optimal. If the definition were left like this, it would be an open loop control. To make it closed loop control, the first control input is applied (sent to an actuator) and the rest is discarded. In the time $t+1$ we the procedure is repeated, now over time horizon from $t+1$ to $N+1$. This window shifting is called receding horizon. So in every step, the same problem is computed again mostly over the same variables as in the previous step. Only the variables at time t are removed since they are now in the past, and new variables for time $N+1$ are added.

Mathematically described:

$$
\begin{aligned}
\underset{\substack{\boldsymbol{x}(t+1),\ldots,\boldsymbol{x}(t+N)\\ \boldsymbol{u}(t),\ldots,\boldsymbol{u}(t+N-1)}}{\text{minimize}} \quad & J = \frac{1}{2}\left(\boldsymbol{x}(t+N) - \boldsymbol{x}_{\text{ref}}(t+N)\right)^T Q_N \left(\boldsymbol{x}(t+N) - \boldsymbol{x}_{\text{ref}}(t+N)\right) \\
& + \frac{1}{2}\sum_{k=0}^{N-1}\Big(\left(\boldsymbol{x}(t+k) - \boldsymbol{x}_{\text{ref}}(t+k)\right)^T Q \left(\boldsymbol{x}(t+k) - \boldsymbol{x}_{\text{ref}}(t+k)\right) + \boldsymbol{u}^T(t+k) R \boldsymbol{u}(t+k)\Big) \\
\text{subject to} \quad & \boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k) \qquad k = t,\ t+1,\ldots,\ t+N \\
& \boldsymbol{x}(k_0) = \text{given} \\
& \boldsymbol{x}_{min} \le \boldsymbol{x}(k) \le \boldsymbol{x}_{max} \\
& \boldsymbol{u}_{min} \le \boldsymbol{u}(k) \le \boldsymbol{u}_{max}
\end{aligned}
\tag{5.1}
$$

5.1 Building thermal model

In chapter 3 about static optimization, how to calculate optimal water flows through the system to achieve demanded heat flow rates was discussed. In this section, how to compute those heat flows required to achieve, in steady state, demanded temperatures of the rooms, will be shown. To do that, a thermal building model will be needed. How to construct and identify such models was extensively described in [19] from which the models used here are derived. The simpler variant *R1C0* of a room model is used.

The R1C0 is a model of a room that consists of air described by its temperature and capacitance, four walls characterized by their resistances, and ambient environment with its temperature which can be either outside open air or another room. Each zone also has its source of heat. Example of this model is in the fig. 5.2.

Its state space representation can be written as

$$
\begin{aligned}
\dot{T}_z(t) &= AT_z(t) + \mathbf{B}\boldsymbol{u}(t) \\
y &= CT + \mathbf{D}\boldsymbol{u}
\end{aligned}
\tag{5.2}
$$

where $A \in \mathbb{R},\ \ \mathbf{B} \in \mathbb{R}^{1\times5}, C \in \mathbb{R},\ \ \mathbf{D} \in \mathbb{R}^{1\times5}$ and

$$A = -\frac{1}{C_z}\sum_{i=1}^{4} U^i \cdot A_w^i, \qquad i \text{ meaning wall index, not power}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{C_z} & \frac{U^1\cdot A_w^1}{C_z} & \frac{U^2\cdot A_w^2}{C_z} & \frac{U^3\cdot A_w^3}{C_z} & \frac{U^4\cdot A_w^4}{C_z} \end{bmatrix}, \qquad \boldsymbol{u}(t) = \begin{bmatrix} \dot{Q}_{HX}(t) \\ T_{n1}(t) \\ T_{n2}(t) \\ T_{n3}(t) \\ T_{n4}(t) \end{bmatrix}$$

C_z $[\mathrm{J}\cdot\mathrm{K}^{-1}]$ is the air thermal capacity, U $[\mathrm{W}\cdot\mathrm{m}^{-2}\cdot\mathrm{K}^{-1}]$ is the wall thermal conductance and A_w $[\mathrm{m}^2]$ is the wall's surface. The termal capacity is calculated as:

$$C_z = \rho_a \cdot V \cdot c_a \tag{5.3}$$

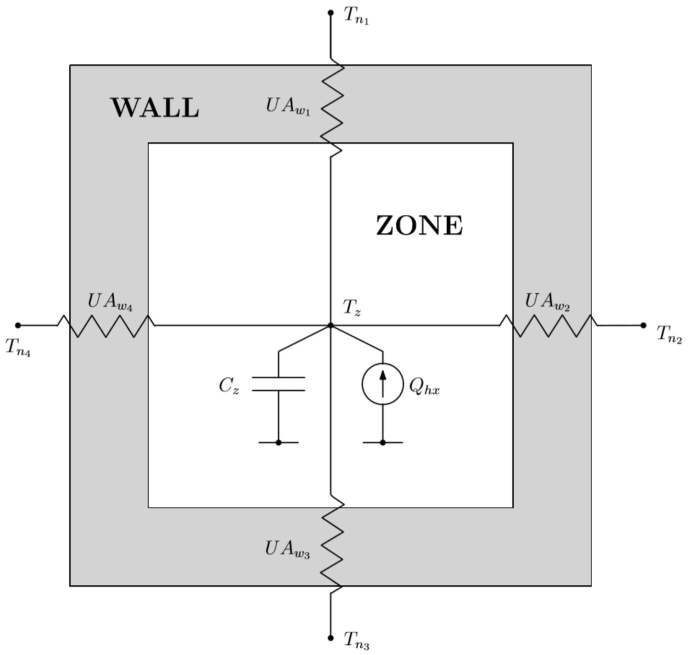


Figure 5.2: Scheme of an R1C0 room model.[19]

where ρ_a $[\mathrm{kg} \cdot \mathrm{m}^{-3}]$ is air density, V $[\mathrm{m}^3]$ is volume of air in the zone, c_a $[J \cdot Kg^{-1} \cdot K^{-1}]$ is air specific heat. The thermal conductance of a wall is:

$$U = k/l \tag{5.4}$$

where k $[\mathrm{W} \cdot \mathrm{m}^{-1} \cdot \mathrm{K}^{-1}]$ is the wall's material thermal conductivity and l [m] is the wall's width.

When writing this thesis I had no centralized model of a building. However I had tools tools to create one from a distributed model. An algorithm for fast generation of the zone models for a whole building or at least of a floor was developed by Tomáš Bäumelt and is described in [19]. It takes an *adjacency matrix* representing the zones arrangement in a building and returns appropriate one-zone models with general variables representing zone characteristics (e.g., zone's size, wall's width & conductivity). Variables that are common to two or more zones (typically a wall between two rooms) are treated as one. The adjacency matrix for a building with n rooms is:

$$\begin{aligned} \mathbf{A}_{i,j} &= \begin{cases} 1 & \text{if } i\text{-th room shares a wall with} j\text{-th room} \\ 0 & \text{otherwise} \end{cases} \\ i,j &= 1,\ldots,n \end{aligned} \tag{5.5}$$

An algorithm that connects these zones into one centralized system was one of the outcomes of the thesis developed by Jan Hauser and is described in [20]. Zones from the algorithm developed by Bäumelt must be passed as an input along with an extended adjacency matrix. That has a similar structure as the adjacency matrix defined in (5.5), but instead of 1 when two zones share a wall, there is an index of the common wall in the zone's object's array. More formally:

$$
\begin{aligned}
\mathbf{A}_{\text{ext\_}i,j} &= \begin{cases} k & \text{if } i\text{-th room shares a wall with} j\text{-th room} \\ 0 & \text{otherwise} \end{cases} \\
i,j &= 1,\ldots,n \\
k &= \text{index of the wall in the } i\text{-th zone's array}
\end{aligned}
\tag{5.6}
$$

Result of this algorithm is a centralized state space model represented by its $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times 5}$, $\mathbf{C} \in \mathbb{R}^{n\times n}$, $\mathbf{D} \in \mathbb{R}^{n\times 5}$ matrices. The state vector and the input vector are:

$$
\boldsymbol{T} = \begin{bmatrix} T_{z1} \\ T_{z2} \\ \vdots \\ T_{zn} \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} Q_{HX1} \\ Q_{HX2} \\ \vdots \\ Q_{HXn} \\ T_{amb} \end{bmatrix}
\tag{5.7}
$$

For an example of creation of a building thermal model see chapter 6.

5.2 Linear MPC

The centralized system model of a building must be discretized because MPC is a discrete control technique. Then the controller can be designed. Firstly, a simple MPC controller already implemented by Hauser in [20] will be discussed. It has a quadratic objective function with linear constraints. Inputs to the controller are desired temperatures in every room over a specified period. Outputs of the controller are heat flow rates that have to be supplied to each room (to achieve the desired temperatures or more precisely: to minimize the objective function).

The objective function for one step is:

$$
\begin{aligned}
J(k) = &\left(\boldsymbol{C}\boldsymbol{x}(k) - \boldsymbol{x_s}(k)\right)^T Q \left(\boldsymbol{C}\boldsymbol{x}(k) - \boldsymbol{x_s}(k)\right) + \boldsymbol{u}(k)^T R\boldsymbol{u}(k) \\
&+ \left(\mathbf{1}^T \cdot \boldsymbol{u}(k) - \mathbf{1}^T \cdot \boldsymbol{u_s}(k)\right)^T q \left(\mathbf{1}^T \cdot \boldsymbol{u}(k) - \mathbf{1}^T \cdot \boldsymbol{u_s}(k)\right)
\end{aligned}
\tag{5.8}
$$

the equality constraints are:

$$
\boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}\boldsymbol{u}(k) \tag{5.9}
$$

and the inequality constraints are:

$$
\begin{aligned}
&\boldsymbol{x}(k_0) = \text{given} \\
&\boldsymbol{r}_{min}(k) \le \boldsymbol{x_s}(k) \le \boldsymbol{r}_{max}(k) \\
&\boldsymbol{u}_{min} \le \boldsymbol{u_s}(k) \\
&\mathbf{1} \cdot \boldsymbol{u_s}(k) \le \boldsymbol{u}_{totMax} \\
&\boldsymbol{u}_{min} \le \boldsymbol{u}(k)
\end{aligned}
\tag{5.10}
$$

Heat source power that must be optimally distributed into individual rooms is limited. This limit is represented by the soft constraint $\boldsymbol{u}_{totMax}$. To give the optimizer more freedom, the desired temperatures are not represented by lines but by a band. This band is represented in each time step by constraints $\boldsymbol{r}_{min}(k)$ and $\boldsymbol{r}_{max}(k)$. To increase feasibility so called *slack variables* $\boldsymbol{x_s}$, $\boldsymbol{u_s}$ are added(subsec. 5.2.1). Q is the penalty coefficient that is applied when the predicted temperature is not in the desired band, R is the penalty coefficient that tries to minimize input usage, and q is the coefficient that penalizes input even more in case it exceeds the soft constraint $\boldsymbol{u}_{totMax}$.

5.2.1 Slack variables

Variables representing the behavior of the system, in the case mentioned above $\boldsymbol{x}$ or $\boldsymbol{u}$, are unbounded. To apply some kind of restrictions on them, so called slack variables $\boldsymbol{x_s}$, $\boldsymbol{u_s}$ are introduced. The slack variables are bounded and their boundaries are specified by given band. When $\boldsymbol{x}$ is in the band, it has the same value as the slack variable $\boldsymbol{x_s}$. Their difference is zero, therefore the penalty Q (eq. 5.8) has no effect. When it's outside of the band, the penalty Q is multiplied by the distance from the border. Analogically it is also implemented with the total heat rate limit. For an illustration see figure 5.3.

