Control of heating in a low energy single-family house

10$^{\text{th}}$ semester project

Christian Mølgaard Nielsen
Kristian Helmer Kjær Larsen

AALBORG UNIVERSITY
MASTER THESIS

Aalborg University
Control and Automation

Department of Automation and Control
Aalborg University
http://www.aau.dk/

AALBORG UNIVERSITY
MASTER THESIS

Title:
Control of heating in a low energy single-family house

Project Period:
Spring 2022

Project Group:
1032

Participant(s):
Christian Mølgaard Nielsen
Kristian Helmer Kjær Larsen

Supervisor(s):
Jan Dimon Bendtsen
Simon Thorsteinsson

Copies:

Page Numbers: 133

Date of Completion:
June 1, 2022

Abstract:

This master thesis deals with designing a control scheme for heating of a low energy single-family house in connection with the OPSYS 2.0. The house is heated using hydronic underfloor heating circuits supplied with hot water from a partially solar powered heat pump. A two-level hierarchical control scheme has been devised. The upper level calculates a heating budget based on average house dynamics and forecasts of energy prices, power production of the house's photovoltaic panels, and the weather. The lower level determines how to distribute this heating budget to the different rooms using a lumped parameter multi-zone model of the house and forecasts of the weather. To facilitate the design, discrete time, non-linear, grey-box models of the system has been developed for each of the two levels. Using these models, state observers have been designed in the form of Kalman filters. Both levels contain a model predictive controller, that have been developed by reformulating the models into the mixed logical dynamical framework, which has allowed the two control problems to be formulated as mixed integer quadratic programming problems that are more efficiently solved. The control scheme has been shown to achieve disturbance rejection and stabilisation of the system, by testing it using a high-fidelity model of the house supplied by the OPSYS 2.0 participants.

Foreword

This master thesis has been written during the spring of 2022 by two master students on the master Control and Automation at the Technical Faculty of IT and Design at Aalborg University.

It is based on a project proposal in connection with the ongoing OPSYS 2.0 project concerning the control of heat pumps in a BK2020 house. Further information will be given in Chapter 1. The OPSYS 2.0 project's participants are:

- Teknologisk Institut:
  - Ivan Katic
  - Søren Østergaard Jensen
- NEOGRID TECHNOLOGIES ApS
  - Henrik Lund Stærmose
  - Alex Arash Sand Kalaee
  - Pierre Vogler-Finck
- NORDISK WAVIN A/S
  - Søren Dueholm
- ROBERT BOSCH A/S
  - Brian Nielsen
- Aalborg Universitet
  - Jan Dimon Bendtsen
  - Simon Thorsteinsson

Throughout this master thesis, technical information concerning simulation models and experimental setups has been given by Simon Thorsteinsson. When this information is used in the report, a reference to [1] is made.

Appendix A describes various state estimation algorithms that are not actually used in the main report, in somewhat greater detail than what would normally be expected from a master thesis in the field of Control Engineering. This is deliberately included to fulfil certain unique curriculum requirements for this particular project, and should not be considered necessary for comprehension of the main text. Nonetheless, some readers may find it interesting as supplementary information.

This report was written in LaTeX and has been shared online in-between the group members using Overleaf licensed to all students at Aalborg University.

Notation

Citations are marked in square-brackets with a number corresponding to a source in the bibliography and a potential page number, e.g. [1, p. 2].

The report contains figures, tables, and equations which are numbered individually and dependent on the chapter. Captions for figures are positioned below and captions for tables are positioned above.

Multiplication is noted by a space between two variables, or a variable and a number, e.g. $a\,b$ or $2\,a$. Multiplication between two numbers are written with a dot, e.g. $2 \cdot 3$.

Throughout the report, decimal separators will be noted with a dot ".", e.g. 100 mV corresponds to 0.1 V.

Leibniz' notation for derivatives is used, e.g. $\frac{\mathrm{d}y}{\mathrm{d}x}$. The derivative with respect to time is often noted by Newton's notation, e.g. $\dot{x}$.

Variables are written in italic and units in roman. Variable subscripts are written in roman. Subscript indexing is written in italic. For example $T_{\mathrm{r},i,j}$ refers to the j^{th} room on the i^{th} floor.

Contents

Part I

Preliminary study

1 Introduction

This chapter contains an analysis and description of the factors and background needed for the formulation of the problem statement for this project. It contains the following:

- Section 1.1 describes the motivation for reducing required energy for heating of residential homes.
- Section 1.2 gives an introduction to to zero emission buildings and the Danish BK2020 building class.
- Section 1.3 presents the OPSYS 2.0 project and introduces the problem of heating houses.
- Section 1.5 covers the problem statement.
- Section 1.6 describes the developed control strategy in an attempt to motivate and give an overview of the following chapters.

1.1 Motivation

Climate change is a global concern caused by the increasing Carbon dioxide (CO_2) emissions from fossil fuels. In spite of the 2020 decline caused by the global pandemic, CO_2 emissions are yet again increasing [2, p. 11]. The European Union (EU) commission considers climate change "the defining challenge of our time" [3] and plan to fight it through reduced energy consumption among other actions. Moreover, due to increasing energy prices [4], reducing energy consumption is also advantageous for the consumer.

In light of the recent conflict in Ukraine, the EU commission is advocating an accelerated transition to clean energy, due increased energy prices caused by the conflict and reliance on "unreliable suppliers" [5].

One of the largest consumers of energy world wide is the residential building sector, taking up approximately 21% globally in 2019 [6]. Therefore, focusing on reducing the energy consumption of residential buildings is an important part of battling climate change.

Approximately 63.6% of energy consumed in residential buildings in the EU in 2019 was used for space heating [7], thus a lot of effort has been made to reduce the required energy for space heating of residential homes as well as using energy from renewable sources.

In the following section, an EU commission defined highly energy efficient building type.

1.2 Zero emission buildings

Nearly Zero-Emission Building (NZEB) have been defined by the EU commission as buildings that have a very high energy performance. The energy required should be covered by renewable sources from 2020. Furthermore, the European commission has proposed a

Zero-Emission Building (ZEB) requirement starting by 2030. The definition of the ZEB requirement is similar to the NZEB definition, but stricter in terms of efficiency [8].

1.2.1 BK2020

In Denmark, a version of NZEB building class, is the BK2020 Building class [9]. This building class has been introduced as a voluntary set of low energy requirements for new buildings. It focuses especially on stricter requirements for ventilation and energy efficiency. In [9] it is stated that the requirement for total energy usage including heating, ventilation, cooling and hot water usage per m^2 has to be less than 20 kWh/m^2 per year. Furthermore, it specifies a set of requirements for the energy balance of the windows, doors and how airtight the house is.

Therefore, research is being made on managing the heating of these houses in the most efficient manner.

This project is based on a project proposal by Simon Thorsteinsson in connection with his PhD work on the OPSYS 2.0 project concerning the control of heat pumps in a BK2020 house [1].

1.3 The OPSYS 2.0 project

In 2019 the OPSYS 2.0 project started. It focuses on increasing the efficiency of both existing and new heat pump installations by developing new control methods for optimising the forward temperature from the heat pump and the flow rate through the heat emitting system. Based on this, the project is focusing on developing a control system capable of creating flexibility services for stabilisation of the electricity grid and optimising the self-consumption of PV generated electricity in private houses.

The project is a joint programme between Danish Technological Institute, Aalborg University, Neogrid Technologies, Nordisk Wavin A/S and Robert Bosch A/S [10].

1.3.1 House components

A BK2020 house has been built in 2016. It is used for surveying different methods for managing heating of BK2020 houses in the OPSYS 2.0 project [10]. Figure 1.1 shows a sketch of the house with the main components relevant for this project. A more detailed floor plan can be found in Figure 1.2.

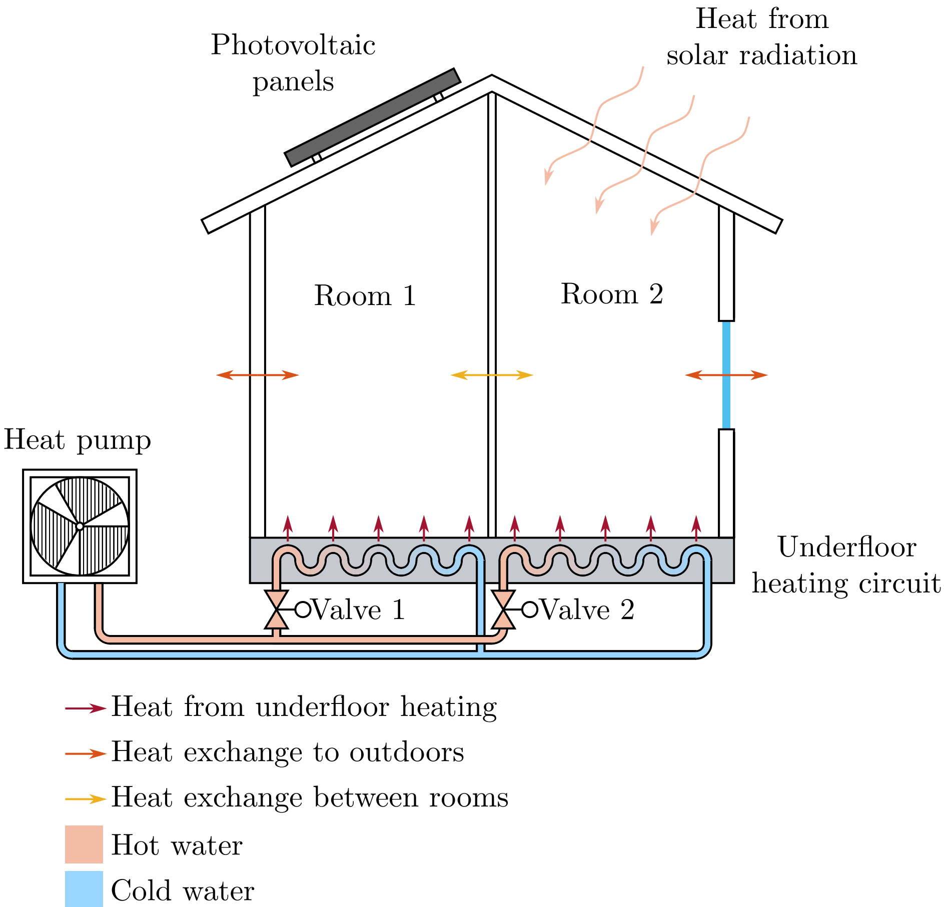

Figure 1.1: Simple sketch of the main components of a two room house and the different heat flows.

The house is equipped with an electrical Air to Water Heat Pump (AWHP) connected to hydronic underfloor heating throughout the house.

A heat pump uses energy, in this case electric power, to transfer heat from a cold reservoir, the outdoors, to a hot reservoir, the water in the hydronic underfloor heating circuit. This is generally more efficient than directly using the electric energy for heating through Joule heating. A more detailed description of the working principles of a heat pump is given in Section 4.3.

Hydronic underfloor heating circuits uses water as a transport medium for distributing heat throughout a building. Pipes are laid in the floor of each room, thus allowing heat exchange between the water in the pipes and the floor, which will then radiate the heat to the air and inhabitants of the room above. Valves are used to control which rooms receive hot water and therefore heat. The house also has a mechanical ventilation system and a Photovoltaic (PV) system with 35.8 m^2 panels on the south facing roof [11].

PV panels generate electricity based on solar radiation and is used to offset some of the electric energy requirement of the house, thus lowering the electric energy that needs to be bought from the electrical grid.

1.3.2 Dymola model

A Dymola model of the house has been made as a part of the OPSYS 2.0 project. The model has been made as a digital twin, to allow for fast control method experimentation without interfering with the inhabitants of the real house. Figure 1.2 shows an approximate floor plan of the house. The real floor plan is not shown for privacy reasons of the inhabitants of the real house.

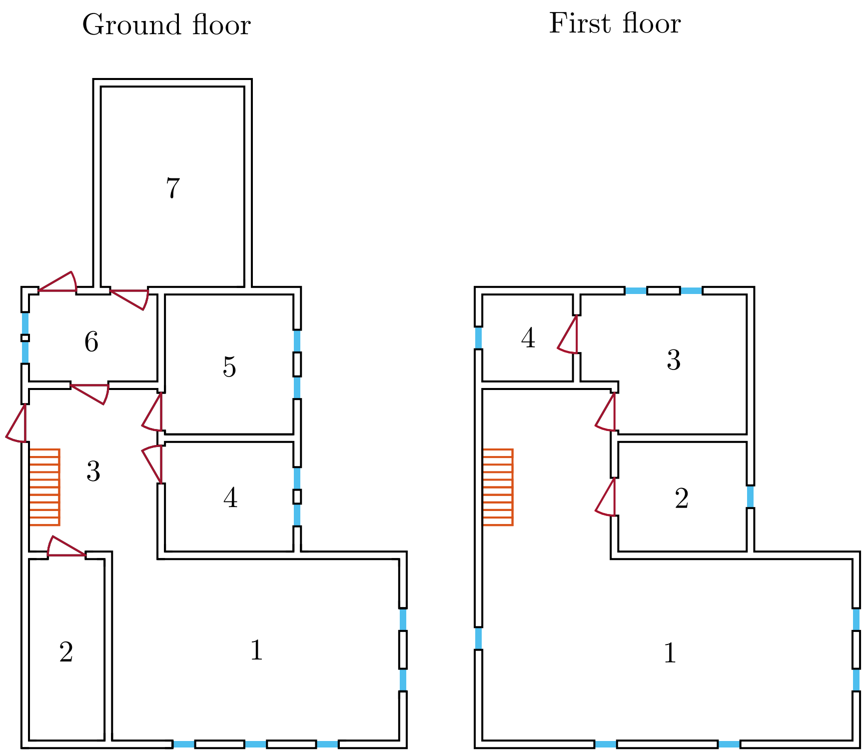

Figure 1.2: Floor plan of the house showing both the ground floor and the first floor.

The features of the Dymola model of the house are described in [11]. The house is a single family house with two floors and a total heated gross area of 230 m^2. The ground floor is 129 m^2 and the first floor is 101 m^2.

It has 7 rooms on the ground floor and 4 rooms on the first floor. All rooms have windows, except for room 7 on the ground floor. The house is heated by an air-to-water heat pump that is connected to a Thermal Energy Storage (TES), which is then connected to underfloor heating in each room. The TES unit is not included in the Dymola model. The consequence of this omission can be found in Subsection 11.1.1.

The underfloor heating in each room is controlled by an ON/OFF valve corresponding each room. All rooms except room 1 on the first floor and room 1 on the ground floor have 1 valve. These two rooms have 3 and 2 underfloor heating circuits respectively.

Lastly, there is a PV system on the roof that can be used during daytime in order to reduce the amount of power bought from the grid. The PV is also used for powering the ventilation system during daytime. The ventilation system has a heat recovery system for some of the rooms, which is used during the heating season.

The Dymola model has been written in the coding language Modelica, which is an object oriented language made for modelling multi-domain systems [12].

The development of the Dymola model of the house has been a collaborative effort by:

- Simon Thorsteinsson
- Søren Østergaard Jensen
- Kasper Vinther
- Ivan Katic

The model is exported from Dymola as a Functional Mock-up Unit (FMU), which can be interfaced through the Functional Mock-up Interface (FMI) PyFMI [13] by a simulation environment written in Python by Simon Thorsteinsson [1]. The general setup is seen in Figure 1.3.

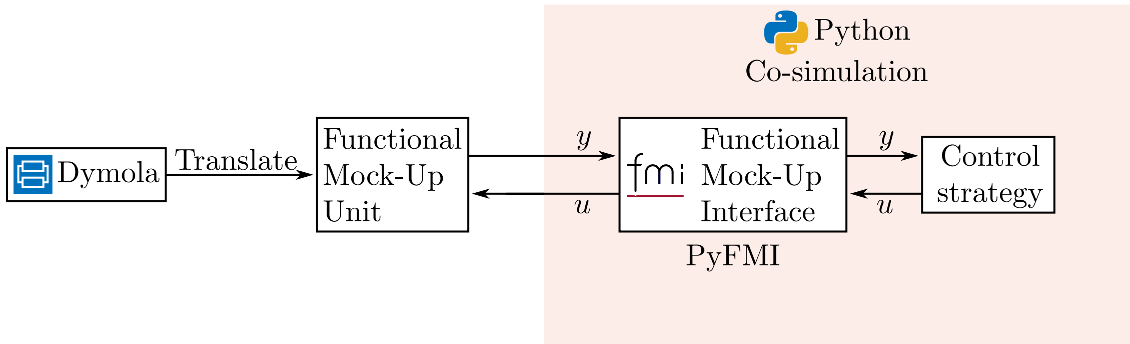

Figure 1.3: Simplified diagram of the simulation environment used in this project.

As the Dymola model has been designed with the purpose of being a digital twin of the real house, the complexity of it is very high. This means that the model is fit for validating control designs, but a simplified model has to be made for designing a control system for the house.

1.3.3 Energy prices

Based on the OPSYS 2.0 project focus it has been chosen that this report will focus on minimising the amount of money spent on heating based on electricity prices. This choice is based on the fact that the electricity prices in Denmark are constantly fluctuating based on supply and demand. Figure 1.4 shows the fluctuation of the raw spot electricity price in Denmark over a day [14].

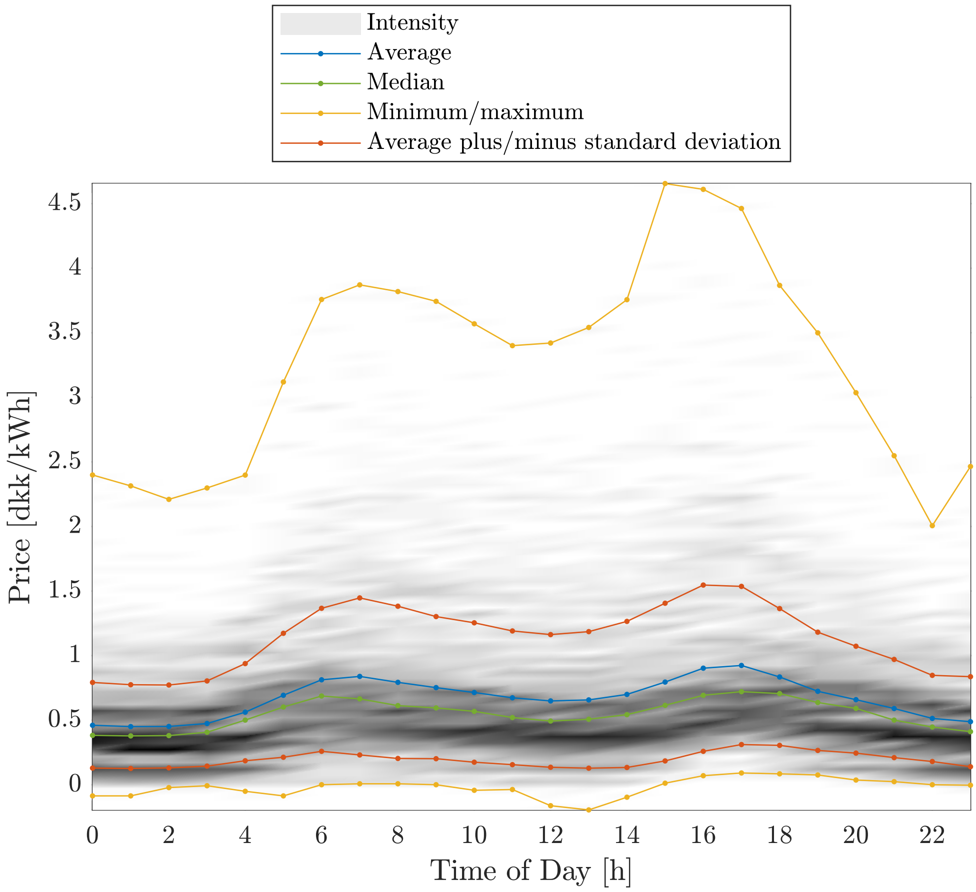

Figure 1.4: Statistical data of the raw spot electricity price in Denmark over 24 hours using 423 days from the 01-01-2021 to 27-02-2022.

From Figure 1.4 it can be seen that the prices fluctuates both from day to day, and over the course of 24 hours. In general there is a pattern in peak hours being around 8:00 and 18:00. The price is lower during the night.

This means that in order to save money on heating with a heat pump it is important to take the varying electricity price into account when designing a control system for heating of the house with respect to economic flexibility.

It has to be noted that the actual price for electricity, paid by the costumer, is a bit different as it depends on various other factors such as as taxes, how much power is consumed etc.

1.3.4 Thermal comfort

Since the goal is to minimise the amount of money spent on heating, it is important to define a requirement for the thermal comfort of the occupants of the building. Without this, the optimal solution would simply be turning off the heating, which would be undesirable for the occupants.

No rules are present in Denmark concerning temperatures in residential buildings. [15] however states average temperatures for different room types in residential buildings:

- 21°C in living rooms and kitchens.

- 20.5°C in children's bedrooms.
- 19°C in bedrooms.

Furthermore, [15] also states an optimal temperature range is 20-22°C, but 18-25°C should not be problematic. Temperature differences greater than 5°C between adjacent rooms is undesirable due to possibility of water condensation leading to mold growth.

The Danish Working Environment Authority further states that temperature changes of more than 4°C during a workday is uncomfortable [16].

Furthermore, the maximum floor heating temperature is 29 °C for European standards and the minimum is 19 °C [17]. The water in the underfloor heating has a maximum permissible temperature of 35°C as suggest by [18]. Moreover, the heat pump can only deliver water with a temperature of more 20 °C or hotter [19].

1.4 Background

This section contains an overview of studies made in the field of control strategies for heating buildings with respect to minimising heating costs.

According to [20] the two main categories of control strategies for heating buildings are Rule-Based Control (RBC) and Model Predictive Control (MPC).

The RBC's are simple heuristic methods, that generally have the form "if(condition is verified), then(action is triggered)". In [20] the RBC for flexibility objectives can be split up into three categories.

1. Control strategies focusing on load shifting with fixed scheduling, where the controller tries to avoid or force the operation of the heat pump during fixed hours.
2. Control strategies relying on the variations of energy price in time with the objective of reducing the energy costs for the end users.
3. Control strategies focusing on improving the consumption of renewable energy sources.

Common traits for all of the RBC strategies are firstly that they depend a lot on choice of the thresholds values placed on trigger parameters. As the trigger parameters or threshold parameters are normally fixed, it is difficult for RBC to adapt to changing external conditions. Secondly it is difficult to evaluate the performance of RBCs, as the usually do not have explicit cost defined a priori.

MPC strategies, on the other hand, are optimisation based methods where an explicit cost is defined a priori. The main idea is to construct a cost function that takes flexibility objectives into account. These could be consumption, cost of heating, renewable energy consumption etc.

According to [20] Economic Model Predictive Control (EMPC) strategies are a promising way to define the MPC cost function. Some of the studies with similar setups, to the OPSYS 2.0 project, claim to reduce energy costs by 13-40% depending on the building type [21] [22].

EMPC's have promising results for some building types and heat source setups, but most studies on only heat pump heated buildings are simulation based [20].

Furthermore, MPC has challenges concerning the implementation of the controller. One of the biggest is obtaining a satisfactory building model, as it can be costly and a complicated process [20]. Lastly awareness of the trade-off between model accuracy and simplicity is important. This is mainly because the computational complexity of MPC implementation quickly becomes impractical as the model becomes more complex [20]. Thus an important performance measurement for an implemented MPC is the computation time, which will be taken into consideration for the design of the model used in the control strategy designed in this project.

Following the analysis of the problem at hand and general solutions to similar problems, the problem statement is formulated in the following section.

1.5 Problem statement

In Section 1.1 it has been argued that buildings should be more energy efficient. Furthermore, in Section 1.2 the new BK2020 building class has been introduced as one of the voluntary initiatives to achieve better energy efficiency in new buildings in Denmark. The OPSYS 2.0 project has been introduced in Section 1.3, which is a project on making intelligent heating for buildings heated by heat pumps and powered PV panels. In Section 1.4 the state of the art in RBC and MPC has been presented. Moreover, requirements for comfort of building occupants has been set. Lastly some of most important trade-offs has been discussed for the design of a proposed MPC for different flexibility objectives have been made. This leads to the following problem statement:

"How can a control system be designed that minimises the energy cost, while maintaining comfort levels for inhabitants in the house."

In the following section, an overview of the solution strategy will be given.

1.6 Control strategy

This section contains a general description of the control strategy in an attempt to give the reader an overview and motivate the following chapters.

This master thesis does not use any experimental setups or data. Instead, as stated in Section 1.3, a model of the system developed in Dymola has been made available from the start of the project. This model is used instead, both for generating data for parameter estimation and for co-simulation, where it acts as a stand in for the plant of the system, during testing of the control strategy developed in this master thesis. The control strategy in developed in this project has the following goals:

- Set points:
  - $T_{\mathrm{r,ref}}$: Individual reference temperatures for each room in the house decided by the user.
- Control outputs:
  - $T_{\mathrm{out,ref}}$: Reference value for the outflow of the heat pump.
  - v: Individual valve positions for the underfloor heating circuit in the house.
- Maintain comfort levels, i.e. keep the room temperatures T_{r} close to their reference $T_{\mathrm{r,ref}}$.
- Take weather forecast data in the form of outdoor temperatures T_{a} and solar intensity I_{s} into account.
- Reduce cost for the consumer by taking forecasts for the spot price of electricity p, and the power generated by the Photovoltaic (PV) panels P_{pv} into account.

Since it is desired to utilise forecasts for weather, energy prices, and PV power, it has been decided to use Model Predictive Control (MPC). In an MPC optimum control outputs are calculated based on a cost function over a prediction horizon. Thus, forecast data can be used in the prediction of a MPC, while for example a PID controller is only capable of reacting to the changes as they happen. MPC strategies requires state observers if not all model states are measured. Thus, Kalman Filters are used for online estimation of model states.

Due to the complexity of the system it has been decided to design a hierarchical control scheme consisting of two levels as seen on Figure 1.5.

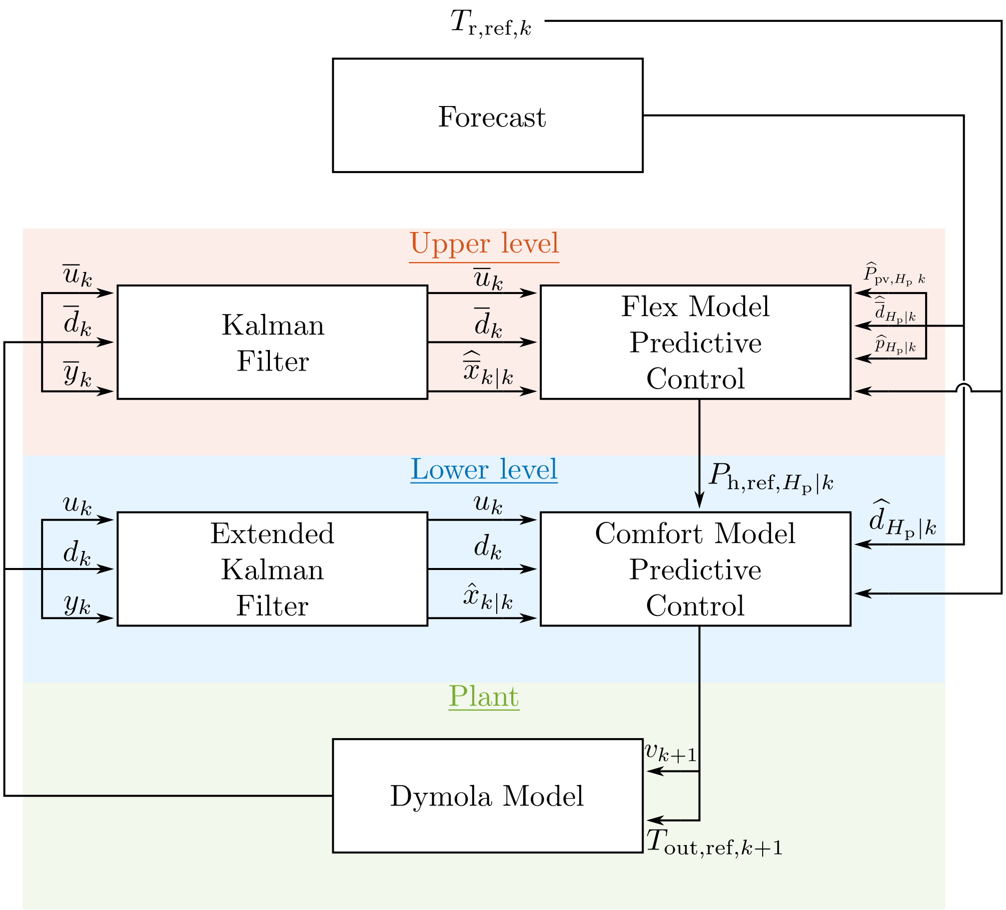

Figure 1.5: Simplified block diagram of how the different control modules are connected with each other, forecast data, and the Dymola model described in Subsection 1.3.2.

The additional variables on Figure 1.5 are as follows:

y	:	Outputs
x	:	States
u	:	Inputs
d	:	Disturbances from the weather

An overline, e.g. $\overline{y}$ signifies the variable is connected to the upper layer. A hat, e.g. $\hat{x}$ means that it is a estimate or prediction of the variable. Subscript $_k$ means that it is the value of the variable at sample k. Subscript for a variable like $\hat{x}_{k|k}$ means that it is the estimate of the value at time k given information until time k. $\widehat{p}_{H_\mathrm{p}|k}$ means that it is the estimate for the prediction horizon H_p given data up until time k. The layers can be summarised as follows:

- The upper level:
  - Goal: Determine a heating power budget based on the avarage house dynamics and forecast data of weather, electricity prices, and power generation of the PV panels.
  - Sample time: 60 minutes.

  - Prediction horizon: 24 hours.

- The lower level:
  - Goal: Maintain individual room temperatures while following the power budget and taking weather forecasts into account and distribute the heating budget such that discomfort is mitigated.
  - Sample time: 15 minutes.
  - Prediction horizon: 1 hour.

Due to the two different control layers with different goals and sample times, two different dynamic models are needed for the control design.

The following chapters describes first the development of the models followed by the design of the control strategy and are organised as follows:

- Part II Modelling
  - Chapter 2 gives an overview of the different submodels and how the data for parameter estimation is generated.
  - Chapter 3 describes the lower layer's model and parameter estimation.
  - Chapter 4 describes the upper layer's model and parameter estimation.
- Part III Control design
  - Chapter 5 describes in detail how the different modules of the control strategy interact.
  - Chapter 6 concerns the Extended Kalman Filter for the lower layer.
  - Chapter 7 describes the Kalman Filter for the upper layer.
  - Chapter 8 describes the comfort MPC for the lower layer.
  - Chapter 9 describes the flex MPC for the upper layer.

Part II

Modelling

2 Modelling overview

This chapter gives an overview over the developed models used for observer and controller design in later chapters. It contains the following sections:

- Section 2.1 gives a description of the motivation for the derivation and development of the models.
- Section 2.2 describes the different submodels the models are parted into and their relations.
- Section 2.3 describes the simulation setup used for generating data for the parameter estimation of the models.

2.1 Motivation

As stated in Section 1.3 a model of the system developed in Dymola has been made available from the start of the project. This model is a high fidelity, first-principle model consisting of approximately 22500 equations. The high complexity results in the model not lending itself well to controller design, as for example real-time Model Predictive Control (MPC) becomes intractable.

Instead, it will be used for generating data to produce two simpler models suitable for observer and control design in Chapter 6 to 9. The following section describes how the system is parted up in two main models to allow for development of submodels in the next chapters.

2.2 Overview

Figure 2.1 shows a sketch of the major components of the house models, their most relevant states, inputs and outputs. In the following chapters, the subscripts i and j refer to the floor and room respectively.

Due to the complexity of the real system, the modelling process has been split up into two 2 models, one for each of the control scheme's levels, to allow for an easier modelling process. Figure 2.2 shows an overview of how the different submodels of the two models interact.

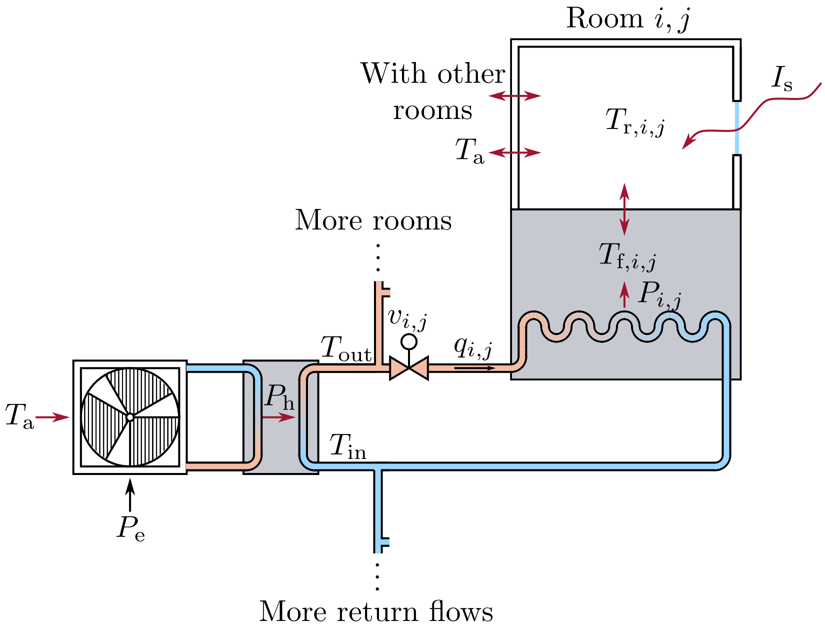

Figure 2.1: Sketch of major elements of the model, states, and inputs and outputs of the different submodels. Heat flows are shown with red arrows.

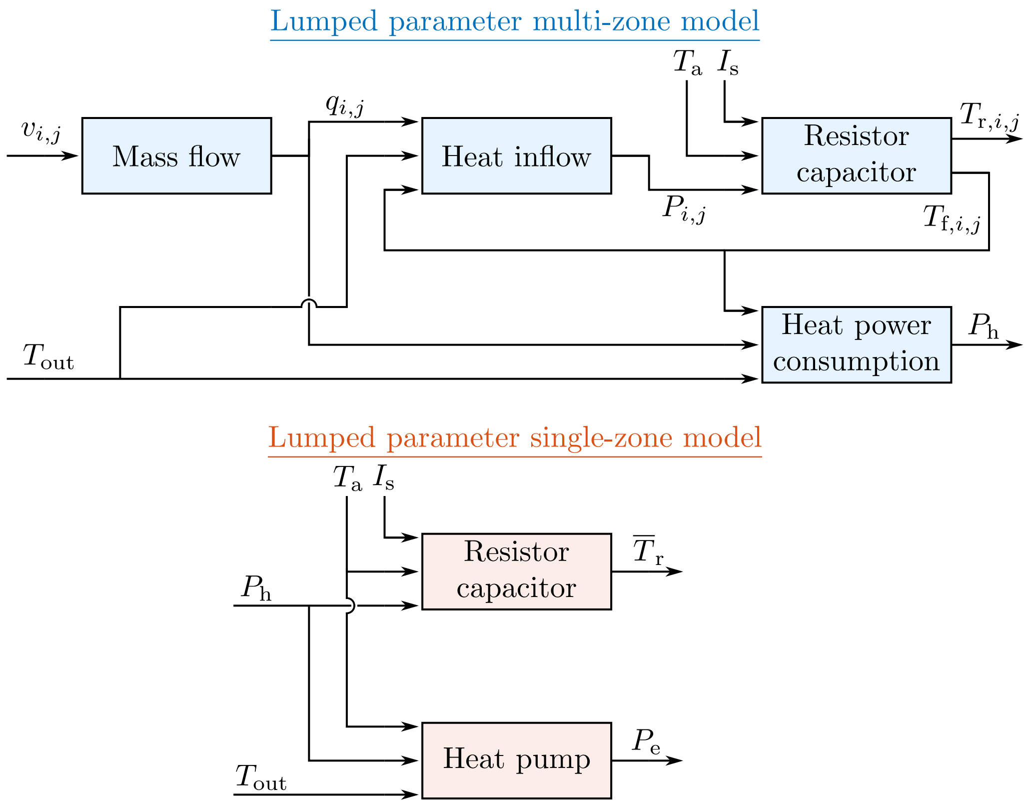

Figure 2.2: Block diagram of the how the different submodels of are connected. In blue are submodels used for the lumped parameter multi-zone model and in orange are submodels used for the lumped parameter single-zone model.

The submodels under lumped parameter multi-zone model are used for modelling the individual room temperatures and the heat power consumption of the house. The submodels for the lumped parameter multi-zone model are developed in Chapter 3.

1. A simple model of how the valve position controls the mass flows in the underwater heating circuits. This is found in Section 3.1.
   - Inputs: Valve positions $v_{i,j}$
   - Outputs: Mass flow of each underwater heating circuit $q_{i,j}$
2. A resistor capacitor equivalent model of the thermal characteristics of the different rooms and walls. This is found in Section 3.2.
   - Inputs: Rate of heat flows from underfloor heating $P_{i,j}$, solar intensity I_{s}, and ambient temperature T_{a}.
   - Outputs: Room temperatures $T_{\mathrm{r},i,j}$ and floor slab temperatures $T_{\mathrm{f},i,j}$.
3. An advection model for the heat transfer from the water to the underfloor heating. This is found in Section 3.3.
   - Inputs: Outflow temperature from the heat pump T_{out}, mass flows in each underfloor heating circuit $q_{i,j}$, and the floor temperature $T_{\mathrm{f},i,j}$.
   - Outputs: Rate of heat flows from underfloor heating $P_{i,j}$.
4. A mixing model, that calculates the return temperature from the underfloor heating which is sent back to the heat pump and calculates the heating energy consumption of the house. This submodel can be found in Section 3.5.
   - Inputs: Mass flows in each underfloor heating circuit $q_{i,j}$, outflow temperature of the heat pump and the floor temperature $T_{\mathrm{f},i,j}$.
   - Outputs: Heating power supplied by the heat pump P_{h}.

The submodels under lumped parameter single-zone model are used for modelling the heat power consumption of the house together with heat pump dynamics in order to determine a heating power budget based on the general house dynamics. The submodels for the lumped parameter single-zone model are developed in Chapter 4.

1. A resistor capacitor equivalent model of the thermal characteristics of the different rooms and walls. This is found in Section 3.2.
   - Inputs: Heating power supplied by the heat pump P_{h}, solar intensity I_{s}, and ambient temperature T_{a}.
   - Outputs: Weighted average of room temperatures $\overline{T}_{\mathrm{r}}$.
2. An energy based heat pump model with variable efficiency. This is the heat pump model found in Section 4.3.
   - Inputs: Heating power supplied by the heat pump P_{h}, outflow temperature from the heat pump T_{out}, and ambient temperature T_{a}.
   - Outputs: Electric power supplied to the heat pump P_{e}

In the following section the data generation necessary for parameter estimation for the different submodels is described.

2.3 Data generation

A grey box approach has been decided for modelling of the system. Thus, input-output data is needed for the estimation of the model parameters.

The input-output data used for the parameter estimation of the submodels is generated using the Dymola model described in Subsection 1.3.2. The Dymola model is exported as a Functional Mock-up Unit (FMU) which then can be used in a co-simulation setup with a Python script also developed by Simon Thorsteinsson [1].

The Python script is configured to output the following signals necessary for the parameter estimation:

$$y = \begin{bmatrix} P_{\text{e}} & q_{i,j} & P_{i,j} & T_{\text{in}} & T_{\text{out}} & T_{\text{a}} & T_{\text{r},i,j} & I_{\text{s}} \end{bmatrix}^{\text{T}} \tag{2.1}$$

Note, due to the way the heat pump is formulated in the Dymola model, the electric power consumed by the heat pump is not viewed as an input. Instead a reference for the output temperature $T_{\text{out,ref}}$ is given, and the heat pump calculates the necessary electric power to achieve this reference within its operating range.

Thus, the inputs to the system are:

$$u = \begin{bmatrix} v_{i,j} & T_{\text{out,ref}} \end{bmatrix}^{\text{T}} \tag{2.2}$$

Note, some rooms do have multiple underfloor heating circuits controlled by individual valves. However, these will be handled as if there was only one valve for simplicity. For example, room 1 on the ground floor has two individual circuits. If $v_{0,1} = 1$ it means that both valves are open.

The FMU runs with a sample time of 1 minute with the co-simulation setup with Python. Input and output data from these open loop simulations are then exported to .csv files. Figure 2.3 shows the overview of the simulation setup used for generating the data sets.