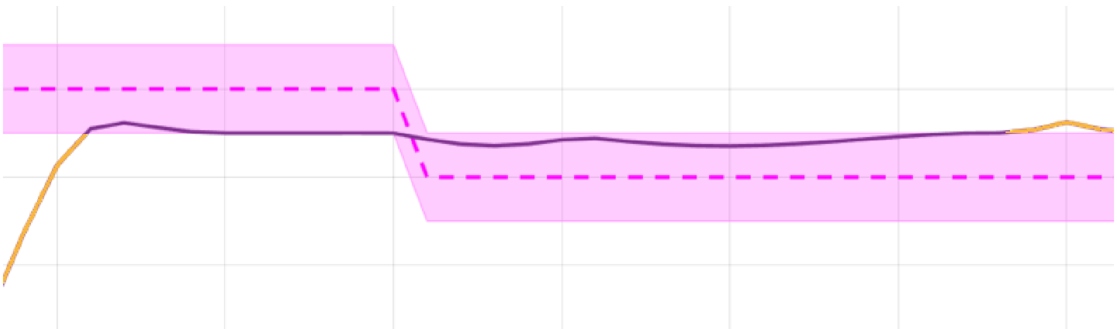


Figure 5.3: The slack variable can move only in the pink band unlike the real variable that can move freely. However, since their difference is penalized, it's best for the real variable to stay in the band where the difference is zero. The solid line (the real variable) represents the state with the orange parts representing the penalized trajectory. The dashed line is just the middle of the band.

5.3 Nonlinear MPC

The linear MPC controller described in 5.2 could be connected with the static optimizer (3). Necessary heat flow rates computed by the MPC controller would be passed into the optimizer that would produce final mass flows. However, instantiating a static optimizer for every time step would be computationally inefficient. Moreover, the MPC controller doesn't take into account the connection of heat exchangers and could demand heat flow rates that would be impossible to achieve. Therefore, an MPC controller that combines both into one entity is designed. The inputs are the requested temperature bands, and the outputs are necessary water flows through the one-pipe network.

In the following subsection, the underlying optimization problem is presented in the form of the objective function and constraints. Referring to multiple variables of the same kind is done via vectors (bold font); for example, $\boldsymbol{T}_p$ means water temperatures flowing into all heat exchangers while $T_p^i(k)$ means just the water temperature flowing into i-th one. For better understanding, a table of variables used in this section is shown in table 5.1.

symbol	dimension	description
N		control horizon
n		number of zones
k	$1,\dots,N$	step
$\boldsymbol{x}$	$n \times (N+1)$	(predicted) zone temperatures
$\boldsymbol{x}_s$	$n \times (N+1)$	slack variable of (predicted) zone temperatures
$\boldsymbol{u}$	$n \times N$	heat flow rates being transferred into rooms
$\dot{\boldsymbol{m}}_s$	$n \times N$	water flows in secondary pipes
$\dot{m}_p$	$1 \times N$	water flow in primary pipe
$\boldsymbol{T}_p$	$n \times N$	temperatures in the primary tube
$\boldsymbol{r}_{min}$	$n \times N$	lower bound of desired temperatures
$\boldsymbol{r}_{max}$	$n \times N$	upper bound of desired temperatures

Table 5.1: NMPC variables.

5.3.1 Objective function

The objective function is the most fundamental part of the controller. It defines how the system should be controlled. There are more possible objectives that can be applied to control. Those that were applied during the development of this thesis are listed below, however, note that not all of them have to be used. They are combined simply by summing them. If two objectives are contradictory, then one with bigger weight will have more influence.

- **Reaching desired states.** This is a function that penalizes if the state is not in the desired band. It will be active every time.

$$J_r(k) = \left(\boldsymbol{C}\boldsymbol{x}(k) - \boldsymbol{x_s}(k)\right)^T Q \left(\boldsymbol{C}\boldsymbol{x}(k) - \boldsymbol{x_s}(k)\right) \tag{5.11}$$

- **Total input cost.** This objective function minimizes the total inputs usage (water flows). If active, it will generally tend to minimize flows, therefore, the temperatures will tend to their lower bounds. It is advised to have R much smaller then Q.

$$J_c(k) = \boldsymbol{u}(k)^T R \boldsymbol{u}(k) \tag{5.12}$$

- **Secondary water flows smoothness.** It might be inefficient or even impossible for the pumps to provide sharp, fast changes of the demanded water flows in the secondary circuits. This objective function penalizes this behavior.

$$J_{ss}(k) = \left(\boldsymbol{m}_s(k) - \boldsymbol{m}_s(k-1)\right)^T \boldsymbol{P}_s \left(\boldsymbol{m}_s(k) - \boldsymbol{m}_s(k-1)\right) \tag{5.13}$$

- **Primary water flows smoothness**. Same as the above for the primary flow.

$$J_{ps}(k) = \left(\boldsymbol{m}_p(k) - \boldsymbol{m}_p(k-1)\right)^T \boldsymbol{P}_p \left(\boldsymbol{m}_p(k) - \boldsymbol{m}_p(k-1)\right) \tag{5.14}$$

5.3.2 Constraints

Unlike the objective functions that can be combined, all the constraints are in use all the time.

- The dynamical model of the building. Linear equality constraint.

$$\mathcal{C}_1: \quad \boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}\boldsymbol{u}(k) \tag{5.15}$$

- Conversion of heat flows into water flows and temperatures. Nonlinear equality constraint.

$$\mathcal{C}_2: \quad \boldsymbol{u}(k) = \dot{\boldsymbol{Q}}_{HX} = \left[1 - \exp\left(-\frac{U(\dot{\boldsymbol{m}}_s(k), \dot{V}_{air}) \cdot A}{\min(\dot{\boldsymbol{m}}_s(k) \cdot c_w, \dot{V}_{air} \cdot c_{air})}\right)\right] \cdot \min(\dot{\boldsymbol{m}}_s(k) \cdot c_w, \dot{V}_{air} \cdot c_{air}) \cdot (\boldsymbol{T}_p(k) - \boldsymbol{x}(k)) \tag{5.16}$$

- Water temperature dependence on preceding HX. Nonlinear equality constraint.

$$\mathcal{C}_3: \quad T_p^i(k) = T_p^{i-1}(k) - \frac{\dot{Q}_{HX}^{i-1}(k)}{c_w \cdot \dot{m}_p(k)} \tag{5.17}$$

- State(zone temperatures) slack variables linear inequality constraints.

$$\mathcal{C}_4: \quad \boldsymbol{r}_{min}(k) \leq \boldsymbol{x}_s(k) \leq \boldsymbol{r}_{max}(k) \tag{5.18}$$

- Water flow in any secondary pipe is less or equal to the one in the primary pipe to avoid reverse flow.

$$\mathcal{C}_5: \quad \dot{m}_s^i(k) \leq \dot{m}_p(k) \tag{5.19}$$

- Heat flow rate provided by an HX is always positive. Active cooling is not considered, although an extension would be fairly straightforward.

$$\mathcal{C}_6: \quad 0 < \boldsymbol{u}(k) \tag{5.20}$$

- Water in any pipe can't flow in the opposite direction and is capped at a maximum.

$$\begin{aligned} &\mathcal{C}_7: \quad 0 \leq \dot{m}_s^i(k) \leq \dot{m}_{s\_\max}, \quad i = 1, \dots, \#\ \text{HXs} \\ &\mathcal{C}_8: \quad 0 \leq \dot{m}_p(k) \leq \dot{m}_{p\_\max} \end{aligned} \tag{5.21}$$

- Water temperature at the boiler outlet is fixed.

$$\mathcal{C}_9: \quad T_p^1(k) = T_{p\_\text{designed}}^1(k) \tag{5.22}$$

- The temperature of water between HXs in the primary pipe must be less than boiler temperature. Lower boundary is introduced because the water temperature is not supposed to go under e.g. 10 °C (in fact, under the minimal temperature of all zones). By doing this, the optimizer search space is more limited which might speed up the optimization. In general, it's advised to limit as many optimization variables as possible.

$$\mathcal{C}_{10}: \quad \boldsymbol{T}_{p\_\min}^{2,\dots,n}(k) \leq \boldsymbol{T}_p^{2,\dots,n}(k) \leq \boldsymbol{T}_{p\_\text{designed}}^{2,\dots,n}(k) \tag{5.23}$$

5.3.3 Min function smoothening

Optimization problem solvers often use first derivatives (gradients) or even second derivatives (Hessians). Therefore, not differentiable functions such as the min function in 5.16 have to be approximated with differentiable variants. For two dimensions it holds that $\min(x, y) = \frac{1}{2}(x + y - |x - y|)$. Therefore, an approximation to $z \to |z|$ needs to be found. A suitable one can be $|z| \doteq \sqrt{z^2 + \alpha}$, where α is a positive real number. An example of this approximation is shown in the fig. 5.4. The approximation of the original function:

$$\min(x, y) = \frac{1}{2}\left(x + y - \sqrt{(x - y)^2 + \alpha}\right) \tag{5.24}$$

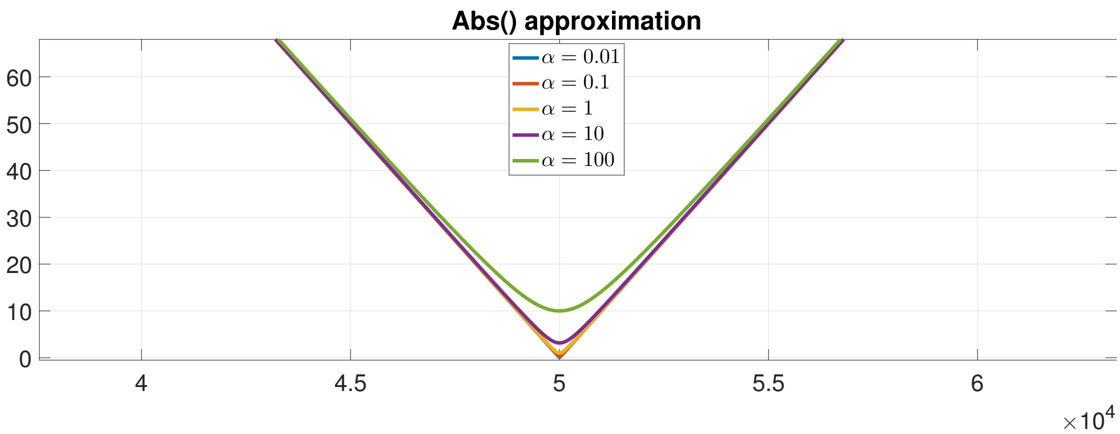


Figure 5.4: Approximation of an absolute function for different α.

Chapter 6

MPC Experiments and Implementation

The following experiments were implemented in Matlab®. A three-zone building model (fig. 6.1) was chosen as a minimal representative problem, which could be used to show all interesting behaviors while not having the model too complicated for a paper presentation.