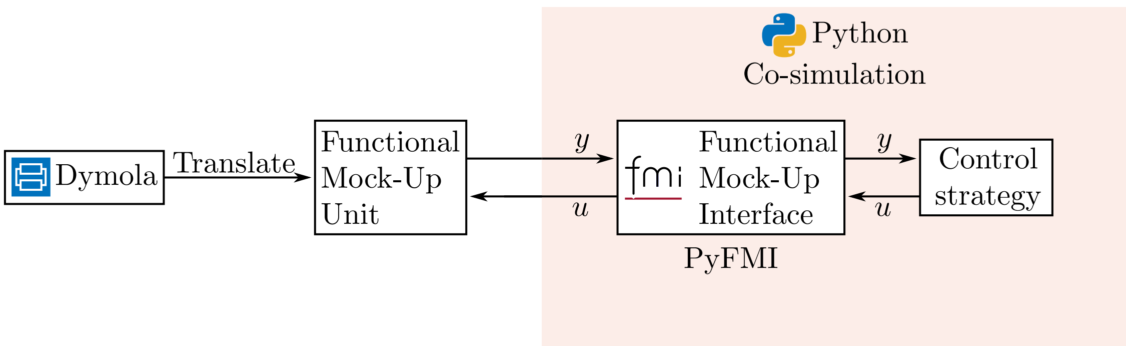

Figure 2.3: Overview of the simulation setup.

Two data sets are generated for the parameter estimation: one for training and one for validation. Both data sets are 30 days long, using disturbance data starting January 1$^{\text{st}}$ 2018 [1].

To allow for easier parameter estimation, the inputs for the data sets are constructed to give high exitation of the system.

Both sets use $T_{\text{out,ref}} = 30°\text{C} = 303.15\,\text{K}$. All valves are all initially set closed. Then every four hours the valves for one room are opened. This continues till all valves are open, which takes 2 days. Then every four hours, the valves for one room are closed. This pattern is then repeated for the entire 30 day period.

This gives a sort of staggered step input to the system as seen on Figure 2.4.

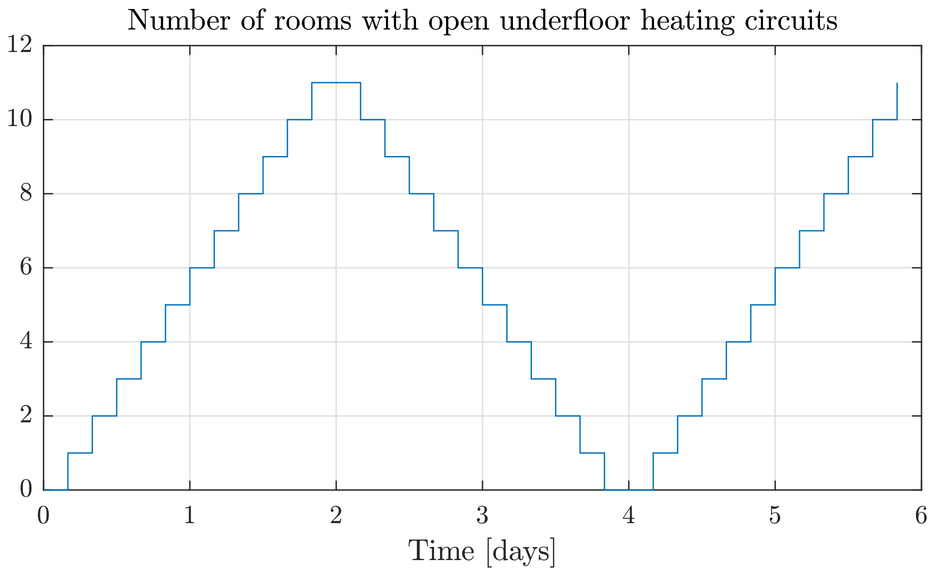

Figure 2.4: Staggered valve openings used for parameter estimation.

The order of opening and closing is randomised for both data sets. The order can be found in Table 2.1.

Table 2.1: Valve opening order for the different rooms for the training and validation set used for the parameter estimation.

Opening order	1	2	3	4	5	6	7	8	9	10	11
Training set	0,3	1,4	1,3	0,5	0,4	0,1	0,6	1,2	0,2	1,1	0,7
Validation set	0,5	0,2	1,3	1,4	1,2	0,6	0,1	0,4	1,1	0,3	0,7

Both data sets are down sampled to a sample time of 15 minutes for the parameter estimation of the lumped parameter multi-zone model. Due to the complexity and relatively high sample time of the multi-zone model, only the initial 48 hours are used.

However, for the lumped parameter single-zone model, described in Chapter 4, the sampling time is 1 hour, and the entire 30 day simulation data is used due to the very low complexity of this model.

Next in Chapter 3, the submodels describing the dynamics of the multi-zone model are developed.

3 Lumped parameter multi-zone

This chapter concerns the development of the submodels describing the lumped parameter multi-zone model. The chapter contains the following sections:

- Section 3.1 describes the mass flow submodel.
- Section 3.2 concerns the development of the resistor capacitor submodel.
- Section 3.3 develops the heat inflow submodel.
- Section 3.4 describes how the three previous submodels are combined into a final discrete time state space model.
- Section 3.5 describes the development of a heat power consumption model.
- Section 3.6 deals with the parameter estimation of the different submodels for the lumped parameter multi-zone model.

Figure 3.1 shows how the different submodels of the lumped parameter multi-zone model relates to each other.

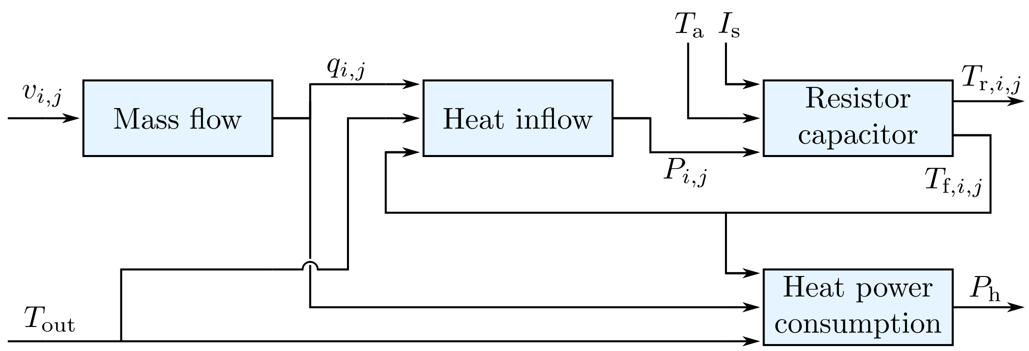

Figure 3.1: Sketch of how the different submodels of the lumped parameter multi-zone model relates to each other.

Next in Section 3.1 the model that describes the mass flow will be described.

3.1 Mass flow model

This section describes the valve positions' relationships to the mass flows in the underfloor heating circuits. The section only contains the following subsection:

- Subsection 3.1.1 describes the simple valve - mass flow relation.

Figure 3.2 shows how the model of this section relates to other submodels for the lumped parameter multi-zone model.

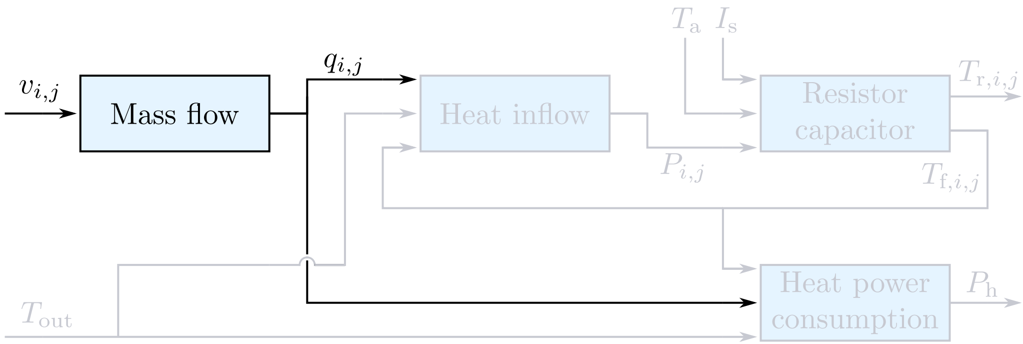

Figure 3.2: Sketch of how the model of this section relates to the other submodels.

3.1.1 Model

The relation between mass flow, pressure and resistance in a hydraulic system can be described by

$$q = \frac{1}{R_{\mathrm{p}}} \Delta p^{1/\alpha} \qquad [23, \text{ p. } 50] \tag{3.1}$$

where:

q	:	Mass flow	$\frac{\mathrm{kg}}{\mathrm{s}}$
Δp	:	Pressure difference	Pa
R_{p}	:	Constant whose value depend on the type of restriction.	$\frac{1}{\mathrm{m\,s}}$
α	:	Constant whose value depends on the type of flow.	$\frac{1}{\mathrm{m\,s}}$

It is assumed that the flow rates in the floor heating pipes are slow, which means that the flows are assumed to be laminar. This means that α will be set to 1 [23, p. 50], which yields the electrical analogy

$$q = \frac{1}{R_{\mathrm{p}}} \Delta p \approx I = \frac{1}{R} \Delta V \tag{3.2}$$

where:

ΔV	:	Voltage difference on each side of the resistor	V
I	:	Current	A

With this framework a resistor model can be setup for describing the resistance in the floor heating in each room as shown in Figure 3.3.

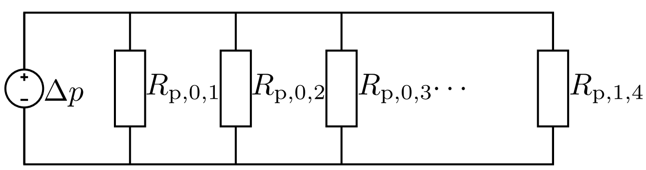

Figure 3.3: Resistor model describing the resistance in each floor.

It is assumed that the pressure is constant and the resistance between pump and parallel circuit is zero. This assumption consequently means that it is assumed that there is a circulation pump installed maintaining constant pressure in the floor heating pipes. Thus, the flow in each circuit can be described by:

$$q_{i,j} = \overline{q}_{i,j}\, v_{i,j} \tag{3.3}$$

where:

$\overline{q}_{i,j}$	:	Nominal mass flow	$\frac{\text{kg}}{\text{s}}$
$v_{i,j}$	:	Valve position	$\{0, 1\}$

Table 3.1 shows the number of individually controlled underfloor heating circuits per room, and their corresponding total mass flows when all are open. Note again, that for rooms with multiple circuits, they are handled as if there was only one, i.e. when e.g. $v_{1,1} = 1$ it means that all three valves in that room are open.

Table 3.1: Number of tubes per heating circuit.

Room	Number of circuits	Total Mass flow $\left[\frac{\text{kg}}{\text{s}}\right]$
0,1	2	0.0906
0,2	1	0.0188
0,3	1	0.0187
0,4	1	0.0187
0,5	1	0.0188
0,6	1	0.0188
0,7	1	0.0187
1,1	3	0.0324
1,2	1	0.0108
1,3	1	0.0108
1,4	1	0.0108

This concludes the mass flow model of the system. In the following section a resistor capacitor model of the rooms, walls and floors of the house will be developed.

3.2 Resistor capacitor model

This section concerns the development of a model of the temperatures of the rooms. Figure 3.4 shows how the model designed in this chapter will relate to the other submodels.

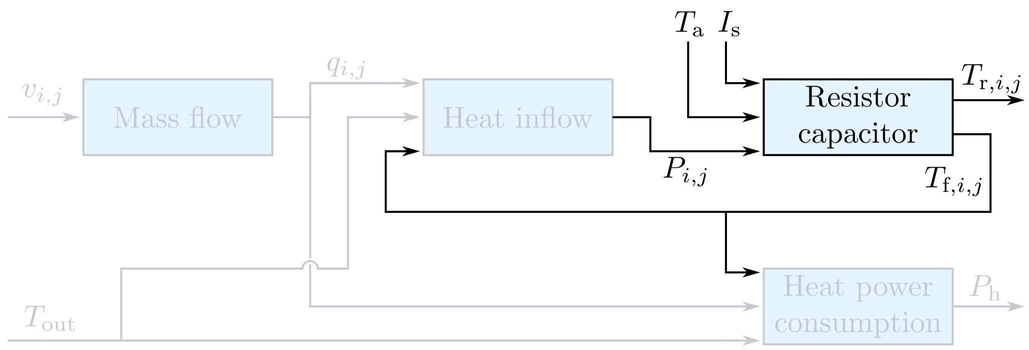

Figure 3.4: Sketch of how the model of this section relates to the other submodels.

The section contains the following subsections:

- Subsection 3.2.1 explains the basic ideas behind the modelling of the house as a resistor-capacitor network.
- Subsection 3.2.2 describes how the house is parted up into three types of components: Rooms, inner walls and outer walls.
- Subsection 3.2.3 develops the state space model of the resistor capacitor submodel consisting of 11 rooms connected through walls to each other and the outdoors.

3.2.1 Introduction

The model of the house' thermal dynamics, from the heat flow from the underfloor heating to the rooms and walls of the house will be modelled using a resistor-capacitor model. Here equivalents between thermal systems and electrical systems are used to allow for easier analysis. Equivalents are seen in Table 3.2.

Table 3.2: Electrical and thermal equivalent.

Electrical systems	**Thermal systems**
Current I	Heat flow rate/power P
Voltage V	Temperature T
Electrical resistance R	Thermal resistance R
Capacitance C	Heat capacity C
Electrical conductance G	Thermal conductance U

The basic idea is based around the formula for heating of a single phase system

$$\begin{aligned} Q &= m\,c\,\Delta T \qquad [24, \text{p. } 608] \\ &\Downarrow \text{ Assuming constant } m \text{ and } c \\ P &= m\,c\,\frac{\mathrm{d}T}{\mathrm{d}t} \end{aligned} \tag{3.4}$$

where:

Q	:	Heat	J
m	:	Mass	kg
c	:	Specific heat capacity	$\frac{\mathrm{J}}{\mathrm{kg\,K}}$
ΔT	:	Change in temperature	K
P	:	Power	W

and the formula for thermal conduction through a conductive slab

$$\begin{aligned} P &= k\,A\,\frac{\mathrm{d}T}{\mathrm{d}x} \qquad [24, \text{p. } 624] \\ &\approx \frac{k\,A}{w}\,\Delta T \end{aligned} \tag{3.5}$$

where:

k	:	Thermal conductivity	$\frac{\mathrm{W}}{\mathrm{m\,K}}$
A	:	Cross-sectional area of the slab	m^2
$\frac{\mathrm{d}T}{\mathrm{d}x}$	:	Temperature gradient across the slab	$\frac{\mathrm{K}}{\mathrm{m}}$
w	:	Width of the slab	m
ΔT	:	Temperature difference across the slab	K

This yields the electrical analogies

$$\begin{aligned} P = m\,c\,\frac{\mathrm{d}T}{\mathrm{d}t} \approx I = C\,\frac{\mathrm{d}V}{\mathrm{d}t} \quad &\text{Current-voltage relation for capacitors} \\ P \approx \frac{k\,A}{w}\,\Delta T \approx I = \frac{1}{R}\,V = G\,V \quad &\text{Ohm's law for resistors} \end{aligned} \tag{3.6}$$

where:

V	:	Voltage	V
I	:	Current	A
G	:	Electric conductance	S

which forms the basis for the electrical equivalent diagrams in this chapter.

$$\begin{aligned} C &= m\,c \quad \text{Heat capacity} \\ U &= \frac{1}{R} = \frac{k\,A}{w} \quad \text{Thermal conductance} \end{aligned} \tag{3.7}$$

Note, that the thermal conductance U is used instead of its reciprocal, the thermal resistance R, to simplify the expressions.

This idea is based on similar models presented in [25, 26, 27, 28, 29].

- [25] simplifies the house model to consist of three temperature states, the floor slab, the walls and the rooms. Each have a separate heat capacity and are interconnected using thermal resistances with each other and ambient surroundings. Heat rate from solar radiation is added to the wall and to the room, heat rate from occupants are added to the room, and heat rate from the underfloor heating is added to the floor slab.

- [26] similarly lumps the house into one room temperature state and one floor temperature state, but no wall temperature. Both states have separate heat capacities and are connected using thermal resistances. No heat is added from occupants, only solar radiation and thermal conductivity to ambient surroundings is considered as external heat rate inputs to the room. Again heat rate from the underfloor heating is added to the floor slab.

- [27] develops a way more complex model in TRNSYS of a 4 room apartment. For example, the floor slab consists of different heat capacities for the ceramic tiles, the moisture proof layer, thermal insulation etc. Moreover, convective and radiant heat transfer is also taken into account.

- [28] develops an intermediate model. Like [27] floors and walls are parted up into multiple states with separate capacities connected with resistances.

- [29] creates a model of a 5 room house, with each room having a single temperature state, but no floors or walls. Each state is connected with a thermal resistance and a separate heat capacity.

These models have varying levels of complexity. Since the goal of the model developed in this project is not for simulation, but for observer and controller design, limiting the complexity of the model is key. However, several requirements are deemed necessary:

1. Individual room temperatures.
2. Heat inflow from underfloor heating per room.
3. Varying outdoor temperatures.
4. Solar heating on individual rooms.

For more modularity, the model has been partitioned into three different components, that can be combined to model different floor plans. These are:

1. Room with underfloor heating, subsubsection 3.2.2.1.
2. Outer wall, subsubsection 3.2.2.2.
3. Inner wall, subsubsection 3.2.2.3.

The floor plan as presented in Figure 1.2 can then be parted up as seen on Figure 3.5.

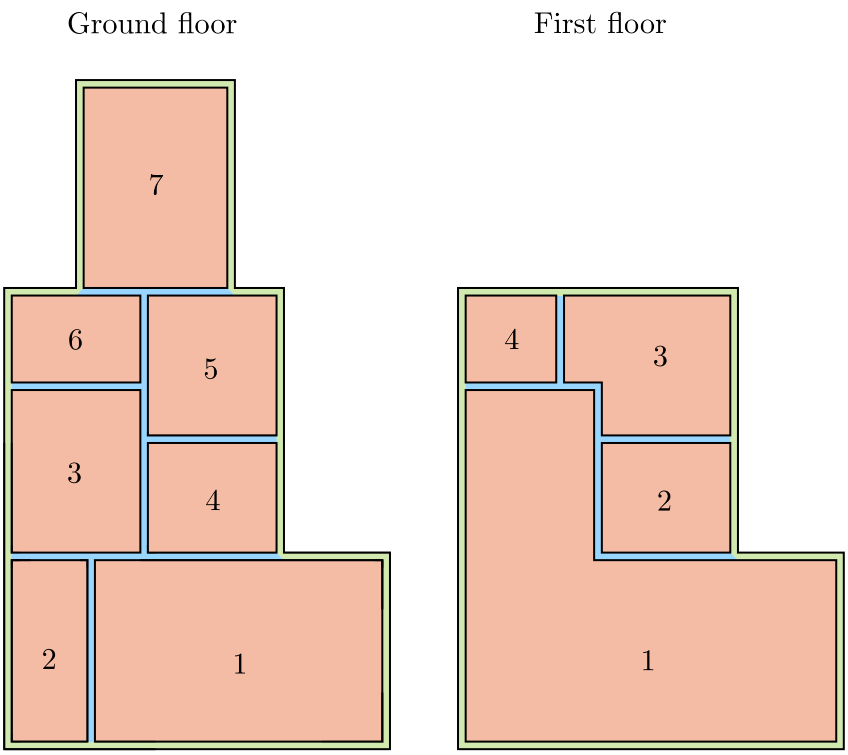

Figure 3.5: Parting up the floor plan into submodels. Orange: Room with underfloor heating. Green : Outer wall. Blue: Inner wall.

In the following section, the three different types of components of Figure 3.9 will be given.

3.2.2 Components

In this section, the three different types of components of Figure 3.9 will be described and combined.

3.2.2.1 Room with underfloor heating

The submodel of a room with underfloor heating consists of the following parts:

- A lumped parameter model of a room, with a thermal mass and power inflow from the sun hitting the room.
- A lumped parameter model of the floor with a thermal mass and power inflow from the heating pipes.

This results in an electrical equivalent diagram as seen Figure 3.6 where i and j denotes floor number and room number respectively.

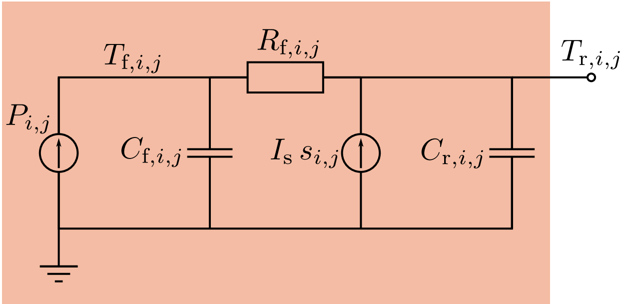

Figure 3.6: Electrical equivalent diagram of a room with underfloor heating, where: $R_{\mathrm{f},i,j} = U_{\mathrm{f},i,j}^{-1}$.

Which leads to the equations:

$$
\begin{aligned}
C_{\mathrm{f},i,j}\,\dot{T}_{\mathrm{f},i,j} &= -U_{\mathrm{f},i,j}\,T_{\mathrm{f},i,j} + U_{\mathrm{f},i,j}\,T_{\mathrm{r},i,j} + P_{i,j} \\
C_{\mathrm{r},i,j}\,\dot{T}_{\mathrm{r},i,j} &= -\left(U_{\mathrm{f},i,j} + \sum_l U_l\right) T_{\mathrm{r},i,j} + U_{\mathrm{f},i,j}\,T_{\mathrm{f},i,j} + \sum_l U_l\,T_l + I_\mathrm{s}\,s_{i,j}
\end{aligned}
\tag{3.8}
$$

where:

$T_{\mathrm{f},i,j}$	:	Temperature of the floor	K
$T_{\mathrm{r},i,j}$	:	Temperature of the air in the room	K
$C_{\mathrm{f},i,j}$	:	Heat capacity of the floor	$\frac{\mathrm{J}}{\mathrm{K}}$
$C_{\mathrm{r},i,j}$	:	Heat capacity of the air in the room	$\frac{\mathrm{J}}{\mathrm{K}}$
$U_{\mathrm{f},i,j}$	:	Thermal conductance between floor and room	$\frac{\mathrm{W}}{\mathrm{K}}$
$P_{i,j}$	:	Heat inflow from underfloor heating	W
I_s	:	Solar intensity	-
$s_{i,j}$	:	Solar scaling factor	W
T_l	:	Temperature of room adjacent components	K
U_l	:	Thermal conductance between the room and adjacent components	$\frac{\mathrm{W}}{\mathrm{K}}$

3.2.2.2 Outer wall

The submodel of the outer wall consists of a resistor, allowing exchange of heat between a time-varying ambient temperature and the connecting room.

This results in the assumption that the outdoors have an infinitely large thermal capacity, and thus is not affected by the any heat flow from the house.

Figure 3.7 shows the electrical equivalent of the outer wall model.

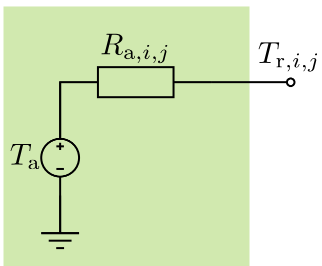

Figure 3.7: Electrical equivalent diagram of an outer wall, where: $R_{\mathrm{a},i,j} = U_{\mathrm{a},i,j}^{-1}$.

T_{a}	:	Temperature of the ambient outdoors	K
$U_{\text{a},i,j}$	:	Thermal conductance between the adjacent room and ambient surroundings	$\frac{\text{W}}{\text{K}}$

3.2.2.3 Inner wall

The inner walls are like the outer walls only considered thermal resistances, but now between adjacent rooms. Thus, the inner walls have no associated temperature states.

This results in a electrical equivalent diagram as seen Figure 3.8. Here two rooms i, j and i', j' are connected by inner wall number l.

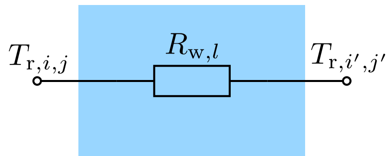

Figure 3.8: Electrical equivalent diagram of an inner wall, where: $R_{\text{w},l} = U_{\text{w},l}^{-1}$.

$U_{\text{w},l}$	:	Thermal conductance between adjacent rooms	$\frac{\text{W}}{\text{K}}$

3.2.3 Combined resistor-capacitor model

The connections the different components of the resistor-capacitor model of the house can be seen on Figure 3.9 and 3.10.

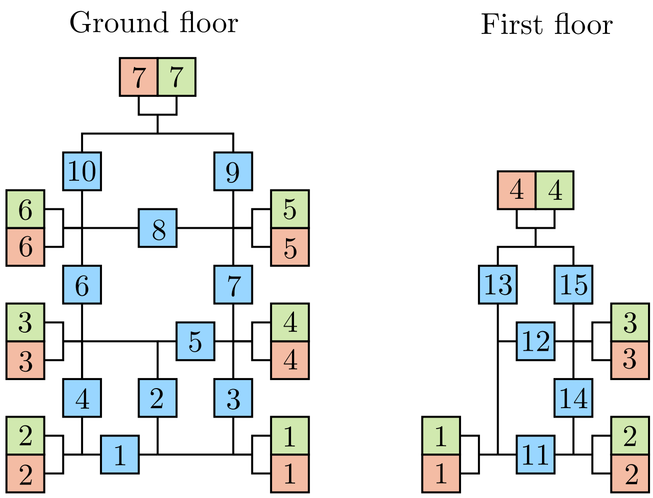

Figure 3.9: Electrical equivalent diagram of the floor plan of the house from Figure 3.5. Note that the inter floor connections are not shown here. Orange: Room with underfloor heating. Green: Outer wall. Blue: Inner wall.

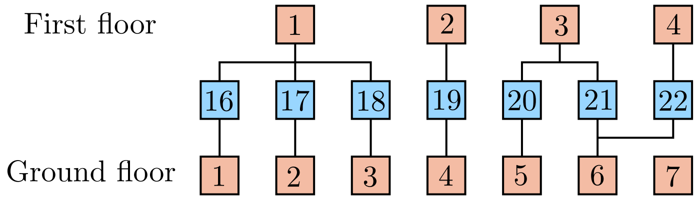

Figure 3.10: Inner wall connections between the two floors Orange: Room with underfloor heating. Blue: Inner wall.

By connection the different components as seen on Figure 3.9 and Figure 3.10 a linear, dynamic model of the system can be built.

$$M\,\dot{x} = A_{\mathrm{rc}}\,x + B_{\mathrm{rc}}\,u_{\mathrm{rc}} + E_{\mathrm{rc}}\,d \tag{3.9}$$

where the states x_{rc}, the inputs u_{rc} and the disturbances d are defined as:

$$x = \begin{bmatrix} T_{\mathrm{f},0,1} \\ \vdots \\ T_{\mathrm{f},1,4} \\ \hline T_{\mathrm{r},0,1} \\ \vdots \\ T_{\mathrm{r},1,4} \end{bmatrix}_{22\times 1} \qquad u_{\mathrm{rc}} = \begin{bmatrix} P_{0,1} \\ \vdots \\ P_{1,4} \end{bmatrix}_{11\times 1} \qquad d = \begin{bmatrix} I_{\mathrm{s}} \\ T_{\mathrm{a}} \end{bmatrix}_{2\times 1} \tag{3.10}$$

The mass matrix M is a diagonal matrix with the capacities of the temperature states:

$$M = \left[\begin{array}{cccc|cccc} C_{\mathrm{f},0,1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & C_{\mathrm{f},0,2} & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{\mathrm{f},1,4} & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & C_{\mathrm{r},0,1} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & C_{\mathrm{r},0,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & C_{\mathrm{r},1,4} \end{array}\right]_{22\times 22} \tag{3.11}$$

The B_{rc} matrix defines which floor states receive the different heat inflows from u_{rc}, and E_{rc} defines how the outside disturbances affect each room:

$$B_{\mathrm{rc}} = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \hline 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{array}\right]_{22\times 11} \qquad E_{\mathrm{rc}} = \left[\begin{array}{cc} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \hline s_{0,1} & R_{\mathrm{a},0,1} \\ \vdots & \vdots \\ s_{1,4} & R_{\mathrm{a},1,4} \end{array}\right]_{22\times 2} \tag{3.12}$$

The symmetric A_{rc} matrix contains the conductances connecting the different temperature states and the ambient surroundings:

$$A_{\mathrm{rc}} = \left[\begin{array}{c|c} -A_{\mathrm{rc},1} & A_{\mathrm{rc},1} \\ \hline A_{\mathrm{rc},1} & A_{\mathrm{rc},2} \end{array}\right]_{22\times 22} \qquad A_{\mathrm{rc},1} = \begin{bmatrix} U_{\mathrm{f},0,1} & 0 & \cdots & 0 \\ 0 & U_{\mathrm{f},0,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & U_{\mathrm{f},1,4} \end{bmatrix} \tag{3.13}$$

$$A_{\mathrm{rc},2} = \begin{bmatrix}
a_{\mathrm{d},1} & U_{\mathrm{w},1} & U_{\mathrm{w},2} & U_{\mathrm{w},3} & 0 & 0 & 0 & U_{\mathrm{w},16} & 0 & 0 & 0 \\
U_{\mathrm{w},1} & a_{\mathrm{d},2} & U_{\mathrm{w},4} & 0 & 0 & 0 & 0 & U_{\mathrm{w},17} & 0 & 0 & 0 \\
U_{\mathrm{w},2} & U_{\mathrm{w},4} & a_{\mathrm{d},3} & U_{\mathrm{w},5} & 0 & U_{\mathrm{w},6} & 0 & U_{\mathrm{w},18} & 0 & 0 & \\
U_{\mathrm{w},3} & 0 & U_{\mathrm{w},5} & a_{\mathrm{d},4} & U_{\mathrm{w},7} & 0 & 0 & 0 & U_{\mathrm{w},19} & 0 & 0 \\
0 & 0 & 0 & U_{\mathrm{w},7} & a_{\mathrm{d},5} & U_{\mathrm{w},8} & U_{\mathrm{w},9} & 0 & 0 & U_{\mathrm{w},20} & 0 \\
0 & 0 & U_{\mathrm{w},6} & 0 & U_{\mathrm{w},8} & a_{\mathrm{d},6} & U_{\mathrm{w},10} & 0 & 0 & U_{\mathrm{w},21} & U_{\mathrm{w},22} \\
0 & 0 & 0 & 0 & U_{\mathrm{w},9} & U_{\mathrm{w},10} & a_{\mathrm{d},7} & 0 & 0 & 0 & 0 \\
U_{\mathrm{w},16} & U_{\mathrm{w},17} & U_{\mathrm{w},18} & 0 & 0 & 0 & 0 & a_{\mathrm{d},8} & U_{\mathrm{w},11} & U_{\mathrm{w},12} & U_{\mathrm{w},13} \\
0 & 0 & 0 & U_{\mathrm{w},19} & 0 & 0 & 0 & U_{\mathrm{w},11} & a_{\mathrm{d},9} & U_{\mathrm{w},14} & \\
0 & 0 & 0 & 0 & U_{\mathrm{w},20} & U_{\mathrm{w},21} & 0 & U_{\mathrm{w},12} & U_{\mathrm{w},14} & a_{\mathrm{d},10} & U_{\mathrm{w},15} \\
0 & 0 & 0 & 0 & 0 & U_{\mathrm{w},22} & 0 & U_{\mathrm{w},13} & 0 & U_{\mathrm{w},15} & a_{\mathrm{d},11}
\end{bmatrix} \tag{3.14}$$

The diagonal terms $a_{\mathrm{d},\iota}$, $\iota \in \{1, \ldots, 11\}$ in $A_{\mathrm{rc},2}$ are the negative sums of the off diagonal terms of the corresponding $\iota + 11^{\mathrm{th}}$ row of the A_{rc} matrix minus $U_{\mathrm{a},\iota}$. For example:

$$a_{\mathrm{d},1} = -\left(U_{\mathrm{w},1} + U_{\mathrm{w},2} + U_{\mathrm{w},3} + U_{\mathrm{w},16} + U_{\mathrm{f},0,1} + U_{\mathrm{a},1}\right) \tag{3.15}$$

Next in Section 3.3 the heat inflow submodel will be developed, which concerns the modelling of the $P_{i,j}$ used as inputs.

3.3 Heat inflow model

This section concerns the design how a model for calculating the heat flow rate from each underfloor heating circuit to floor slabs. Figure 3.11 shows how the model of this section relates to the other submodels.

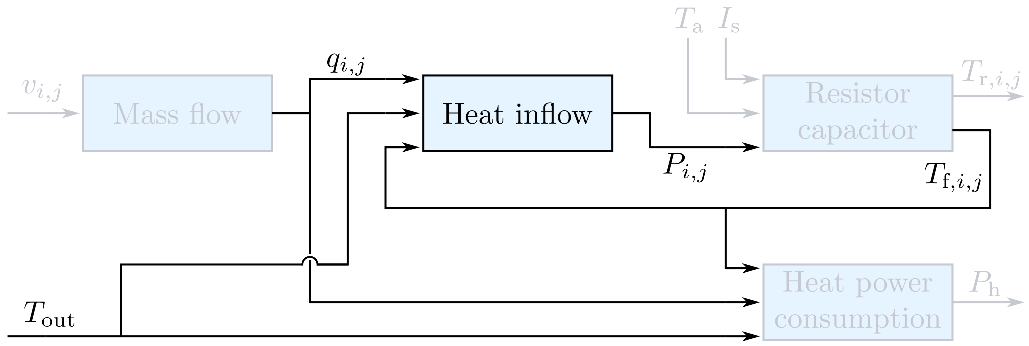

Figure 3.11: Sketch of how the model of this section relates to the other submodels.

The section contains the following subsection:

- Subsection 3.3.1 describes the idea and physics behind this submodel.

3.3.1 Model

In order to model the heat inflow from underfloor heating the formula for heating of a single phase system is used.

$$
\begin{aligned}
Q &= m\, c\, \Delta T \qquad [24, \text{p. } 608] \\
&\Downarrow \text{ Assuming constant } \Delta T \text{ and } c \\
P &= q\, c\, \Delta T
\end{aligned}
\tag{3.16}
$$

where:

Q	:	Heat	J
m	:	Mass	kg
q	:	Mass flow	$\frac{\text{kg}}{\text{s}}$
c	:	Specific heat capacity	$\frac{\text{J}}{\text{kg K}}$
ΔT	:	Change in temperature	K
P	:	Power	W

For the heat inflow from the underfloor heating water to the underfloor slap the inflow temperature to each circuit is assumed to be the same, T_{out}, i.e. the outflow temperature from the heat pump.

Additionally, the outflow of each circuit is assumed to be that of the temperature of the floor from Section 3.2, $T_{\text{f},i,j}$ biased with a constant $b_{i,j}$.

Thus, Equation 3.16 becomes:

$$
P_{i,j} = c\, q_{i,j}\, b_{i,j}\, (T_{\text{out}} - T_{\text{f},i,j}) \tag{3.17}
$$

where:

$q_{i,j}$	:	Mass flow	$\frac{\mathrm{kg}}{\mathrm{s}}$
c	:	Specific heat capacity of water	$\frac{\mathrm{J}}{\mathrm{kg\,K}}$
T_{out}	:	Temperature of water out of the heat pump	K
$T_{\mathrm{f},i,j}$	:	Temperature of the floor	K
$b_{i,j}$	:	Floor temperature bias	
$P_{i,j}$	:	Heat inflow from underfloor heating	W

In Section 3.4 this submodel will be combined with the submodel from Section 3.2.

3.4 Dynamic multi-zone model

In this section the submodels from Section 3.1, Section 3.2 and Section 3.3 will be combined and discretised. To facilitate the control and observer design, the model needs to be converted into a discrete time system on the form:

$$x_{k+1} = f(x_k, u_k, d_k) \tag{3.18}$$

where:

$$\begin{aligned} x_k &= \begin{bmatrix} T_{\mathrm{f},i,j} \\ T_{\mathrm{r},i,j} \end{bmatrix}, \qquad y_k = \begin{bmatrix} T_{\mathrm{r},i,j} \end{bmatrix} \\ u_k &= \begin{bmatrix} q_{i,j} \\ T_{\mathrm{out}} \end{bmatrix}, \qquad d_k = \begin{bmatrix} I_s \\ T_{\mathrm{a}} \end{bmatrix} \end{aligned} \tag{3.19}$$

To achieve this, the resistor capacitor model in Equation 3.9 is converted into the following form first:

$$\dot{x} = A\,x + B\,u_{\mathrm{rc}} + E\,d \tag{3.20}$$

where:

$$A = M^{-1}\,A_{\mathrm{rc}} \qquad B = M^{-1}\,B_{\mathrm{rc}} \qquad E = M^{-1}\,E_{\mathrm{rc}} \tag{3.21}$$

This is now a linear continuous time state space model. This can then be converted into a discrete time model on the form.

$$x_{k+1} = A_{\mathrm{d}}\,x_k + B_{\mathrm{d}}\,u_{\mathrm{rc},k} + E_{\mathrm{d}}\,d \tag{3.22}$$

This is done using the Zero Order Hold (ZOH) method:

$$A_{\mathrm{d}} = \mathrm{e}^{A\,T_{\mathrm{s}}}, \qquad B_{\mathrm{d}} = A^{-1}\,(A_{\mathrm{d}} - \mathbf{I}_{22\times 22})\,B, \qquad E_{\mathrm{d}} = A^{-1}\,(A_{\mathrm{d}} - \mathbf{I}_{22\times 22})\,E \tag{3.23}$$

where:

T_{s}	:	Sample time of the system	s
$\mathbf{I}$	:	Identity matrix	

A sample time of $T_{\mathrm{s}} = 900\,\mathrm{s}$ (15 minutes) has been chosen based on general system responsiveness. Next, Equation 3.17 is put into matrix-vector form:

$$\begin{aligned} P_{i,j} &= c\,q_{i,j}\,b_{i,j}\,(T_{\mathrm{out}} - T_{\mathrm{f},i,j}) \\ &\Downarrow \\ P &= F\,u_k\,G\,u_k - F\,u_k \odot H x_k \end{aligned} \tag{3.24}$$

where:

$$F = \begin{bmatrix} c\,b_{0,1} & 0 & \cdots & 0 & 0 \\ 0 & c\,b_{0,2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & c\,b_{1,4} & 0 \end{bmatrix}_{11\times 12}$$
$$G = \begin{bmatrix}\mathbf{0}_{1\times 11} & 1\end{bmatrix} \qquad H = \begin{bmatrix}\mathbf{I}_{11\times 11} & \mathbf{0}_{11\times 11}\end{bmatrix} \tag{3.25}$$

and P is the $P_{i,j}$ put into an ordered vector. $\odot$ is used for element-wise multiplication. Finally, it is assumed that for the purposes of the controller only the room temperature states are measured.

Combining yields the final dynamic, discrete time model:

$$\begin{aligned} x_{k+1} &= A_\mathrm{d}\,x_k + E_\mathrm{d}\,d_k + B_\mathrm{d}\,\left(F\,u_k\,G\,u_k - F\,u_k \odot H\,x_k\right) \\ y_k &= C_\mathrm{d}\,x_k \end{aligned} \tag{3.26}$$

where:

$$C_\mathrm{d} = \begin{bmatrix}\mathbf{0}_{11\times 11} & \mathbf{I}_{11\times 11}\end{bmatrix}, \qquad \begin{aligned} x_k &= \begin{bmatrix} T_{\mathrm{f},i,j} \\ T_{\mathrm{r},i,j} \end{bmatrix}, & y_k &= \begin{bmatrix} T_{\mathrm{r},i,j} \end{bmatrix} \\ u_k &= \begin{bmatrix} q_{i,j} \\ T_\mathrm{out} \end{bmatrix}, & d_k &= \begin{bmatrix} I_s \\ T_\mathrm{a} \end{bmatrix} \end{aligned} \tag{3.27}$$

Next in Section 3.5 the heat power consumption submodel will be described. The reason why it has not been included in the dynamic multi-zone model is that it needs the floor temperatures which are not measured for parameter estimation. Therefore the Extended Kalman Filter (EKF) is needed together with the dynamic multi-zone model from Section 3.4 in order to use estimates of the floor temperatures $\hat{T}_{\mathrm{f},i,j,k|k}$, from the Extended Kalman Filter (EKF) from Chapter 6 using the 48 hour training data set described in Section 2.3.