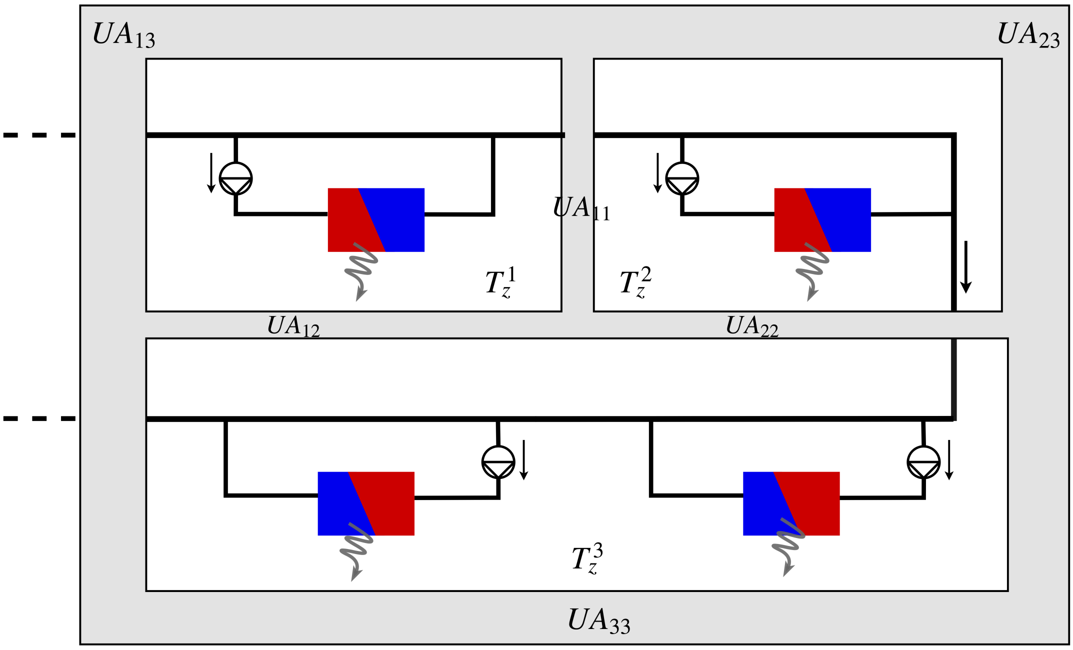


Figure 6.1: Three-zone building model with the one-pipe network. The model of the hypothetical building which was used for the experiments. The model has two zones of the same size and one zone of double the size of the smaller. Outer walls are also twice as wide as than the walls between rooms.

The adjacency matrix of this model is:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{6.1}$$

The single zones obtained as the output of the algorithm by Bäumelt[19] have the following general properties (table 6.1):

zone	capacitance	wall 1	wall 2	wall 3
1	C_{z1}	UA_{11}	UA_{12}	UA_{12}
2	C_{z2}	UA_{11}	UA_{22}	UA_{23}
3	C_{z3}	UA_{12}	UA_{22}	UA_{33}

Table 6.1: Single zones' parameters in the building model.

these variables are later on substituted with real values. The extended adjacency matrix is:

$$\mathbf{A}_{ext} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix} \tag{6.2}$$

6.1 Linear MPC: Supplied heat flow rate

The linear MPC controller described in 5.2 was used. The goal was to find heat flow rates for all rooms necessary to achieve desired temperatures in the rooms. The one-pipe network was not considered. The limit of possible heat flow rates that could be transferred altogether was set to 5000 [W]. The resulting (optimal) distribution of heat flow rates into individual rooms is shown in the fig. 6.3. The system was discretized with the discretization period 10 min, and the prediction of the evolution of the zone temperatures is made for 50 time steps, that means over 8 *hours*. The underlying optimization problem was solved in 0.21 seconds. The results of the prediction from the first step of the zone temperatures are shown in fig. 6.2.

6.1.1 Implementation

The problem falls under the category of quadratic programming which modern solvers usually have no problem of solving. However, comparing this problem which has 461 optimization variables to the Static Optimizations that had between 5 to 10 depending on the chosen scenario, it's clear that this is much more complex. Even in the linear case, the `fmincon` function is incapable of solving it. Therefore the `Gurobi`[21] solver is used. For efficient problem description, the code is written with `Yalmip toolbox`[1] for Matlab [22]. `Yalmip` is a "mediator" that makes defining a control oriented optimization problem easier, preprocesses it and passes it to solvers.

6.2 Nonlinear MPC: Computing water flows

As opposed to the linear MPC controller, in the nonlinear MPC the one-pipe network with heat exchangers is considered (fig. 6.1). The goal is to compute water flows through the network that would provide demanded temperatures in the rooms. However, computing water flows and heat exchangers efficiencies introduces nonlinear constraints. The optimization problem behind this example was described in section 5.3. Switching

[1] `https://yalmip.github.io/`

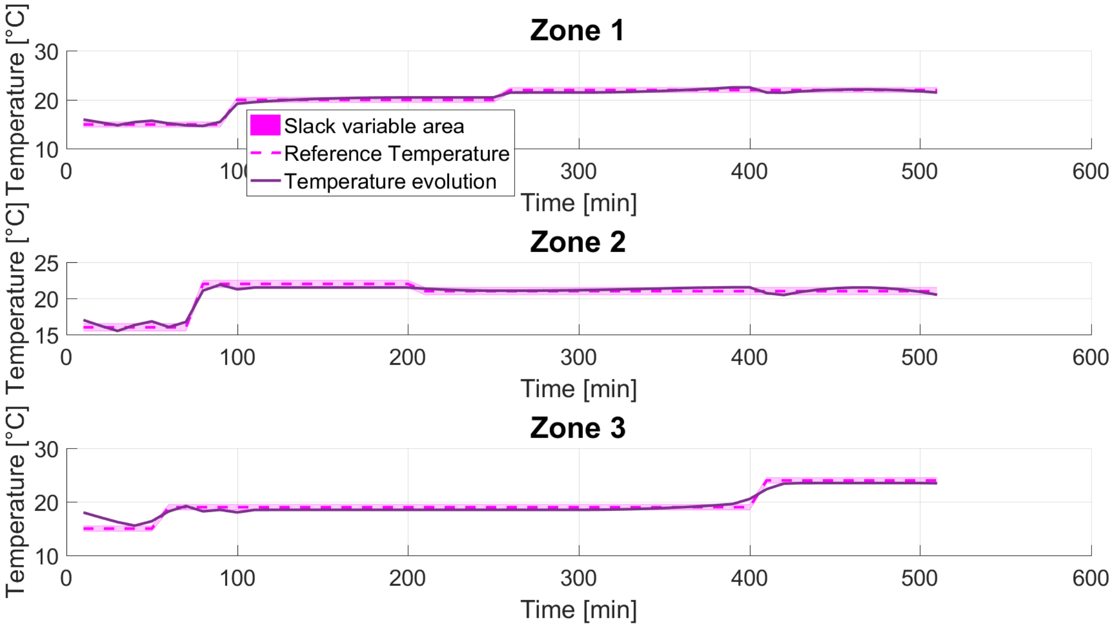


Figure 6.2: Temperature evolution of the room temperatures. Prediction from the first step of the linear MPC controller. The presence of the system model in the controller is visible around time 400min, when the controller pre-heats zone 1&2 and then heats only the third zone for some time.

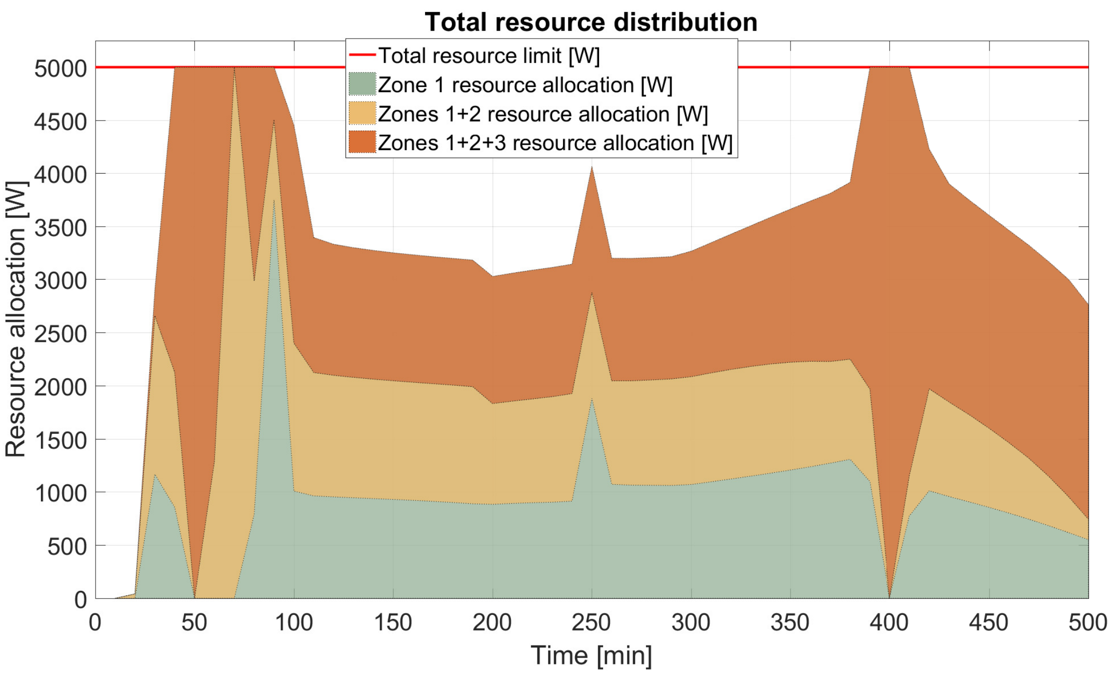


Figure 6.3: Total heat flow rate distribution among rooms. Prediction from the first step of the linear MPC controller. The presence of the system model in the controller is visible around time 400min, when the controller pre-heats zone 1&2 and then at $Time = 400\text{min}$ it assigns the whole heat power resource to the third zone for some time.

to water flows introduced another optimization variables (table 5.1), and there is now 906 of them in total, which is double when compared to the linear MPC.

6.2.1 Implementation

Gurobi® is undoubtedly one of the best optimizer on the market for linear, quadratic and mixed integer programming [23]. However this is a highly nonlinear problem, optimization variables regarding secondary circuits farther in the net depend on all the previous ones and conducted experiments showed that no out-of-the-box nonlinear solver can solve it. Therefore a new approach had to be taken.

CasADi

Typically solvers call the objective function (or the modified objective function with barriers from constraints) and numerically (using finite differences) try to estimate gradients and Hessians. Joel Andersson Ph.D. presented in his dissertation thesis [24] software framework compatible with Matlab called `CasADi` (a tool for **a**lgorithmic **di**fferentiation using a syntax borrowed from **c**omputer **a**lgebra **s**ystems), that allows evaluating the first and second order derivatives efficiently and without errors. It exploits the fact that in the end, every expression executes as a sequence of elementary arithmetic operations and elementary functions. It decomposes the expression into these basic terms and constructs a computational graph (fig. 6.4). Then by applying the chain rule (eq. 6.3) for derivatives, derivatives of arbitrary order can be computed. CasADi can handle even not smooth functions such as $\min(x, y)$ so implementation of the smoothening approximation described in 5.3.3 is no longer required.

$$(f \circ g)' = (f' \circ g) \cdot g' \tag{6.3}$$

Similarly to Yalmip, it preprocesses the optimization problem before passing it to solvers, which may speed up the optimization. Notably, by using graph coloring techniques, it discovers sparsity of matrices (fig. 6.5) representing optimization variables and reformulates the problem into a simpler one. The comparison with other optimization toolboxes is shown in figure 6.6. We managed to solve the optimization problem using CasADi with the `ipopt`[2] solver. A nice feature of `ipopt` is that it automatically scales optimization variables therefore manual scaling is no longer required.

6.2.2 Experiments

The experiments regarding the nonlinear MPC were conducted at the same hypothetical building model as the linear MPC. (fig. 6.1). To allow multiple heat exchangers in one zone, the optimization problem described in 5.3 had to be slightly modified. A structure that maps heat exchangers onto zones was added, and in the constraint equation 5.16 the term $\boldsymbol{u}(k) = \dot{\boldsymbol{Q}}_{HX}$ was replaced with $u(k)^i = \sum\limits_{j=M(i)} \dot{Q}^j_{HX}$, where $M()$ is the mapping function and i means the i-th room.