3.5 Heat power consumption model

This section concerns the development of the submodel describing the heat power consumption of the lumped parameter multi-zone model.

The section contains the following sections:

- Subsection 3.5.1 describes how the return water temperature will be modelled.
- Subsection 3.5.2 describes how the return flow model is used for modelling the heat power consumption of the lumped parameter multi-zone model.

Figure 3.12 shows how the model of the section relates to the other submodels.

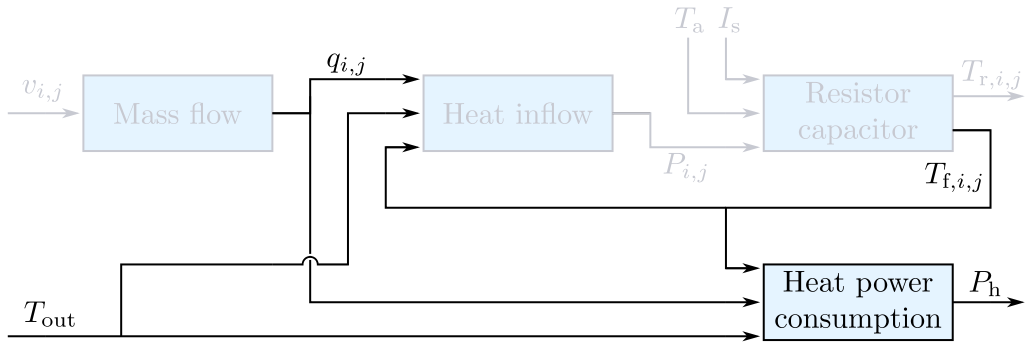

Figure 3.12: Sketch of how the model of this section relates to the other submodels.

3.5.1 Return flow model

The water flow returning to the heat pump is a mixture of the water flows from each underfloor heating circuit. Thus, the idea for modelling the return flow temperature T_{in} is a weighted average of the underfloor heating circuit temperatures.

$$T_{in} = \frac{1}{\sum_{i,j} q_{i,j}} \sum_{i,j} q_{i,j} T_{in,i,j} \tag{3.28}$$

However, these temperatures are not modelled. Instead the corresponding floor slab temperature $T_{f,i,j}$ is used, as it is assumed these are approximately the same. This yields

$$T_{in} = \frac{1}{\sum\limits_{i,j} q_{i,j}} \sum_{i,j} q_{i,j}\, T_{f,i,j} \tag{3.29}$$

where:

T_{in}	:	Temperature of heat pump inflow water	K
$T_{f,i,j}$	:	Temperature of the floor	K
$q_{i,j}$	:	Mass flow in the different underfloor heating circuits	$\frac{kg}{s}$

However, since $T_{f,i,j}$ and the return flow temperature from each circuit is not exactly the same, a scaling factor $a_{i,j}$ multiplied on each $T_{f,i,j}$.

$$T_{in} = \frac{1}{\sum_{i,j} q_{i,j}} \sum_{i,j} q_{i,j}\, T_{f,i,j}\, a_{i,j} \tag{3.30}$$

Due to the mathematical problems of dividing with zero, it has been found to be more beneficial to reformulate this part of the model as:

$$T_{\text{in}}\, q_{\text{tot}} = \sum_{i,j} q_{i,j}\, T_{\text{f},i,j}\, a_{i,j}, \qquad q_{\text{tot}} = \sum_{i,j} q_{i,j} \tag{3.31}$$

In Subsection 3.6.6 the estimation of the scaling factors $a_{i,j}$ will be described.

3.5.2 Heating of the water

This part of the modelling follows that of Section 3.3. Again the heating of a single phase system is used as the basis:

$$\begin{aligned} &Q = m\, c\, \Delta T \qquad [24, \text{p. } 608] \\ &\quad \Downarrow \text{ Assuming constant } \Delta T \text{ and } c \\ &P = q\, c\, \Delta T \end{aligned} \tag{3.32}$$

where:

Q	:	Heat	J
m	:	Mass	kg
q	:	Mass flow	$\frac{\text{kg}}{\text{s}}$
c	:	Specific heat capacity	$\frac{\text{J}}{\text{kg K}}$
ΔT	:	Change in temperature	K
P	:	Power	W

In the case of the heat pump, after substituting in the relevant variables to Equation 3.32, the relation is achieved.

$$\begin{aligned} P_{\text{h}} &= \sum_{i,j} q_{i,j}\, c\, (T_{\text{out}} - T_{\text{in}}) \\ &\Downarrow \\ P_{\text{h}} &= c\, q_{\text{tot}}\, T_{\text{out}} - c\, q_{\text{tot}}\, T_{\text{in}} \end{aligned} \tag{3.33}$$

where:

$q_{i,j}$	:	Mass flow in the different underfloor heating circuits	$\frac{\text{kg}}{\text{s}}$
q_{tot}	:	Total mass flow in the underfloor heating circuits	$\frac{\text{kg}}{\text{s}}$
c	:	Specific heat capacity of water	$\frac{\text{J}}{\text{kg K}}$
T_{out}	:	Temperature of the heat pump outflow water	K
T_{in}	:	Temperature of the heat pump inflow water	K
P_{h}	:	Heating power supplied by the heat pump to the water	W

The specific heat capacity of water c is 4182 $\frac{\text{J}}{\text{kg K}}$ [24, p. 608]. This can then be combined with Equation 3.31 to achieve:

$$P_{\text{h}} = c\, q_{\text{tot}}\, T_{\text{out}} - c \sum_{i,j} q_{i,j}\, T_{\text{f},i,j}\, a_{i,j} \tag{3.34}$$

Next in Section 3.6 the parameter estimation for the dynamic multi-zone model from Section 3.4 and the heat power consumption model will be described.

3.6 Parameter estimation

This section aims to estimate parameters for dynamic multi-zone model from Section 3.4 as well as the the return flow model described in Section 3.5 For both of these, the data described in Section 2.3 is used.

The reason why the parameter estimation of the lumped parameter multi-zone model is split up into two parts is that the heat power consumption model of the return flow has no dynamics. The section contains the following subsections:

- Subsection 3.6.1 gives a introduction to the setup for the parameter estimation of the dynamic multi-zone model.
- Subsection 3.6.2 briefly describes the optimisation algorithm used by the parameter estimation method.
- Subsection 3.6.3 explains how the initial values for the parameter estimation is found.
- Subsection 3.6.4 shows open loop simulation response for the validation set using the model with the found parameters.
- Subsection 3.6.5 briefly analysis the linear part of the obtained model.
- Subsection 3.6.6 contains the parameter estimation of the return flow model from Subsection 3.5.1.

3.6.1 Introduction

Because of the developed dynamic multi-zone model, a grey box approach is used. Thus, the parameters to be estimated are:

$$\begin{aligned} &\theta = \begin{bmatrix} C_{\mathrm{f},i,j} & C_{\mathrm{r},i,j} & U_{\mathrm{f},i,j} & U_{\mathrm{a},i,j} & U_{\mathrm{w},l} & s_{i,j} & b_{i,j} \end{bmatrix}^{\mathrm{T}} \\ &\theta \geq 0 \end{aligned} \tag{3.35}$$

There are 11 of each except for $U_{\mathrm{w},l}$ for which there are 22. This results in a total of 88 parameters to be estimated.

For the parameter estimation MATLAB's `nlgreyest` command from the System Identification toolbox has been used. This command allows grey box estimation of systems on the form:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, \theta) \\ y_k &= g(x_k, u_k, \theta) \end{aligned} \tag{3.36}$$

This form of Equation 3.26 is achieved by combing the input and disturbance vectors:

$$u'_k = \begin{bmatrix} u_k \\ d_k \end{bmatrix} = \begin{bmatrix} q_{i,j} \\ T_{\mathrm{out}} \\ I_s \\ T_{\mathrm{a}} \end{bmatrix} \tag{3.37}$$

Additionally, to improve the parameter estimation results the values of $P_{i,j}$ are added as additional outputs since, they are outputted by the Dymola model. The `lsqnonlin` algorithm used by `nlgreyest` does not allow for weights of systems with multiple outputs. Since $P_{i,j}$ and $T_{\mathrm{f},i,j}$ have widely different ranges, the output of $T_{\mathrm{f},i,j}$ is scaled with a factor of 600 to bring the typical range of $T_{\mathrm{f},i,j}$ in the training set of 5 K to be more similar to that $P_{i,j}$ which typically has a range of about 3000 W.

Thus, the model for the purpose of parameter estimation becomes:

$$\begin{aligned} x_{k+1} &= f(x_k, u'_k, \theta) \\ y'_k &= \begin{bmatrix} 600\, C_\mathrm{d}\, x_k \\ P_{i,j} \end{bmatrix} \end{aligned} \tag{3.38}$$

The data from the Dymola model described in Section 2.3 is downsampled to run at a sample time of $T_\mathrm{s} = 900\,\mathrm{s}$, i.e. the same as that of the model.

In the following subsection the algorithm used by `nlgreyest` is described.

3.6.2 lsqnonlin

`lsqnonlin` is a trust-region reflective Newton method from MATLAB's Optimization toolbox, where the cost function V is the sum of squares of the simulation error [30]:

$$V(\theta) = \frac{1}{N} \sum_{k=1}^{N} e^\mathrm{T}(k, \theta)\, e(k, \theta) \tag{3.39}$$

where:

θ	:	Parameters to be estimated
N	:	Number of samples
k	:	Discrete time
e	:	Error between measured and simulated outputs

I.e. the algorithms attemps to find the local optimum θ^*

$$\begin{gathered} \theta^* = \arg\min V(\theta) \\ \text{subject to:} \\ a_i(\theta) \geq 0 \quad \text{for } i = 1, 2, \ldots, p \end{gathered} \tag{3.40}$$

where:

a_i	:	Inequality constraints on θ
p	:	Number of inequality constraints

MATLAB's `lqsnonlin` is based on algorithms presented in [31] and [32].

3.6.3 Approach

Due to the large number of parameters without reasonable initial guesses, the parameter estimation has been parted up into multiple steps. The final values from each step is used as initial guesses for the following step. The initial guess for the parameters for the first step was found manually, so as to achieve a stable response.

$$\begin{aligned} U_\mathrm{f} &= 1.00\mathrm{E}+03\,\frac{\mathrm{W}}{\mathrm{K}}, & U_\mathrm{a} &= 1.00\mathrm{E}+01\,\frac{\mathrm{W}}{\mathrm{K}}, & C_\mathrm{f} &= 1.00\mathrm{E}+07\,\frac{\mathrm{J}}{\mathrm{K}} \\ C_\mathrm{r} &= 1.00\mathrm{E}+06\,\frac{\mathrm{J}}{\mathrm{K}}, & s &= 1.00\mathrm{E}+03, & b &= 1.00\mathrm{E}+00 \end{aligned} \tag{3.41}$$

1. For each room build a model consisting of one outer wall, one room, and one heat inflow, resulting in 11 separate models.
   $\theta_1 = \begin{bmatrix} C_{\mathrm{f},i,j} & C_{\mathrm{r},i,j} & U_{\mathrm{f},i,j} & U_{\mathrm{a},i,j} & s_{i,j} & b_{i,j} \end{bmatrix}^\mathrm{T}$
2. Connect the rooms on each floor with inner walls to create two separate models according to Figure 3.9. Initialise the created $U_{\mathrm{w},l}$ as $1.00\mathrm{E}+01\,\frac{\mathrm{W}}{\mathrm{K}}$.
   $\theta_2 = \begin{bmatrix} U_{\mathrm{w},l} \end{bmatrix}^\mathrm{T}, \quad l = \{1, \ldots, 15\}$
3. With the same two separate models do parameter estimation with
   $\theta_3 = \begin{bmatrix} C_{\mathrm{f},i,j} & C_{\mathrm{r},i,j} & U_{\mathrm{f},i,j} & U_{\mathrm{a},i,j} & U_{\mathrm{w},l} & s_{i,j} & b_{i,j} \end{bmatrix}^\mathrm{T}$
4. Connect the two floors with inner walls to one combined model according to Figure 3.10. Initialise the created $U_{\mathrm{w},l}$ as $1.00\mathrm{E}+01\,\frac{\mathrm{W}}{\mathrm{K}}$.
   $\theta_4 = \begin{bmatrix} U_{\mathrm{w},l} \end{bmatrix}^\mathrm{T}, \quad l = \{16, \ldots, 22\}$
5. Do parameter estimation on all the parameters in the last model.
   $\theta_5 = \begin{bmatrix} C_{\mathrm{f},i,j} & C_{\mathrm{r},i,j} & U_{\mathrm{f},i,j} & U_{\mathrm{a},i,j} & U_{\mathrm{w},l} & s_{i,j} & b_{i,j} \end{bmatrix}^\mathrm{T}$

The results of the final step will be presented in Subsection 3.6.4.

3.6.4 Results

Figure 3.13 and 3.14 shows the open-loop simulation response of the system on the validation data set compared to the measured temperature output for the ground and first floor respectively.

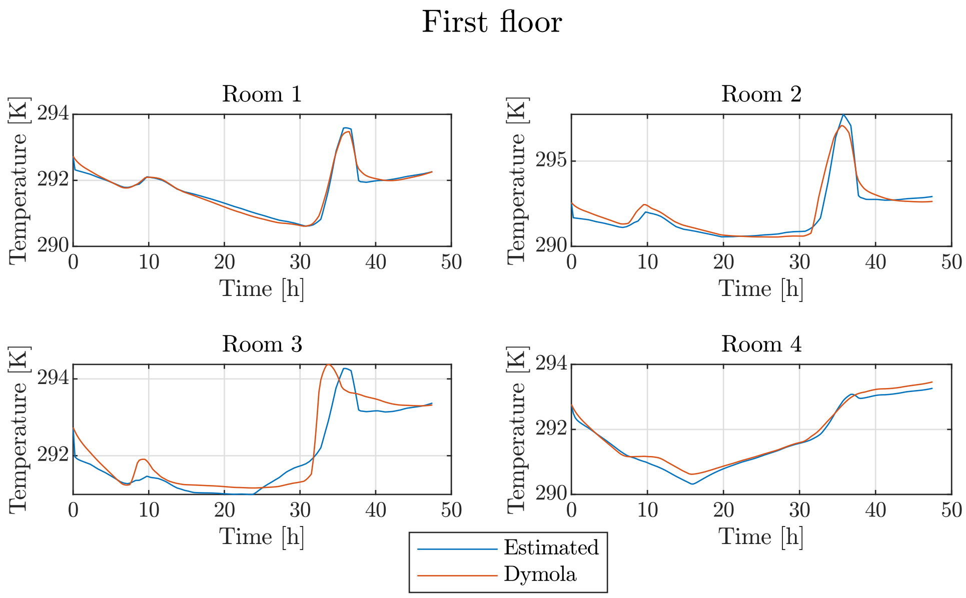

Figure 3.13: Room temperature open-loop simulations for the rooms on the first floor using the validation data set.

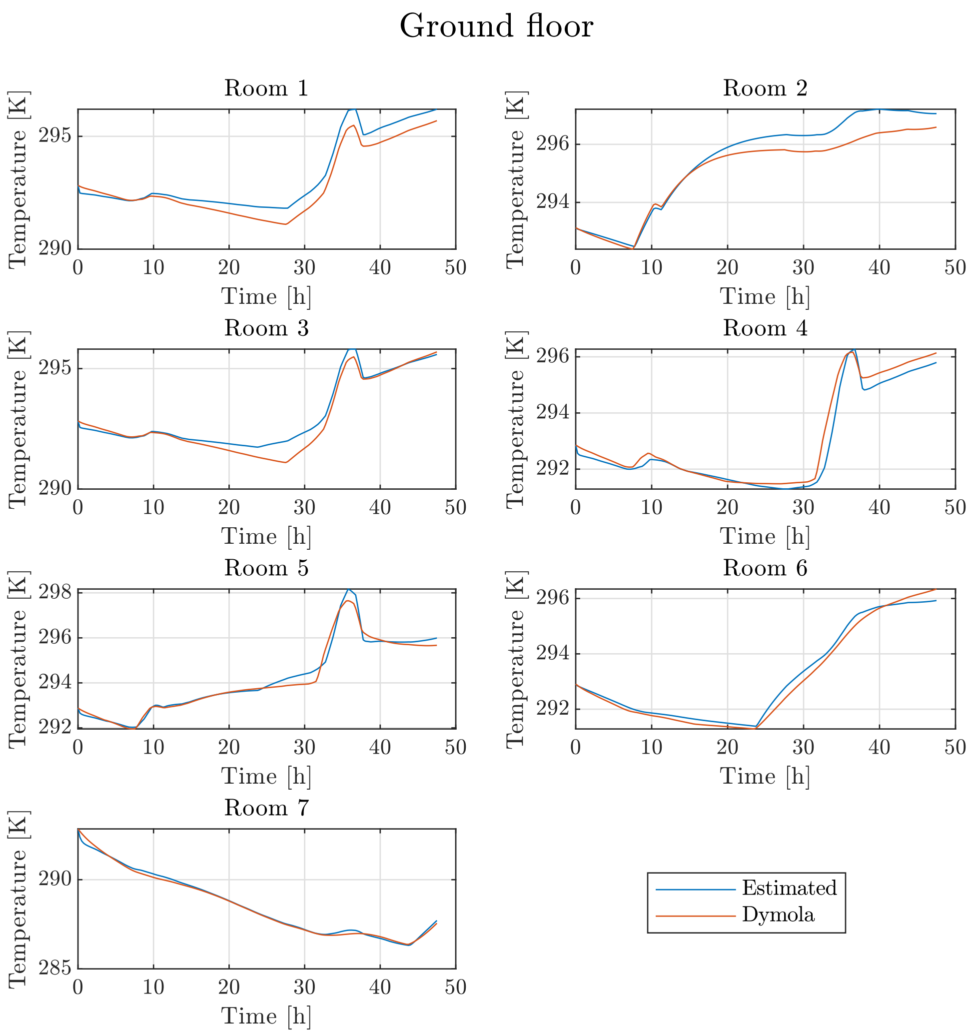

Figure 3.14: Room temperature open-loop simulations for the rooms on the ground floor using the validation data set.

Additionally, Figure 3.15 and 3.16 show the heat flow rate for the same.

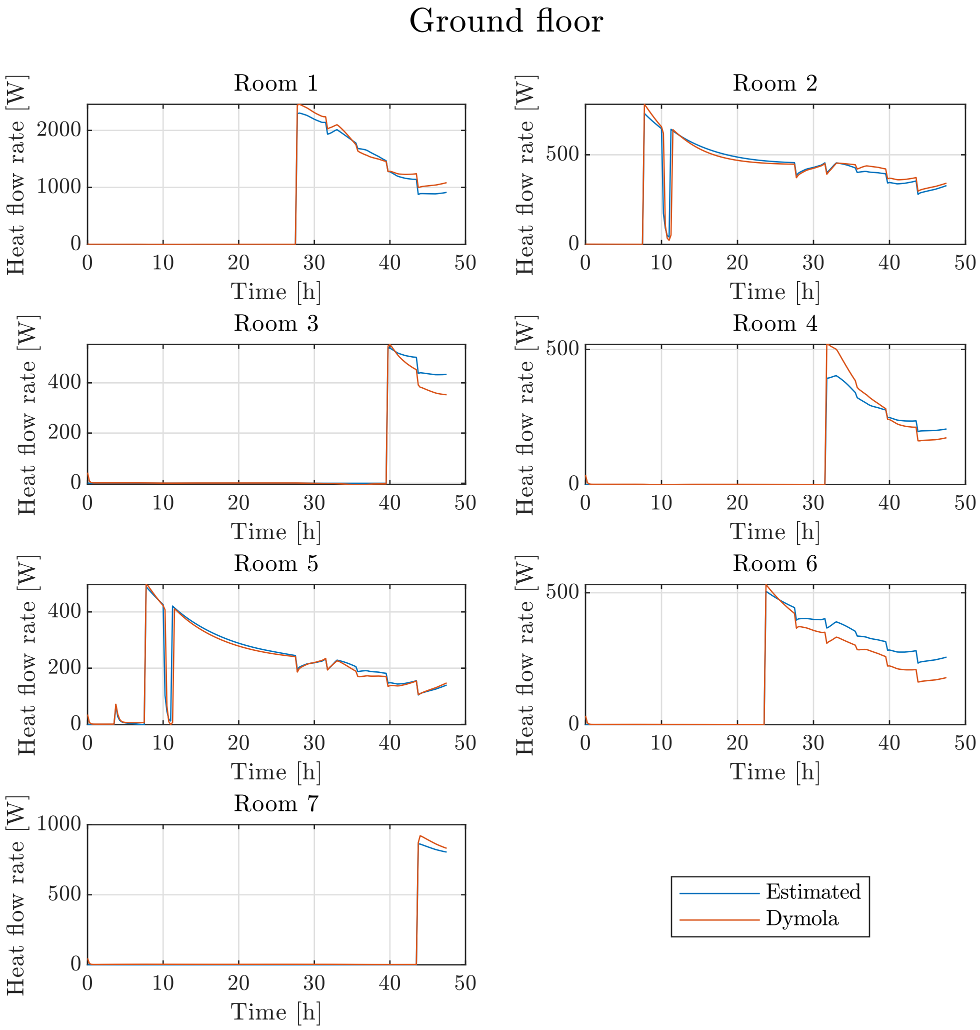

Figure 3.15: Heat flow rate open-loop simulations for the rooms on the ground floor using the validation data set.

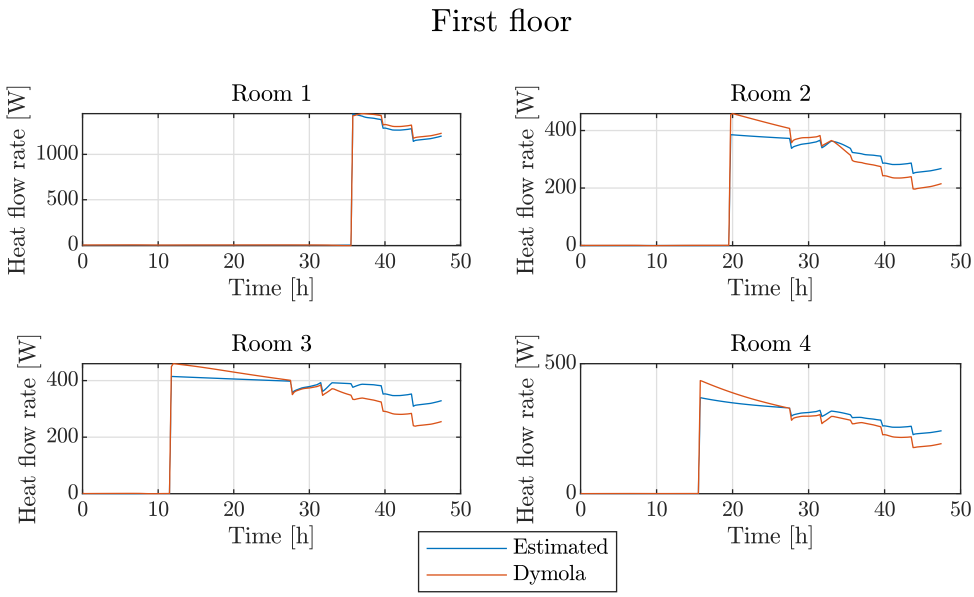

Figure 3.16: Heat flow rate open-loop simulations for the rooms on the first floor using the validation data set.

The final values found by the parameter estimation can be seen in Table 3.3 and 3.4.

Table 3.3: Parameters for the rooms, outer walls and heat inflow.

Room	C_f $\left[\frac{\mathrm{J}}{\mathrm{K}}\right]$	C_r $\left[\frac{\mathrm{J}}{\mathrm{K}}\right]$	s	U_f $\left[\frac{\mathrm{W}}{\mathrm{K}}\right]$	U_a $\left[\frac{\mathrm{W}}{\mathrm{K}}\right]$	b
0,1	1.68E+07	1.59E+04	3.33E+02	1.88E+02	3.83E+00	6.06E−01
0,2	3.41E+06	4.63E+05	6.06E+01	2.40E+03	1.82E+00	8.76E−01
0,3	9.63E+06	1.09E+04	8.63E+01	2.40E+01	5.95E−01	7.20E−01
0,4	3.71E+06	1.45E+05	6.27E+02	2.44E+02	4.07E+00	4.59E−01
0,5	2.48E+06	2.17E+04	2.64E+02	4.81E+01	6.95E−01	5.96E−01
0,6	3.19E+06	4.63E+05	6.06E+01	2.40E+03	3.81E+00	5.49E−01
0,7	5.37E+06	4.63E+05	6.16E+01	3.81E+02	1.76E+01	7.49E−01
1,1	5.01E+07	4.60E+05	5.75E+03	2.36E+03	5.37E+01	9.40E−01
1,2	1.06E+07	4.63E+03	1.27E+03	2.05E+02	1.06E+01	7.35E−01
1,3	4.49E+07	1.50E+04	9.71E+01	3.94E+01	3.91E+00	8.78E−01
1,4	3.96E+06	4.62E+05	1.63E+02	3.99E+02	1.00E+01	6.53E−01

Table 3.4: Parameters for the inner walls.

Inner wall	Connecting rooms	$U_{\mathrm{w}}\ \left[\frac{\mathrm{W}}{\mathrm{K}}\right]$
1	0,1 and 0,2	1.61E+01
2	0,1 and 0,3	5.46E+01
3	0,1 and 0,4	5.46E+01
4	0,2 and 0,3	5.46E−01
5	0,3 and 0,4	2.31E+01
6	0,3 and 0,6	2.91E+01
7	0,4 and 0,5	5.72E+00
8	0,5 and 0,6	3.14E+01
9	0,5 and 0,7	1.46E+01
10	0,6 and 0,7	1.17E+00
11	1,1 and 1,2	1.27E+00
12	1,1 and 1,3	2.23E+00
13	1,1 and 1,4	5.46E−01
14	1,2 and 1,3	9.85E+00
15	1,3 and 1,4	2.65E+01
16	0,1 and 1,1	5.46E+01
17	0,2 and 1,1	5.46E+01
18	0,3 and 1,1	8.80E−01
19	0,4 and 1,2	9.21E+00
20	0,5 and 0,3	6.16E−01
21	0,6 and 0,3	3.24E+01
22	0,6 and 0,4	7.91E+00

As can be seen from the plots, the model closely follows the measured output from the Dymola model.

This concludes the parameter estimation part of the dynamic model. In the next section the linear part of the model is briefly analysed.

3.6.5 Linear model analysis

The linear part of the model is:

$$\begin{aligned} x_{k+1} &= A_{\mathrm{d}}\, x_k + B_{\mathrm{d}}\, u_{\mathrm{rc},k} \\ y_k &= C_{\mathrm{d}}\, x_k \end{aligned} \tag{3.42}$$

where:

$$x_k = \begin{bmatrix} T_{\mathrm{f},i,j} \\ T_{\mathrm{r},i,j} \end{bmatrix}, \qquad y_k = \begin{bmatrix} T_{\mathrm{r},i,j} \end{bmatrix}, \qquad u_{\mathrm{rc},k} = \begin{bmatrix} P_{i,j} \end{bmatrix} \tag{3.43}$$

The invariant zeros for the linear part of the model can be calculated using the matrix:

$$\begin{bmatrix} A_{\mathrm{d}} - s\,\mathbf{I}_{22\times 22} & B_{\mathrm{d}} \\ C_{\mathrm{d}} & 0 \end{bmatrix} \tag{3.44}$$

The invariant zeros are the complex values of s for which the rank of the matrix drops below its normal value. For the discrete time model there are 11 positive, real valued, invariant zeros in the range $[0, 0.5)$.

The poles of the system is calculated as the 22 eigenvalues of the A_d. These are all positive, real valued in the range $[0, 1)$, and thus the system is stable.

Next, the controllability and observability of the model is investigated. To determine if a system is controllable and observable the controllability matrix $\mathfrak{C}$ and observability matrix $\mathfrak{O}$ is needed:

$$\mathfrak{C} = \begin{bmatrix} B_\mathrm{d} & A_\mathrm{d}\,B_\mathrm{d} & A_\mathrm{d}^2\,B_\mathrm{d} & \cdots & A_\mathrm{d}^{n-1}\,B_\mathrm{d} \end{bmatrix}, \qquad \mathfrak{O} = \begin{bmatrix} C_\mathrm{d} \\ C_\mathrm{d}\,A_\mathrm{d} \\ C_\mathrm{d}\,A_\mathrm{d}^2 \\ \vdots \\ C_\mathrm{d}\,A_\mathrm{d}^{n-1} \end{bmatrix} \tag{3.45}$$

where:

n : Number of states

The system is controllable and observable if:

$$\begin{aligned} \mathrm{rank}(\mathfrak{C}) &= n \Rightarrow \text{Controllable} \\ \mathrm{rank}(\mathfrak{O}) &= n \Rightarrow \text{Observable} \end{aligned} \tag{3.46}$$

Which is the case for the discrete time system, as both have rank $n = 22$. Rank calculations of matrices can be subject to numeric instability if the matrices are ill-conditioned. Thus, the condition number is calculated for both.

$$\begin{aligned} \mathrm{cond}(\mathfrak{C}) &= 2.68\mathrm{E}{+}03 \\ \mathrm{cond}(\mathfrak{O}) &= 4.02\mathrm{E}{+}04 \end{aligned} \tag{3.47}$$

Which seem reasonably low. It is thus concluded that the linear part of the dynamic model of the system is controllable and observable.

Next in Subsection 3.6.6 the parameter estimation for the return flow from Subsection 3.5.1 will be described.

3.6.6 Parameter estimation for return flow

In order to estimate the $a_{i,j}$ scaling factors, the state estimates of the floor temperatures, $\hat{T}_{\mathrm{f},i,j,k|k}$, from the Extended Kalman Filter (EKF) from Chapter 6 using the training data set described in Section 2.3 is used.

$$T_{\mathrm{in},k}\, q_{\mathrm{tot},k} = \sum_{i,j} q_{i,j,k}\, \hat{T}_{\mathrm{f},i,j,k|k}\, a_{i,j} \tag{3.48}$$

which can then be reformulated into a matrix-vector equation for all the time step over the two day simulation period sampled every 15 minutes, i.e. 192 samples:

$$X\, a = Y \tag{3.49}$$

where:

$$X = \begin{bmatrix} q_{0,1,0}\, \hat{T}_{\mathrm{f},0,1,0|0} & q_{0,2,0}\, \hat{T}_{\mathrm{f},0,2,0|0} & \cdots & q_{1,4,0}\, \hat{T}_{\mathrm{f},1,4,0|0} \\ q_{0,1,1}\, \hat{T}_{\mathrm{f},0,1,1|1} & q_{0,2,1}\, \hat{T}_{\mathrm{f},0,2,1|1} & \cdots & q_{1,4,1}\, \hat{T}_{\mathrm{f},1,4,1|1} \\ \vdots & \vdots & \ddots & \vdots \\ q_{0,1,191}\, \hat{T}_{\mathrm{f},0,1,191|191} & q_{0,2,191}\, \hat{T}_{\mathrm{f},0,2,191|191} & \cdots & q_{1,4,191}\, \hat{T}_{\mathrm{f},1,4,191|191} \end{bmatrix} \tag{3.50}$$

$$Y = \begin{bmatrix} q_{\mathrm{tot},0}\, T_{\mathrm{in},0} \\ q_{\mathrm{tot},1}\, T_{\mathrm{in},1} \\ \vdots \\ q_{\mathrm{tot},191}\, T_{\mathrm{in},191} \end{bmatrix}, \qquad a = \begin{bmatrix} a_{0,1} \\ a_{0,2} \\ \vdots \\ a_{1,4} \end{bmatrix} \tag{3.51}$$

This is then solved using MATLAB's `mldivide` command , which finds the values of a using a least-squares method [33]. The $a_{i,j}$ values found can be seen in Table 3.5.

Table 3.5: Fitted values of $a_{i,j}$.

$a_{i,j}$	Value
$a_{0,1}$	1.0120
$a_{0,2}$	1.0010
$a_{0,3}$	0.9988
$a_{0,4}$	1.0180
$a_{0,5}$	1.0159
$a_{0,6}$	1.0051
$a_{0,7}$	1.0011
$a_{1,1}$	0.9951
$a_{1,2}$	0.9778
$a_{1,3}$	1.0154
$a_{1,4}$	1.0104

This can then be validated by comparing the left and right hand side of Equation 3.48 for the validation data set from Section 2.3. The plot of this can be seen on Figure 3.17.

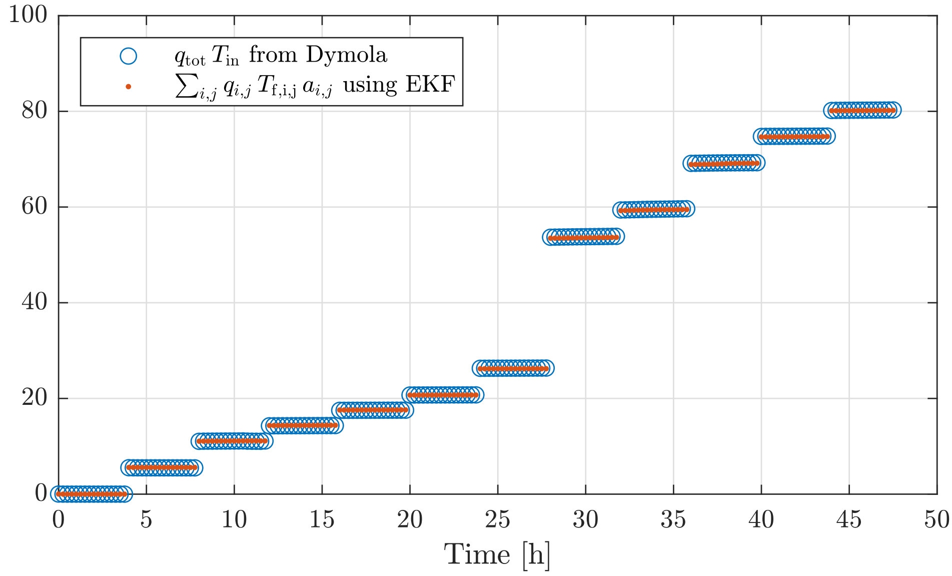

Figure 3.17: Plot of the left and right hand side of Equation 3.48 for a validation data set.

As can be seen from the plot, the deviation between the two is small, and the submodel is thus considered satisfactory.

Figure 3.18 shows the deviation for the calculated heat power consumption P_{h} using Equation 3.34

$$P_{\text{h}} = c\,q_{\text{tot}}\,T_{\text{out}} - c\sum_{i,j} q_{i,j}\,T_{\text{f},i,j}\,a_{i,j} \tag{3.52}$$

compared to the measured P_{h} from the Dymola model.

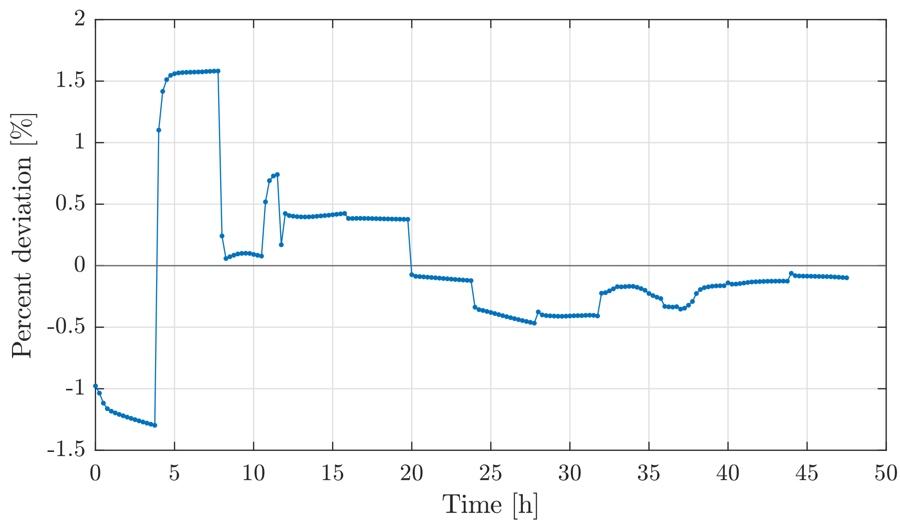

Figure 3.18: Percentage deviation between measured and calculated P_{h} for the validation data set.

Again, it can be seen that the deviation is very low. This concludes the modelling of the lumped parameter multi-zone model. In Chapter 4 the lumped parameter single-zone model will be described.

4 Lumped parameter single-zone

This chapter concerns the development of a lumped parameter single-zone model of the house.

The chapter contains the following sections:

- Section 4.1 concerns the development of the resistor capacitor submodel.
- Section 4.2 describes the parameter estimation of the submodel described in Section 4.1.
- Section 4.3 describes the development of the heat pump model.

Figure 4.1 shows an overview of the two submodels that will be described in this chapter.

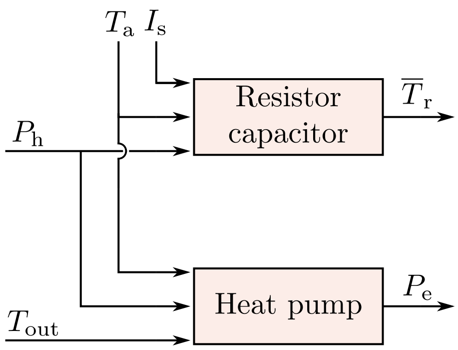

Figure 4.1: Overview of the submodels of the lumped parameter single-zone model.

Next in Section 4.1 the development of a simplified resistor capacitor model for the lumped single-zone model will be described.

4.1 Resistor capacitor model

This section concerns the development of a simplified resistor capacitor model for the lumped single-zone model of the house. Figure 4.2 shows how the model relates to the other submodel of the lumped parameter single-zone model.

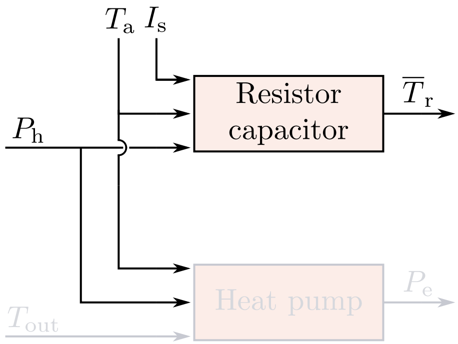

Figure 4.2: Sketch of how the resistor capacitor model relates to the other submodel of the lumped parameter single-zone model.

It follows the ideas and methods presented in Section 3.2 and thus little detail will be given. To achieve the simplicity required by the control strategy, the model only contains a single room component connected to an outer wall component. See subsubsection 3.2.2.1 and 3.2.2.2 respectively. This structure allows modelling of the average behaviour of the entire house. Figure 4.3 shows the electrical equivalent diagram used for this model.

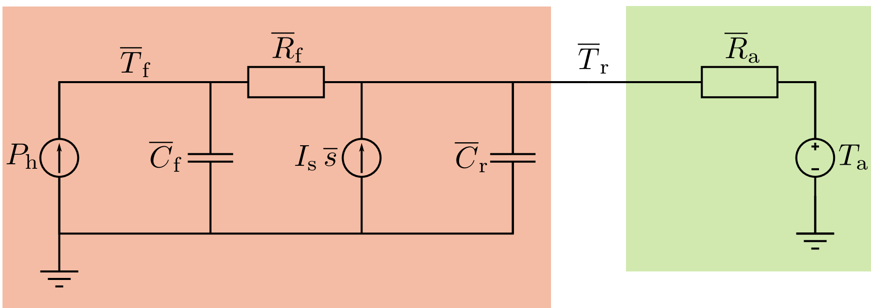

Figure 4.3: Electrical equivalent diagram of the model used for the single-zone model.$\overline{R}_\text{f} = \overline{U}_\text{f}^{-1}$ and $\overline{R}_\text{a} = \overline{U}_\text{a}^{-1}$.