[2] `https://en.wikipedia.org/wiki/IPOPT`

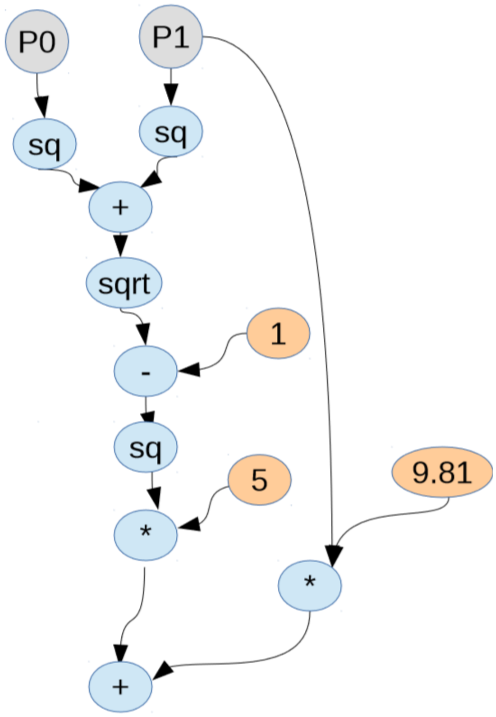


Figure 6.4: An example of a computational graph. [25]. The original expression is clearly: $5\left(\sqrt{P_0^2+P_1^2}-1\right)^2+P_1\cdot 9.81$.

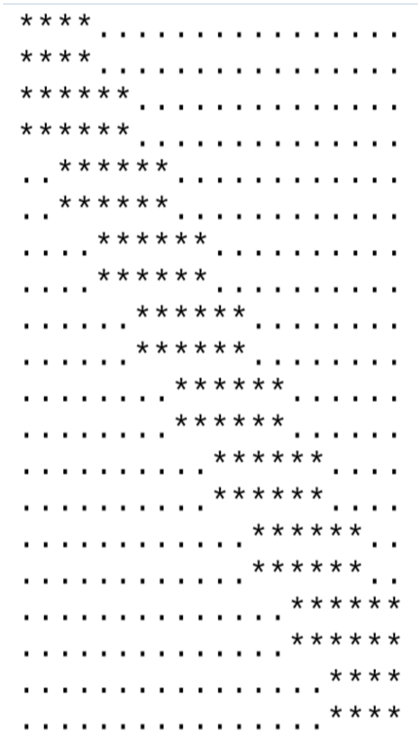


Figure 6.5: An example of a matrix sparsity. [25]. This is a typical look of a matrix, that represents a problem where variables are chained and depend on previous ones.

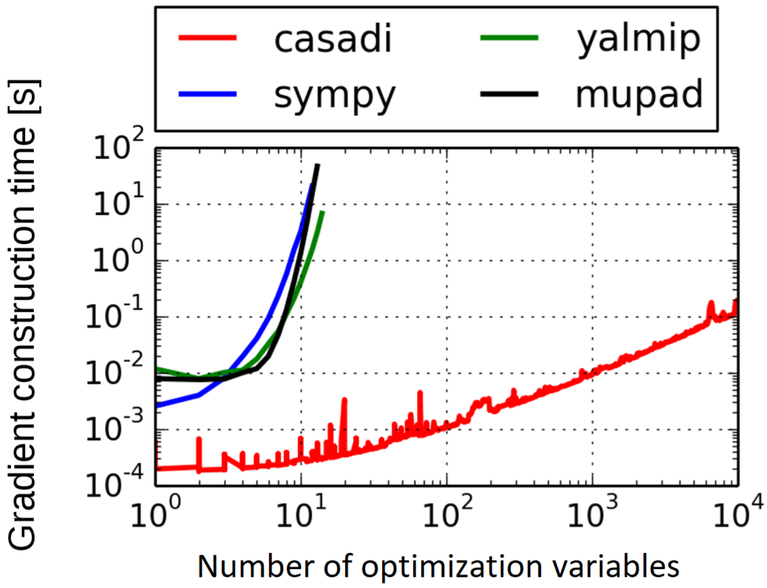


Figure 6.6: Various optimization toolboxes comparison. [25] It shows how time scales with the number of optimization variables when constructing the gradient. Results for an optimal control problem with one state, solved with the single shooting method. Prediction horizon is on the horizontal axis.

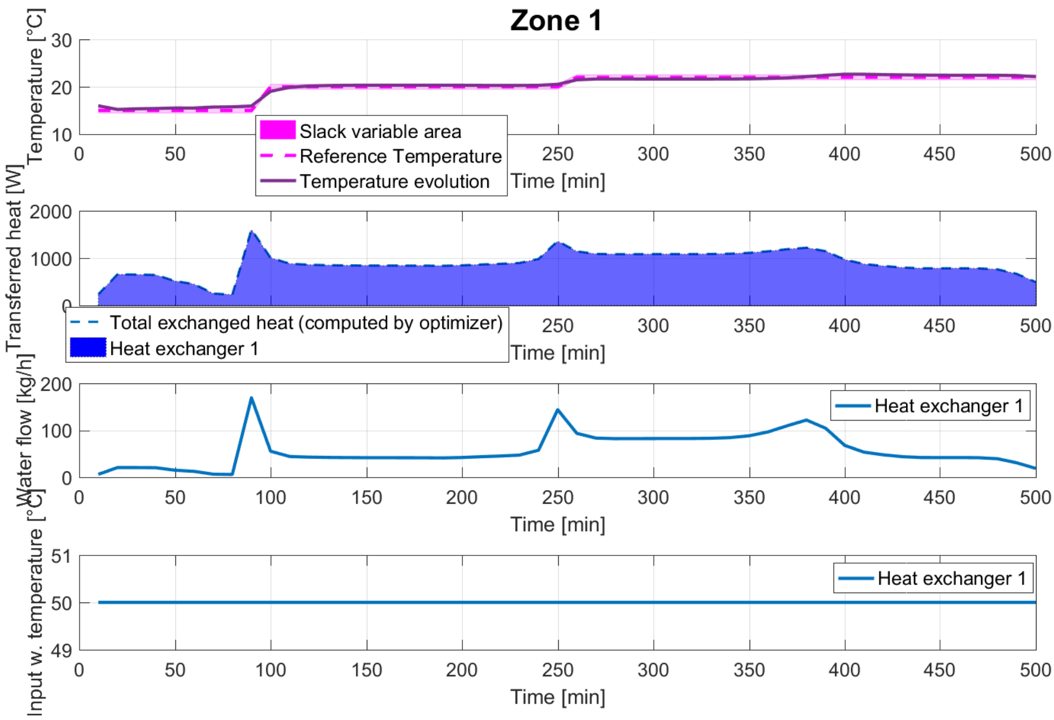
Zones pre-heating and boiler performance planning

This example is similar to the linear variant (sec. 6.1). It illustrates the presence of the model of the system in the controller. Because it "knows" that at time 400 min in the third zone there will be high demand for heat flow to raise temperature from 19°C to 25°C it pre-heats the first and second room to upper temperature boundaries and then (at time approx. 370 min) reduces water flows to the first and second heat exchanger while increasing water flow to the third and fourth HXs. This is shown in figures 6.7 and 6.8. The penalization for rapid changes in both primary and secondary pipes was active. By looking at the fig. 6.8 we can see that also the boiler power is (indirectly) planned by the controller by changing the flow in the primary pipe.

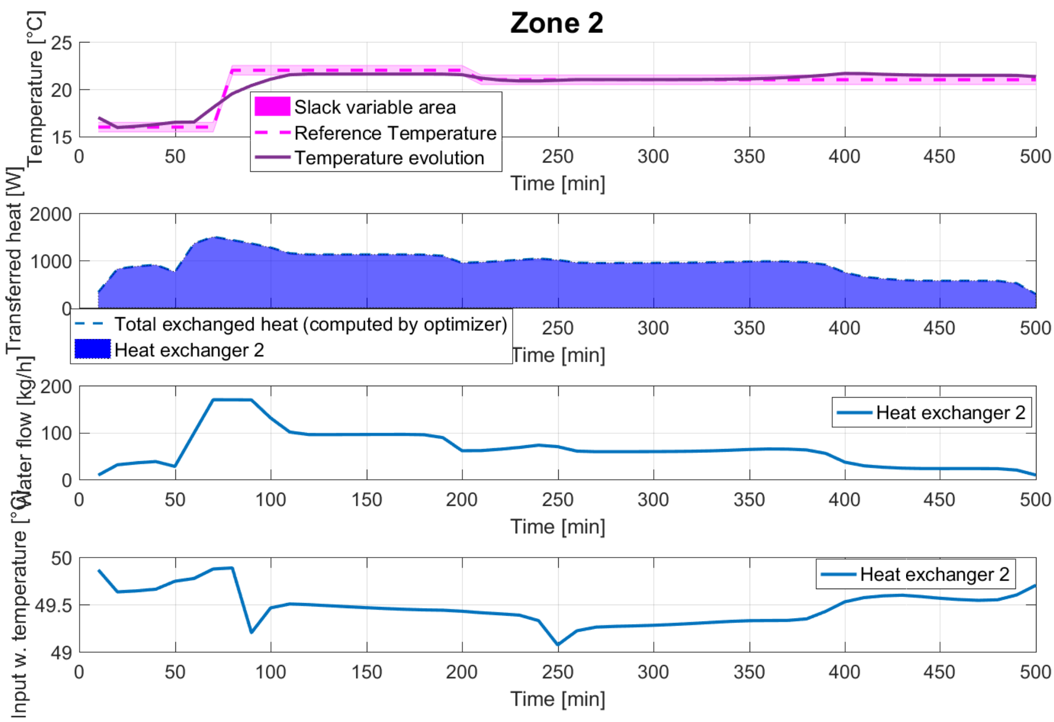
Effectiveness and bottom boundaries attraction

In the next experiment, the penalization of changes in both primary and secondary pipes was removed. As an expectable consequence, the evolution in time of water flows in both types of tubes was not that smooth. But also another phenomenon is visible - in the room with two exchangers, they both supplied the same amount of heat flow, instead of heating the room with only one with larger water flow. That is compliant with that how conductivity U scales with mass flow, which is shown in the fig. 6.9. This phenomenon only holds when the penalization of flows changes is not active.

Another behavior pattern is visible at the end of the prediction horizon in all zones. The temperature references are constant, and the real temperatures tend to fall to the bottom boundaries. That is because the controller's objective function contains the term that minimizes the heat flows usage. The outcomes of this experiment are presented in fig. 6.10 and 6.11.