Based on this diagram the model can be described by the equations:

$$\begin{aligned} \overline{C}_\text{f}\,\dot{\overline{T}}_\text{f} &= -\overline{U}_\text{f}\,\overline{T}_\text{f} + \overline{U}_\text{f}\,\overline{T}_\text{r} + P_h \\ \overline{C}_\text{r}\,\dot{\overline{T}}_\text{r} &= -\left(\overline{U}_\text{f} + \overline{U}_\text{a}\right)\overline{T}_\text{r} + \overline{U}_\text{f}\,\overline{T}_\text{f} + \overline{U}_\text{a}\,T_\text{a} + I_\text{s}\,\overline{s} \end{aligned} \tag{4.1}$$

where:

$\overline{T}_\text{f}$	:	Temperature of the floor	K
$\overline{T}_\text{r}$	:	Temperature of the air in the room	K
$\overline{C}_\text{f}$	:	Heat capacity of the floor	$\frac{\text{J}}{\text{K}}$
$\overline{C}_\text{r}$	:	Heat capacity of the air in the room	$\frac{\text{J}}{\text{K}}$
$\overline{U}_\text{f}$	:	Thermal conductance between floor and room	$\frac{\text{W}}{\text{K}}$
P_h	:	Heating power supplied by the heat pump to the water	W
I_s	:	Solar intensity	-
$\overline{s}$	:	Solar scaling factor	W
T_a	:	Temperature of ambient outdoors	K
$\overline{U}_\text{a}$	:	Thermal conductance between the room and ambient surroundings	$\frac{\text{W}}{\text{K}}$

This can then be converted to a linear state space form:

$$\begin{aligned} \dot{x} &= A\,x + B\,u + E\,d \\ y &= C\,x \end{aligned} \tag{4.2}$$

where:

$$\begin{aligned} &x = \begin{bmatrix} \overline{T}_\mathrm{f} \\ \overline{T}_\mathrm{r} \end{bmatrix} && u = \begin{bmatrix} P_\mathrm{h} \end{bmatrix} \quad d = \begin{bmatrix} I_\mathrm{s} \\ T_\mathrm{a} \end{bmatrix} && y = \begin{bmatrix} \overline{T}_\mathrm{r} \end{bmatrix} \\ &A = \begin{bmatrix} -\frac{\overline{U}_\mathrm{f}}{\overline{C}_\mathrm{f}} & \frac{\overline{U}_\mathrm{f}}{\overline{C}_\mathrm{f}} \\ \frac{\overline{U}_\mathrm{f}}{\overline{C}_\mathrm{r}} & -\frac{\overline{U}_\mathrm{f}+\overline{U}_\mathrm{a}}{\overline{C}_\mathrm{r}} \end{bmatrix} && B = \begin{bmatrix} \frac{1}{\overline{C}_\mathrm{f}} \\ 0 \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 \\ \frac{\overline{s}}{\overline{C}_\mathrm{r}} & \frac{\overline{U}_\mathrm{a}}{\overline{C}_\mathrm{r}} \end{bmatrix} && C = \begin{bmatrix} 0 & 1 \end{bmatrix} \end{aligned} \tag{4.3}$$

This concludes the development of the resistor capacitor model structure for the lumped single-zone model of the house. Next Section 4.2 concerns the parameter estimation of the parameters for the simplified resistor capacitor model for the lumped parameter single-zone model.

4.2 Parameter estimation

This section aims to estimate the paramters for the resistor capacitor model for the lumped parameter single-zone model of the house from Section 4.1.

The methods used in this section follows that of Section 3.6. MATLAB's `nlgreyest` command from the System Identification toolbox is used again. It requires a model on the form:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, \theta) \\ y_k &= g(x_k, u_k, \theta) \end{aligned} \tag{4.4}$$

The model from Section 4.1 is on the form:

$$\begin{aligned} \dot{x} &= A\,x + B\,u + E\,d \\ y &= C\,x \end{aligned} \tag{4.5}$$

This is converted to the form

$$\begin{aligned} x_{k+1} &= f(x_k, u'_k, \theta) = A_\mathrm{d}\,x_k + B'_\mathrm{d}\,u'_k \\ y_k &= g(x_k, \theta) = C_\mathrm{d}\,x_k \end{aligned} \tag{4.6}$$

by setting

$$u' = \begin{bmatrix} u \\ d \end{bmatrix} = \begin{bmatrix} P_\mathrm{h} \\ I_\mathrm{s} \\ T_\mathrm{a} \end{bmatrix} \qquad B' = \begin{bmatrix} B & E \end{bmatrix} \tag{4.7}$$

and discretizing the matrices using the Zero Order Hold (ZOH) method

$$A_\mathrm{d} = \mathrm{e}^{A\,T_\mathrm{s}}, \qquad B'_\mathrm{d} = A^{-1}\,(A_\mathrm{d} - \mathbf{I}_{2\times 2})\,B' \tag{4.8}$$

where:

T_s	:	Sample time of the system	s
$\mathbf{I}$	:	Identity matrix	

The input P_{h} is calculated as described in Subsection 3.5.2:

$$P_{\mathrm{h}} = c\, q_{\mathrm{tot}}\, (T_{\mathrm{out}} - T_{\mathrm{in}}) \tag{4.9}$$

The output $\overline{T}_{\mathrm{r}}$ is calculated as a weighted average of all the room temperatures. The weights used are the room's floor area.

$$\overline{T}_{\mathrm{r}} = \frac{1}{\sum_{i,j} S_{i,j}} \sum_{i,j} T_{\mathrm{r},i,j}\, S_{i,j} \tag{4.10}$$

where:

$S_{i,j}$: Floor area of room i, j m^2

All signals are downsampled to the sample time $T_{\mathrm{s}} = 3600\,\mathrm{s}$.
The parameters to be estimated are:

$$\begin{aligned} &\theta = \begin{bmatrix}\overline{C}_{\mathrm{f}} & \overline{C}_{\mathrm{r}} & \overline{U}_{\mathrm{f}} & \overline{U}_{\mathrm{a}} & \overline{s}\end{bmatrix}^{\mathrm{T}} \\ &\theta \geq 0 \end{aligned} \tag{4.11}$$

The initial guesses for these are the same as from Subsection 3.6.3:

$$\begin{aligned} &\overline{U}_{\mathrm{f}} = 1.00\mathrm{E}+03\,\frac{\mathrm{W}}{\mathrm{K}}, \quad &&\overline{U}_{\mathrm{a}} = 1.00\mathrm{E}+01\,\frac{\mathrm{W}}{\mathrm{K}}, \quad &&\overline{C}_{\mathrm{f}} = 1.00\mathrm{E}+07\,\frac{\mathrm{J}}{\mathrm{K}} \\ &\overline{C}_{\mathrm{r}} = 1.00\mathrm{E}+06\,\frac{\mathrm{J}}{\mathrm{K}}, \quad &&\overline{s} \quad = 1.00\mathrm{E}+03 \end{aligned} \tag{4.12}$$

In the following subsection the results of the parameter estimation is presented.

4.2.1 Results

Figure 4.4 shows the linear model's open loop response on the validation data set.

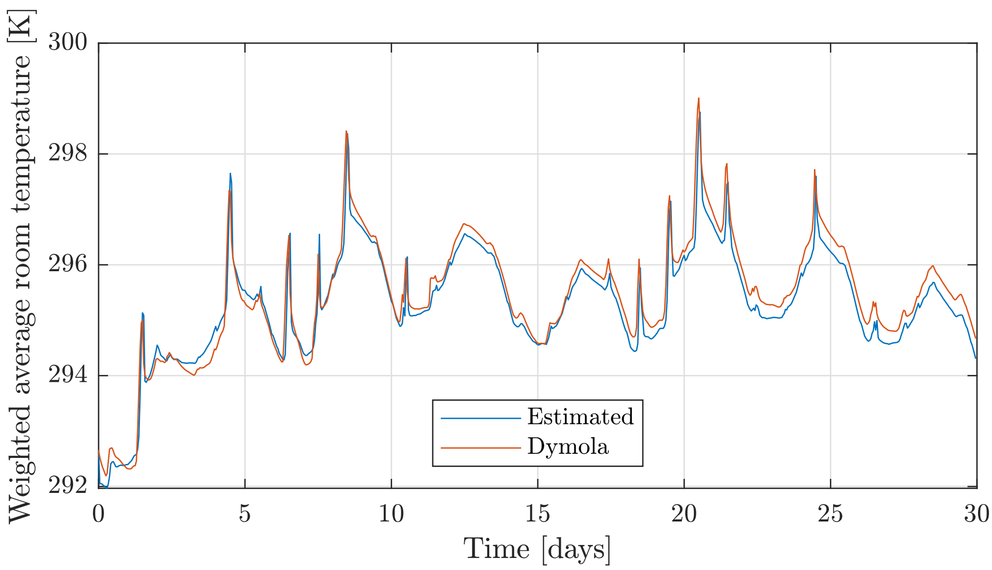

Figure 4.4: Open loop response using the validation data set.

The simulated weighted average room temperature closely follows that of the Dymola model and is therefore considered satisfactory. Table 4.1 shows the found parameters.

Table 4.1: Parameters for the resistor capacitor model of the lumped single-zone model of the house.

$\overline{C}_\mathrm{f}$ $\left[\frac{\mathrm{J}}{\mathrm{K}}\right]$	$\overline{C}_\mathrm{r}$ $\left[\frac{\mathrm{J}}{\mathrm{K}}\right]$	s	$\overline{U}_\mathrm{f}$ $\left[\frac{\mathrm{W}}{\mathrm{K}}\right]$	$\overline{U}_\mathrm{a}$ $\left[\frac{\mathrm{W}}{\mathrm{K}}\right]$
9.05E+07	1.80E+06	6.14E+03	3.48E+03	1.04E+02

Next, in Section 4.3 the heat pump model will be described.

4.3 Heat pump model

This section contains the description and model of the heat pump. Figure 4.5 shows how the model of this section relates to the resistor capacitor model described in Section 4.1.

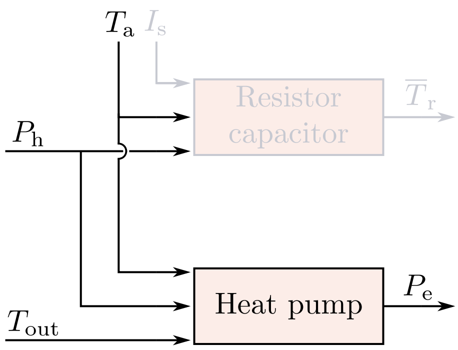

Figure 4.5: Sketch of how the model of this section relates to the other submodels.

It contains the following subsections

- Subsection 4.3.1 gives a description of the working principle behind heat pumps, and their theoretical optimum the Carnot engine.
- Subsection 4.3.2 describes a model of how electric power supplied to the heat pump is transformed into the heat power supplied by the heat pump.
- Subsection 4.3.3 concerns the validation of the heat pump using the the validation data set for 2 days and the estimated floor temperatures from the EKF developed in Chapter 6.

4.3.1 Heat pump description

A heat engine produces useful work when energy flows in its natural direction from a hot reservoir to a cold reservoir as per the second law of thermodynamics.

A heat pump is the opposite of a heat engine. A heat pump uses work to transfer heat from a cold reservoir to a hot reservoir. Heat pumps are used extensively to heat and cool buildings. This report focuses on using the heat pump to generate heat for the heating of a residential buildings. Thus, the cooling cycle will not be further discussed.

Figure 4.6 shows a diagram of the heat pump Bosch Compress 6000 AW installed at the house. It is capable of both heating and cooling, but is here shown with valve positions as per a heating cycle.

AW refers to the fact it is an Air to Water Heat Pump (AWHP), where heat is taken from outside air and transferred to the hot water network used for underfloor heating.

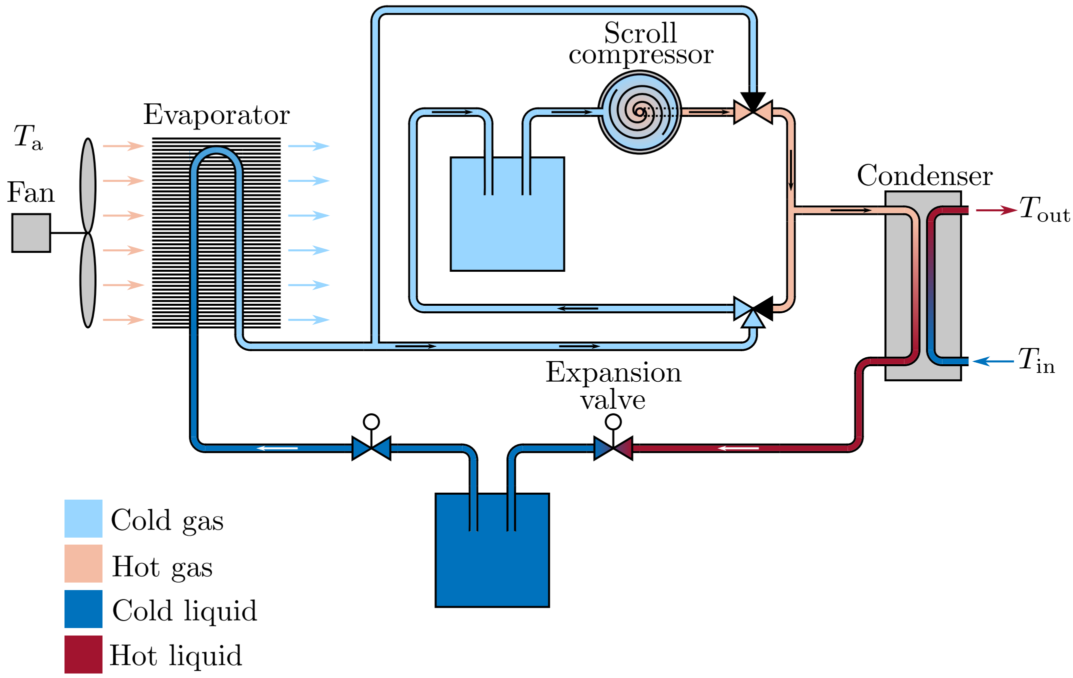

Figure 4.6: Sketch of the working components of the Bosch Compress 6000 AW in its heating configuration based on diagram from [19, p. 8].

A heating cycle can be described as a 4 step process:

1. Inside the evaporator, the liquid refrigerant evaporates into gas by absorbing energy from the outdoor air.
2. The compressor increases the pressure and thus the temperature of the refrigerant gas.
3. In the condenser, the hot refrigerant releases energy by condensing back into a liquid.
4. An expansion valve reduces the pressure of the liquid, thus lowering its temperature before returning to the evaporator.

The main point of interest for the modelling of the heat pump is the conversion from work done on the system to output, in this case electricity, energy. Thus we first examine the upper limit for the efficiency of work to energy transfer of the system: a Carnot engine, as per Carnot's theorem: "No real heat engine operating between two energy reservoirs can be more efficient than a Carnot engine operating between the same two reservoirs." [24, p. 675]. Coefficient of Performance (COP) is a often used measure for the efficiency of a heat pump defined as

$$\mathrm{COP} = \frac{Q}{W} \tag{4.13}$$

where:

Q	:	Heat supplied by the heat pump	J
W	:	Net work put into the system	J

The COP of a Carnot heat pump is given by

$$\mathrm{COP}_{\mathrm{carnot}} = \frac{T_{\mathrm{h}}}{T_{\mathrm{h}} - T_{\mathrm{c}}} \tag{4.14}$$

where:

T_{h}	:	Temperature of the hot reservoir	K
T_{c}	:	Temperature of the cold reservoir	K

As per Carnot's theorem, the heat pump installed is of course not ideal. Thus, to model the heat pump in question a heuristic model is developed to modify the COP based on operating conditions.

The heat outputted from the heat pump is used for heating of the water circulating the floor heating system.

4.3.2 Coefficient of Performance

In this section the The model used for the COP of the heat pump is a heuristic model developed by Simon Thorsteinsson using data from the real system [1]. First, Equation 4.13 is rewritten and the time derivative is found.

$$\begin{aligned} Q &= \mathrm{COP}\, W \\ &\Downarrow \text{ Assuming constant COP} \\ P_{\mathrm{h}} &= \mathrm{COP}\, P_{\mathrm{e}} \end{aligned} \tag{4.15}$$

where:

P_{e}	:	Electric power supplied to the heat pump	W

A good fit for the COP of the system has been found by Simon Thorsteinsson [1] as:

$$\mathrm{COP} = \left(c_2\, P_{\mathrm{e}} + c_1 + \frac{c_0}{P_{\mathrm{e}}} \right) \mathrm{COP}_{\mathrm{carnot}} + \frac{k_0}{P_{\mathrm{e}}} \tag{4.16}$$

where c_0, c_1, c_2, and k_0 are coefficients found through the fitting process. The COP formula can be seen as consisting of a $\mathrm{COP}_{\mathrm{carnot}}$ term modified by a function dependent on the work W and an offset term also dependent on W. Figure 4.7 shows the P_{e} and COP relation for different values of $\mathrm{COP}_{\mathrm{carnot}}$.

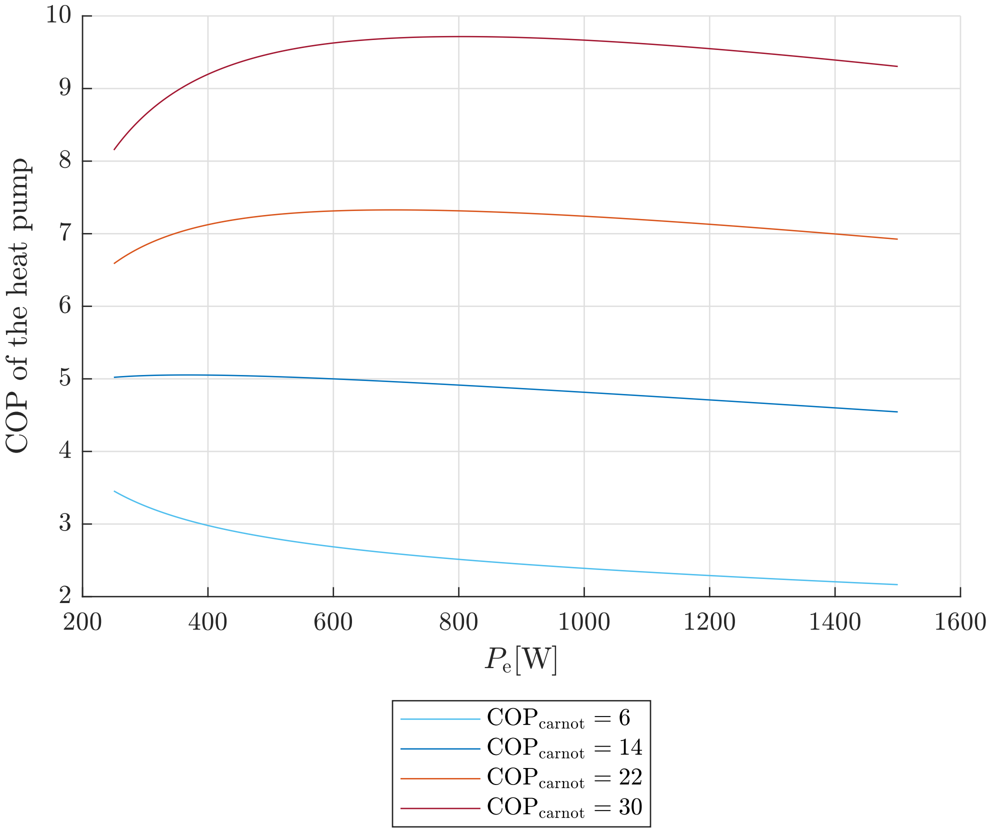

Figure 4.7: COP values for different P_e and COP_{carnot}.

In the case of the heat pump Equation 4.14 can be written as:

$$COP_{carnot} = \frac{T_{out}}{T_{out} - T_a} \tag{4.17}$$

where:

T_a	:	Temperature of the ambient outdoors	K

Figure 4.8 shows the COP_{carnot} for different values of T_a and T_{out}.

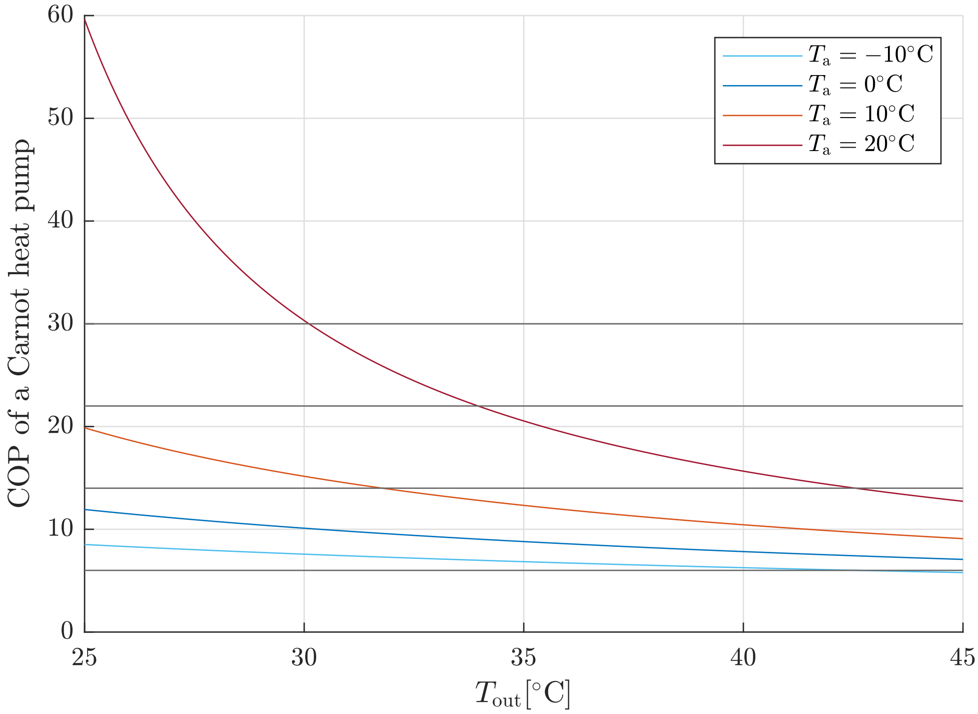

Figure 4.8: $\text{COP}_{\text{carnot}}$ for different values of T_a and T_{out}. Horizontal black lines are for the $\text{COP}_{\text{carnot}}$ values used in Figure 4.7.

Substituting in and rewriting yields

$$P_h = k_0 + \left(c_0 + c_1\, P_e + c_2\, P_e^2\right) \frac{T_{\text{out}}}{T_{\text{out}} - T_a} \tag{4.18}$$

The values of the coefficients in the model can be seen in Table 4.2.

Table 4.2: Coefficient for the heat pump COP model.

k_0	c_0	c_1	c_2
5.703E+02	−4.644E+01	3.921E−01	−4.245E−05

The heat pump can only operate in certain P_e range. The minimum and maximum values of this range is:

$$\min P_e = 250\,\text{W} \qquad \max P_e = 1500\,\text{W} \tag{4.19}$$

If the heat pump turns off, if for example less electric power is needed than in the operating range, then the heat pump can not be restarted for 3 hours. It also requires a minimum up time of 1 hour.

Figure 4.9 shows a the P_h and P_e relation for the operating range of the heat pump for different $\text{COP}_{\text{carnot}}$ values.

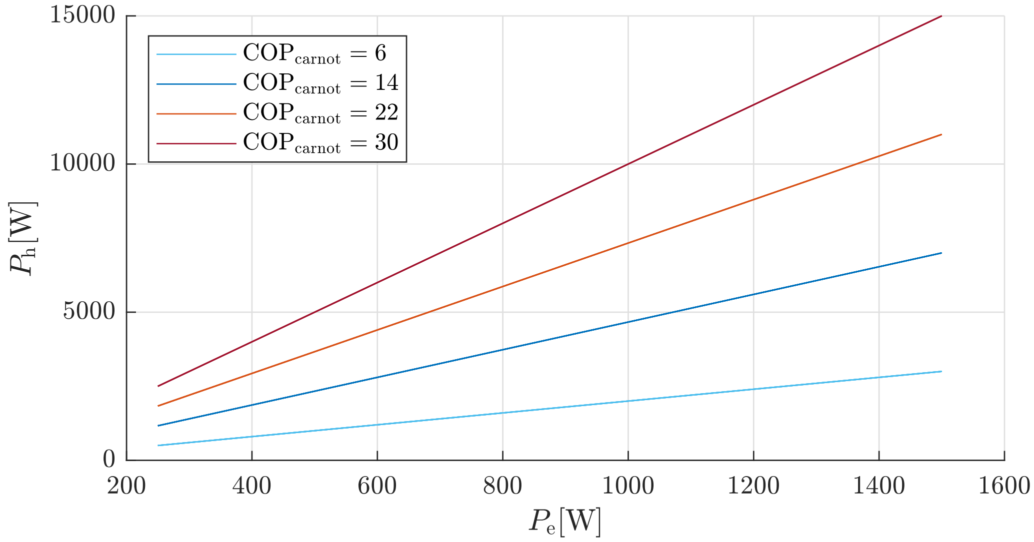

Figure 4.9: P_{h} and P_{e} relation for the operating range of the heat pump for different $\mathrm{COP}_{\mathrm{carnot}}$ values.

Finally, to validate the heat pump model, the $T_{\mathrm{in}}\, q_{\mathrm{tot}}$ estimates from the model from Subsection 3.5.1 and $T_{\mathrm{in}}\, q_{\mathrm{tot}}$ from the Dymola model output are used to calculate the required P_{e}.

4.3.3 Results

Taking q_{tot} and T_{out} directly from the validation data set, P_{h} is calculated using:

$$P_{\mathrm{h}} = c\, q_{\mathrm{tot}}\, T_{\mathrm{out}} - c\, q_{\mathrm{tot}}\, T_{\mathrm{in}} \tag{4.20}$$

both using $T_{\mathrm{in}}\, q_{\mathrm{tot}}$ from the data set and using $\sum_{i,j} q_{i,j,k}\, \hat{T}_{\mathrm{f},i,j,k|k}\, a_{i,j}$ as described in Subsection 3.5.1.

Next Equation 4.18 is solved for P_{e} using the quadratic formula:

$$P_{\mathrm{e}} = \frac{-c_1 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}} \pm \sqrt{\left(c_1 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}\right)^2 - 4\left(c_2 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}\right)\left(-P_{\mathrm{h}} + k_0 + c_0 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}\right)}}{2\, c_2 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}} \tag{4.21}$$

Only the solution where $\pm$ is $+$ is required.

$$P_{\mathrm{e}} = \frac{-c_1 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}} + \sqrt{\left(c_1 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}\right)^2 - 4\left(c_2 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}\right)\left(-P_{\mathrm{h}} + k_0 + c_0 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}\right)}}{2\, c_2 \frac{T_{\mathrm{out}}}{T_{\mathrm{out}}-T_{\mathrm{a}}}} \tag{4.22}$$

If the calculated P_{e} values goes below $\min P_{\mathrm{e}} = 250\,\mathrm{W}$ it is set equal to $0\,\mathrm{W}$, and if it goes above $\max P_{\mathrm{e}} = 1500\,\mathrm{W}$ it is set equal to $1500\,\mathrm{W}$. A plot of this is shown in Figure 4.10, where the P_{e} from the validation data set is compared to calculated values both using $T_{\mathrm{in}}\, q_{\mathrm{tot}}$ from the data set and using $\sum_{i,j} q_{i,j,k}\, \hat{T}_{\mathrm{f},i,j,k|k}\, a_{i,j}$.

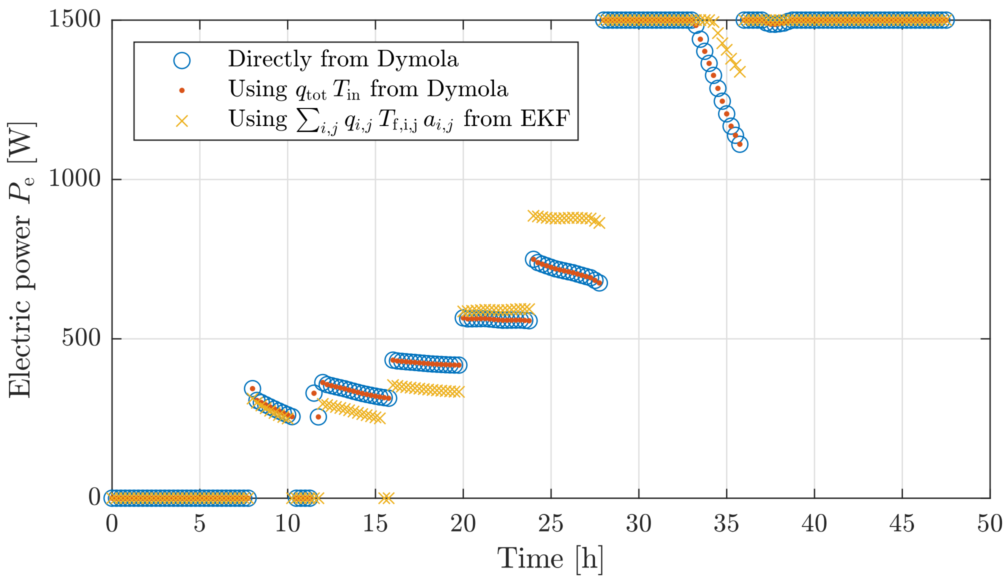

Figure 4.10: Electric power consumption of the heat pump using the validation data set.

On the figure it can be seen that there is a very good correspondance between the P_{e} directly from Dymola and calculated using $T_{\mathrm{in}}\, q_{\mathrm{tot}}$. However, even though there is very little difference between $T_{\mathrm{in}}\, q_{\mathrm{tot}}$ and $\sum_{i,j} q_{i,j,k}\, \hat{T}_{\mathrm{f},i,j,k|k}\, a_{i,j}$ as seen in Figure 3.17, the calculated P_{e} using $\sum_{i,j} q_{i,j,k}\, \hat{T}_{\mathrm{f},i,j,k|k}\, a_{i,j}$ deviates a lot. This high sensitivity to the error of the calculation of the return flow submodel is assumed to be caused by the non-linear nature of the heat pump model.

As an example, consider the heat pump using parameters as seen in Table 4.3.

Table 4.3: Example values.

T_{a}	T_{out}	T_{in}	q_{tot}
$0°\mathrm{C} = 273.15\,\mathrm{K}$	$35°\mathrm{C} = 308.15\,\mathrm{K}$	$25°\mathrm{C} = 298.15\,\mathrm{K}$	$0.0648\,\frac{\mathrm{kg}}{\mathrm{s}}$

Calculating $T_{\mathrm{in}}\, q_{\mathrm{tot}}$ yields 19.32. Using this the electric power consumption is calculated as 809.1 W. Increasing $T_{\mathrm{in}}\, q_{\mathrm{tot}}$ by 1% to 19.51 changes the calculated power consumption to 535.2 W, which is almost 34% lower.

This concludes the modelling part of this master thesis. Next, the control strategy is developed.

Part III

Control design

5 Control design overview

This chapter gives an overview over the developed control strategy. Figure 5.1 shows the general overview of the different elements in the control strategy.

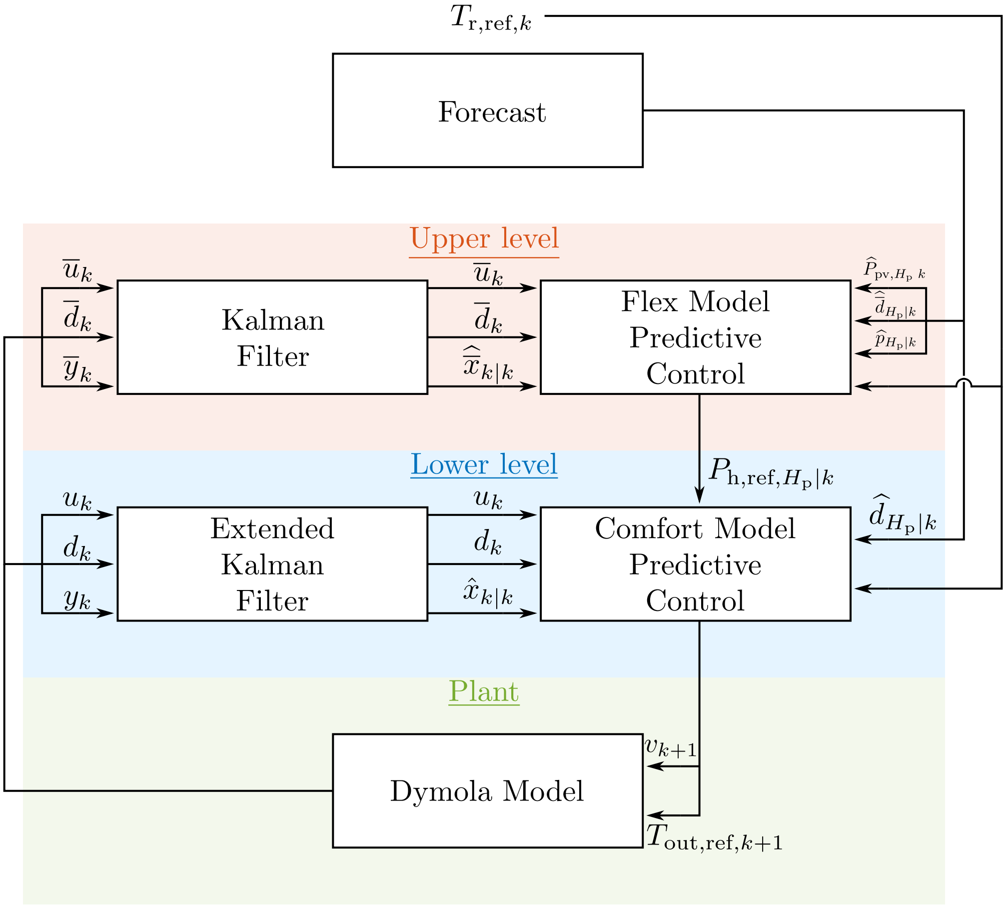

Figure 5.1: Simplified block diagram of how the different control modules are connected with each other, forecast data, and the Dymola model described in Subsection 1.3.2. See Table 5.1 for a description of the parameters.

Table 5.1: Variables of the control strategy

$\widehat{P}_{\mathrm{pv},H_\mathrm{p}\mid k}$	Predicted power production of the PV panels over the entire prediction horizon.
$\widehat{\overline{d}}_{H_\mathrm{p}\mid k}$	Prediction of the disturbances T_a and I_s over the entire prediction horizon.
$\widehat{p}_{H_\mathrm{p}\mid k}$	Predicted electricity prices over the entire prediction horizon.
$\widehat{d}_{H_\mathrm{p}\mid k}$	Prediction of the disturbances T_a and I_s over the entire prediction horizon.
$\overline{u}_k$	Current input consisting of P_h calculated as $P_\mathrm{h} = c\, q_\mathrm{tot}\,(T_\mathrm{out} - T_\mathrm{in})$.
$\overline{d}_k$	Current disturbances consisting of T_a and I_s.
$\overline{y}_k$	Current output consisting of $\overline{T}_\mathrm{r}$ calculated as the weighted average of $T_{\mathrm{r},i,j}$.
u_k	Current input consisting of T_out and $q_{i,j}$ calculated as $q_{i,j} = \overline{q}_{i,j}\, v_{i,j}$.
d_k	Current disturbances consisting of T_a and I_s.
y_k	Current output consisting of $T_{\mathrm{r},i,j}$.
$\overline{x}_{k\mid k}$	State estimate of the lumped parameter single-zone model.
$x_{k\mid k}$	State estimate of the lumped parameter multi-zone model.
$\overline{T}_{\mathrm{r,ref},k}$	Weighted average room temperature reference.
$P_{\mathrm{h,ref},H_\mathrm{p}\mid k}$	Heat power reference over the entire prediction horizon.
$T_{\mathrm{r,ref},i,j,k}$	Individual room temperature references.
v_{k+1}	Optimal valve positions $v_{i,j}$ at time $k+1$.
$T_{\mathrm{out,ref},k+1}$	Optimal forward temperature of the heat pump at time $k+1$.

- Forecast:
  - Goal: Supply forecast data to the model predictive controllers of the two layers of the hierarchical control scheme.
  - Note, no predictor for the forecast has been developed and no noise has been added. Thus, the predictions used in the following chapters are the actual true values in the future. See Subsection 11.1.2 for a brief discussion of this.
- Extended Kalman Filter (EKF):
  - Goal: Estimate the states x of the lumped parameter multi-zone model to be used in the comfort MPC.
  - Sample time: 15 minutes.
  - Chapter 6.
- Kalman Filter (KF):
  - Goal: Estimate the states $\overline{x}$ of the lumped parameter single-zone model to be used in the flex MPC.
  - Sample time: 60 minutes.
  - Chapter 7.
- Comfort Model Predictive Control (MPC):
  - Goal: Maintain individual room temperature references while following the heating power budget from the flex MPC.
  - Sample time: 15 minutes.
  - Chapter 8.

- Flex Model Predictive Control (MPC):
    - Goal: Calculate an optimal heating power budget for the comfort MPC while maintaining the average room temperature at its reference.
    - Sample time: 60 minutes.
    - Chapter 9.
- Dymola model
    - Goal: Simulate house dynamics given the inputs from the control scheme.
    - Sample time: 1 minute, but uses an adaptive step solver.

The following chapter concerns the development of the state observer for the lumped parameter multi-zone model which is the Extended Kalman Filter (EKF).

6 State observer for multi-zone model

This chapter concerns the development of a state observer for the lumped parameter multi-zone model described in Chapter 3.

The chapter contains the following sections:

- Section 6.1 gives an introduction and motivation for the choice of state observer.
- Section 6.2 presents the algorithm used for the state observer.
- Section 6.3 shows the results from the state observers.

6.1 Introduction

In order to observe the states that are not measureable i.e the floor temperatures $T_{\mathrm{f},i,j}$ and to filter potential noise from the measureable states $T_{\mathrm{r},i,j}$ it has been chosen to use an Extended Kalman Filter (EKF) as the final model derived in Chapter 2 has non-linear elements as seen in Equation 6.1. An EKF is a non-linear version of a Kalman Filter (KF) which linearises state equations about an estimate of the current mean and covariance for each time step.

The system for which the EKF is developed is:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, d_k) \\ &= A_\mathrm{d}\, x_k + E_\mathrm{d}\, d_k + \underbrace{B_\mathrm{d}(F\, u_k\, G\, u_k - F\, u_k \odot H x_k)}_{\text{Non-linear part}} \\ y_k &= C_\mathrm{d}\, x_k \end{aligned} \tag{6.1}$$

where:

$$\begin{aligned} x_k &= \begin{bmatrix} T_{\mathrm{f},i,j} \\ T_{\mathrm{r},i,j} \end{bmatrix}, \qquad y_k = \begin{bmatrix} T_{\mathrm{r},i,j} \end{bmatrix} \\ u_k &= \begin{bmatrix} q_{i,j} \\ T_\mathrm{out} \end{bmatrix}, \qquad d_k = \begin{bmatrix} I_s \\ T_\mathrm{a} \end{bmatrix} \end{aligned} \tag{6.2}$$

6.2 Extended Kalman filter algorithm

In the following the implemented EKF algorithm is described. For an in depth description of the method and argumentation for the choice of EKF see Appendix A.

State equations:

$$x_{k+1} = f(x_k, u_k, d_k) + w_k, \qquad w_k \overset{i.i.d.}{\sim} \mathcal{N}(0, Q_k) \tag{6.3}$$

$$y_k = C_\mathrm{d}\, x_k + v_k, \qquad v_k \overset{i.i.d.}{\sim} \mathcal{N}(0, R_k) \tag{6.4}$$

Initial condition:

$$x_0 \in \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{6.5}$$

Measurement update after receiving y_k, u_k:

$$\begin{aligned}
\hat{y}_{k|k-1} &= C_\mathrm{d}\, \hat{x}_{k|k-1} \\
\tilde{y}_{k|k-1} &= y_k - \hat{y}_{k|k-1} \\
K_k &= P_{k|k-1}\, C_\mathrm{d}^\mathrm{T} \left(C_\mathrm{d}\, P_{k|k-1}\, C_\mathrm{d}^\mathrm{T} + R_k \right)^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k\, \tilde{y}_{k|k-1} \\
P_{k|k} &= (I - K_k\, C_\mathrm{d})\; P_{k|k-1}\; (I - K_k\, C_\mathrm{d})^\mathrm{T} + K_k\, R_k\, K_k^\mathrm{T}
\end{aligned} \tag{6.6}$$

Time update:

$$\begin{aligned}
\hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, u_k, d_k) \\
A_k &= \left. \frac{\partial f(x, u, d)}{\partial x^\mathrm{T}} \right|_{\hat{x}_{k|k}, u_k, d_k} \\
P_{k+1|k} &= A_k\, P_{k|k}\, A_k^\mathrm{T} + Q_k
\end{aligned} \tag{6.7}$$

For initial values of covariance $P_{0|0}$ is set to be the identity matrix and initial values of x_0 is set to be 293.15 K.

The covariance matrices Q_k and R_k from Equation 6.3 and Equation 6.4 are initially assumed to be time independent and diagonal. This means that only the variances Q_{ii} and R_{ii} of w_k and v_k have an influence on the filter. They are set to be identity matrices based on [34].

In order to linearise $f(x_k, u_k, d_k)$ for Equation 6.7 partial derivative for x^T has been derived as

$$A_k = A_\mathrm{d} + B_\mathrm{d} \left[\begin{array}{cccc:cccc}
-c\, b_{0,1}\, q_{0,1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & -c\, b_{0,2}\, q_{0,2} & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & -c\, b_{1,4}\, q_{1,4} & 0 & 0 & \cdots & 0
\end{array} \right] \tag{6.8}$$

In the next section the results of the EKF are presented.

6.3 Results

Figure 6.1 and Figure 6.2 shows test results of the implemented EKF used on a validation data set.

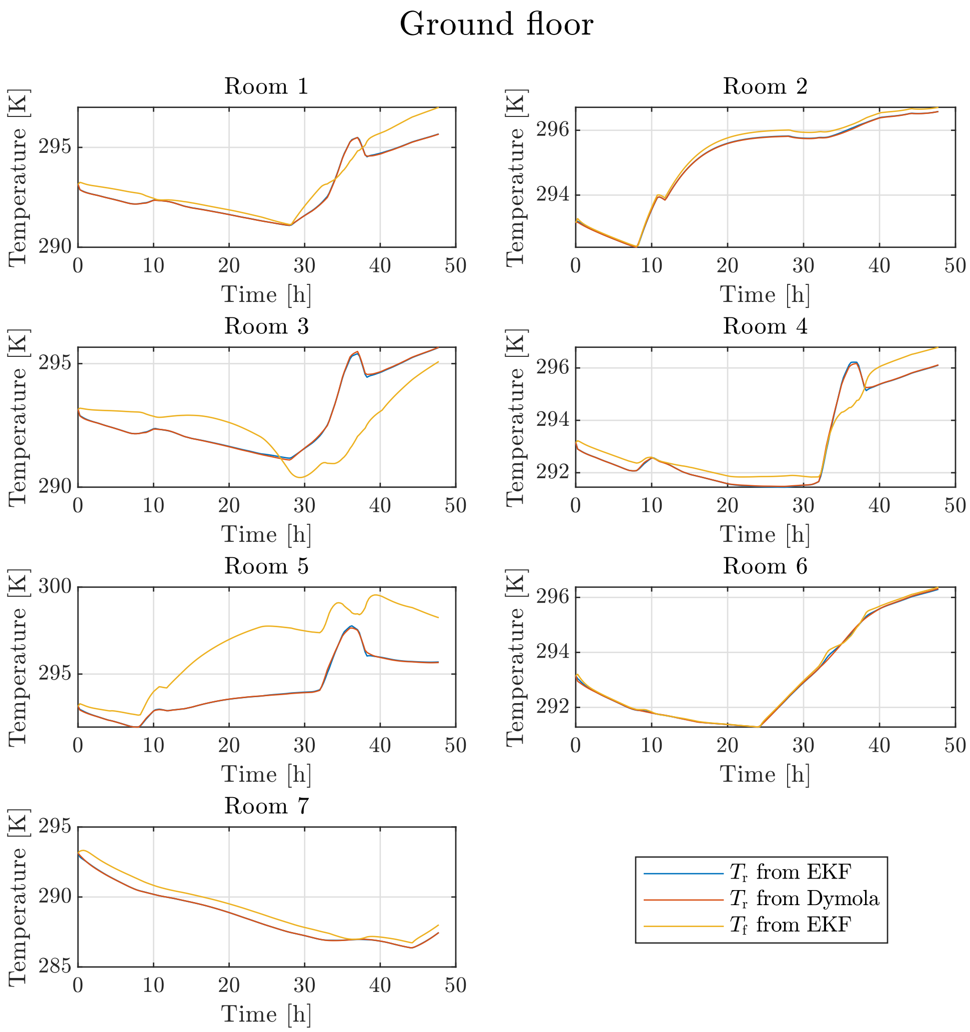

Figure 6.1: Ground floor results from the EKF using the validation data set.

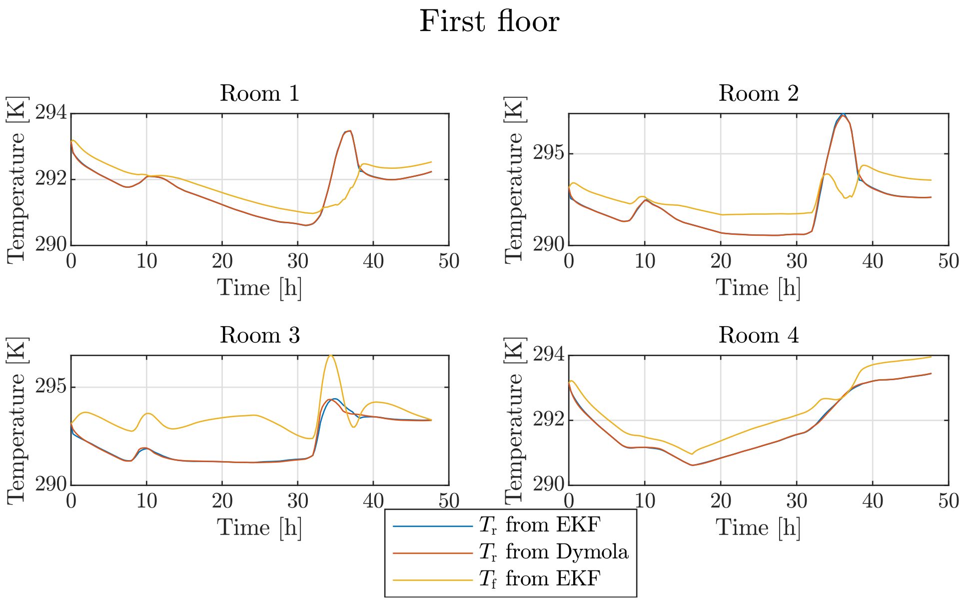

Figure 6.2: First floor results from the EKF using the validation data set.

For both floors it can be seen that the estimated states $T_{r,i,j}$ follows closely the measured $T_{r,i,j}$ from Dymola very closely. Furthermore, $T_{f,i,j}$ estimates from the EKF are mostly estimated higher than $T_{r,i,j}$ as expected since the floor is used for heating.

In the follow chapter the state observer for the upper level using the lumped parameter single-zone model is developed.

7 State observer for single-zone model

In this chapter a state observer for the resistor capacitor model from the lumped parameter single-zone model will be developed. As the model is linear a normal Kalman Filter (KF) will be used.

The system is on the following form from Section 4.2:

$$\begin{aligned} x_{k+1} &= A\,x_k + B'\,u'_k \\ y_k &= C\,x_k \end{aligned} \tag{7.1}$$

where:

$$x = \begin{bmatrix} \overline{T}_{\mathrm{f}} \\ \overline{T}_{\mathrm{r}} \end{bmatrix} \qquad u' = \begin{bmatrix} u \\ d \end{bmatrix} = \begin{bmatrix} P_{\mathrm{h}} \\ I_{\mathrm{s}} \\ T_{\mathrm{a}} \end{bmatrix} \tag{7.2}$$

and recall the input P_{h} is calculated as described in Subsection 3.5.2:

$$P_{\mathrm{h}} = c\,q_{\mathrm{tot}}\,(T_{\mathrm{out}} - T_{\mathrm{in}}) \tag{7.3}$$

The output $\overline{T}_{\mathrm{r}}$ is calculated as a weighted average of all the room temperatures. The weights used are the room's floor area.

$$\overline{T}_{\mathrm{r}} = \frac{1}{\sum_{i,j} S_{i,j}} \sum_{i,j} T_{\mathrm{r},i,j}\, S_{i,j} \tag{7.4}$$

where:

$S_{i,j}$	:	Floor area of room i, j	m^2

All signals are downsampled to the sample time $T_{\mathrm{s}} = 3600\,\mathrm{s}$. Following shows the KF algorithm used for computing the estimated weighted average room temperature $\overline{T}_{\mathrm{r}}$ and weighted average floor temperature $\overline{T}_{\mathrm{F}}$.

State equations:

$$\begin{aligned} x_{k+1} &= A\,x_k + B'\,u_k + w_k, && w_k \overset{i.i.d.}{\sim} \mathcal{N}(0,Q) \\ y_k &= C\,x_k + v_k, && v_k \overset{i.i.d.}{\sim} \mathcal{N}(0,R) \end{aligned} \tag{7.5}$$

Initial condition:

$$x_0 \in \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{7.6}$$

Measurement update after receiving y_k and u_k:

$$\begin{aligned} \hat{y}_{k|k-1} &= C\,\hat{x}_{k|k-1} \\ \tilde{y}_{k|k-1} &= y_k - \hat{y}_{k|k-1} \\ K_k &= P_{k|k-1}\,C^{\mathrm{T}}\,\left(C\,P_{k|k-1}\,C^{\mathrm{T}} + R\right)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k\,\tilde{y}_{k|k-1} \\ P_{k|k} &= (I - K_k\,C)\;P_{k|k-1}\;(I - K_k\,C)^{\mathrm{T}} + K_k\,R\,K_k^{\mathrm{T}} \end{aligned} \tag{7.7}$$

Time update:

$$\begin{aligned} \hat{x}_{k+1|k} &= A\,\hat{x}_{k|k} + B'\,u_k \\ P_{k+1|k} &= A\,P_{k|k}\,A^{\mathrm{T}} + Q \end{aligned} \tag{7.8}$$

For the weights Q and R identity matrices has been used with satisfactory results. Next in Chapter 8 the model predictive controller for the comfort MPC will be described.

8 Comfort MPC

This chapter will concern the development of a Model Predictive Control (MPC) for the lumped parameter multi-zone model developed in Chapter 3.

It contains the following sections:

- Section 8.1 describes the motivation and goals of the Model Predictive Control (MPC) developed in this chapter.
- Section 8.2 gives an introduction to the Mixed Logical Dynamical (MLD) framework, how it results in a Mixed Integer Quadratic Programming (MIQP) problem and how it is solved.
- Section 8.3 concerns the conversion of the lumped parameter multi-zone model from Chapter 3 into the MLD framework.
- Section 8.4 describes the control law for the comfort MPC formulated as a MIQP problem.

8.1 Motivation

As stated in Section 1.6 and Chapter 5 the overall goal of the comfort MPC presented in this chapter is to track the temperature reference for each room in the house, while also following a heating power budget calculated by the flex MPC.

The model developed for the use in this layer of the control scheme is the lumped parameter multi-zone model from Section 3.4:

$$\begin{aligned} x_{k+1} &= A_\mathrm{d}\, x_k + E_\mathrm{d}\, d + B_\mathrm{d}\, \left(F\, u_k\, G\, u_k - F\, u_k \odot H x_k\right) \\ T_{\mathrm{r},i,j} &= C_\mathrm{d}\, x_k \end{aligned} \tag{8.1}$$

where:

$$x = \begin{bmatrix} T_{\mathrm{f},i,j} \\ T_{\mathrm{r},i,j} \end{bmatrix} \qquad y = \begin{bmatrix} T_{\mathrm{r},i,j} \end{bmatrix} \qquad u = \begin{bmatrix} q_{i,j} \\ T_\mathrm{out} \end{bmatrix} \qquad d = \begin{bmatrix} I_s \\ T_\mathrm{a} \end{bmatrix} \tag{8.2}$$

and the heat consumption model of Section 3.5:

$$P_\mathrm{h} = c\, q_\mathrm{tot}\, T_\mathrm{out} - c \sum_{i,j} q_{i,j}\, T_{\mathrm{f},i,j}\, a_{i,j} \tag{8.3}$$

Equation 8.1 contains two non-linear terms which are undesirable for model predictive control: $F\, u_k\, \begin{bmatrix} \mathbf{0}_{1\times 11} & 1 \end{bmatrix}\, u_k$ and $F\, u_k \odot \begin{bmatrix} \mathbf{I}_{11\times 11} & \mathbf{0}_{11\times 11} \end{bmatrix}\, x_k$.

These consist of inputs multiplied with inputs and states multiplied with inputs respectively. Non-linear terms are undesirable in MPC strategies, as they result in non-linear optimisation problems that are potentially much harder to solve and without guarantee of globally optimal solutions.

However, the non-linear terms in the multi-zone model can be converted into a more desirable form by doing the following linear substitution from Section 3.1 in the input vector:

$$q_{i,j} = \overline{q}_{i,j}\, v_{i,j} \tag{8.4}$$

where:

$q_{i,j}$	:	Mass flow	$\frac{\text{kg}}{\text{s}}$
$\overline{q}_{i,j}$	:	Nominal mass flow	$\frac{\text{kg}}{\text{s}}$
$v_{i,j}$	:	Valve position	$\{0,1\}$

This results in the non-linear terms now consisting of a real-valued input multiplied with binary-valued inputs and real-valued states multiplied with binary-valued inputs:

$$\begin{aligned} \underbrace{T_{\text{out}}}_{\substack{\text{real}\\\text{valued}\\\text{input}}}\ \underbrace{q_{i,j}}_{\substack{\text{real}\\\text{valued}\\\text{input}}} &\Rightarrow \underbrace{\overline{q}_{i,j}}_{\text{constant}}\ \underbrace{T_{\text{out}}}_{\substack{\text{real}\\\text{valued}\\\text{input}}}\ \underbrace{v_{i,j}}_{\substack{\text{binary}\\\text{valued}\\\text{input}}} \\ \underbrace{T_{\text{f},i,j}}_{\substack{\text{real}\\\text{valued}\\\text{state}}}\ \underbrace{q_{i,j}}_{\substack{\text{real}\\\text{valued}\\\text{input}}} &\Rightarrow \underbrace{\overline{q}_{i,j}}_{\text{constant}}\ \underbrace{T_{\text{f},i,j}}_{\substack{\text{real}\\\text{valued}\\\text{state}}}\ \underbrace{v_{i,j}}_{\substack{\text{binary}\\\text{valued}\\\text{input}}} \end{aligned} \tag{8.5}$$

This allows the use of a version of Model Predictive Control known as Mixed Integer Preditive Control (MIPC), which will be discussed in the following section.

8.2 Method

The substitution from real-valued mass flows to binary-valued valve positions in the input vector allows for the use of a framework for modelling and controlling systems known as Mixed Logical Dynamical (MLD) systems. The MLD framework was first proposed by Alberto Bemporad and Manfred Morari in 1998 [35]. The use of MLD systems in MPC strategies results in optimization problems that can be solved through Mixed Integer Quadratic Programming (MIQP), which has efficient methods for finding the global optimal solution.

The section first explains the MLD framework in Subsection 8.2.1 followed by the optimal solution strategy MIQP in Subsection 8.2.2.

8.2.1 MLD systems

The main idea behind the MLD form is to transform logic relations into mixed-integer linear inequalities. The general form of the MLD system is:

$$\begin{aligned} x_{k+1} &= A_k\, x_k + B_{1,k}\, u_k + B_{2,k}\, \delta_k + B_{3,k}\, z_k \\ y_k &= C_k\, x_k + D_{1,k}\, u_k + D_{2,k}\, \delta_k + D_{3,k}\, z_k \\ E_{2,k}\, \delta_k + E_{3,k}\, z_k &\le E_{1,k}\, u_k + E_{4,k}\, x_k + E_{5,k} \end{aligned} \tag{8.6}$$

where:

$$\begin{aligned} x &= \begin{bmatrix} x_c \\ x_l \end{bmatrix},\ x_c \in \mathbb{R}^{n_c},\ x_l \in \{0,1\}^{n_l},\ n = n_c + n_l \\ y &= \begin{bmatrix} y_c \\ y_l \end{bmatrix},\ y_c \in \mathbb{R}^{p_c},\ y_l \in \{0,1\}^{p_l},\ p = p_c + p_l \\ u &= \begin{bmatrix} u_c \\ u_l \end{bmatrix},\ u_c \in \mathbb{R}^{m_c},\ u_l \in \{0,1\}^{m_l},\ m = m_c + m_l \end{aligned} \tag{8.7}$$

and $\delta \in \{0,1\}^{r_l}$ and $z \in \mathbb{R}^{r_c}$ are auxiliary binary and continuous variables respectively. The MLD framework allows for modelling of a broad selection of broad range systems [35, 36]. To exemplify the use of the MLD framework, consider the following example adapted from [35]:

$$\begin{gathered} x_{k+1} = \begin{cases} 0.8\, x_k + u_k & \text{if } x_k \ge 0, \\ -0.8\, x_k + u_k & \text{if } x_k < 0 \end{cases} \\ x_k \in [-10, 10] \quad u_k \in [-1, 1] \end{gathered} \tag{8.8}$$

To deal with the inequality $x_k \ge 0$, we define an binary auxiliary variable $\delta_k \in \{0,1\}$:

$$\delta_k = 1 \Leftrightarrow x_k \ge 0 \tag{8.9}$$

This logic relation can then be transformed in to the inequalities by using the following rule for $\delta = 1 \Leftrightarrow f(x) \ge 0$, where:

$$\begin{aligned} -m\,\delta &\le f(x) - m \\ -(M + \varepsilon)\,\delta &\le -f(x) - \varepsilon \end{aligned} \tag{8.10}$$

where:

$$\begin{aligned} M &\ge \max_{x \in \mathcal{X}} f(x) \\ m &\le \min_{x \in \mathcal{X}} f(x) \end{aligned} \tag{8.11}$$

and ε is a small positive scalar. Thus Equation 8.8 can be rewritten to:

$$x_{k+1} = 1.6\, \delta_k\, x_k - 0.8\, x_k + u_k \tag{8.12}$$

We now define a continuous auxiliary variable:

$$z_k = \delta_k\, x_k \tag{8.13}$$

This multiplication can be converted as a series of linear inqualities using the following rule for $z = \delta\, f(x)$:

$$\begin{aligned} c_1 &: \quad z \le M\,\delta \\ c_2 &: \quad z \ge m\,\delta \\ c_3 &: \quad z \le f(x) - m\,(1 - \delta) \\ c_4 &: \quad z \ge f(x) - M\,(1 - \delta) \end{aligned} \tag{8.14}$$

These constraints can be visualised as seen on Figure 8.1.

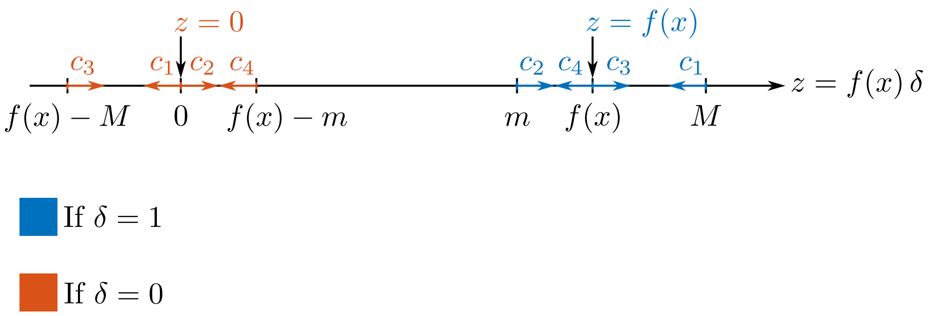

Figure 8.1: Sketch of how the constraints work for continuous auxiliary variables defined as a product of a binary and continuous variable.

Using this, we can write the final system as:

$$\begin{aligned} x_{k+1} &= 1.6\, z_k - 0.8\, x_k + u_k \\ z &\leq 10\,\delta \\ z &\geq -10\,\delta \\ z &\leq x + 10\,(1-\delta) \\ z &\geq x - 10\,(1-\delta) \\ 10\,\delta &\leq x + 10 \\ -(10+\varepsilon)\,\delta &\leq -x - \varepsilon \\ -1 &\leq u \leq 1 \\ -10 &\leq x \leq 10 \end{aligned} \tag{8.15}$$

which is now on MLD form. Note how the dynamic equation is linear, as well as the constraints.

This can now be used in a MPC problem. If the cost function V of the MPC is quadratic, for example:

$$V_k = \sum_{i=1}^{H_\mathrm{p}} \|x_{k+i} - x_{\mathrm{ref},k+i}\|^2_{Q_i} + \sum_{i=1}^{H_\mathrm{u}} \|u_{k+i}\|^2_{R_i}\,, \quad Q \geq 0, R \geq 0 \tag{8.16}$$

where:

H_p	:	Prediction horizon
H_u	:	Control horizon
x_ref	:	State reference
$Q\,,\ R$	:	Weight matrices

then the resulting optimisation problem will be quadratic with both real-valued and binary valued variables. This can then be solved efficiently using MIQP, which will be discussed in the following section.

8.2.2 Mixed integer quadratic programming

Mixed Integer Quadratic Programming (MIQP) problems can be written on the general form:

$$\begin{aligned} &\min x^{\mathrm{T}} Q\, x + q^{\mathrm{T}} x \\ &\text{subject to:} \\ &\text{linear constraints:} && A\,x \;= b \\ &\text{bound constraints:} && x_{\min} \leq x \leq x_{\max} \\ &\text{inequality constraints:} && q_i^{\mathrm{T}}\, x \leq b_i \end{aligned} \tag{8.17}$$

where some or all of x are integer-valued [37]. To solve the comfort MPC's optimisation problem, CasADi [38] is used in Python. CasADi is an open-source tool for numeric optimisation. It is setup to use the open-source solver Basic Open-source Nonlinear Mixed INteger programming (BONMIN) [39]. BONMIN used a method known as branch and bound to solve the optimisation problem.

8.2.2.1 Branch bound

Mixed Integer Quadratic Programming (MIQP) problems can be solved using a quadratic programming branch and bound method [35]. This can be illustrated using a tree graph as seen on Figure 8.2.

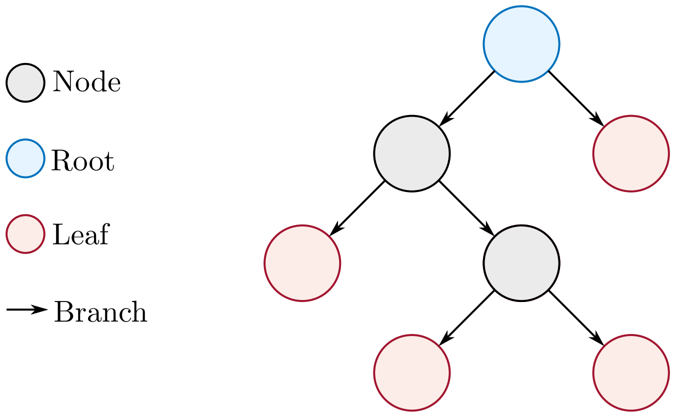

Figure 8.2: Illustration of a tree graph. Note, the roots and leaves are also nodes.

The branch and bound method can be summarised as follows:

1. Since there is no direct way of solving the MIQP, the problem is relaxed, by assuming all integer variables are continuous. This yields a QP problem. Due to the integer variables, the constraints of the original control problem is non-convex. However, the relaxation causes the linear non-convex constraints to become convex. The new QP problem can then be solved efficiently via e.g. an interior point method. This optimisation problem is the root node.

2. If the optimum solution is integer valued for the integer variables, then the node is labelled as fathomed and is a leaf node. If not, choose a integer variable x_i which has a non-integer value solution x_i^*. Create two new QP problems: one with an additional constraint of $x_i \leq \lfloor x_i^* \rfloor$ and another with $x_i \geq \lceil x_i^* \rceil$ .This is called branching. $\lfloor x \rfloor$ and $\lceil x \rceil$ are the floor and ceiling function of x respectively.

3. Solve these new QP problems.
4. Continue performing step 3 and 4 until all branches are fully explored and fathomed.
5. Choose the optimum value among the leaf nodes as the global solution.

The main strength of the branch and bound method lies in the fact that whole subtrees can be excluded from further exploration by fathoming the corresponding root nodes. That can happen if an integer solution is obtained or if the the QP subproblem is infeasible. In the case of an integer solution is obtained for a node, then the corresponding value of the function can serve as an upper bound on the optimal solution of the MIQP. This upper bound can be used for fathoming other nodes if it has a greater optimal value, else it can be used as lower bound [35].

In the following subsubsection a simple example will be given.

8.2.2.2 Branch and bound example

To explain the solving process of the branch and bound algorithm, consider the following optimisation problem adapted from [40]:

$$
\begin{aligned}
&\max x_1 + 2\,x_2 \\
&\text{subject to:} \\
&\qquad x_1 + 4\,x_2 \le 8 \\
&\qquad 4\,x_1 + x_2 \le 8 \\
&\qquad x_1 \ge 0 \\
&\qquad x_2 \ge 0
\end{aligned}
\tag{8.18}
$$

where x_1 and x_2 are integers.

First, in the root node solve the optimisation problem assuming the integers to be continuous. See Figure 8.3.

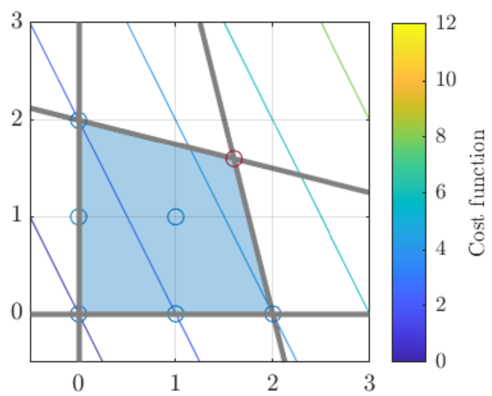

Figure 8.3: Blue circles: integers satisfying constraints. Red circle: optimum point. Grey lines: constraints. Blue area: Feasible region.

The cost in the optimum point $(1.6, 1.6)$ is 4.8. Since the optimum is not integer valued we continue by branching the problem. Since $x_1 = 1.6$ we choose branches $x_1 \le 1$ and $x_1 \ge 2$.

This leads to two new optimisation problems as seen on Figure 8.4 and 8.5 that can each be easily solved.

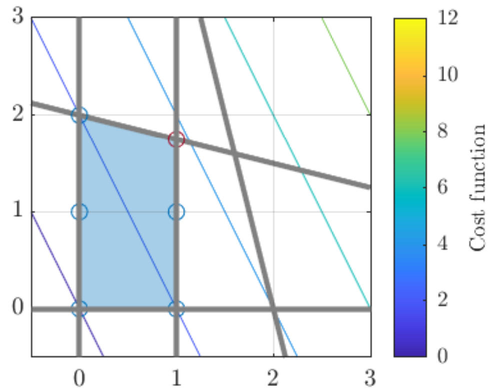

Figure 8.4: Same as Figure 8.3, but with additional constraint $x_1 \leq 1$.

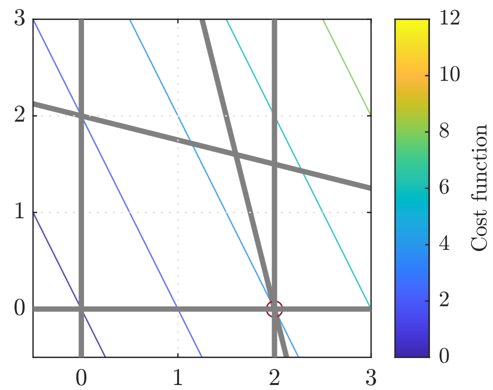

Figure 8.5: Same as Figure 8.3, but with additional constraint $x_1 \geq 2$.

The cost in the optimum point $(1, 1.75)$ is 4.5 for the branch where $x_1 \leq 1$. For $x_1 \geq 2$ the cost in the optimum point $(2, 0)$ is 2. Since the optimum point for the branch with $x_1 \geq 2$ is integer valued for x_1 and x_2, the branch is labeled as fathomed.

Next the branch with $x_1 \leq 1$ is branched again. Since $x_2 = 1.75$ in the optimum point, we choose branches $x_2 \leq 1$ and $x_2 \geq 2$. This leads to two new optimisation problems as seen on Figure 8.6 and 8.7 that can each be easily solved.

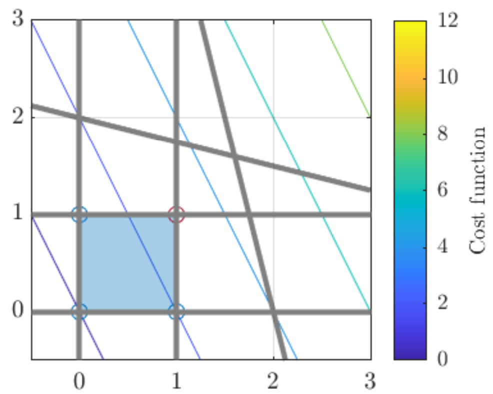

Figure 8.6: Same as Figure 8.4, but with additional constraint $x_2 \leq 1$.

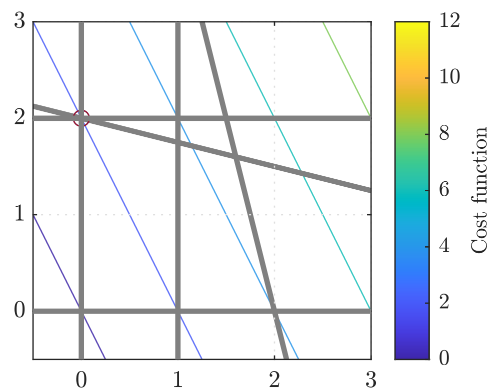

Figure 8.7: Same as Figure 8.4, but with additional constraint $x_2 \geq 2$.

The cost in the optimum point $(1, 1)$ is 3 for the branch where $x_3 \leq 1$. For $x_2 \geq 2$ the cost in the optimum point $(0, 2)$ is 4. Since both optimum points are integer valued, they are both labeled as fathomed. There are no more branches to search and the algorithm stops. The highest cost function is 4 for the integer valued optimum point $(0, 2)$. This procedure can be illustrated as a tree graph as seen on Figure 8.8.

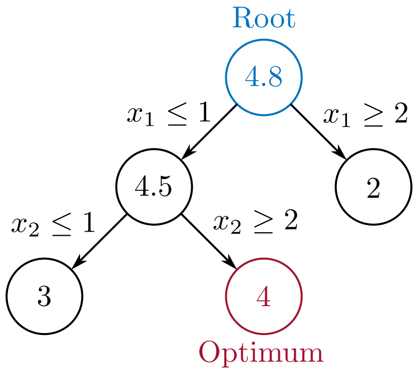

Figure 8.8: Tree graph of the branch and bound example with optimum cost shown for each node's maximisation problem.

In the following Section 8.3 the multi-zone model from Chapter 3 will be converted to MLD form. In Section 8.4 the optimal control problem for the comfort MPC will be formulated and it will be shown that it is a MIQP problem.

8.3 Conversion of the multi-zone model

This section covers the conversion of the lumped multi-zone model to a MLD system. The model from Section 3.4 will be combined with the heat power consumption model of Section 3.5. We start by looking at Equation 3.17 from the heat inflow model:

$$P_{i,j} = c\, q_{i,j}\, b_{i,j}\, (T_{\text{out}} - T_{\text{f},i,j}) \tag{8.19}$$

First the $q_{i,j} = \overline{q}_{i,j}\, v_{i,j}$ is substituted in:

$$P_{i,j} = c\, \overline{q}_{i,j}\, b_{i,j}\, T_{\text{out}}\, v_{i,j} - c\, \overline{q}_{i,j}\, b_{i,j}\, T_{\text{f},i,j}\, v_{i,j} \tag{8.20}$$

We now introduce the auxiliary variables

$$z = \begin{bmatrix} T_{\text{f},0,1}\, v_{0,1} \\ \vdots \\ T_{\text{f},1,4}\, v_{1,4} \\ \hdashline T_{\text{out}}\, v_{0,1} \\ \vdots \\ T_{\text{out}}\, v_{1,4} \end{bmatrix}_{22\times 1} = \begin{bmatrix} z_{\text{f},i,j} \\ \hdashline z_{\text{out},i,j} \end{bmatrix} \tag{8.21}$$

Substituting Equation 8.21 into 8.20 results in:

$$P_{i,j} = c\, \overline{q}_{i,j}\, b_{i,j}\, z_{\text{out},i,j} - c\, \overline{q}_{i,j}\, b_{i,j}\, z_{\text{f},i,j} \tag{8.22}$$

This can then be vectorised as:

$$P = \left[\begin{array}{cccc:cccc} -c\, b_{0,1}\, \overline{q}_{0,1} & 0 & \cdots & 0 & c\, b_{0,1}\, \overline{q}_{0,1} & 0 & \cdots & 0 \\ 0 & -c\, b_{0,2}\, \overline{q}_{0,2} & \cdots & 0 & 0 & c\, b_{0,2}\, \overline{q}_{0,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -c\, b_{1,4}\, \overline{q}_{1,4} & 0 & 0 & \cdots & c\, b_{1,4}\, \overline{q}_{1,4} \end{array}\right] z_k \tag{8.23}$$

Now we can define D_1 using B_d from Equation 8.1:

$$D_1 = B_\text{d} \left[\begin{array}{cccc|cccc} -c\,b_{0,1}\,\overline{q}_{0,1} & 0 & \cdots & 0 & c\,b_{0,1}\,\overline{q}_{0,1} & 0 & \cdots & 0 \\ 0 & -c\,b_{0,2}\,\overline{q}_{0,2} & \cdots & 0 & 0 & c\,b_{0,2}\,\overline{q}_{0,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -c\,b_{1,4}\,\overline{q}_{1,4} & 0 & 0 & \cdots & c\,b_{1,4}\,\overline{q}_{1,4} \end{array} \right] \tag{8.24}$$

Equation 8.24 is combined with the state equation from Equation 8.1 to obtain:

$$x_{k+1} = A_\text{d}\,x_k + E_\text{d}\,d + D_1\,z_k \tag{8.25}$$

Next, we consider Equation 8.3:

$$P_\text{h} = c\,q_\text{tot}\,T_\text{out} - c\sum_{i,j} q_{i,j}\,T_{\text{f},i,j}\,a_{i,j} \tag{8.26}$$

Using the same approach as for the previous we can obtain:

$$P_\text{h} = \begin{bmatrix} -c\,a_{0,1}\,\overline{q}_{0,1} & \cdots & -c\,a_{1,4}\,\overline{q}_{1,4} & \overline{q}_{0,1} & \cdots & \overline{q}_{1,4} \end{bmatrix} z_k \tag{8.27}$$

The output equation from Equation 8.1 is expanded to include Equation 8.27:

$$\begin{bmatrix} T_{\text{r},i,j} \\ P_h \end{bmatrix} = \underbrace{\begin{bmatrix} C_\text{d} \\ \mathbf{0}_{1\times 22} \end{bmatrix}}_{C} x_k + \underbrace{\begin{bmatrix} & & \mathbf{0}_{11\times 22} & & & \\ -c\,a_{0,1}\,\overline{q}_{0,1} & \cdots & -c\,a_{1,4}\,\overline{q}_{1,4} & \overline{q}_{0,1} & \cdots & \overline{q}_{1,4} \end{bmatrix}}_{D_2} z_k \tag{8.28}$$

Cleaning up the notation by dropping $_\text{d}$ subscripts in Equation 8.25 we gain the final MLD form of the lumped parameter multi-zone model of the house:

$$\begin{aligned} x_{k+1} &= A\,x_k + E\,d + D_1\,z_k \\ y_k &= C\,x_k + D_2\,z_k \end{aligned} \tag{8.29}$$

where:

$$x_k = \begin{bmatrix} T_{\text{f},i,j} \\ T_{\text{r},i,j} \end{bmatrix}, \qquad y_k = \begin{bmatrix} T_{\text{r},i,j} \\ P_\text{h} \end{bmatrix}, \qquad d_k = \begin{bmatrix} I_s \\ T_\text{a} \end{bmatrix} \tag{8.30}$$

The auxiliary variables z are as per the example from Subsection 8.2.1 handled using a set of inequality constraints from Equation 8.14.

It was found that these constraints were very restrictive when solving the MIQP problem using the numeric solver BONMIN, which lead to problems with infeasibility.

However in an attempt to help problems with infeasibility of the optimisation problem a small positive scalar $\varepsilon = 0.01\,\text{K}$ is added:

$$\begin{aligned} c_1 :&\quad z \leq M\,\delta + \varepsilon \\ c_2 :&\quad z \geq m\,\delta - \varepsilon \\ c_3 :&\quad z \leq f(x) - m\,(1-\delta) + \varepsilon \\ c_4 :&\quad z \geq f(x) - M\,(1-\delta) - \varepsilon \end{aligned} \tag{8.31}$$

An illustration of Equation 8.31 can be found on Figure 8.9.

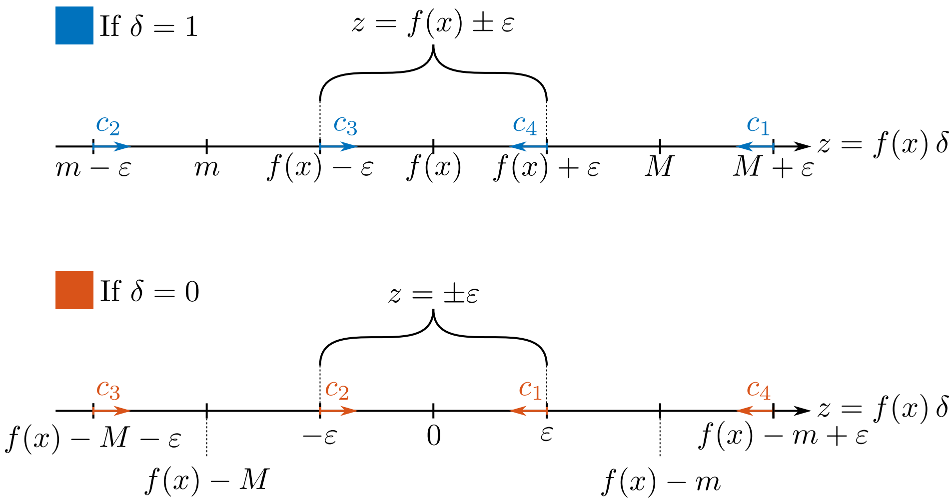

Figure 8.9: The inequality constraints on z illustrated. It has to be noted that the ε is illustrated as being quite large, but the ε is set to be 0.01 K.

This leads to the following inequality constraints for z:

$$
\begin{aligned}
z_{\mathrm{f},i,j} &\leq T_{\mathrm{f,max}}\, v_{i,j} + \varepsilon & z_{\mathrm{out},i,j} &\leq T_{\mathrm{out,max}}\, v_{i,j} + \varepsilon \\
z_{\mathrm{f},i,j} &\geq T_{\mathrm{f,min}}\, v_{i,j} - \varepsilon & z_{\mathrm{out},i,j} &\geq T_{\mathrm{out,min}}\, v_{i,j} - \varepsilon \\
z_{\mathrm{f},i,j} &\leq T_{\mathrm{f},i,j} - T_{\mathrm{f,min}}\,(1 - v_{i,j}) + \varepsilon & z_{\mathrm{out},i,j} &\leq T_{\mathrm{out}} - T_{\mathrm{out,min}}\,(1 - v_{i,j}) + \varepsilon \\
z_{\mathrm{f},i,j} &\geq T_{\mathrm{f},i,j} - T_{\mathrm{f,max}}\,(1 - v_{i,j}) - \varepsilon & z_{\mathrm{out},i,j} &\geq T_{\mathrm{out}} - T_{\mathrm{out,max}}\,(1 - v_{i,j}) - \varepsilon
\end{aligned}
\tag{8.32}
$$

In the following section the Mixed Integer Preditive Control (MIPC) for the comfort MPC will be defined.

8.4 Mixed integer predictive control

The general control law for a MPC is:

At time k solve:

$$\begin{bmatrix} u^*_{k+1} \\ \vdots \\ u^*_{k+H_\mathrm{p}} \end{bmatrix} = \underset{u_{k+1},\dots u_{k+H_\mathrm{p}}}{\arg\min} \quad V_k, \qquad \text{subject to constraints} \tag{8.33}$$

At time $k+1$ apply u^*_{k+1} to the plant.