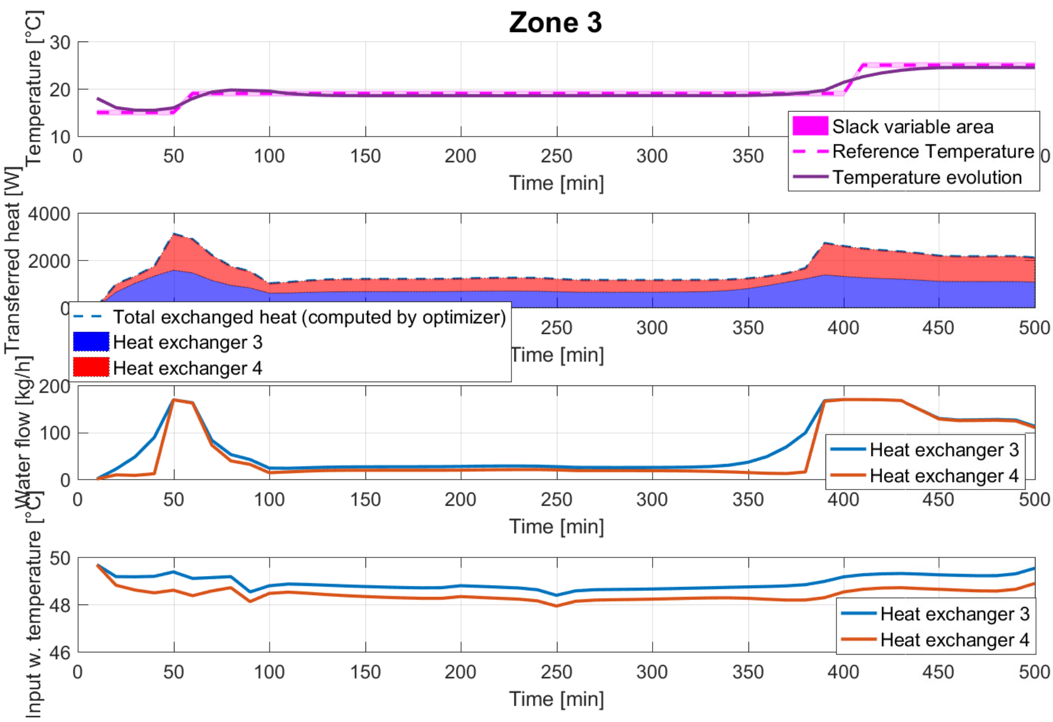


(a) : Zone 1

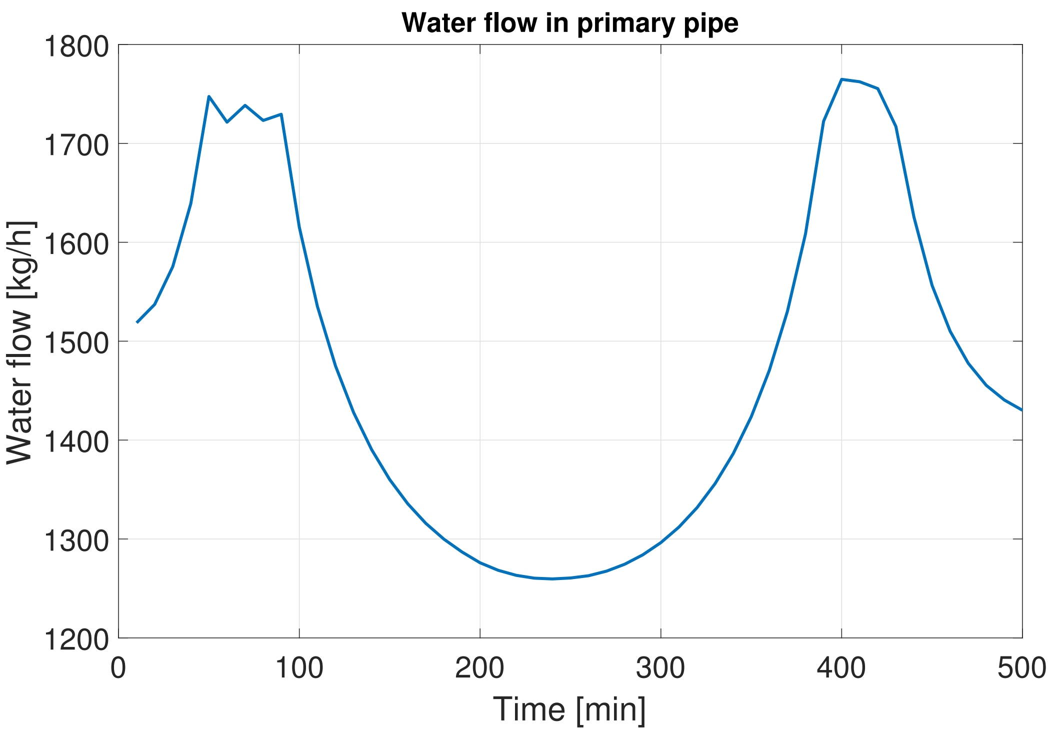


(b) : Zone 2

Figure 6.7: The "zones pre-heating" experiment part 1



(a) : Zone 3



(b) : Primary pipe

Figure 6.8: The "zones pre-heating" experiment part 2

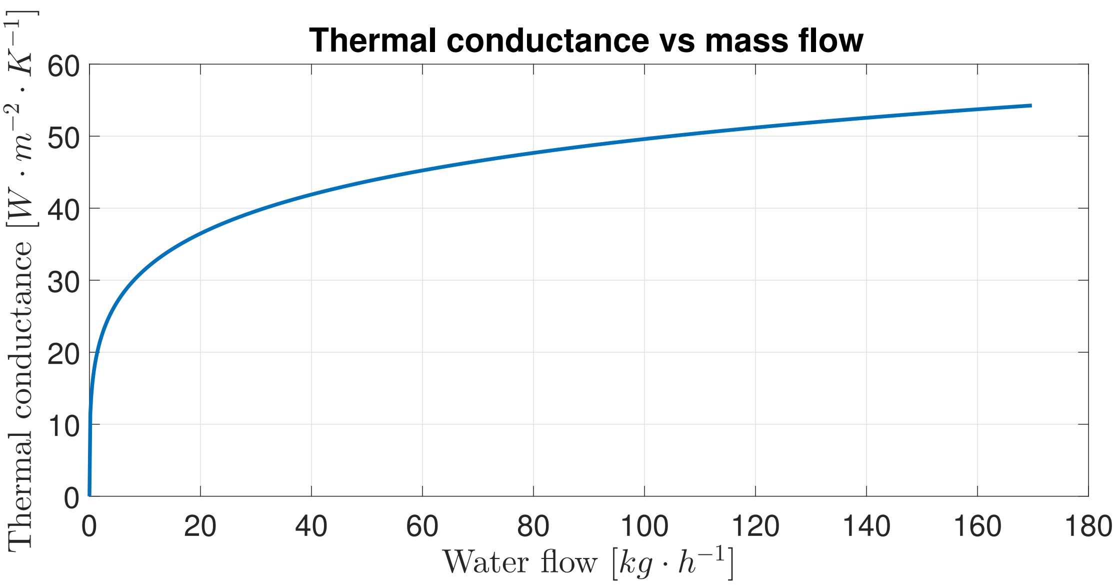
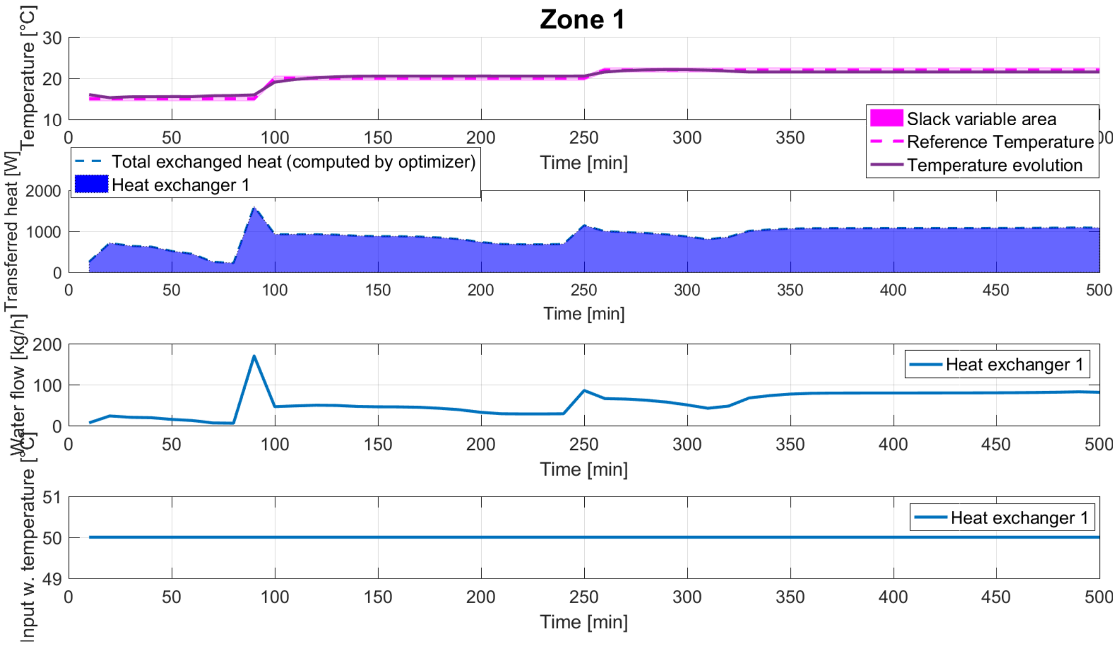
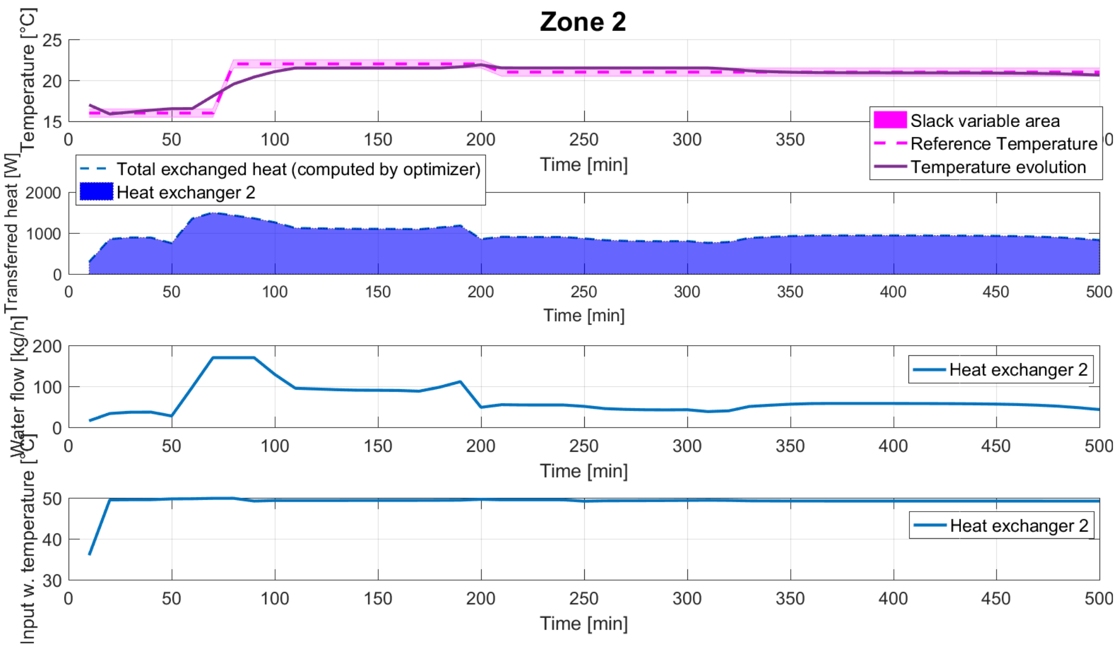


Figure 6.9: Scaling of conductance with mass flow.