The cost function for the comfort MPC is defined as:

$$V_k = \sum_{\iota=1}^{H_\mathrm{p}} \|y_{k+\iota} - y_{\mathrm{ref},k+\iota}\|^2_{Q_\iota} + \sum_{\iota=1}^{H_\mathrm{p}} \|T_{\mathrm{out},k+\iota}\|^2_{R_\iota} + \gamma^\mathrm{T}\, \underline{S}\, \gamma \tag{8.34}$$

where:

H_p	:	Prediction horizon
y_ref	:	Output reference $y_\mathrm{ref} = \begin{bmatrix} T_{\mathrm{r},i,j,\mathrm{ref}} & P_{\mathrm{h,ref}} \end{bmatrix}^\mathrm{T}$
$Q_\iota, R_\iota, \mathcal{S}$	:	Weights
γ	:	Slack variables

This cost function formulation follows those of [41]. The idea behind each term is as follows:

- $\sum_{\iota=1}^{H_\mathrm{p}} \|y_{k+\iota} - y_{\mathrm{ref},k+\iota}\|^2_{Q_\iota}$: Achieve reference tracking for the room temperatures $T_{\mathrm{r},i,j}$ and heat power consumption P_h. Reference for the room temperatures come from the user. The reference for the heat power consumption is calculated by the flex MPC in Chapter 9.
- $\sum_{\iota=1}^{H_\mathrm{p}} \|T_{\mathrm{out},k+\iota}\|^2_{R_\iota}$: Reduce the forward temperature from the heat pump T_out, which increases the Coefficient of Performance (COP) for the heat pump thus increasing efficiency and thus potentially decrease the economic cost of running the heat pump.
- $\gamma^\mathrm{T}\, \underline{S}\, \gamma$: Penalise breaking of certain softened constraints, which will be discussed later in this section.

Next, we show that this cost function formulation results in a convex Quadratic Programming (QP) problem. The term $\gamma^\mathrm{T}\, \underline{S}\, \gamma$ is convex and quadratic if $\underline{S} \geq 0$.

We first consider the two other terms using a method known as lifting. The method presented in the following subsection follows those presented in [41].

8.4.1 Lifting

First the state equation of Equation 8.29 is expanded through the prediction horizon:

$$\begin{aligned}
x_{k+1} &= A\,x_k + E\,d_k + D_1\,z_k \\
x_{k+2} &= A^2\,x_k + A\,E\,d_k + A\,D_1\,z_k + E\,d_{k+1} + D_1\,z_{k+1} \\
x_{k+3} &= A^3\,x_k + A^2\,E\,d_k + A^2\,D_1\,z_k + A\,E\,d_{k+1} + E\,d_{k+2} + A\,D_1\,z_{k+1} + D_1\,z_{k+2} \\
\vdots \quad & \quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots
\end{aligned} \tag{8.35}$$

This can then be vectorised as:

$$\underline{x} = \underline{A}_0\,x_k + \underline{E}_0\,d_k + \underline{D}_{1,0} z_k + \underline{E}\,\underline{d} + \underline{D}_1\,\underline{z} \tag{8.36}$$

where:

$$\underline{x} = \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+H_\mathrm{p}} \end{bmatrix} \qquad \underline{d} = \begin{bmatrix} d_{k+1} \\ \vdots \\ d_{k+H_\mathrm{p}} \end{bmatrix} \qquad \underline{z} = \begin{bmatrix} z_{k+1} \\ \vdots \\ z_{k+H_\mathrm{p}} \end{bmatrix} \tag{8.37}$$

and

$$\underline{A}_0 = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_\mathrm{p}} \end{bmatrix} \qquad \underline{E}_0 = \begin{bmatrix} E \\ A\,E \\ \vdots \\ A^{H_\mathrm{p}-1}\,E \end{bmatrix} \qquad \underline{D}_{1,0} = \begin{bmatrix} D_1 \\ A\,D_1 \\ \vdots \\ A^{H_\mathrm{p}-1}\,D_1 \end{bmatrix} \tag{8.38}$$

$$\underline{E} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ E & 0 & 0 & \cdots & 0 \\ A\,E & E & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{H_\mathrm{p}-2}\,E & A^{H_\mathrm{p}-3}\,E & A^{H_\mathrm{p}-4}\,E & \cdots & 0 \end{bmatrix} \tag{8.39}$$

$$\underline{D}_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ D_1 & 0 & 0 & \cdots & 0 \\ A\,D_1 & D_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{H_\mathrm{p}-2}\,D_1 & A^{H_\mathrm{p}-3}\,D_1 & A^{H_\mathrm{p}-4}\,D_1 & \cdots & 0 \end{bmatrix} \tag{8.40}$$

The same procedure can be used on the output equation of Equation 8.29 to obtain:

$$\underline{y} = \underline{C}\,\underline{x} + \underline{D}_2\,\underline{z} \tag{8.41}$$

where:

$$\underline{y} = \begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+H_\mathrm{p}} \end{bmatrix} \quad \underline{C} = \begin{bmatrix} C & 0 & \cdots & 0 \\ 0 & C & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C \end{bmatrix} \quad \underline{D}_2 = \begin{bmatrix} D_2 & 0 & \cdots & 0 \\ 0 & D_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_2 \end{bmatrix} \tag{8.42}$$

Defining

$$\underline{Q} = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{H_\mathrm{p}} \end{bmatrix} \qquad \underline{r} = \begin{bmatrix} y_{\mathrm{ref},k+1} \\ \vdots \\ y_{\mathrm{ref},k+H_\mathrm{p}} \end{bmatrix} \qquad Q_\iota \geq 0,\ \forall \iota \tag{8.43}$$

we can rewrite

$$\sum_{\iota=1}^{H_\mathrm{p}} \|y_{k+\iota} - y_{\mathrm{ref},k+\iota}\|_{Q_\iota}^2 = (\underline{y} - \underline{r})^\mathrm{T}\, \underline{Q}\, (\underline{y} - \underline{r}) \tag{8.44}$$

Which is convex and quadratic since $\underline{Q} \geq 0$ due to it being a block-diagonal matrix with positive semi-definite matrices in the diagonal.

Next, defining:

$$\underline{T}_\mathrm{out} = \begin{bmatrix} T_{\mathrm{out},k+1} \\ \vdots \\ T_{\mathrm{out},k+H_\mathrm{p}} \end{bmatrix} \qquad \underline{R} = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{H_\mathrm{p}} \end{bmatrix} \tag{8.45}$$

We can rewrite:

$$\sum_{\iota=1}^{H_\mathrm{p}} \|T_{\mathrm{out},k+\iota}\|_{R_\iota}^2 = \underline{T}_\mathrm{out}^\mathrm{T}\, \underline{R}\, \underline{T}_\mathrm{out} \tag{8.46}$$

which again is convex and quadratic so long as $\underline{R} \geq 0$.

Next, we describe the constraints added to the optimisation problem.

8.4.2 Constraints

The previously listed constraints from Equation 8.32 are of course included, with a set for each time step in the prediction horizon.

However, they do require additional constraints since the used solver BONMIN does not directly support binary variables, but instead integer variables.

By defining the linear, inequality constraints:

$$0 \leq v_{i,j} \leq 1 \tag{8.47}$$

for the integer variables they are forced to be binary as desired.

Additionally, terminal constraints are added to the problem. Terminal constraints is one of the ways of insuring stability of the system [41, p. 169]. These are usually on the form:

$$y_{k+H_\mathrm{p}} - y_{\mathrm{ref},k+H_\mathrm{p}} = 0 \tag{8.48}$$

However, this might lead to infeasibility problems. Thus, the constraints are softned by the addition of the previously mentioned slack variables γ:

$$-\gamma \leq y_{k+H_\mathrm{p}} - y_{\mathrm{ref},k+H_\mathrm{p}} \leq \gamma, \qquad \gamma \geq 0 \tag{8.49}$$

Lifting was also used to convert most of the constraints into matrix-vector form. This was found to speed up the solving of the optimal control problem by a factor of approximately 3 to 4. As an example, the terminal constraint in Equation 8.49 is formulated as:

$$-K_2 \leq K_1\, \underline{z} - \gamma \qquad -K_2 \geq K_1\, \underline{z} + \gamma \tag{8.50}$$

where:

$$\begin{aligned} K_1 &= C_\mathrm{end}\, \underline{D}_1 + D_{2,\mathrm{end}} \\ K_2 &= C_\mathrm{end}\, \underline{A}_0\, x_k + C_\mathrm{end}\, \underline{E}_0\, d_k + C_\mathrm{end}\, \underline{D}_{1,0} z_k + C_\mathrm{end}\, \underline{E}\, \underline{d} - y_{\mathrm{ref},k+H_\mathrm{p}} \\ C_\mathrm{end} &= \begin{bmatrix} \mathbf{0}_{12\times 66} & C \end{bmatrix} \qquad D_{2,\mathrm{end}} = \begin{bmatrix} \mathbf{0}_{12\times 66} & D_2 \end{bmatrix} \end{aligned} \tag{8.51}$$

In the next subsection the optimal control problem for the comfort MPC is defined.

8.4.3 Optimal control problem

The prediction horizon H_p is set to 4, i.e. 1 hour. This has been chosen to limit computation time while still allowing some prediction of the dynamics of the system.

The weights $\underline{Q}$, $\underline{R}$, and $\underline{S}$ are here given an initial value but will be further tuned in Section 10.2. For Q and R Bryson's rule [34] is used:

$$Q_{ii} = \frac{1}{\left(\max\left(y_i - y_{\mathrm{ref},i}\right)\right)^2}, \quad R = \frac{1}{\left(\max T_\mathrm{out}\right)^2} \tag{8.52}$$

Where Q_{ii} is the i^th diagonal term in the Q matrix. The max refers to the maximum permissible value. For all the $T_\mathrm{r} - T_\mathrm{r,ref}$ it is set to 1 K, for $P_\mathrm{h} - P_\mathrm{h,ref}$ it is set to 100 W, and for T_out it is set to $35 + 273.15$K, as described in Subsection 1.3.4.

Moreover, the weights are kept constant over the prediction horizon.

$$Q_1 = \ldots = Q_{H_\mathrm{p}}, \qquad R_1 = \ldots = R_{H_\mathrm{p}} \tag{8.53}$$

Lastly the weight for the slack variables $\underline{S}$ is set to a diagonal matrix of comparatively high value.

$$\underline{S} = 10^6 \times \mathbb{I}_{12\times 12} \tag{8.54}$$

These choices also guarantees $\underline{Q} \geq 0$, $\underline{R} \geq 0$, and $\underline{S} \geq 0$.

The constraints for the auxiliary variables, Equation 8.32, requires the values in Table 8.1, which are based on Subsection 1.3.4.

Table 8.1: Minimum and maximum values used for constraints on z.

$T_\mathrm{out,max}$	$T_\mathrm{out,min}$	$T_\mathrm{f,max}$	$T_\mathrm{f,min}$
$35 + 273.15$ K	$20 + 273.15$ K	$29 + 273.15$ K	$19 + 273.15$ K

$\varepsilon = 0.01$ K in Equation 8.32 as previously stated.

Due to the way MIQP problems are solved, the computation time will in a worst case scenario depend exponentially on the number of binary variables as all leaves of the search tree in the branch and bound method needs to be solved. Thus, to limit computation time it has been decided that for the prediction horizon the valve positions are the same.

$$v_{i,j,k+1} = \ldots = v_{i,j,k+H_\mathrm{p}} \tag{8.55}$$

which limits the number of binary variables for the comfort MPC to 11.

The optimal control problem of the comfort MPC can now be formulated as:

At time k solve the Mixed Integer Quadratic Programming (MIQP) problem:

$$\min(\underline{y} - \underline{r})^{\mathrm{T}} \underline{Q} \, (\underline{y} - \underline{r}) + \underline{T}_{\mathrm{out}}^{\mathrm{T}} \, \underline{R} \, \underline{T}_{\mathrm{out}} + \gamma^{\mathrm{T}} \, \underline{S} \, \gamma \tag{8.56}$$

Subject to:

- Dynamics:

$$\begin{gathered}
\underline{x} = \underline{A}_0 \, x_k + \underline{E}_0 \, d_k + \underline{D}_{1,0} z_k + \underline{E} \, \underline{d} + \underline{D}_1 \, \underline{z} \qquad \underline{y} = \underline{C} \, \underline{x} + \underline{D}_2 \, \underline{z} \\
\underline{x} = \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+H_{\mathrm{p}}} \end{bmatrix} \quad \underline{d} = \begin{bmatrix} d_{k+1} \\ \vdots \\ d_{k+H_{\mathrm{p}}} \end{bmatrix} \quad \underline{z} = \begin{bmatrix} z_{k+1} \\ \vdots \\ z_{k+H_{\mathrm{p}}} \end{bmatrix} \quad \underline{y} = \begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+H_{\mathrm{p}}} \end{bmatrix} \quad \underline{r} = \begin{bmatrix} y_{\mathrm{ref},k+1} \\ \vdots \\ y_{\mathrm{ref},k+H_{\mathrm{p}}} \end{bmatrix} \\
x = \begin{bmatrix} T_{\mathrm{f},i,j} \\ T_{\mathrm{r},i,j} \end{bmatrix} \quad d = \begin{bmatrix} I_s \\ T_{\mathrm{a}} \end{bmatrix} \quad z = \begin{bmatrix} z_{\mathrm{f},i,j} \\ z_{\mathrm{out},i,j} \end{bmatrix} \quad y = \begin{bmatrix} T_{\mathrm{r},i,j} \\ P_{\mathrm{h}} \end{bmatrix} \quad y_{\mathrm{ref}} = \begin{bmatrix} T_{\mathrm{r,ref},i,j} \\ P_{\mathrm{h,ref}} \end{bmatrix}
\end{gathered} \tag{8.57}$$

- Auxiliary variables:

$$\begin{aligned}
z_{\mathrm{f},i,j,k+\iota} &\le T_{\mathrm{f,max}} \, v_{i,j} + \varepsilon \\
z_{\mathrm{out},i,j,k+\iota} &\le T_{\mathrm{out,max}} \, v_{i,j} + \varepsilon \\
z_{\mathrm{f},i,j,k+\iota} &\ge T_{\mathrm{f,min}} \, v_{i,j} - \varepsilon \\
z_{\mathrm{out},i,j,k+\iota} &\ge T_{\mathrm{out,min}} \, v_{i,j} - \varepsilon \\
z_{\mathrm{f},i,j,k+\iota} &\le T_{\mathrm{f},i,j,k+\iota} - T_{\mathrm{f,min}} \, (1 - v_{i,j}) + \varepsilon \\
z_{\mathrm{out},i,j,k+\iota} &\le T_{\mathrm{out},k+\iota} - T_{\mathrm{out,min}} \, (1 - v_{i,j}) + \varepsilon \\
z_{\mathrm{f},i,j,k+\iota} &\ge T_{\mathrm{f},i,j,k+\iota} - T_{\mathrm{f,max}} \, (1 - v_{i,j}) - \varepsilon \\
z_{\mathrm{out},i,j,k+\iota} &\ge T_{\mathrm{out},k+\iota} - T_{\mathrm{out,max}} \, (1 - v_{i,j}) - \varepsilon \\
&\text{for } \iota \in 1, \ldots, H_{\mathrm{p}} \\
&0 \le v_{i,j} \le 1
\end{aligned} \tag{8.58}$$

- Terminal constraints:

$$-\gamma \le y_{k+H_{\mathrm{p}}} - y_{\mathrm{ref},k+H_{\mathrm{p}}} \le \gamma, \qquad \gamma \ge 0 \tag{8.59}$$

And apply at the next time step $k+1$ the optimal value of $v_{i,j}$ and $T_{\mathrm{out},k+1}$ from the MIQP problem.

As can be seen from the above, the comfort MPC is a Mixed Integer Preditive Control (MIPC) consisting of a Mixed Integer Quadratic Programming (MIQP) problem with a quadratic and convex cost function, and mixed integer linear inequality constraints. This guarantees a globally optimal solution to the control problem when solved through the branch and bound method.

This concludes the development of the comfort MPC. In the following chapter the flex MPC is designed and combined with the comfort MPC.

9 Flex MPC

This chapter will concern the development of a Model Predictive Control (MPC) for the lumped parameter single-zone model developed in Chapter 4.

It contains the following sections:

- Section 9.1 describes the purpose and model for the flex MPC.
- Section 9.2 concerns the conversion of the model equations into the Mixed Logical Dynamical (MLD) framework.
- Section 9.3 describes, the cost function, constraints, and final formulation of the optimal control problem of the flex MPC.

9.1 Motivation

As stated in Section 1.6 and Chapter 5 the overall goal of the flex MPC presented in this chapter is to determine a heating power budget based on general house dynamics and forecast data for electricity prices and power generation of the Photovoltaic (PV) panels of the house.

The model developed for the use in this layer of the control scheme is the lumped parameter single-zone model from Section 4.1 and Section 4.2:

$$\begin{aligned} x_{k+1} &= A_\mathrm{d}\, x_k + E_\mathrm{d}\, d_k + B_\mathrm{d}\, u_k \\ y_k &= C\, x_k \end{aligned} \tag{9.1}$$

where:

$$x = \begin{bmatrix} \overline{T}_\mathrm{f} \\ \overline{T}_\mathrm{r} \end{bmatrix} \quad d = \begin{bmatrix} I_\mathrm{s} \\ T_\mathrm{a} \end{bmatrix} \quad u = \begin{bmatrix} P_\mathrm{h} \end{bmatrix} \quad y = \begin{bmatrix} \overline{T}_\mathrm{r} \end{bmatrix} \tag{9.2}$$

and the heat pump model of Section 4.3

$$P_\mathrm{h} = k_0 + \left(c_0 + c_1\, P_\mathrm{e} + c_2\, P_\mathrm{e}^2\right) \frac{T_\mathrm{out}}{T_\mathrm{out} - T_\mathrm{a}} \tag{9.3}$$

The heat pump model will be dealt with in the constraints of the MPC. However, to include the limit operation range and the ability to turn of the heat pump, the dynamics of Equation 9.1 is converted to the Mixed Logical Dynamical (MLD) framework described in Section 8.2.

9.2 Conversion of the single-zone model

This section covers the conversion of the lumped parameter single-zone model to a MLD system.

We start with defining a continuous auxiliary variable z_h:

$$z_\mathrm{h} = P_\mathrm{h}\,\delta_\mathrm{hp}, \quad z_\mathrm{h} \geq 0 \tag{9.4}$$

where:

z_h	:	Auxiliary variable for heating power of the heat pump	W
δ_hp	:	ON/OFF variable for the heat pump	$\{0, 1\}$

$\delta_\mathrm{hp} = 1$ if the heat pump is on and 0 if it is off. Equation 8.31 is again used to convert the multiplicative relation of Equation 9.4 to a set of linear inequality constraints for every time step in the prediction horizon of the MPC:

$$\begin{aligned} z_\mathrm{h} &\leq P_\mathrm{h,max}\,\delta_\mathrm{hp} + \varepsilon \\ z_\mathrm{h} &\geq P_\mathrm{h,min}\,\delta_\mathrm{hp} - \varepsilon \\ z_\mathrm{h} &\leq P_\mathrm{h} - P_\mathrm{h,min}\,(1 - \delta_\mathrm{hp}) + \varepsilon \\ z_\mathrm{h} &\geq P_\mathrm{h} - P_\mathrm{h,max}\,(1 - \delta_\mathrm{hp}) - \varepsilon \end{aligned} \tag{9.5}$$

where ε is a small positive scalar, in this case 1 W. Thus Equation 9.1 becomes:

$$\begin{aligned} x_{k+1} &= A_\mathrm{d}\,x_k + E_\mathrm{d}\,d_k + B_\mathrm{d}\,z_{\mathrm{h},k} \\ y_k &= C\,x_k \end{aligned} \tag{9.6}$$

In the following section the Mixed Integer Preditive Control (MIPC) for the flex MPC is defined.

9.3 Mixed integer predictive control

As per Section 8.4, the general control law a MPC is:

At time k solve:

$$\begin{bmatrix} u^*_{k+1} \\ \vdots \\ u^*_{k+H_\mathrm{p}} \end{bmatrix} = \underset{u_{k+1},\dots u_{k+H_\mathrm{p}}}{\arg\min}\; V_k, \qquad \text{subject to constraints} \tag{9.7}$$

At time $k+1$ apply u^*_{k+1} to the plant.

The cost function for the flex MPC is defined as:

$$V_k = \sum_{\iota=1}^{H_\mathrm{p}} \|y_{k+\iota} - y_{\mathrm{ref},k+\iota}\|^2_{Q_\iota} + p^\mathrm{T}\,\underline{P}_\mathrm{grid} + \gamma^\mathrm{T}\,\underline{S}\,\gamma \tag{9.8}$$

where:

H_{p}	:	Prediction horizon	
y_{ref}	:	Output reference	
$Q_\iota, \mathcal{S}$	:	Weights	
γ	:	Slack variables	
p	:	Electricity price during the prediction horizon	$\frac{\mathrm{DKK}}{\mathrm{W}}$
$\underline{P}_{\mathrm{grid}}$	:	Power bought from the electric grid during the prediction horizon	W

The idea behind each term is as follows:

- $\sum_{\iota=1}^{H_{\mathrm{p}}} \|y_{k+\iota} - y_{\mathrm{ref},k+\iota}\|_{Q_\iota}^2$: Achieve reference tracking for the weighted average room temperature $\overline{T}_{\mathrm{r}}$ using a weighted average of the room temperature references, $\overline{T}_{\mathrm{r,ref}}$.
- $p^{\mathrm{T}}\, \underline{P}_{\mathrm{grid}}$: Calculate the monetary cost of the heating of the house. Thus, reduce the amount money spent on heating of the house.
- $\gamma^{\mathrm{T}}\, \underline{S}\, \gamma$: Penalise breaking of certain softened constraints, which will be discussed later in this section.

Next, we show that this cost function formulation results in a convex Mixed Integer Quadratic Programming (MIQP) problem.

As for the comfort MPC, the term $\gamma^{\mathrm{T}}\, \underline{S}\, \gamma$ is convex and quadratic if $\underline{S} \geq 0$. Additionally, the term $p^{\mathrm{T}}\, \underline{P}_{\mathrm{grid}}$ is linear and convex if the elements of P_{grid} and p are non-negative valued. I.e. we restrict us from selling to the grid and limiting us to prices that are non-negative. Note, negative prices for electricity do occur as seen on Figure 1.4.

We again use lifting to reformulate the last term in the cost function.

9.3.1 Lifting

Using the method from Subsection 8.4.1 the state and output equations are lifted to obtain:

$$\begin{aligned} \underline{x} &= \underline{A}_0\, x_k + \underline{E}_0\, d_k + \underline{B}_0\, z_{\mathrm{h},k} + \underline{E}\, \underline{d} + \underline{B}\, \underline{z}_{\mathrm{h}} \\ \underline{y} &= \underline{C}\, \underline{x} \end{aligned} \tag{9.9}$$

where:

$$\underline{x} = \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+H_{\mathrm{p}}} \end{bmatrix} \qquad \underline{d} = \begin{bmatrix} d_{k+1} \\ \vdots \\ d_{k+H_{\mathrm{p}}} \end{bmatrix} \qquad \underline{z}_{\mathrm{h}} = \begin{bmatrix} z_{\mathrm{h},k+1} \\ \vdots \\ z_{\mathrm{h},k+H_{\mathrm{p}}} \end{bmatrix} \qquad \underline{y} = \begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+H_{\mathrm{p}}} \end{bmatrix} \tag{9.10}$$

and

$$\underline{A}_0 = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_{\mathrm{p}}} \end{bmatrix} \quad \underline{E}_0 = \begin{bmatrix} E \\ A\,E \\ \vdots \\ A^{H_{\mathrm{p}}-1}\, E \end{bmatrix} \quad \underline{B}_0 = \begin{bmatrix} B \\ A\,B \\ \vdots \\ A^{H_{\mathrm{p}}-1}\, B \end{bmatrix} \quad \underline{C} = \begin{bmatrix} C & 0 & \cdots & 0 \\ 0 & C & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C \end{bmatrix} \tag{9.11}$$

$$\underline{E} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ E & 0 & 0 & \cdots & 0 \\ A\,E & E & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{H_{\mathrm{p}}-2}\, E & A^{H_{\mathrm{p}}-3}\, E & A^{H_{\mathrm{p}}-4}\, E & \cdots & 0 \end{bmatrix} \tag{9.12}$$

$$\underline{B} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B & 0 & 0 & \cdots & 0 \\ A\,B & B & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{H_\mathrm{p}-2}\,B & A^{H_\mathrm{p}-3}\,B & A^{H_\mathrm{p}-4}\,B & \cdots & 0 \end{bmatrix} \tag{9.13}$$

Defining

$$\underline{Q} = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{H_\mathrm{p}} \end{bmatrix} \qquad \underline{r} = \begin{bmatrix} y_{\mathrm{ref},k+1} \\ \vdots \\ y_{\mathrm{ref},k+H_\mathrm{p}} \end{bmatrix} \tag{9.14}$$

we can rewrite

$$\sum_{\iota=1}^{H_\mathrm{p}} \|y_{k+\iota} - y_{\mathrm{ref},k+\iota}\|^2_{Q_\iota} = (\underline{y} - \underline{r})^\mathrm{T}\,\underline{Q}\,(\underline{y} - \underline{r}) \tag{9.15}$$

which is convex and quadratic so long as $\underline{Q} \geq 0$.

Next, we describe the constraints added to the optimisation problem.

9.3.2 Constraints

The previously listed constraints from Equation 9.5 are of course included, with a set for each time step in the prediction horizon. By defining the linear, inequality constraints:

$$0 \leq \delta_{\mathrm{hp},k+\iota} \leq 1, \quad \iota \in 1, \ldots, H_\mathrm{p} \tag{9.16}$$

for the integer variables they are forced to be binary as desired. As for the comfort MPC, softened terminal constraints are added to the flex MPC for stability using the previously mentioned slack variables γ:

$$-\gamma \leq y_{k+H_\mathrm{p}} - y_{\mathrm{ref},k+H_\mathrm{p}} \leq \gamma, \qquad \gamma \geq 0 \tag{9.17}$$

The remaining constraints are parted into two groups:

- Heat pump properties in subsubsection 9.3.2.1
- Electricity balance in subsubsection 9.3.2.2

9.3.2.1 Heat pump properties

As stated in Section 9.1, the heat pump model of Section 4.3 is implemented in the flex MPC as constraints.

For convex optimisation problems we generally require inequality constraints on the form $g(x) \leq 0$, where g is a convex function [42]. For the heat pump we achieve this by defining as per Equation 4.22:

$$\begin{aligned} P_{\mathrm{e},k+\iota} &\geq f(P_{\mathrm{h},k+\iota}) \\ P_{\mathrm{e},k+\iota} &\geq \frac{-b + \sqrt{b^2 - 4\,a\,c}}{2\,a} \end{aligned} \tag{9.18}$$

where:

$$a = c_2 \frac{T_{\mathrm{out},k}}{T_{\mathrm{out},k} - T_{\mathrm{a},k+\iota}}, \qquad b = c_1 \frac{T_{\mathrm{out},k}}{T_{\mathrm{out},k} - T_{\mathrm{a},k+\iota}}$$
$$c = -P_{\mathrm{h},k+\iota} + k_0 + c_0 \frac{T_{\mathrm{out},k}}{T_{\mathrm{out},k} - T_{\mathrm{a},k+\iota}} \tag{9.19}$$

Note, the use of $\geq$ in Equation 9.18. For the forward temperature of the heat pump T_{out} the current value, i.e. at time k, is used throughout the entire prediction horizon. For the outdoor ambient temperature T_{a} forecast data is used as per the definition of the $\underline{d}$-vector from Equation 9.10.

We check the convexity of Equation 9.18 by reformulating it to:

$$0 \geq \frac{-b + \sqrt{b^2 - 4\,a\,c}}{2\,a} - P_{\mathrm{e},k+\iota} = g(P_{\mathrm{e},k+\iota}, P_{\mathrm{h},k+\iota}) \tag{9.20}$$

Next the Hessian matrix $\mathcal{H}$ of g is calculated:

$$\mathcal{H} = \begin{bmatrix} \dfrac{\partial^2 g}{\partial P_{\mathrm{e},k+\iota}^2} & \dfrac{\partial^2 g}{\partial P_{\mathrm{e},k+\iota} \partial P_{\mathrm{h},k+\iota}} \\ \dfrac{\partial^2 g}{\partial P_{\mathrm{h},k+\iota} \partial P_{\mathrm{e},k+\iota}} & \dfrac{\partial^2 g}{\partial P_{\mathrm{h},k+\iota}^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{-c_2}{2\sqrt{2\,c_2\,P_{\mathrm{h},k+\iota} - 4\,c_2\,c_0 + c_1^2 - 2\,c_2\,k_0}^{\,3}} \end{bmatrix} \tag{9.21}$$

which has one eigenvalue at 0, and one that is positive for reasonable values of $P_{\mathrm{h},k+\iota}$. Thus, the Hessian of g is positive semi-definite and the function g is therefore convex, i.e. Equation 9.18 is a convex constraint. This is part of the reason why the heat pump model of Section 4.3 has the form it has, according to Simon Thorsteinsson [1]. However it should be noted that the constraint in Equation 9.18 potentially could lead to too high values of $P_{\mathrm{e},k+\iota}$, but this is handled later in subsubsection 9.3.2.2.

As stated in Section 4.3, the heat pump has certain restrictions. These are an operating range for the electric power as in Equation 4.19:

$$P_{\mathrm{h,min}} = 250\,\mathrm{W} \qquad P_{\mathrm{h,max}} = 1500\,\mathrm{W} \tag{9.22}$$

And if the heat pump turns off, then the heat pump can not be restarted for 3 hours. It also requires a minimum up time of 1 hour. First, the restrictions on the on the operating range, while allowing turning off the heat pump is implemented as:

$$\delta_{\mathrm{hp},k+\iota}\,P_{\mathrm{e,min}} \leq P_{\mathrm{e},k+\iota} \leq \delta_{\mathrm{hp},k+\iota}\,P_{\mathrm{e,max}} \tag{9.23}$$

which forces it to be in the range $[P_{\mathrm{e,min}}, P_{\mathrm{e,max}}]$ if the heat pump is on, and 0 if the heat pump is off.

Next, the restrictions on on/off times for the heat pump. Since the sample time of the flex MPC is equal to the minimum on-time of the heat pump, this restriction will not be considered here. The restriction on the minimum off-time is handled through the linear constraints:

$$-\sum_{\lambda=2}^{M} \left(\delta^*_{\mathrm{hp},k+\iota-\lambda}\right) + M\,\delta^*_{\mathrm{hp},k+\iota-1} - (M-1)\,\delta^*_{\mathrm{hp},k+\iota} \geq -(M-1), \quad \iota = 1 \ldots H_{\mathrm{p}} \tag{9.24}$$

where:

M : Number of samples the heat pump needs to be shut off at a time.

Since the minimum off-time is 3 hours and the sample time is 1 hour $M = 3$. δ^*_{hp} is a combination of previous, current and future states of the heat pump:

$$\delta^*_{\mathrm{hp}} = \begin{bmatrix} \delta_{\mathrm{hp},k-2} \\ \delta_{\mathrm{hp},k-1} \\ \delta_{\mathrm{hp},k} \\ \hdashline \delta_{\mathrm{hp},k+1} \\ \vdots \\ \delta_{\mathrm{hp},k+H_\mathrm{p}} \end{bmatrix} \tag{9.25}$$

To show how Equation 9.24 works, we consider a couple of examples:

$$\begin{aligned} \delta^*_{\mathrm{hp}} &= \begin{bmatrix} 0 & 0 & 0 & \delta_{\mathrm{hp}} \end{bmatrix}^\mathrm{T} \\ -0 - 0 + 3 \cdot 0 - (3-1)\,\delta_{\mathrm{hp}} &\geq -(3-1) \\ -2\,\delta_{\mathrm{hp}} &\geq -2 \\ \delta_{\mathrm{hp}} &= \{0, 1\} \end{aligned} \tag{9.26}$$

$$\begin{aligned} \delta^*_{\mathrm{hp}} &= \begin{bmatrix} 1 & 1 & 0 & \delta_{\mathrm{hp}} \end{bmatrix}^\mathrm{T} \\ -1 - 1 + 3 \cdot 0 - (3-1)\,\delta_{\mathrm{hp}} &\geq -(3-1) \\ -2\,\delta_{\mathrm{hp}} &\geq 0 \\ \delta_{\mathrm{hp}} &= 0 \end{aligned} \tag{9.27}$$

This concludes the constraints due the heat pump's properties.

9.3.2.2 Electricity balance

We want to achieve the following relation using constraints:

$$P_{\mathrm{grid},k+\iota} = \max\left(P_{\mathrm{e},k+\iota} - P_{\mathrm{pv},k+\iota}, 0\right) \tag{9.28}$$

where:

P_{grid}	:	Electric power that needs to be bought from the grid	W
P_{pv}	:	Electric power supplied from the PV panels	W

This formulation means that the power supplied by the PV panels is considered free, and that only the excess that requires to be bought from the grid.

We define the variable P_{grid} as:

$$P_{\mathrm{grid},k+\iota} = P_{\mathrm{e},k+\iota} - P_{\mathrm{pv},k+\iota} + z_{\mathrm{e},k+\iota} \tag{9.29}$$

where:

z_e	:	Auxiliary variable	W

Defining the two linear inequality constraints:

$$P_{\mathrm{grid},k+\iota} \geq 0 \qquad z_{\mathrm{e},k+\iota} \geq 0 \tag{9.30}$$

results in the desired relation from Equation 9.28. Note, here how a too high value of $P_{\text{e},k+\iota}$ in the definition of $P_{\text{grid},k+\iota}$ would increase the price in the cost function, thus pushing $P_{\text{e},k+\iota}$ towards the lower bound of $P_{\text{e},k+\iota} \geq f(P_{\text{h},k+\iota})$ from Equation 9.18, thereby yielding the desired $P_{\text{e},k+\iota} = f(P_{\text{h},k+\iota})$ relation from Section 4.3.

Next the MIQP problem with non-linear constraints for the flex MPC is formulated.

9.3.3 Optimal control problem

The prediction horizon H_p is set to 24, i.e. 1 day. This has been chosen to limit computation time while still allowing some prediction of the rather slow averaged dynamics of the system using the lumped parameter single-zone model.

The weights $\underline{Q}$ and $\underline{S}$ are here given an initial value but will be further tuned in Section 10.2. For $\underline{Q}$ Bryson's rule [34] is again used. The maximum allowed value is here set to $25 + 273.15$ K.

Moreover, the different Q's are kept constant over the prediction horizon.

$$Q_1 = \ldots = Q_{H_\text{p}} \tag{9.31}$$

Lastly the weight for the slack variables $\underline{S}$ is set to a diagonal matrix of comparatively high value.

$$\underline{S} = 10^3 \times \mathbb{I}_{1\times 1} \tag{9.32}$$

These choices also guarantees $\underline{Q} \geq 0$ and $\underline{S} \geq 0$.

For the constraints for the auxiliary variables z_h, Equation 9.5 the following parameters are defined:

$$P_{\text{h,max}} = 10^5 \qquad P_{\text{h,min}} = -10^3 \tag{9.33}$$

These are set relatively high and low, since their only purpose is the constraints of z_h. The limitations of the heat pump is handled through the constraints of subsubsection 9.3.2.1.

The optimal control problem of the comfort MPC can now be formulated as:

At time k solve the Mixed Integer Quadratic Programming (MIQP) problem:

$$\min(\underline{y} - \underline{r})^{\mathrm{T}}\, \underline{Q}\, (\underline{y} - \underline{r}) + p^{\mathrm{T}}\, \underline{P}_{\mathrm{grid}} + \gamma^{\mathrm{T}}\, \underline{S}\, \gamma \tag{9.34}$$

Subject to:

- Dynamics:

$$\underline{x} = \underline{A}_0\, x_k + \underline{E}_0\, d_k + \underline{B}_0\, z_{\mathrm{h},k} + \underline{E}\, \underline{d} + \underline{B}\, \underline{z}_{\mathrm{h}} \qquad \underline{y} = \underline{C}\, \underline{x} \tag{9.35}$$

$$\underline{x} = \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+H_{\mathrm{p}}} \end{bmatrix} \qquad \underline{d} = \begin{bmatrix} d_{k+1} \\ \vdots \\ d_{k+H_{\mathrm{p}}} \end{bmatrix} \qquad \underline{z}_{\mathrm{h}} = \begin{bmatrix} z_{\mathrm{h},k+1} \\ \vdots \\ z_{\mathrm{h},k+H_{\mathrm{p}}} \end{bmatrix} \qquad \underline{y} = \begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+H_{\mathrm{p}}} \end{bmatrix} \tag{9.36}$$

$$x = \begin{bmatrix} \overline{T}_{\mathrm{f}} \\ \overline{T}_{\mathrm{r}} \end{bmatrix} \quad d = \begin{bmatrix} I_{\mathrm{s}} \\ T_{\mathrm{a}} \end{bmatrix} \quad y = \begin{bmatrix} \overline{T}_{\mathrm{r}} \end{bmatrix} \tag{9.37}$$

- Auxiliary variables:

$$\begin{aligned} z_{\mathrm{h},k+\iota} &\leq P_{\mathrm{h,max}}\, \delta_{\mathrm{hp},k+\iota} + \varepsilon \\ z_{\mathrm{h},k+\iota} &\geq P_{\mathrm{h,min}}\, \delta_{\mathrm{hp},k+\iota} - \varepsilon \\ z_{\mathrm{h},k+\iota} &\leq P_{\mathrm{h},k+\iota} - P_{\mathrm{h,min}}\, (1 - \delta_{\mathrm{hp},k+\iota}) + \varepsilon \\ z_{\mathrm{h},k+\iota} &\geq P_{\mathrm{h},k+\iota} - P_{\mathrm{h,max}}\, (1 - \delta_{\mathrm{hp},k+\iota}) - \varepsilon \\ \underline{z}_{\mathrm{h}} &\geq 0 \end{aligned} \tag{9.38}$$

- Terminal constraints:

$$-\gamma \leq y_{k+H_{\mathrm{p}}} - y_{\mathrm{ref},k+H_{\mathrm{p}}} \leq \gamma, \qquad \gamma \geq 0 \tag{9.39}$$

- Heat pump properties:

$$\begin{aligned} P_{\mathrm{e},k+\iota} &\geq \frac{-b + \sqrt{b^2 - 4\,a\,c}}{2\,a}, \qquad a = c_2\, \frac{T_{\mathrm{out},k}}{T_{\mathrm{out},k} - T_{\mathrm{a},k+\iota}} \\ b &= c_1\, \frac{T_{\mathrm{out},k}}{T_{\mathrm{out},k} - T_{\mathrm{a},k+\iota}}, \qquad c = -P_{\mathrm{h},k+\iota} + k_0 + c_0\, \frac{T_{\mathrm{out},k}}{T_{\mathrm{out},k} - T_{\mathrm{a},k+\iota}} \end{aligned} \tag{9.40}$$

$$\begin{aligned} &\delta_{\mathrm{hp},k+\iota}\, P_{\mathrm{e,min}} \leq P_{\mathrm{e},k+\iota} \leq \delta_{\mathrm{hp},k+\iota}\, P_{\mathrm{e,max}} \\ &-\sum_{\lambda=2}^{M} \left(\delta^*_{\mathrm{hp},k+\iota-\lambda}\right) + M\, \delta^*_{\mathrm{hp},k+\iota-1} - (M-1)\, \delta^*_{\mathrm{hp},k+\iota} \geq -(M-1) \\ &\delta^*_{\mathrm{hp}} = \begin{bmatrix} \delta_{\mathrm{hp},k-2} & \delta_{\mathrm{hp},k-1} & \delta_{\mathrm{hp},k} & \delta_{\mathrm{hp},k+1} & \cdots & \delta_{\mathrm{hp},k+H_{\mathrm{p}}} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{9.41}$$

- Electricity balance:

$$\begin{gathered} P_{\mathrm{grid},k+\iota} = P_{\mathrm{e},k+\iota} - P_{\mathrm{pv},k+\iota} + z_{\mathrm{e},k+\iota} \\ P_{\mathrm{grid},k+\iota} \geq 0 \qquad z_{\mathrm{e},k+\iota} \geq 0 \\ \text{for } \iota \in 1, \ldots, H_{\mathrm{p}} \end{gathered} \tag{9.42}$$

And update at the next time step $k+1$ the state of the heat pump and the $P_{\mathrm{h,ref}}$ used by the comfort MPC, using the optimal values of $\delta_{\mathrm{hp},k+1}$ and $P_{\mathrm{h},k+1}$ from the MIQP problem.

As can be seen from the above, the flex MPC is a Mixed Integer Preditive Control (MIPC) consisting of a Mixed Integer Quadratic Programming (MIQP) problem with a quadratic and convex cost function, and both mixed integer linear inequality constraints, and one set of convex non-linear constraints for the P_h-P_e relation. This guarantees a globally optimal solution to the control problem when solved through the branch and bound method.

This concludes the development of the flex MPC. In the next chapter, the different parts of the overall control strategy is combined and tested using the Dymola model as the plant of the system.