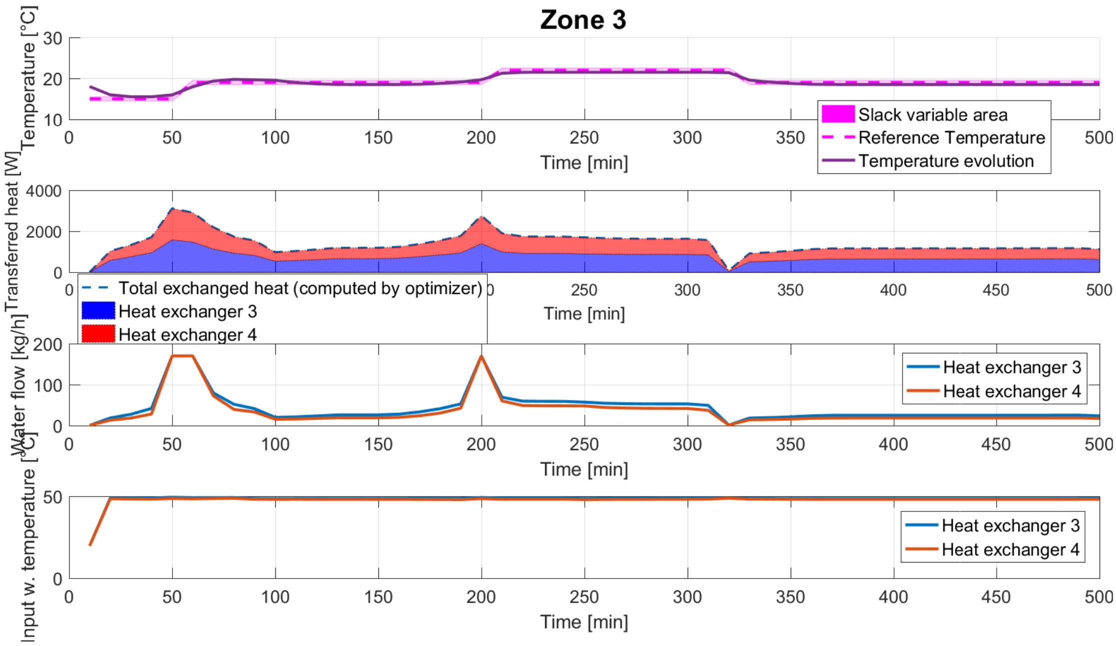


(a) : Zone 1

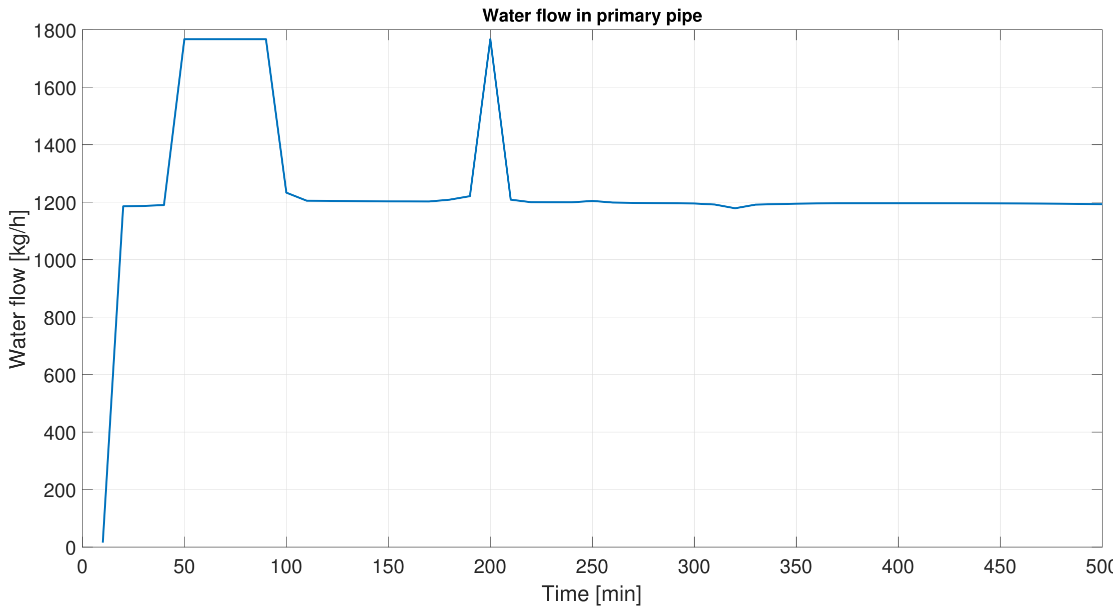


(b) : Zone 2

Figure 6.10: The "Effectiveness and bottom boundaries attraction" experiment part 1



(a) : Zone 3



(b) : Primary pipe

Figure 6.11: The "Effectiveness and bottom boundaries attraction" experiment part 2

6.3 Control mode

Both the linear and non linear MPC experiments were in fact implemented in Matlab as "predictors". Only the prediction of the states from the first step is shown in the presented figures. The prediction horizon was $N = 50$ steps. Therefore, Simulink was used to fully simulate the control mode. The MPC controller was connected in a closed loop to a state space model of the controlled building (as shown in fig. 5.1).

6.3.1 Implementation

For simulation in Simulink, `Matlab system block` that can be imported into Simulink was used. The block is then associated with Matlab `System object`. That is a class “designed specifically for implementing and simulating dynamic systems with inputs that change over time”[26]. Implementation of the System object is described in Appendix A.

There are also Matlab callbacks that can be implemented. Those are Matlab scripts that are called before, during or after the simulation. They are not required like the functions described above but can be useful for things that might be confusing or impossible to do in Simulink, e.g., preparing the desired temperatures or advanced plotting of simulation results.

The Simulink model of the controller is presented in the figure 6.12. Prediction from the first step of the controller is shown in the figure 6.13. The result of control mode after 50 steps are shown in figure 6.14. The results of control mode after 50 steps with added white noise to the output of the building model is captured in the figure 6.15. Standard deviation of the noise was

$$\delta = \sqrt{\frac{P}{T_s}} \doteq 0.4 \tag{6.4}$$

where $P = 100$ is the noise power specified in the Band-Limited White Noise Simulink block, and $T_s = 600$ is the simulation sample time.

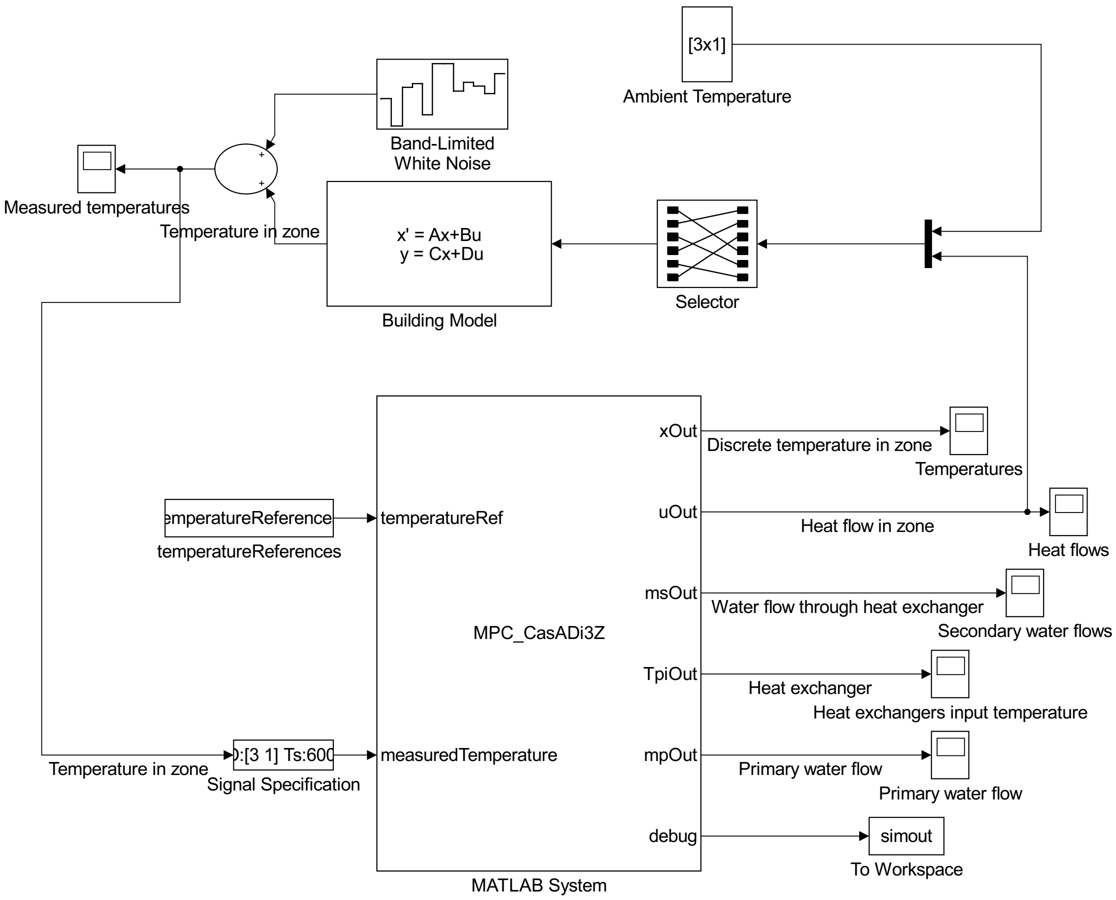


Figure 6.12: Simulink model of the nonlinear MPC controller.

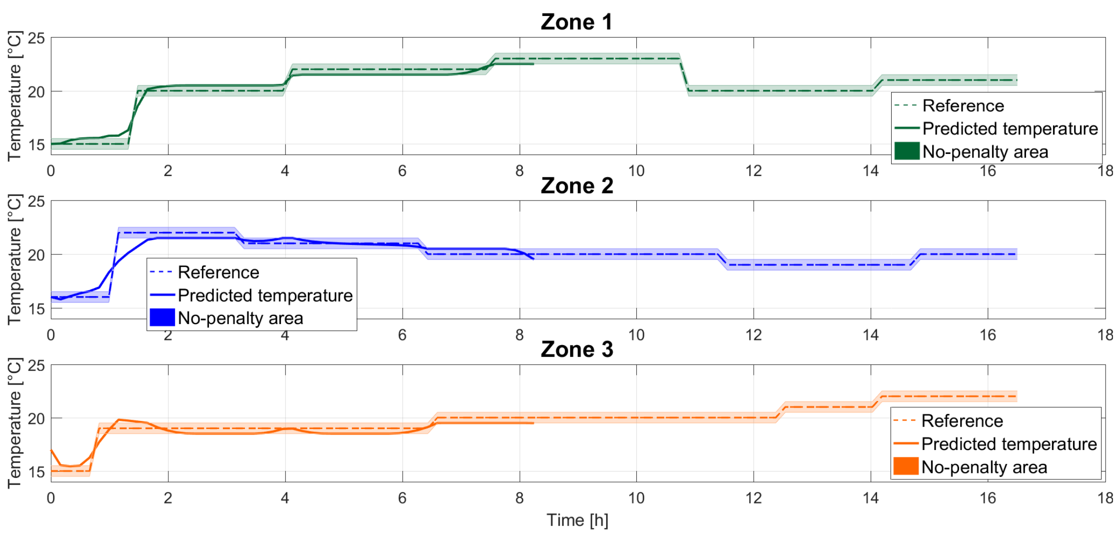


Figure 6.13: Prediction from the first step of the Simulink controller.