10 Implementation and test

This chapter concerns the implementation, tuning, and test of the control strategy developed in the previous chapters.

This chapter contains the following sections:

- Section 10.1 briefly describes how the different components of the control strategy is implemented in the Python framework from Simon Thorsteinsson [1].
- Section 10.2 concerns the tuning of the controller to improve its performance.
- Section 10.3 shows and discusses the results from the final four-week test.

10.1 Implementation

As stated in Subsection 1.3.2 a Python framework had been made available for the simulation. To incorporate the designed control strategy, each of the four components, KF, EKF, flex MPC, and comfort MPC, are implemented as separate Python classes. Each class then consist of a number of functions. For example the EKF consists of:

- `__init__`: Initialisation function, run the first time step to load constants such as model matrices.
- `getMeasurementUpdate`: Calculates the measurement update of the EKF as described in Equation 6.6.
- `getTimeUpdate_AL_matrix`: Calculates the linearisation of the state equation using Equation 6.8.
- `getTimeUpdate_states`: Performs the time update as described by Equation 6.7 while calling `getTimeUpdate_AL_matrix`.
- `Update_vars`: Moving forward one time step by $\hat{x}_{k+1|k} \Rightarrow \hat{x}_{k|k-1}$ and $P_{k+1|k} \Rightarrow P_{k|k-1}$.
- `Update_filter`: Call `getMeasurementUpdate`, then `getTimeUpdate_states` and then `Update_vars`.

As previously stated the KF and the flex MPC has a sample time of 60 minutes. These two parts of the control scheme are always updated every hour.

The EKF and the comfort MPC has a sample time of 15 minutes. The EKF is always updated every 15 minutes, but the comfort MPC is only calculated when the heat pump is active, i.e. when the flex MPC decides power needs to be added to the system.

Next, the references. The temperature references for the rooms are kept constant throughout the prediction horizons. I.e. no prediction of how the inhabitants may change the temperature reference in the future is performed. The heating power budget from the flex MPC used as a reference by the comfort MPC is handled differently, as predictions are available from the flex MPC. As an example consider Figure 10.1.

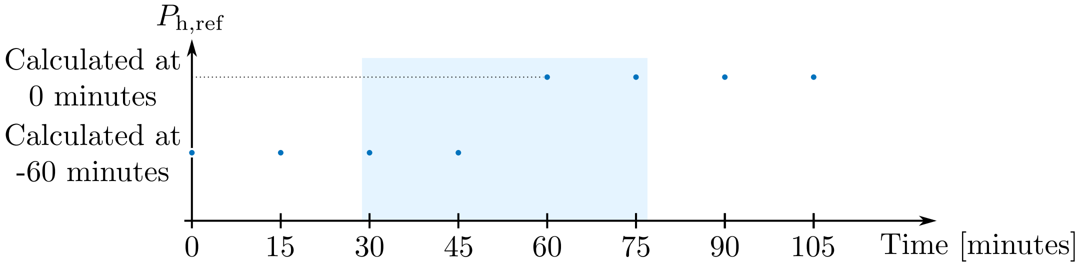

Figure 10.1: Sketch of how the heating power budget reference is implemented. Here the blue region highlights the values used for $P_{\mathrm{h,ref}}$ for the comfort MPC when calculating at time 15 minutes.

Note how the reference is off-set from when it is calculated due to the Zero Order Hold (ZOH) of the control strategy.

Lastly, due to time constraints, the forecast data used by the MPCs is the actual true values in the future. I.e. noise due to predictions are not added to the weather data I_{s} and T_{a}, nor the electricity prices p or the power generated by the Photovoltaic (PV) panels. The weather and PV data used is from 2018 starting January 1$^{\mathrm{st}}$, while the price data is from 2021, also starting January 1$^{\mathrm{st}}$.

In the following section a few of the control strategy's parameters are tuned.

10.2 Tuning

Due to time constraints very little tuning has been done. Only the Q matrices, i.e. the weights on the reference tracking error, of the flex and comfort MPC has been tuned. Further discussion of possible future work with the tuning process of the control strategy can be found in Subsection 11.2.3.

For testing the control strategy during the tuning process, short two-day simulations has been used.

The main difficulty was balancing the weight of the tracking error with the price for heating in the flex MPC. This was partly due to the tracking error being a quadratic term and the heating price in DKK being linear. The initial guess of the weight of the Q matrix in the flex MPC was way too low, leading to the heat pump never being activated and the house just cooling down. Thus, the weight was increased significantly until the flex MPC began properly following the average room temperature reference. The Q matrix in the flex MPC was increased with a factor of 1.50E+09 compared to the initial guess from Subsection 9.3.3.

Next, it was noticed that the comfort MPC did not follow the $P_{\mathrm{h,ref}}$ from the flex MPC adequately, and therefore not lowering the heat input to the system enough before increases in heating from the sun. Thus, the corresponding diagonal entry in the Q matrix of the comfort MPC was increased with a factor of 10 compared to the initial guess from Subsection 8.4.3.

In the following section, the results of the final test after the tuning is shown.

10.3 Final test

For the final, test a 4 week simulation test was preformed to better evaluate the performance of the control strategy designed in this master thesis. The test was set up such that the temperature reference was 20°C for all the rooms the first two weeks, where after it was changed to 22°C, in an attempt to to evaluate its performance in regards to not just following a constant reference but also a sudden change. The values was decided based on Subsection 1.3.4.

Figure 10.2 and 10.3 shows the room temperatures and the estimated floor temperatures of the different rooms on the ground and first floor respectively.

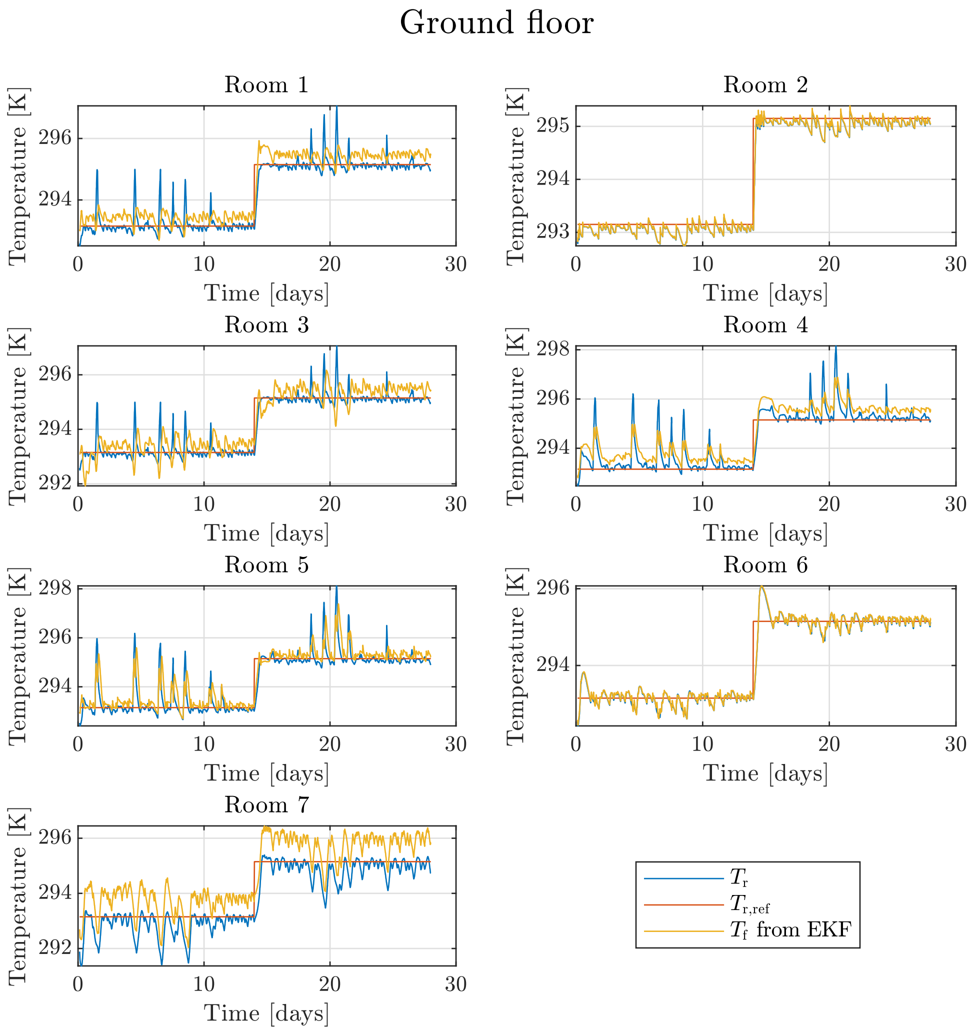

Figure 10.2: Ground floor temperatures.

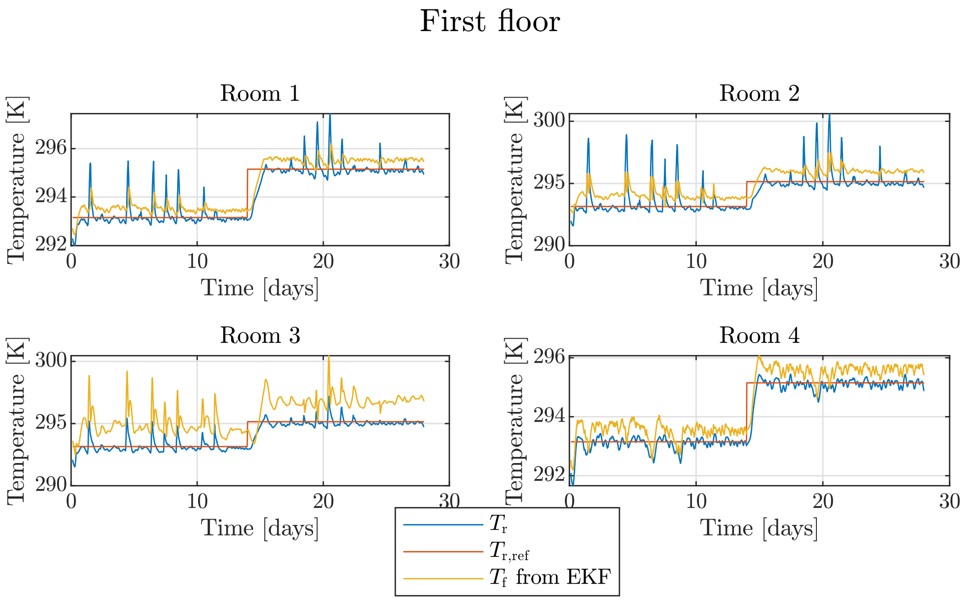

Figure 10.3: First floor temperatures.

As can be seen, the controller generally follows the room temperature references accurately except for a few instances coinciding with very high heating from solar radiation. At the reference change, the temperature of the rooms rapidly change to follow the new reference. Most rooms, have little to no overshoot, but e.g. room 6 on the ground floor is seen to have an overshoot of approximately 1 K.

Figure 10.2 also shows that the temperature in room 7 on the ground floor falls quicker than the rest of the rooms when the heat pump is turned off. This might be due to the fact that the room is not as well insulated [1]. Figure 10.4 shows a close ups of the room temperature and some of the other major signals around the reference change.

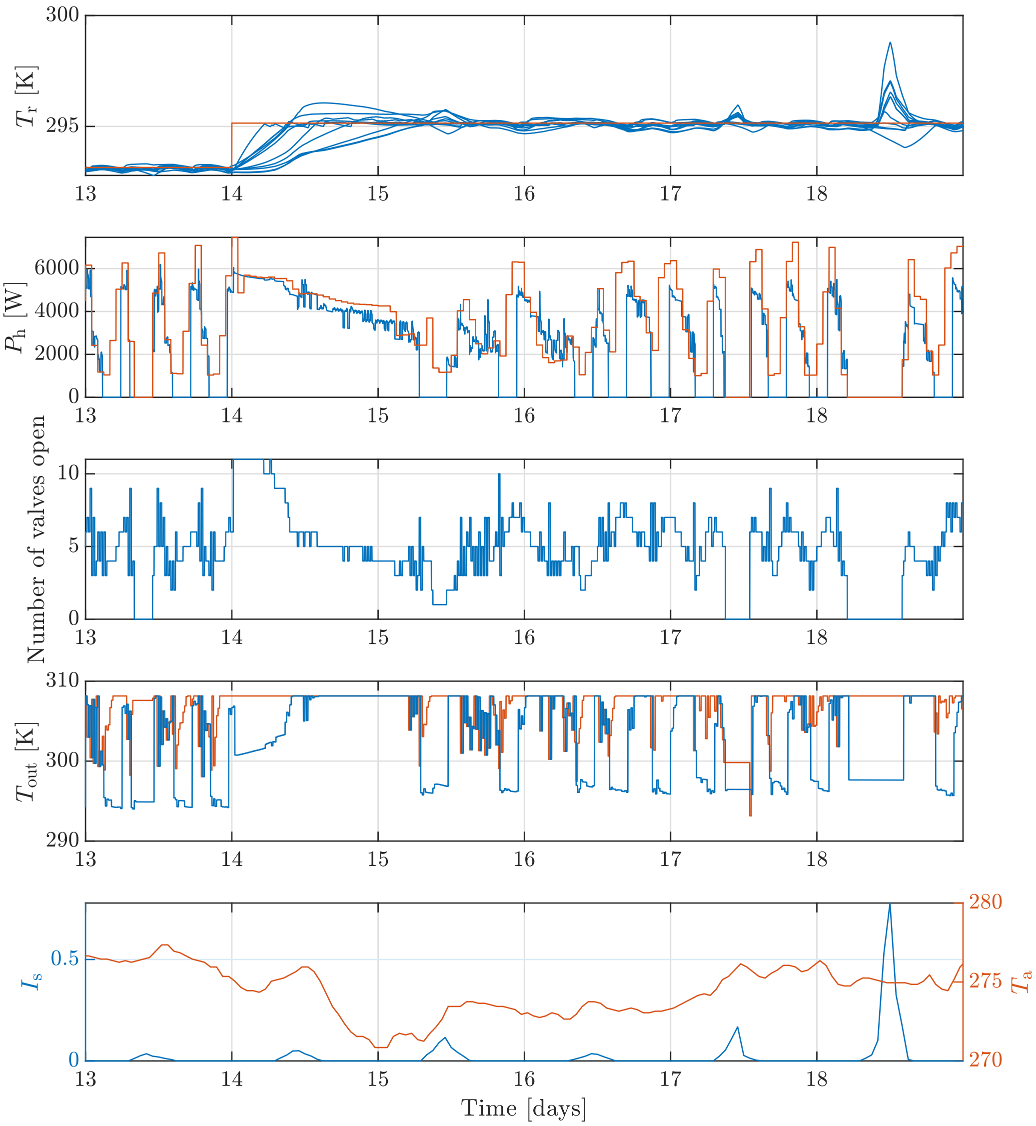

Figure 10.4: Value in blue, reference in orange. Except for the fifth plot of I_s and T_a.

The graph for P_h shows that the comfort MPC generally follows the heating budget from the flex MPC pretty well. However, the comfort MPC does at times turn off the heat pump by decreasing the reference for T_{out} too much. See the big decrease on day 15 for an example of this.

The graph of the number of valves open, shows how generally not all valves are open, thus the comfort MPC dynamically changes the distribution of the energy from the heat pump to the rooms. However, as to be expected, during the reference change on day 14 all valves are opened for a few hours, as all the rooms have a deficit in energy compared to the reference.

As can be seen from the graph for the forward temperature T_{out}, heat pump generally delivers the forward temperature specified by the comfort MPC. The major deviations coincide with when the heat pump is turned off which is to be expected and not considered a problem.

As can be seen from the graph for P_{h} and I_{s}, the flex MPC does reduce the heating budget and then turn off the heat pump before big increases in heating from solar radiation in an attempt to combat this disturbance. An example of this can be seen on day 18. Note, further lowering in heating before the I_{s} spikes could potentially be beneficial, since the room temperatures do not decrease noticeably with the current setup. A small decrease in room temperatures before the spike in I_{s} would decrease the size of the room temperature increase during the spike which would be favourable.

Figure 10.5 shows the major signals in regards to the flex MPC's cost function.

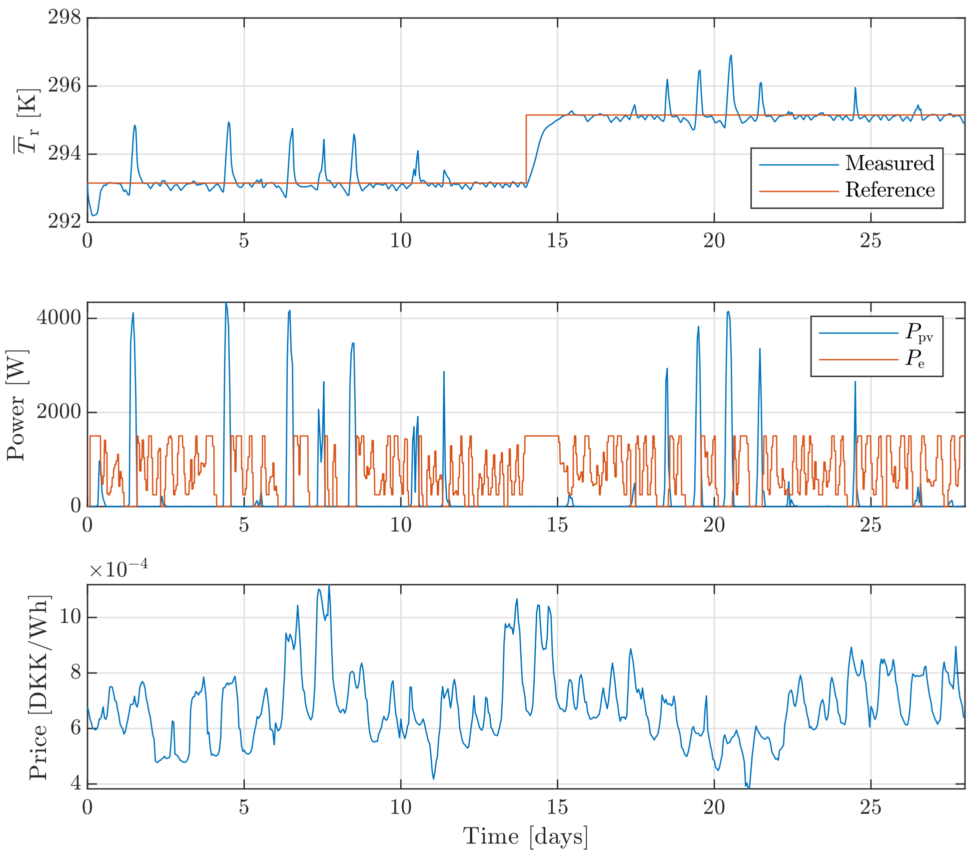

Figure 10.5: Major signals in regards to the flex MPC's cost function during the 28 day test.

As can be seen the average room temperature $\overline{T}_{\mathrm{r}}$ generally follows the reference, but with the same spikes due to the solar heating. Figure 10.5 also shows that during peak production hours for the PV panels, they generate more than the heat pump can even consume. However, this naturally also coincides with when the average room temperature error is the greatest. I.e. free heating is available exactly when heating is not really needed. Further discussion of this will be given in Subsection 11.2.3.

Using the 28 days of data two performance measures for the controller can be calculated. First, as a measure of comfort we calculated root mean square error for the reference tracking of the weighted average room temperature.

$$\mathrm{RMS}(\overline{T}_{\mathrm{r}} - \overline{T}_{\mathrm{r,ref}}) = 0.3651 \tag{10.1}$$

And for the cost of heating the house, the total monetary price is calculated:

$$\text{Total cost} = 213.30 \text{ DKK} \tag{10.2}$$

As no other control strategy is available for comparison, there is little value in these performance measures. This will futher be discussed in Subsection 11.2.4.

To evaluate how well fast the implemented control strategy is, the Python package `ttictoc` was used to time the entire control loop on every time step, i.e. everything except the plant. The logging of computation time was only performed for a two-day test using a 6 core Intel i7-10750H with a base clock speed of 2.6 GHz, but boosting to an average 4.5 GHz during the test. Note, using Windows' task manager it was noticed that the task was primarily single-threaded. Figure 10.6 shows the logged computation times.

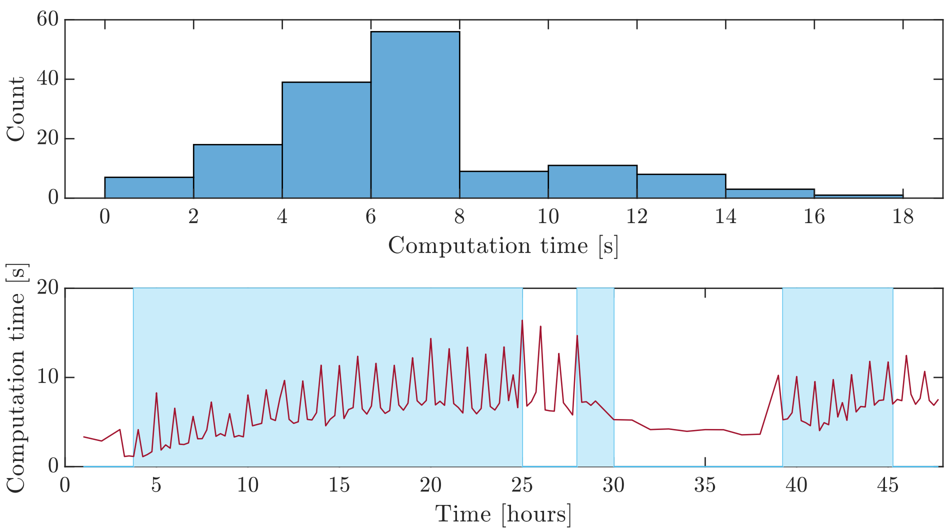

Figure 10.6: Computation times for the control loop during the two-day test. Regions coloured in blue are when the heat pump is turned on, i.e. both the flex and comfort MPC are running.

Figure 10.6 shows the computation times are generally quite low, which allows for more rapid testing of the control strategy. It can also be seen that during time steps where the KF and flex MPC updates the computation time is significantly longer as expected. The average computation time for the control loops is 6.7 s, which is considered more than satisfactory considering the 15 minute sampling time.

This concludes the development and testing of the control strategy. The next chapter concerns a discussion of choices and possible future work for project.

Part IV

Discussion and Conclusion

11 Discussion

This chapter concerns a discussion of different parts of the project and ideas for future work. It contains the following two sections:

- Section 11.1 discusses some of the discrepancies between the Dymola model and the real house.

- Section 11.2 focuses on the different points of the control strategy presented in this report and how it can be improved.

11.1 Discrepancy between model and reality

In this section the main differences between the Dymola model and the real house will be discussed. Furthermore, a discussion of how the current disturbances can be improved in order to make the simulation environment represent reality more closely will be given.

11.1.1 Thermal energy storage

As stated in the documentation on house that has been modelled, the heat pump has a 190 L hot water tank built in [1]. This has not been modelled in the Dymola model [1]. A Thermal Energy Storage (TES) like this hot water tank can be beneficial as it will be possible to store energy for later use, either when the electricity price is low or when there is a surplus of power from the Photovoltaic (PV) system. Therefore, it could be interesting for future work to model a TES in the Dymola model, as well as the models used for design of the control strategy, in order to study the potential gains of the implemented control strategy with a TES included.

Another argument for studying the inclusion of a TES in the house model is that other studies claim that it can be beneficial when applying MPC strategies for heating control of residential buildings. A study made, where a MPC strategy is applied with a house with PV panels and a TES, showed a 14.5% improvement compared to the same setup without the TES included[43].

Next improvements to the disturbances and forecasting data used for the Dymola model will be discussed.

11.1.2 Disturbances and forecasting noise

In the Dymola model some disturbances that might have an impact on the model have not been taken into account such as cooking activities and occupants opening windows once in a while. The mechanical ventilation system also has not been taken into account [1]. These elements could potentially be modelled as disturbances in future work.

In the current implementation of the control strategy presented in Chapter 8 and Chapter 9, the forecasting data used is not actual forecasts, but the real future values used by the Dymola model simulation. In reality, this is normally not the case as forecasting of the weather, ambient temperature and the electricity price is never completely certain. Thus, another point of future work will be to look into methods for adding noise to the forecasting data to test the control scheme's performance in regards to uncertain forecast more like real world data. This added noise would probably require that the uncertainty of the forecast rises with longer time horizons. Moreover, the simulated noise should probably have some type of correlation to the forecasts at previous times. E.g. the weather forecast at 11 o'clock probably has to be somewhat correlated with the weather forecast at 12 o'clock.

Next in Section 11.2 a discussion of the design and implementation of the developed control strategy will be discussed.

11.2 Controller design

In this section the design and implementation of the developed control strategy will be discussed.

11.2.1 Different time of year

At the moment, the implemented control strategy has only been tested with disturbance data from January 2018 and price data from 2021 January i.e in the winter. In order to evaluate the robustness of the control strategy it might be beneficial to check the controller performance during other seasons e.g during summer to see if the controller actually turns off the heat pump during hot summer days where it is not needed.

11.2.2 Choice of MIQP solver

As described in Subsection 8.2.2 CasADi is used with BONMIN in order to solve the MIQP problem for the comfort MPC and the flex MPC. This has proven to be time consuming with regard to the implementation of the control strategy as the documentation for CasADi and BONMIN is very minimalistic and the behaviour of the solver is very sensitive to how the MIQP problem is formulated in the implementation and the settings of the solver. Therefore, it is recommended to look into other options for implementations of MIQP solvers for future work. Furthermore, the solver included in BONMIN utilises a general Mixed Integer Non-Linear Programming (MINLP) solver [39].

In Subsection 8.4.3 it is stated that the comfort MPC is a Mixed Integer Preditive Control (MIPC) consisting of a Mixed Integer Quadratic Programming (MIQP) problem with a quadratic and convex cost function, and linear inequality constraints. This guarantees a globally optimal solution to the control problem. In order to optimise the computation time for solving the comfort MPC optimisation problem, it might be favourable to look into more efficient solvers specifically designed for this problem type instead of a general mixed integer NLP solver

For the flex MPC the MIQP problem also has a convex, quadratic cost function, but with both mixed integer linear inequality constraints and non-linear continuous constraints. Thus, a mixed integer NLP solver is probably still the best option.

11.2.3 Tuning

In this section the choice of the parameters the implemented control strategy i.e prediction horizons, sample times, weights on tuning matrices and valve positions will be discussed. One of the main ideas behind the design choices for the parameters for the control strategy implementation is to be able to run the simulation significantly faster than real time. E.g instead of the simulation running for a month in order to simulate the behaviour of the house during one month, it is desired that a control strategy can be simulated in a few hours for a whole month instead.

For the two controllers designed in Chapter 8 and Chapter 9 the prediction horizons H_p have been chosen as 1 hour for the comfort MPC and 24 hours for the flex MPC. For the comfort MPC the choice of 1 hour i.e $H_\mathrm{p} = 4$, has been made in order for the simulations to be able to run faster than real time. It would for example simply be too time consuming to have a higher computation times due to a longer prediction horizon.

For the flex MPC the 24 hour prediction horizon i.e $H_\mathrm{p} = 24$ a study has been made by [44] where different prediction horizons have been compared on a similar system showing that longer prediction horizons can improve the control strategy up to a prediction horizon on 96 hours.

One problem with having longer prediction horizons for the flex MPC is that it require more data for forecasting and according to [44] the improvement in controller performance also depends on the quality of the data used for forecasting. Therefore it can be beneficial to study the trade-off quality of the forecasting data and the length of the prediction horizon for the flex MPC as a point for future work.

Another point for discussion is the choice of sample time for the the comfort MPC. The choice of $T_\mathrm{s} = 15$ for the comfort MPC is mainly an initial educated guess based on the fact that the system has relatively quick dynamics when exposed to e.g periods of high solar radiation. Another trade-off that has to be looked into is the trade-off between longer sample times and the fit of the model. This could be beneficial as a longer sample time for the comfort MPC will provide longer time for computations. In order to investigate this trade-off a detailed analysis of the system dynamics has to be made in order to determine if e.g. the Nyquist-Shannon sample theorem is respected etc.

In Subsection 8.4.3 the valve positions has also been chosen to be constant throughout the prediction horizon in the comfort MPC. This has been done in order to keep the computation complexity of the optimal control problem from growing exponentially. It might be beneficial to study how the controller behaves if the valve positions are not constrained to be constant throughout the prediction horizon as this might give the comfort MPC more flexibility in how the heat is distributed in the house.

Lastly as mentioned in Section 10.2 the tuning of the weight matrices of the flex MPC and the comfort MPC has only been touched upon briefly as the process of testing different different weights is very time consuming.

11.2.4 Performance gains

Another untouched subject of significant matter is the problem of benchmarking the performance gain of using a control strategy like the one presented in this report. As stated in Section 1.4 a reduction in energy costs can be achieved by utilising optimal control strategies for heating of houses [21] [22] [26] [45]. However, the reduction entirely depends on what developed control strategies are compared to e.g. [45] where an MPC is compared to different varieties of Rule-Based Control (RBC). In this study the reduction also depends heavily on the TES size, tuning of the different controllers and choice of control strategy.

Therefore a framework for finding a good point of reference for benchmarking the performance gain for the implemented control strategy can be developed.

Furthermore, one thing that can be beneficial to assessing the validity of the simulation results is to test the implemented control strategy on the real house that is described in [1].

Next in Chapter 12 the main findings, with regard to the problem statement from Section 1.5, will be presented.

12 Conclusion

This master thesis has attempted to answer the problem statement:

"How can a control system be designed that minimises the energy cost, while maintaining comfort levels for inhabitants in the house."

from Section 1.5.

The house in question is used as part of the OPSYS 2.0 project [1]. It is built according to the Danish Nearly Zero-Emission Building (NZEB) guidelines known as BK 2020. The heating of the house is done using hydronic underfloor heating, with individually controlled circuits for every room. The hot water for the underfloor heating is supplied by an electrically driven Air to Water Heat Pump (AWHP), which gets some of its power requirements fulfilled by the Photovoltaic (PV) panels installed on the roof.

A high-fidelity model of the house created in Dymola has been made available to use as stand-in for the real house. No data from the real house has been used in this project.

A hierarchical control scheme has been developed containing two levels. The two levels of the control scheme can be summarised as follows:

- The upper level:
  - Goal: Determine a heating power budget based on the average house dynamics and forecast data of weather, electricity prices, and power generation of the PV panels.
  - Sample time: 60 minutes.
- The lower level:
  - Goal: Maintain individual room temperatures while following the power budget and taking weather forecasts into account and distribute the heating budget such that discomfort is reduced.
  - Sample time: 15 minutes.

A block diagram of the control scheme can be seen on Figure 12.1.

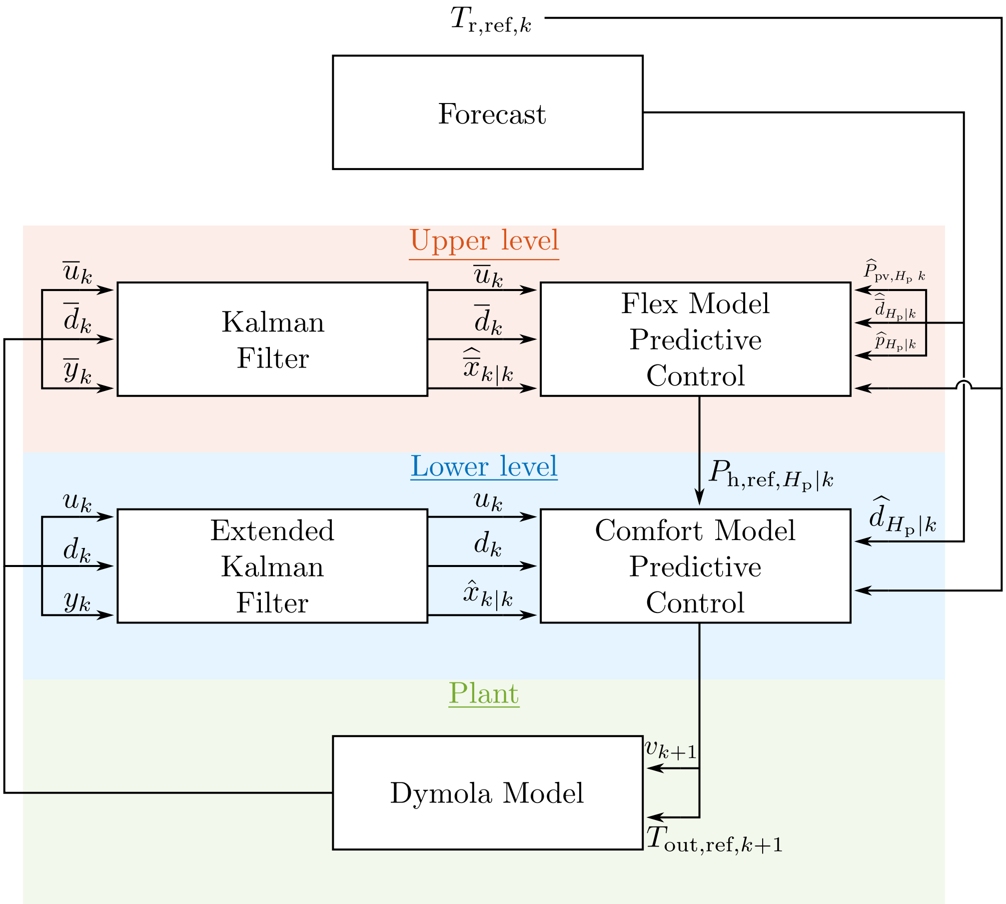

Figure 12.1: Simplified block diagram of how the different control modules are connected with each other, forecast data, and the Dymola model described in Subsection 1.3.2.

To facilitate the design of the two levels in the control strategy, non-linear models have been developed in Chapter 3 and 4. Each of these models have been designed with simplicity in mind, to allow Model Predictive Control (MPC) formulations that are efficiently solvable despite the non-linearities.

For each of the models, state observers have been designed. This is due to not all states being measured and the MPCs requires all states to be known. Due to the non-linear dynamics of the model for the lower level, an Extended Kalman Filter (EKF) has been developed for it. The upper level has linear dynamics and thus a simple Kalman Filter (KF) has been used for it instead.

For each of the two levels a MPC has been developed. It has been decided to use MPCs due to the ease of including disturbance forecasting. In the case of the models, the disturbances, that are forecasted, are weather in the form of ambient temperature and solar radiation. Additionally, forecasts for the spot price of electricity is used in the flex MPC of the upper level for reducing the cost of heating the house for the consumer.

The flex MPC calculates an optimal heating power budget on an hourly basis, based on the average dynamics of the house. The lower level's comfort MPC then attempts to follow this power budget while maintaining the individual room temperatures of the house. This is done by every 15 minutes setting a reference for the forward temperature of the heat pump and setting the valve positions of the underfloor heating circuit in an optimal configuration of open and closed, to direct the heating power to the rooms that require it.

The non-linearities in the model for the lower level has been handled by reformulating the dynamics into the Mixed Logical Dynamical (MLD) system framework. This allows the MPC optimal control problem to be formulated as a Mixed Integer Quadratic Programming (MIQP) problem with linear constraints, which can efficiently solved through the open-source solver Basic Open-source Nonlinear Mixed INteger programming (BONMIN).

To test the control strategy a 28 day simulation with varying room temperature reference was run. The results for each of the 11 rooms in the house can be seen on Figure 12.2 and 12.3.

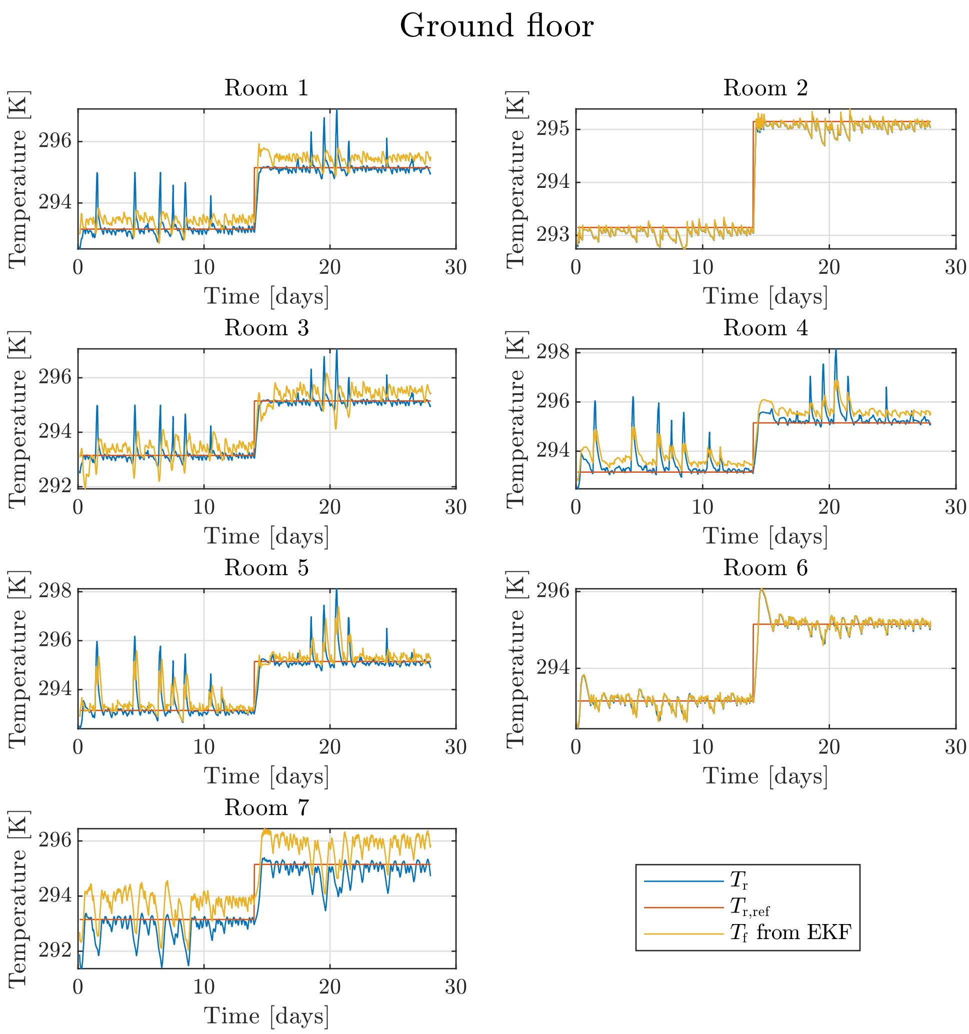

Figure 12.2: Ground floor temperatures.

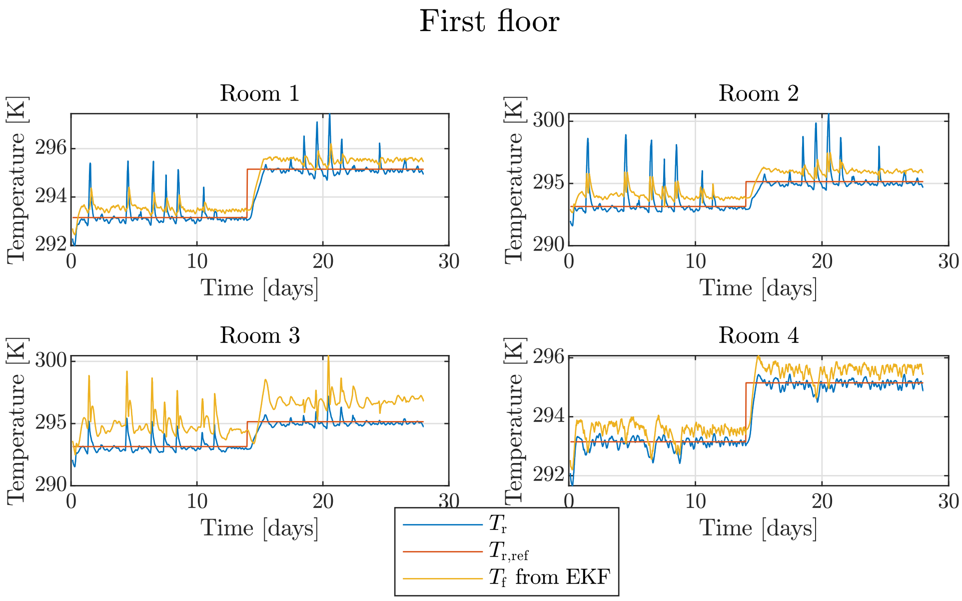

Figure 12.3: First floor temperatures.

As stated in Subsection 11.2.4, no other control strategies have been made available for comparison, but the developed control strategy of this master thesis achieves stability and general disturbance rejection.