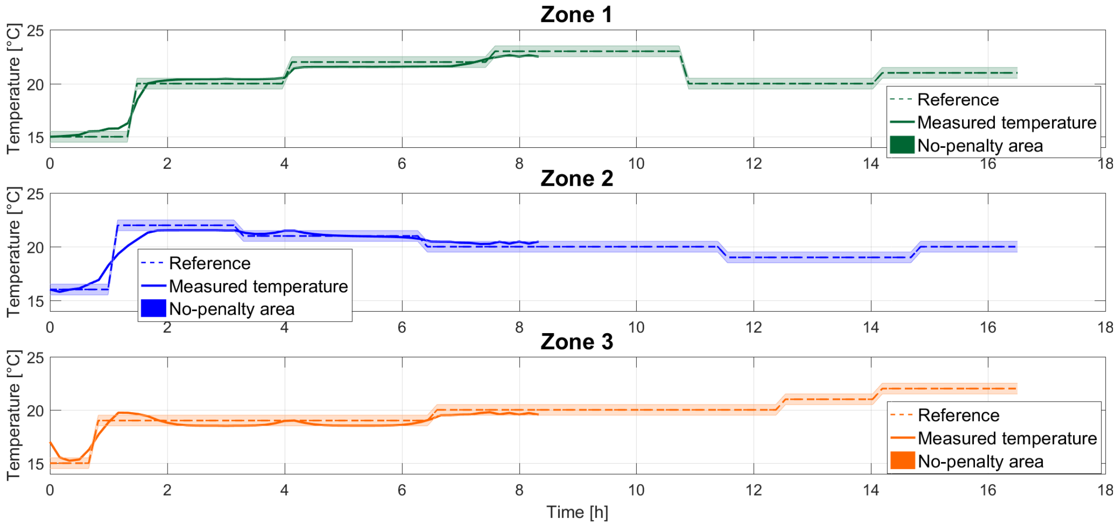


Figure 6.14: Result of control mode after 50 steps.

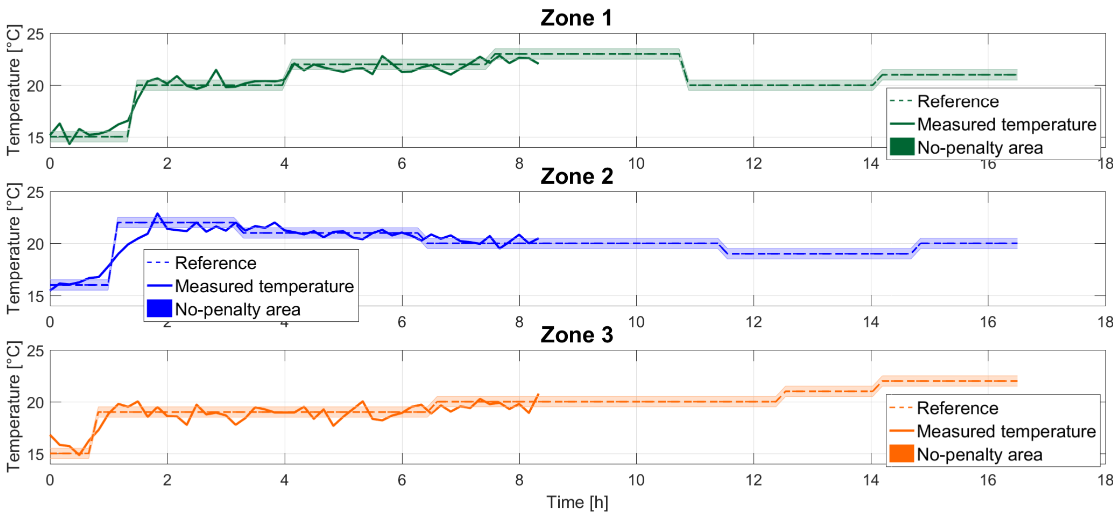


Figure 6.15: Result of control mode experiment after 50 steps with noise added to the measurement of the building model output.

6.4 Verification on a high fidelity building model

The controller was tested on a building model that was mathematically by several orders of magnitude more difficult.

Its walls were modeled by 3 capacities and 4 resistances as opposed to one resistance in the simple model. The heat capacity of furniture was added to the capacity of the air in the room. Another difference was in heat exchangers. Their U-Values were constant (not dependent on water and air flows), but during the simulation they had own dynamics. Therefore, there was a delay between the controller's issued command and its action. A scheme of a zone of such a model is presented in the fig. 6.16.

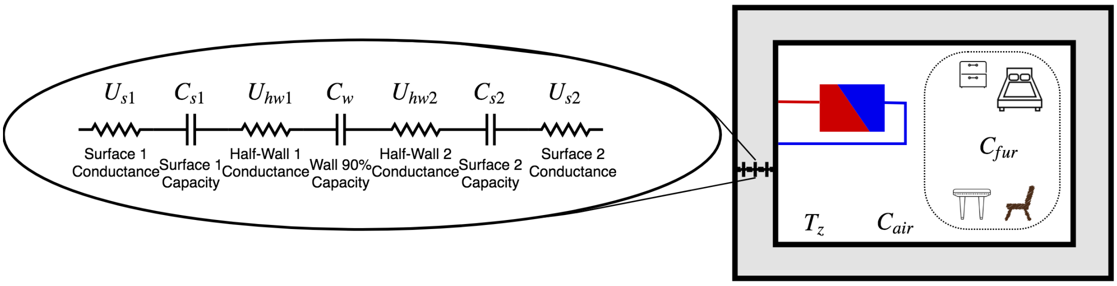


Figure 6.16: Scheme of a zone with several capacities. C_{air} stands for the capacity of air, C_{fur} for the capacity of furniture. U_{s1}, U_{s1} represent conductance of the wall surfaces, C_{s1}, C_{s1} represent their capacities. C_w is the capacity of the majority of the (inner) wall and U_{hw1}, U_{hw2} are conductances of the wall halves.

The model consisted of four rooms of different sizes and capacities. Parameters of the walls also varied. Each radiator was different in size and in thermal transmittance.

Simulink scheme of the thermal building is shown in the figure 6.17. In the left half, the thermal part representing the arrangement of zones and walls is shown. In the right half, the connection of heat exchanger captured. The interconnection between these two parts is done via Simulink Go-To blocks with corresponding colors.

6.4.1 Scenario

The scenario of this experiment was following. The rooms initial temperatures were 15 $°C$. The ambient temperature was kept constant at 14 $°C$. The goal was to immediately heat the temperatures to 20 $°C$ and then heat the kids room to 22 $°C$.

A necessary precondition for a model predictive controller is to have a control model of the controlled object. Simple state space model of the high fidelity building model was identified with the Matlab Identification Toolbox as a system of second order. Suitable fit of a first order was not found. Therefore, the MPC controller was extended to allow thermal state space model of second order in the equality constraints. The additional states (temperature derivatives) were estimated by using default Matlab Kalman filter.

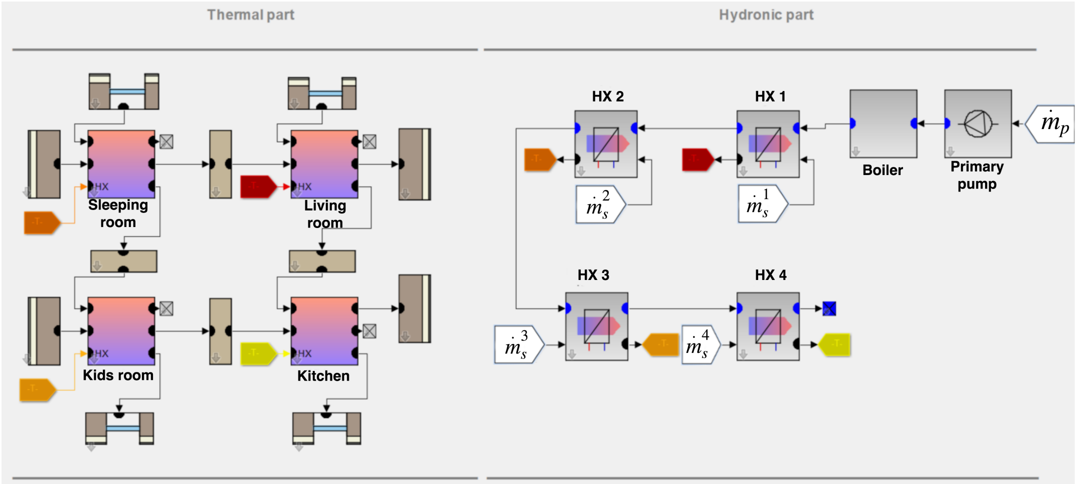


Figure 6.17: Scheme of the 4-zone high fidelity building thermal model. Originally from [27]. Edited. The one pipe hydronic heating network goes from the living room, through the sleeping room, kids room and to the kitchen.

6.4.2 Results

The temperatures of the zones stay in the allowed band most of the time. In fact, they tend to fall to the bottom boundary of the allowed interval because of the controller's objective to minimize power. However, the control has oscillatory characteristic. That's because the model is not properly identified. The evolution of zones temperatures is captured in figure 6.18. The thermal building model used for control has no oscillations. Prediction from the first step is shown in picture 6.19.

Still, the presence of the controller is clearly seen. It preheats all the zones in advance and when the kids room temperature is supposed to rise by two degrees, it supplies majority of hot water to the kids room while the other zones can cool down a bit and still be in the allowed band. Predicted heat flows during the additional kids room heating are captured in the figure 6.20

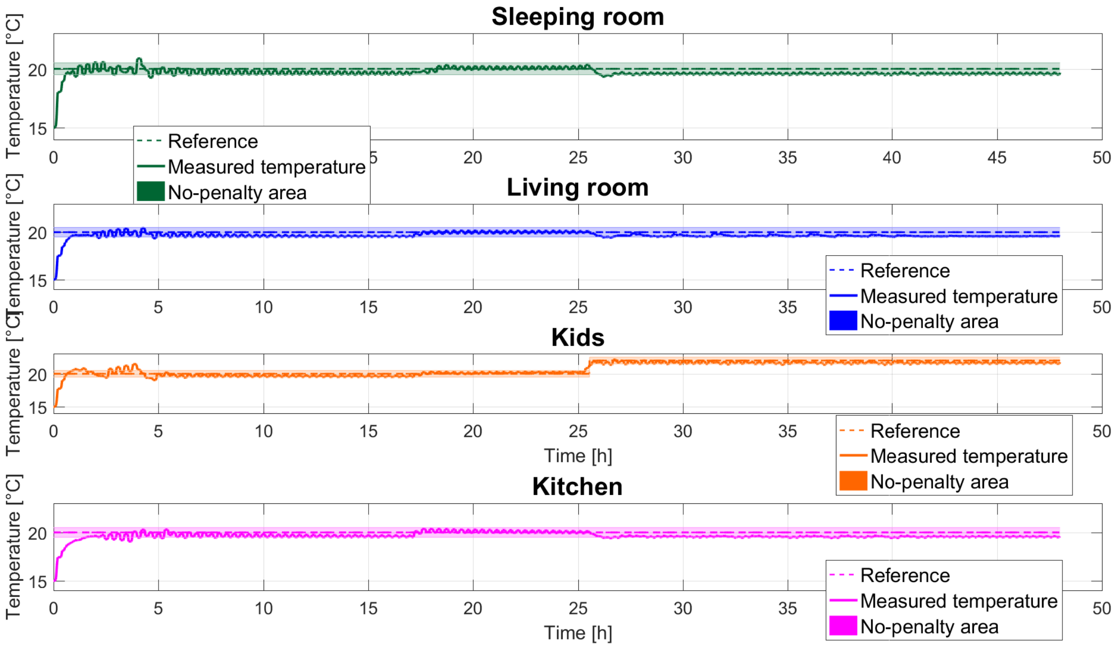


Figure 6.18: Temperatures evolution in the high fidelity thermal building model.

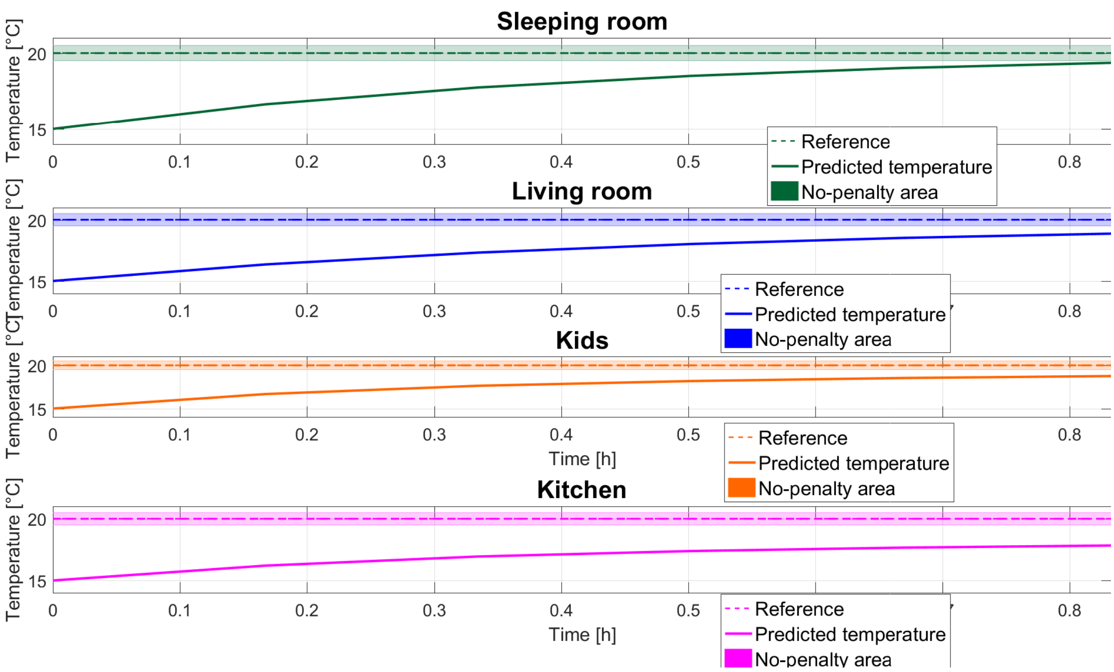


Figure 6.19: Prediction of the zone temperatures from the start of the simulation.

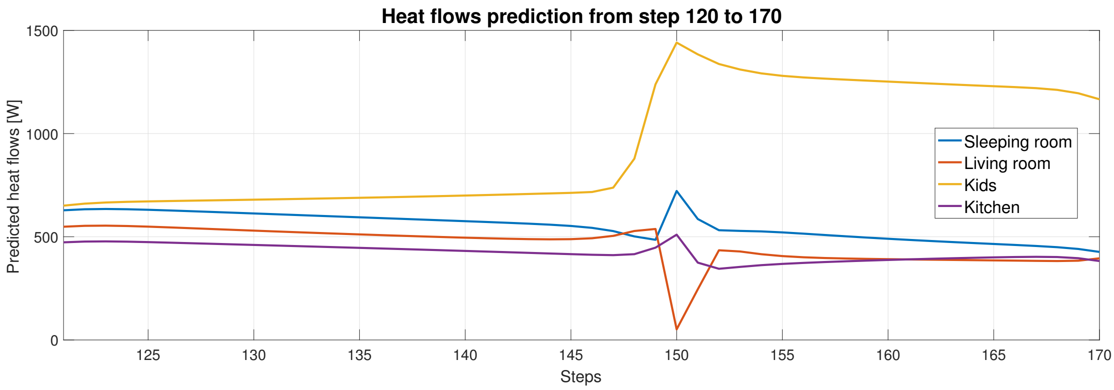


Figure 6.20: Prediction of the heat flows during kids room heating.

Chapter 7
Conclusion

This thesis dealt with problems of control of buildings with a one-pipe water heat delivery systems. Firstly, existing hot water distribution technologies and currently used control approaches were reviewed. Then, the steady-state static optimization tool, that was used as a tool for the network analysis, was successfully designed. A model of a building was constructed and an MPC controller whose results were similar to the optimization tool was implemented. But by using the building model as a constraint representing system's dynamic behavior, the results (inputs and outputs to the model) were computed for the whole prediction horizon and instead of just one step.

The two tools served as a basis for the implementation of the essential model predictive controller whose aim was to compute necessary water flows through the one-pipe network to achieve required temperature levels in the rooms of the building model while minimizing energy consumption and other objectives. That introduced some nonlinearities which made the underlying optimization problem harder to solve and to counter it CasADi framework was presented.

Firstly a predictor that, based on a given state, made a prediction of the states and control actions for future steps was implemented in Matlab. Then, the controller connected to a building state space model in a closed loop was implemented in Simulink. Several experiments were conducted and the results showed some interesting phenomena and advantages of a model-based controller.

In the end, the controller was tested on a high fidelity thermal building model.

7.1 Future work

It's planned that the invented algorithms will run on an embedded device such as Raspberry Pi, which is already used in control testing circuits with heat exchangers. Because of that, it would be useful to explore the possibilities of both Matlab and CasADi code generation, and, if possible, generate code in the C language and integrate the algorithms into the Raspberry Pi. Another thing related to the embedded devices is to research and decide if it's better to make a centralized controller to which all the devices will be connected or develop a distributed controller, that would run (semi-)independently on a device in each secondary circuit.

Concerning the control algorithms, two significant alterations are now in consideration. A minor one is to extend the control (respectively optimization) to the flow of air (which was kept constant in this thesis). A major one is to control directly pumps' and fans' revolutions instead of water and air flows.

Bibliography

[1] Ondřej Zlevor, Jan Předota, and Jiří Dostál. Neznámé čerpadlové otopné sítě a návrhový nástroj pro jednopotrubní variantu. *Vytápění, větrání, instalace*, May 2018. Submitted for publication.

[2] J. Bašta. Regulační armatury - teoretická základna (1). TZB-Info, 2018. https://www.tzb-info.cz/. Accessed: 2018-05-03.

[3] W. Richter, A. Fattah, A. Kremonke, and A. Perschk. Entwicklung und einsatz energiesparender dezentraler heizungspumpen - systemanalyse der heizungsanlage - schlussbericht. Technical report, Dresden, 2005.

[4] Franco Normani. Heat exchanger. https://www.real-world-physics-problems.com/heat-exchanger.html. Accessed: 2018-04-15.

[5] Awad B.S. Alquaity, Salem A. Al-Dini, and Syed M. Zubair. Effectiveness - ntu relations for parallel flow heat exchangers: The effect of kinetic energy variation and heat leak from outside. *International Journal of Refrigeration*, 36:1557–1569, August 2013.

[6] National Programme on Technology Enhanced Learning. Lecoture on heat and mass transfer. https://www.calpoly.edu/~kshollen/ME350/Handouts/Heat_Exchangers.pdf. Accessed: 2018-03-22.

[7] A. Pignotti and R. K. Shah. Effectiveness-number of transfer units relationships for heat exchanger complex flow arrangements. *International Journal of Heat and Mass Transfer*, 35:1275–1291, May 1992.

[8] Magic Heat Reclaimer. http://www.magicheatreclaimer.com/magic-heat-reclaimers/, 2013. Accessed: 2018-04-16.

[9] Wikipedia:Mathematical optimization. https://en.wikipedia.org/wiki/Mathematical_optimization, April 2018. Accessed: 2018-04-16.

[10] ISO 7-1. *Pipe threads where pressure-tight joints are made on the threads*. The Engineering ToolBox, 1994.

[11] Glen and Brown. The darcy-weisbach equation. Available at: https://bae.okstate.edu/faculty-sites/Darcy/DarcyWeisbach/Darcy-WeisbachEq.htm, 2000. Accessed: 2018-04-19.

[12] Cyril Frank Colebrook. Turbulent flow in pipes with particular reference to the transition region between the smooth and rough pipe laws. *Journal of the Institution of Civil Engineers*, 12:393–422, October 1939.

[13] Jiří Dostál, Ivo Herman, Štefan Knotek, Jiří Kubeš, Tomáš Bäumelt, Jan Hauser, Jiří Valtr, and Ondřej Zlevor. Stability study of hydronic network control. Research report, Czech Technical University in Prague, 2016.

[14] Steve Wilson. Pump affinity laws for centrifugal pumps. Grundfos. `https://usdemo01.grundfos.com/content/dam/GPU/training/Pump%20Affinity%20Laws%20for%20Centrifugal%20Pumps.pdf` Accessed: 2018-03-28.

[15] Jiří Dostál. Decentralized control of hydronic building systems. Doctoral minimum, Czech Technical University in Prague, Department of Control Engineering, Prague, 2015.

[16] Grundfos Research and Technology. The centrifugal pump. `https://dk.grundfos.com/content/dam/Global%20Site/Industries%20&%20solutions/Industry/pdf/The_Centrifugal_Pump.pdf`. Accessed: 2018-04-20.

[17] Florian A. Potra and Stephen J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1):281 – 302, 2000. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[18] Inc. The MathWorks. Constrained nonlinear optimization algorithms: fmincon interior point algorithm. `https://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html`. Accessed: 2018-04-15.

[19] Tomáš Bäumelt. *Distributed Building Identification.* MSc thesis, Czech Technical University in Prague, Prague, May 2016.

[20] Jan Hauser. *Distributed Predictive Control of Buildings.* MSc thesis, Czech Technical University in Prague, Prague, May 2017.

[21] Inc. Gurobi Optimization. Gurobi optimizer reference manual. `http://www.gurobi.com`, 2016. Accessed: 2018-04-21.

[22] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[23] Inc. Gurobi Optimization. Gurobi 7.5 performance benchmarks. `http://www.gurobi.com/pdfs/benchmarks.pdf`, 2017. Accessed: 2018-04-21.

[24] Joel Andersson. *A General-Purpose Software Framework for Dynamic Optimization.* PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.

[25] Joris Gillis and Joel Andersson. Eucco. In *CasADi 3volutions.* Katholic University in Leuven, 2016. `https%3A%2F%2Fgithub.com%2Fcasadi%2Fpromo%2Freleases%2Fdownload%2Ftalks%2Fcasadi_3volutions.pdf`. Accessed: 2017-04-20.

[26] Inc. The MathWorks. Matlab system block. https://www.mathworks.com/help/simulink/ug/what-is-matlab-system-block.html, 2018. Accessed: 2018-05-05.

[27] Jiří Dostál. Hydronic heating library 3.0. Technical report, Honeywell, spol. s r.o., 2018.

Appendix A
System object

Implementation of the System object is similar to implementing a standard Matlab class. However, some additional rules and system-oriented requirements have to be met, namely:

- **Empty constructor.** It has to be possible to create an instance of the class by calling the default constructor (without arguments).
- **setupImpl.** The function is called once at the start of the simulation. It's worth it to compute constant and other one-time computations.
- **stepImpl.** This is the core function, it is called every step and during this function the output is computed based on the input.
- **getOutputSizeImpl.** Specifies output size for each output port.
- **getOutputDataTypeImpl.** Specifies type for each output port.
- **isOutputComplexImpl.** For each output port specifies if there are real or complex numbers at the output.
- **isOutputFixedSizeImpl.** Specifies if size of each output port stays fixed or changes during simulation.

Appendix B

Content of attached CD

The CD attached to this thesis contains a pdf version of the thesis.