Bibliography

[1] Ivan Katic et al. *OPSYS 2.0*. Report. Under review. Dansk teknologisk institut, NeoGrid, Wavin, Bosch, Aalborg university.

[2] International Energy Agency. "Global Energy Review". In: (2021). URL: https://iea.blob.core.windows.net/assets/d0031107-401d-4a2f-a48b-9eed19457335/GlobalEnergyReview2021.pdf.

[3] Directorate-General for Climate Action European Commission. *COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS Stepping up Europe's 2030 climate ambition Investing in a climate-neutral future for the benefit of our people*. Sept. 2020. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52020DC0562.

[4] European Statistical Office. *Electricity price statistics*. Oct. 2021. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Electricity_price_statistics#Electricity_prices_for_household_consumers.

[5] European Comission. *REPowerEU: Joint European action for more affordable, secure and sustainable energy*. Mar. 2022. URL: https://ec.europa.eu/commission/presscorner/detail/en/IP_22_1511.

[6] International Energy Agency. *Data and statistics*. 2021. URL: https://www.iea.org/data-and-statistics/data-browser?country=WORLD&fuel=Energy%20consumption&indicator=TFCShareBySector.

[7] European Statistical Office. *Energy consumption in households*. June 2021. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households.

[8] European Commision. *Nearly zero-energy buildings*. 2022. URL: https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/nearly-zero-energy-buildings_en#:~:text=Nearly%20zero%2Demission%20building%20(NZEB,produced%20on%2Dsite%20or%20nearby..

[9] Bolig- og Planstyrelsen. *Bygningsklasser 2020*. URL: https://bygningsreglementet.dk/Historisk/BR18_Version1/Ovrige-bestemmelser/25/Krav (visited on 02/17/2022).

[10] Danish Technological Institute. *Project - OPSYS 2.0*. URL: https://www.dti.dk/projects/project-opsys-2-0/40640 (visited on 02/27/2022).

[11] Søren Østergaard Jensen. *House model for Dymola of a real house with heat pump and underfloor heating.* Tech. rep. 64018-0581. Gregersensvej 1,2630 Taastrup: Danish Technological Institute, Apr. 2020.

[12] Modelica association. *Modelica language.* URL: https://modelica.org/modelicalanguage.html (visited on 03/01/2022).

[13] Christian Andersson, Johan Åkesson, and Claus Führer. *PyFMI: A Python Package for Simulation of Coupled Dynamic Models with the Functional Mock-up Interface.* English. Vol. LUTFNA-5008-2016. Technical Report in Mathematical Sciences 2. Centre for Mathematical Sciences, Lund University, 2016.

[14] Neogrid. *Electricity price data.* URL: https://neogrid.dk/projekter/opsys-2-0/ (visited on 03/01/2022).

[15] Kim Gregersen. *Undersøgelse: Så varmt har danskerne det i deres hjem.* 2022. URL: https://www.bolius.dk/undersoegelse-saa-varmt-har-danskerne-det-i-deres-hjem-39458.

[16] Arbejdstilsynet. *AT-vejledninger - Indeklima.* 2008. URL: https://at.dk/regler/at-vejledninger/indeklima-a-1-2/.

[17] Bjarne W. Olesen. "Radiant Floor Heating In Theory and Practice". In: *Ashrae Journal* 44 (2002), pp. 19–26.

[18] Ida Blomsterberg and Tue Patursson. *Vandbaseret gulvvarme.* 2020. URL: https://www.bolius.dk/vandbaseret-gulvvarme-14346.

[19] Robert Bosch A/S. *Compress 6000 AW - Installationsvejledning.* 2014. URL: https://www.vvs-eksperten.dk/sites/vvs-eksperten.dk/files/2018-05/346768005_installationsvejledning_0.pdf.

[20] Thibault Q. Péan, Jaume Salom, and Ramon Costa-Castelló. "Review of control strategies for improving the energy flexibility provided by heat pump systems in buildings". In: *Journal of Process Control* 74 (2019). Efficient energy management, pp. 35–49. ISSN: 0959-1524. DOI: https://doi.org/10.1016/j.jprocont.2018.03.006. URL: https://www.sciencedirect.com/science/article/pii/S0959152418300489.

[21] Gabrielle Masy et al. "Smart grid energy flexible buildings through the use of heat pumps and building thermal mass as energy storage in the Belgian context". In: *Science and Technology for the Built Environment* 21.6 (2015), pp. 800–811. DOI: 10.1080/23744731.2015.1035590. URL: https://doi.org/10.1080/23744731.2015.1035590.

[22] Roel De Coninck and Lieve Helsen. "Practical implementation and evaluation of model predictive control for an office building in Brussels". In: *Energy and Buildings* 111 (2016), pp. 290–298. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2015.11.014. URL: https://www.sciencedirect.com/science/article/pii/S0378778815303790.

[23] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback control of dynamic systems.* 5.. ed. Prentice Hall, 2006. ISBN: 0131499300.

[24] Raymond A. Serway and John W. Jewett Jr. *Physics for Scientists and Engineers with Modern Physics, Sixth Edition.* Belmont, CA 94002 USA: David Harris, 2004.

[25] Ettore Zanetti et al. "Energy saving potentials of a photovoltaic assisted heat pump for hybrid building heating system via optimal control". In: *Journal of Building Engineering* 27 (2020), p. 100854. ISSN: 2352-7102. DOI: https://doi.org/10.1016/j.jobe.2019.100854. URL: https://www.sciencedirect.com/science/article/pii/S2352710219300701.

[26] Rasmus Halvgaard et al. "Economic Model Predictive Control for building climate control in a Smart Grid". In: *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*. 2012, pp. 1–6. DOI: 10.1109/ISGT.2012.6175631.

[27] Qiong Chen, Nan Li, and Wei Feng. "Model predictive control optimization for rapid response and energy efficiency based on the state-space model of a radiant floor heating system". In: *Energy and Buildings* 238 (2021), p. 110832. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2021.110832. URL: https://www.sciencedirect.com/science/article/pii/S037877882100116X.

[28] B. Lehmann et al. "Intermediate complexity model for Model Predictive Control of Integrated Room Automation". In: *Energy and Buildings* 58 (Dec. 2012), pp. 250–262. DOI: 10.1016/j.enbuild.2012.12.007.

[29] Peter Radecki and Brandon Hencey. "Online building thermal parameter estimation via Unscented Kalman Filtering". In: *2012 American Control Conference (ACC)*. 2012, pp. 3056–3062. DOI: 10.1109/ACC.2012.6315699.

[30] MathWorks. *lsqnonlin - Solve nonlinear least-squares (nonlinear data-fitting) problems*. 2022. URL: https://se.mathworks.com/help/optim/ug/lsqnonlin.html?s_tid=doc_ta.

[31] Thomas F. Coleman and Yuying Li. "An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds". In: *SIAM Journal on Optimization* 6.2 (1996), pp. 418–445. DOI: 10.1137/0806023. eprint: https://doi.org/10.1137/0806023. URL: https://doi.org/10.1137/0806023.

[32] Thomas F. Coleman and Yuying Li. "On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds". In: *Mathematical Programming* 67.1 (Oct. 1994), pp. 189–224. ISSN: 1436-4646. DOI: 10.1007/BF01582221. URL: https://doi.org/10.1007/BF01582221.

[33] Mathworks. *mldivide*. 2022. URL: https://se.mathworks.com/help/matlab/ref/mldivide.html.

[34] John Leth. *A crash course in linear system theory*. 2021. URL: https://www.moodle.aau.dk/pluginfile.php/2151822/course/section/496728/note.pdf.

[35] Alberto Bemporad and Manfred Morari. "Control of systems integrating logic, dynamics, and constraints". In: *Automatica* 35.3 (1999), pp. 407–427. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(98)00178-2.

[36] A. Bemporad, G. Ferrari-Trecate, and M. Morari. "Observability and controllability of piecewise affine and hybrid systems". In: *IEEE Transactions on Automatic Control* 45.10 (2000), pp. 1864–1876. DOI: 10.1109/TAC.2000.880987.

[37] Gurobi Optimization LLC. *Mixed-integer programming (MIP) - A Primer on the Basics*. URL: https://www.gurobi.com/resource/mip-basics/.

[38] Joel A E Andersson et al. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4.

[39] Pierre Bonami et al. *Basic Open-source Nonlinear Mixed INteger programming.* URL: https://www.coin-or.org/Bonmin/.

[40] Marco Chiarandini. *Lecture 9, Integer Linear Programming.* 2017. URL: https://imada.sdu.dk/~marco/Teaching/AY2017-2018/DM559/Slides/dm545-lec9.pdf.

[41] J.M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, 2002.

[42] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* 2009. URL: https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf.

[43] Joan Tarragona, Cèsar Fernández, and Alvaro de Gracia. “Model predictive control applied to a heating system with PV panels and thermal energy storage”. In: *Energy (Oxford)* 197 (2020), pp. 117229–. ISSN: 0360-5442.

[44] “An optimal home energy management system for modulating heat pumps and photovoltaic systems”. eng. In: *Applied energy* 278 (2020), pp. 115661–. ISSN: 0306-2619.

[45] David Fischer et al. “Comparison of control approaches for variable speed air source heat pumps considering time variable electricity prices and PV”. eng. In: *Applied energy* 204 (2017), pp. 93–105. ISSN: 0306-2619.

[46] Mohinder S. Grewal. *Kalman filtering : theory and practice using MATLAB.* eng. 3. ed. Hoboken, N.J: John Wiley and Sons, 2008. ISBN: 9780470173664.

[47] Torben Knudsen and John Leth. “Particle Filters Revisited”. In: *2020 Australian and New Zealand Control Conference (ANZCC).* 2020, pp. 36–41. DOI: 10.1109/ANZCC50923.2020.9318384.

Glossary

AWHP Air to Water Heat Pump. 4, 52, 109

BONMIN Basic Open-source Nonlinear Mixed INteger programming. 73, 77, 81, 105, 111

CO_2 Carbon dioxide. 2
COP Coefficient of Performance. 52, 53, 54, 55, 56, 79

EKF Extended Kalman Filter. 12, 33, 44, 51, 61, 62, 63, 64, 65, 66, 95, 110, 120, 121, 124, 127, 129, 131, 133
EMPC Economic Model Predictive Control. 8
EU European Union. 2

FMI Functional Mock-up Interface. 6
FMU Functional Mock-up Unit. 6, 17

KF Kalman Filter. 10, 12, 61, 63, 67, 95, 101, 110, 127

MINLP Mixed Integer Non-Linear Programming. 105
MIPC Mixed Integer Preditive Control. 70, 78, 83, 86, 93, 105
MIQP Mixed Integer Quadratic Programming. 69, 70, 72, 73, 74, 76, 77, 82, 83, 87, 91, 92, 93, 105, 111
MLD Mixed Logical Dynamical. 69, 70, 71, 72, 76, 77, 85, 86, 111
MPC Model Predictive Control. 8, 9, 10, 12, 14, 61, 62, 68, 69, 70, 72, 73, 76, 78, 79, 81, 82, 83, 85, 86, 87, 88, 89, 91, 92, 93, 95, 96, 99, 100, 101, 104, 105, 106, 110, 111

NLP Non-Linear Programming. 105
NZEB Nearly Zero-Emission Building. 2, 3, 109

PV Photovoltaic. 3, 4, 5, 9, 10, 11, 61, 85, 90, 96, 100, 104, 109

QP Quadratic Programming. 73, 74, 79

RBC Rule-Based Control. 8, 9, 106

TES Thermal Energy Storage. 5, 104, 107

UKF Unscented Kalman Filter. 121, 124, 127

ZEB Zero-Emission Building. 3
ZOH Zero Order Hold. 32, 49, 96

Part V

Appendix

A State Observation methods

This chapter will focus on the different estimation methods for state estimation. Furthermore a brief discussion on model uncertainties validation methods will be discussed. Lastly the implementation of an estimation algorithm used for the project will be described as well. The main focuses are listed in the following:

1. Linear Kalman filters
2. Extended Kalman filters.
3. Unscented Kalman filters
4. Particle filters
5. Implementation of an Extended Kalman filter
6. Model uncertainties and validation

A.1 Kalman filters

Both the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are modified versions of the Kalman Filter (KF).

A KF is a state estimator designed for a system on the form

$$x_{k+1} = A_k\, x_k + B_k\, u_k + w_k, \qquad w_k \overset{i.i.d.}{\sim} \mathcal{N}(0, Q_k) \tag{A.1}$$

$$y_k = C_k\, x_k + D_k\, u_k + v_k, \qquad v_k \overset{i.i.d.}{\sim} \mathcal{N}(0, R_k) \tag{A.2}$$

That is, it is the optimal solution to the discrete-discrete estimation problem, where measurements of outputs y and inputs u up to time k is used to find $\hat{x}_k|k$ which minimises the mean square error between the real states and the estimates at time k. The estimator is unbiased and the state error is uncorrelated to the measurements.

The normal Kalman filter is [46]:

Initial condition:

$$x_0 \in \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{A.3}$$

Measurement update after receiving y_k and u_k:

$$\hat{y}_{k|k-1} = C_k\,\hat{x}_{k|k-1} + D_k\,u_k \tag{A.4}$$

$$\tilde{y}_{k|k-1} = y_k - \hat{y}_{k|k-1} \tag{A.5}$$

$$K_k = P_{k|k-1}\,C_k^{\mathrm{T}}\,\left(C_k\,P_{k|k-1}\,C_k^{\mathrm{T}} + R_k\right)^{-1} \tag{A.6}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\,\tilde{y}_{k|k-1} \tag{A.7}$$

$$P_{k|k} = (I - K_k\,C_k)\;P_{k|k-1}\;(I - K_k\,C_k)^{\mathrm{T}} + K_k\,R_k\,K_k^{\mathrm{T}} \tag{A.8}$$

$$\tag{A.9}$$

Time update:

$$\hat{x}_{k+1|k} = A_k\,\hat{x}_{k|k} + B_k\,u_k \tag{A.10}$$

$$P_{k+1|k} = A_k\,P_{k|k}\,A_k^{\mathrm{T}} + Q_k \tag{A.11}$$

A.2 Extended Kalman filter

The Extended Kalman Filter (EKF) attempts to use the non-linear model where applicable and otherwise linearise the non-linear model around the current state estimates and input measurements. It allows for estimation of non-linear systems on the form [46]:

$$x_{k+1} = f(x_k, u_k) + w_k, \qquad w_k \overset{i.i.d.}{\sim} \mathcal{N}(0, Q_k) \tag{A.12}$$

$$y_k = h(x_k, u_k) + v_k, \qquad v_k \overset{i.i.d.}{\sim} \mathcal{N}(0, R_k) \tag{A.13}$$

(Changes from KF highlighted in red)
Initial conditions:

$$x_0 \sim \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{A.14}$$

$$\tag{A.15}$$

Measurement update:

$$\begin{aligned}
\hat{y}_{k|k-1} &= h(\hat{x}_{k|k-1}, u_k) \\
\tilde{y}_{k|k-1} &= y_k - \hat{y}_{k|k-1} \\
C_k &= \left.\frac{\partial h(x,u)}{\partial x^{\mathrm{T}}}\right|_{\hat{x}_{k|k-1}, u_k} \\
K_k &= P_{k|k-1}\, C_k^{\mathrm{T}} \left(C_k\, P_{k|k-1}\, C_k^{\mathrm{T}} + R_k\right)^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k\, \tilde{y}_{k|k-1} \\
P_{k|k} &= (I - K_k\, C_k)\; P_{k|k-1}\; (I - K_k\, C_k)^{\mathrm{T}} + K_k\, R_k\, K_k^{\mathrm{T}}
\end{aligned} \tag{A.16}$$

Time update:

$$\begin{aligned}
\hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, u_k) \\
A_k &= \left.\frac{\partial f(x,u)}{\partial x^{\mathrm{T}}}\right|_{\hat{x}_{k|k}, u_k} \\
P_{k+1|k} &= A_k\, P_{k|k}\, A_k^{\mathrm{T}} + Q_k
\end{aligned} \tag{A.17}$$

The EKF is neither optimal nor guarantied to be stable and if the initial estimate of the state is wrong, or if the process is modeled incorrectly, the filter may quickly diverge because of the linearisation. Never the less the EKF can give reasonable performance when the the system it is applied to is not too non-linear. Therefore as there are non-linear elements in the model designed in Section 3.4 the EKF for the time update step and a normal kalman filter is used for the measurement update step as the combined model has a linear output equation.

If the model is highly non-linear then an UKF is an option for estimation. This will be covered next.

A.3 Unscented Kalman filter

When the state transition and measurement models are highly nonlinear an Unscented Kalman Filter (UKF) can be used. UKF uses a sampling technique called the unscented transformation to estimate the expectations and covariances. Again the system is on the form [46]:

$$x_{k+1} = f(x_k, u_k) + w_k, \qquad w_k \overset{i.i.d.}{\sim} \mathcal{N}(0, Q_k) \tag{A.18}$$

$$y_k = h(x_k, u_k) + v_k, \qquad v_k \overset{i.i.d.}{\sim} \mathcal{N}(0, R_k) \tag{A.19}$$

We can write the KF for a linear DD problem where:

$$f(x_k, u_k) = A_k\, x_k + B_k\, u_k \tag{A.20}$$
$$h(x_k, u_k) = C_k\, x_k + D_k\, u_k \tag{A.21}$$
$$\tag{A.22}$$

as

(Changes from KF highlighted in red)
Initial conditions:

$$x_0 \sim \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{A.23}$$
$$\tag{A.24}$$

Measurement update:

$$\hat{y}_{k|k-1} = \mathbb{E}(h(\hat{x}_{k|k-1}, u_k) + v_k | \mathcal{Y}_{k-1}) \tag{A.25}$$
$$\tilde{y}_{k|k-1} = y_k - \hat{y}_{k|k-1} \tag{A.26}$$
$$K_k = \operatorname{Cov}(x_k, y_k | \mathcal{Y}_{k-1}) \operatorname{Cov}(y_k | \mathcal{Y}_{k-1})^{-1} \tag{A.27}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\, \tilde{y}_{k|k-1} \tag{A.28}$$
$$P_{k|k} = P_{k|k-1} - K_k \operatorname{Cov}(y_k, x_k | \mathcal{Y}_{k-1}) \tag{A.29}$$

Time update:

$$\hat{x}_{k+1|k} = \mathbb{E}(f(\hat{x}_{k|k}, u_k) + w_k | \mathcal{Y}_{k-1}) \tag{A.30}$$
$$P_{k+1|k} = \operatorname{Cov}(f(x_k, u_k) + w_k | \mathcal{Y}_{k-1}) \tag{A.31}$$

Now, we can use the Unscented Transform (UT) to estimate the expectations and covariances in the above for the non-linear DD problem.

The (scaled) UT algorithm:

Default parameters:

$$\alpha = 1, \quad \kappa = 2, \quad \beta = 0 \tag{A.32}$$

$$\lambda = \alpha^2(n+\kappa) - n, \quad k = \sqrt{n+\lambda}, \quad n = \dim(x) \tag{A.33}$$

u_i is the i^{th} eigenvalue for C_x.
l_i is the i^{th} eigenvector for C_x.
Sigma points:

$$x_i = \begin{cases} \mu_x & , i = 0 \\ \mu_x + k\, l_i \sqrt{u_i} & , 1 \le i \le n \\ \mu_x - k\, l_i \sqrt{u_i} & , n+1 \le i \le 2\,n \end{cases} \tag{A.34}$$

Weights:

$$w_i^t = \begin{cases} \frac{\lambda}{n+\lambda} & , i = 0, \, t = m \\ \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta & , i = 0, \, t = c \\ \frac{\lambda}{2\,(n+\lambda)} & , 1 \le i \le 2\,n \end{cases} \tag{A.35}$$

Expectation:

$$\hat{\mu}_y = \sum_{i=0}^{2\,n} w_i^m f(x_i) \tag{A.36}$$

Covariances:

$$\hat{C}_{yx} = \sum_{i=0}^{2\,n} w_i^c \left((f(x_i) - \hat{\mu}_y)\,(x_i - \hat{\mu}_x)^{\mathrm{T}} \right) \tag{A.37}$$

$$\hat{C}_y = \sum_{i=0}^{2\,n} w_i^c \left((f(x_i) - \hat{\mu}_y)\,(f(x_i) - \hat{\mu}_y)^{\mathrm{T}} \right) \tag{A.38}$$

The UT algorithm takes a function $y = f(x)$, and $\mathbb{E}(x)$ and $\mathrm{Cov}(x)$ as inputs. We can write this as:

$$\begin{bmatrix} \mathbb{E}(y) \\ \mathrm{Cov}(y, x) \\ \mathrm{Cov}(y) \end{bmatrix} = U(f, \mathbb{E}(x), \mathrm{Cov}(x)) \tag{A.39}$$

(Note, it also works for conditional probabilities). The UKF alogrithm then becomes:

Initial conditions:

$$x_0 \sim \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{A.40}$$

$$\tag{A.41}$$

Measurement update:

$$\begin{bmatrix} \hat{y}_{k|k-1} \\ \mathrm{Cov}(\tilde{y}_{k|k-1}, \tilde{x}_{k|k-1}) \\ C_k^h \end{bmatrix} = U(h, \hat{x}_{k|k-1}, P_{k|k-1}) \tag{A.42}$$

$$\mathrm{Cov}(\tilde{y}_{k|k-1}) = C_k^h + R_k \tag{A.43}$$

$$\mathrm{Cov}(\tilde{x}_{k|k-1}, \tilde{y}_{k|k-1}) = \mathrm{Cov}(\tilde{y}_{k|k-1}, \tilde{x}_{k|k-1})^{\mathrm{T}} \tag{A.44}$$

$$\tilde{y}_{k|k-1} = y_k - \hat{y}_{k|k-1} \tag{A.45}$$

$$K_k = \mathrm{Cov}(\tilde{x}_{k|k-1}, \tilde{y}_{k|k-1})\, \mathrm{Cov}(\tilde{y}_{k|k-1})^{-1} \tag{A.46}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\, \tilde{y}_{k|k-1} \tag{A.47}$$

$$P_{k|k} = P_{k|k-1} - K_k\, \mathrm{Cov}(\tilde{y}_{k|k-1}, \tilde{x}_{k|k-1}) \tag{A.48}$$

Time update:

$$\begin{bmatrix} \hat{x}_{k+1|k} \\ X \\ C_k^f \end{bmatrix} = U(f, \hat{x}_{k|k}, P_{k|k}) \tag{A.49}$$

$$P_{k+1|k} = \mathrm{Cov}(\tilde{x}_{k+1|k}) = C_k^f + Q_k \tag{A.50}$$

When having to choose between UKF and EKF, for the project one of the largest benefits of using EKF over UKF, when possible, lies in the complexity of the two algorithms. The UKF is more complex as the Uncented Transform has to be used in order to estimate the expectations and covariance for the time and update step for the UKF.

A.4 Particle Filter

In EKF/UKF the state estimate is described by the (approximation for) conditional mean and covariance of the state given the available measurements.

Mean and covariance are the only necessary parameters to specify a Gaussian probability density function (PDF). To the extend the state CPDF is well approximated by a Gaussian PDF, the EKF and UKF provides the state CPDF. However, for non-linearities in dynamics or measurement so large that the PDF are far from Gaussian the particle filter (PF) is a useful method as it seeks to estimate the CPDF for the state given the measurements without using the Gaussian assumption at all.

For PF we consider systems on the form [47]:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k), \quad w_k \in \mathrm{ID}, \quad p(w_k) \\ y_k &= h(x_k, u_k, v_k), \quad v_k \in \mathrm{ID}, \quad p(v_k) \end{aligned} \tag{A.51}$$

Note, w_k and v_k are no longer required to be additive nor have a Gaussian distribution.

We want to find $p(x_k|Y_k)$.

Bayesian filter
Initial conditions:

$$p(x_0) \triangleq p(x_0|Y_{-1}) \tag{A.52}$$

Measurement update:

$$\begin{aligned} p(y_k|Y_{k-1}) &= \int p(y_k|x_k)p(x_k|Y_{k-1})\,\mathrm{d}x_k \\ p(x_k|Y_k) &= \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \end{aligned} \tag{A.53}$$

Time update:

$$p(x_{k+1}|Y_k) = \int p(x_{k+1}|x_k)p(x_k|Y_k)\,\mathrm{d}x_k \tag{A.54}$$

$p(y_k|x_k)$ and $p(x_{k+1}|x_k)$ can only be found analytically when looking at e.g. a linear Gaussian state space model or a Hidden Markov Chain.

We instead use sampling methods to approximate the distributions.

Approximating continuous PDF with delta Dirac functions

$$\begin{aligned} f_a(x) &= \sum_{i=1}^{N_s} \pi_i \delta(x - x_i),\ \forall i,\ \pi_i = \frac{1}{N_s} \\ F_a(x) &= \int_{-\infty}^{x} f_a(t)\,\mathrm{d}t = \frac{\#(x_i \leq x)}{N_s} \end{aligned} \tag{A.55}$$

$$F_a(x) \to F(x) \text{ for } N_s \to \infty \text{ in a mean square sense}$$

Given $y = g(x)$, then

$$f_a(y) = \sum_{i=1}^{N_s} \pi_i \delta(y - g(x_i)) \tag{A.56}$$

Assume

$$p(x_k|Y_{k-1}) \approx \sum_{i=1}^{N_s} \pi^i_{k|k-1} \delta(x_k - x^i_k) \tag{A.57}$$

Substitute into the Bayes filter to gain

$$p(x_k|Y_k) = \sum_{i=1}^{N_s} \pi^i_{k|k} \delta(x_k - x^i_k) \tag{A.58}$$

where

$$\pi^i_{k|k} = \frac{p(y_k|x^i_k)\pi^i_{k|k-1}}{\sum_{i=1}^{N_s} p(y_k|x^i_k)\pi^i_{k|k-1}} \tag{A.59}$$

Notice:

- Aposteriori probabilities $\pi^i_{k|k}$ equals apriori probabilities $\pi^i_{k|k-1}$ scaled with measurement probabilities $p(y_k|x^i_k)$
- The approximation samples (particles) has not changed i.e. $x^i_k = x^i_{k|k-1} = x^i_{k|k}$

Particle filter Initial conditions: For $i = 1, \ldots, N$ generate $x^i_{1|0}$ using the PDF p_{x_1} and set $\pi^i_{1|0} = \frac{1}{N}$.
Measurement update:

$$\begin{aligned} \pi^i_{k|k} &= \frac{\pi^i_{k|k-1} p(y_k|x^i_{k|k-1})}{\sum_{j=1}^N \pi^j_{k|k-1} p(y_k|x^j_{k|k-1})} \\ x^i_{k|k} &= x^i_{k|k-1} \end{aligned} \tag{A.60}$$

Time update: For $i = 1, \ldots, N$ draw the index j according to the distribution $\pi^i_{k|k}$, and the process noise w according to p_w. Then find $x^i_{k+1|k}$ by mapping $x^j_{k|k}$ and w using the known state transition function:

$$x^i_{k+1|k} = f(x^j_{k|k}, u_k, w) \tag{A.61}$$

and set $\pi^i_{k+1|k} = \frac{1}{N}$.
Then

$$\begin{aligned} \hat{p}(x_k|Y_k) &= \sum_{i=1}^N \pi^i_{k|k} \delta(x_k - x^i_{k|k}) \\ \hat{p}(x_{k+1}|Y_k) &= \sum_{i=1}^N \pi^i_{k+1|k} \delta(x_{k+1} - x^i_{k+1|k}) \end{aligned} \tag{A.62}$$

If v_k is additive, then $p(y_k|x^i_k)$ has the same distribution as v_k with shifted mean value $h(x^i_k, u_k)$

$$p(x_k|Y_{k-1}) = \sum_{i=1}^N \pi^i_{k|k-1} \delta(x_k - x^i_k) \tag{A.63}$$

$$p(x_k|Y_k) = \frac{\sum_{i=1}^N p(y_k|x^i_k) \pi^i_{k|k-1} \delta(x_k - x^i_k)}{p(y_k|Y_{k-1})} \tag{A.64}$$

$$p(y_k|Y_{k-1}) = \sum_{i=1}^N p(y_k|x^i_k) \pi^i_{k|k-1} \tag{A.65}$$

$$p(x_k|Y_k) = \sum_{i=1}^N \pi^i_{k|k} \delta(x_k - x^i_k) \tag{A.66}$$

$$\pi^i_{k|k} = \frac{\pi^i_{k|k-1} p(y_k|x^i_{k|k})}{\sum_{j=1}^N \pi^j_{k|k-1} p(y_k|x^j_{k|k-1})} \tag{A.67}$$

$p(x_{k+1}|Y_k)$ can be approximated by drawing i according to the distribution $\pi^i_{k|k}$ and for this i drawing x^i_{k+1} according to $p(x_{k+1}|x^i_k)$. This is repeated N times given the approximation

$$p(x_{k+1}|Y_k) = \sum_{i=1}^{N} \pi^i_{k+1|k}\delta(x_{k+1} - x^i_{k+1|k})$$

$$\forall i,\ \pi^i_{k+1|k} = \frac{1}{N}$$

When comparing the Particle filter to EKF and UKF, one of the most important benefits is that it does not have the gaussian assumtion for the model. This makes it suitable for broader range of problems compared to Kalman Filters. Furthermore, it can find a global approximation unlike the UKF and EKF which only finds a local approximation. That being said, the Particle filters generally have much higher computational requirements compared to EKF UKF and KF's. Therefore it the EKF UKF and KF's are preferable to use when possible.

A.5 Implementation of EKF

In order to implement an EKF for estimating $T_{\mathrm{r},i,j}$ $T_{\mathrm{f},i,j}$ the EKF has been implemented in python. It has been implemented as a module so for easy interfacing with the Dymola model of the house.

The module has the following structure:

```
class extendedKalmanFilter():
    def __init__(self): # Initialisation of the filter and variables
    def getMeasurementUpdate(self, y): # Calculation of measurement ↩
        update
    def getTimeUpdate_AL_matrix(self,u): # Linarisation of A matrix.
    def getTimeUpdate_states(self, u, d): # Calculation of time update
    def Update_vars(self): # Update variables for next iteration
    def Update_filter(self, y, u, d): # Used to run the filter.
```

`__init__(self)` is a function that pre-allocates all variables for the state equations, initial conditions, measurement update, and time update, when the filter object is created.

```python
def __init__(self):
    # Define system matrices:
    model_data = sio.loadmat('model_matrices.mat');
    self.A = model_data['A']; self.B = model_data['B']
    self.E = model_data['E']; self.b = model_data['b']
    self.c = 4182
    V = np.hstack((np.eye(11), np.zeros((11,1))))
    self.F = self.c * np.diag(self.b[:,0].transpose()) @ V
    self.G = np.zeros((1,12))
    self.G[0,11] = 1
    self.H = np.hstack((np.eye(11),np.zeros((11,11))))

    # EKF signals:
    self.C = np.zeros((11, 22))
    self.C[:, 11:22] = np.eye(11)

    x_init = np.ones((22,1))*293.15 # 20 degree C
    self.xest = x_init        # estimated x_k/k
    self.xpre_old = x_init    # predicted x_k/k-1
    self.xpre = x_init        # predicted x_k+1/k
    self.ypre = 0        # predicted y_k/k-1
    self.yerr_old = 0        # error y_k/k-1
    self.P_init = np.eye(22)     #
    self.Pest = self.P_init      # estimated P_k/k
    self.Ppre = self.P_init       # prediction P_k+1/k
    self.Ppre_old = self.P_init   # predicted P_k/k-1
    self.R = np.eye(11)          #
    self.Q = np.eye(22)          #
```

`getMeasurementUpdate(self,y)` is a function that implements the measurement update step from Equation A.16. But since the $y_k = h(x_k, u_k)$ for the dynamical model designed in Section 3.4 is $y_k = C_d\, x_k$ linearisation of C_k is not needed:

```python
def getMeasurementUpdate(self, y):
    # measurement prediction.
    self.ypre = self.C @ self.xpre_old
    # Calculate measurement error update
    self.yerr_old = y - self.ypre
    temp = self.C @ self.Ppre_old @ self.C.transpose() + self.R
    # Calculate Kalman gain
    K = self.Ppre_old @ self.C.transpose() @ np.linalg.inv(temp)
    self.xest = self.xpre_old + K @ self.yerr_old
    self.Pest = (np.eye(22)-K@ self.C) @ self.Ppre_old @ (np.eye(22)-K @ ↩
        self.C).transpose() + K @ self.R @ K.transpose()
    return self.xest
```

`getTimeUpdate_states(self,u,d)` is a function that implements the time update step from Equation A.17.

```python
def getTimeUpdate_states(self, u, d):
    # Calculate a priori state estimate
    P = self.F @ u * self.G @ u - np.multiply((self.F @ u),(self.H @ self↩
        .xest ))
    self.xpre = self.A @ self.xest + self.E @ d + self.B @ P
    # Linearise A matrix.
    A_linear = self.getTimeUpdate_AL_matrix(u)
    # Calculate Updated a priori estimate covariance
    self.Ppre = A_linear @ self.Pest @ A_linear.transpose() + self.Q
```

`Update_vars(self)` mainly updates the a posteriori state estimate and covariance estimate.

Lastly `Update_filter(self, y, u, d)` is used for running the filter every time step. The function calls the measurement update step, the time update step and updates a posteriori state estimate and covariance estimate:

```
def Update_filter(self, y, u, d):
    self.getMeasurementUpdate(y)
    self.getTimeUpdate_states(u, d)
    self.Update_vars()
```

A.6 Validation

In this section the performance of the implemented EKF described in Section A.5 will be analysed with respect to model uncertainties.

In order to test the performance of the EKF, it has been chosen to add noise to the measurements of the ambient temperature T_{a}, solar radiation I_{s} and measurements of T_{r}. This has been done as the Dymola model used for simulation does not have measurement noise included.

The noise added will initially be Gaussian signals z_{T_a}, z_{T_r} and z_{I_s}. This means that the noisy signals for the simulation will become

$$\begin{aligned} \check{T}_a &= T_a + z_{T_a}, \qquad & z_{T_a} &\overset{i.i.d.}{\sim} \mathcal{N}(0, 0.5) \\ \check{T}_r &= T_r + z_{T_r}, \qquad & z_{T_r} &\overset{i.i.d.}{\sim} \mathcal{N}(0, 0.5) \\ \check{I}_s &= I_s + z_{I_s}, \qquad & z_{I_s} &\overset{i.i.d.}{\sim} \mathcal{N}(0, 0.003) \end{aligned} \tag{A.68}$$

Where I_s has the extra constraint:

$$0 \leq I_s \leq 1 \tag{A.69}$$

A.6.1 Results

Figure A.1 and Figure A.2 shows the the a plot of T_r, $\check{T}_r$ for each room and the estimates from the implemented EKF.

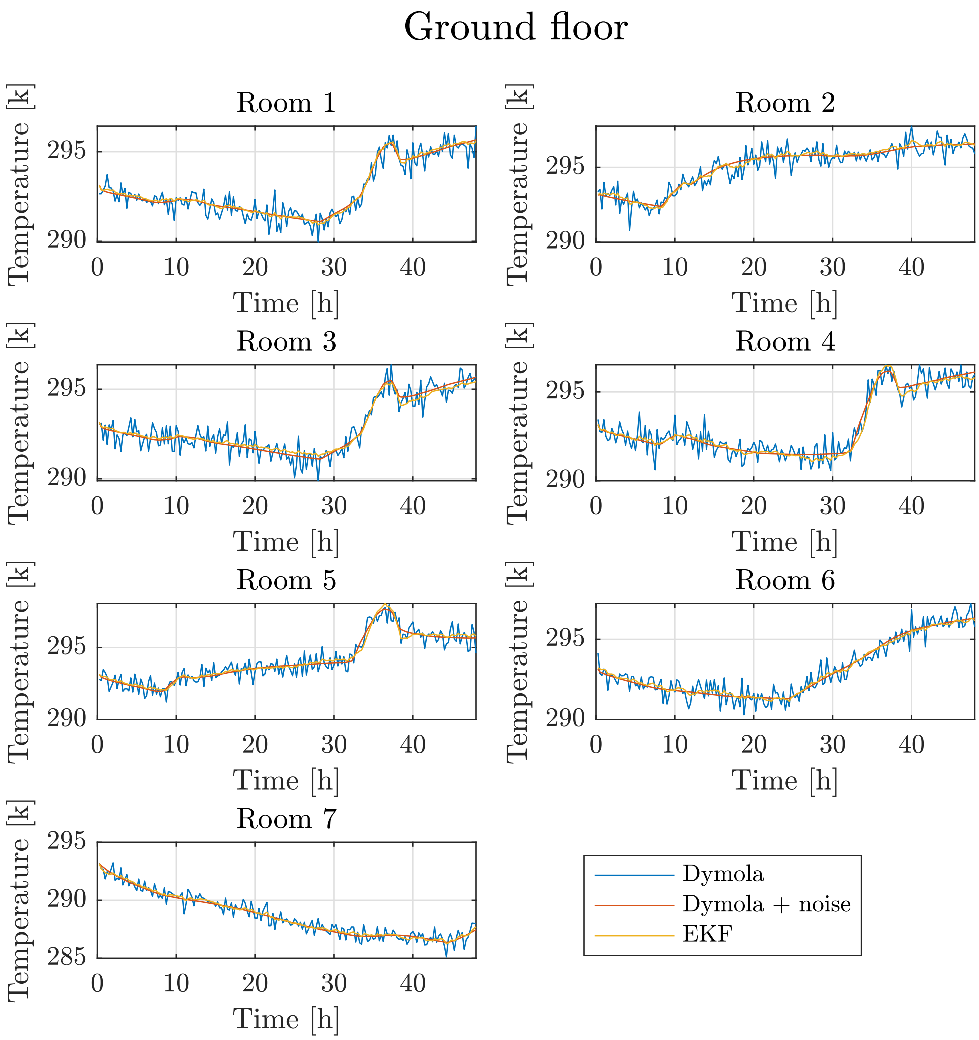

Figure A.1: Test results for ground floor with EKF.

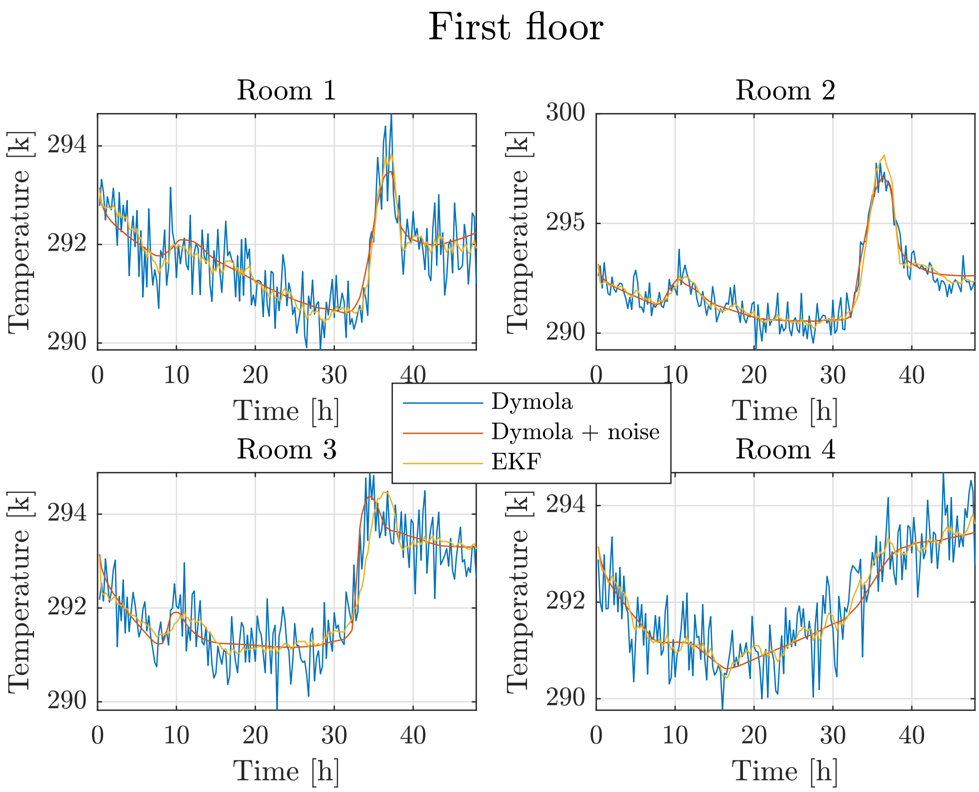

Figure A.2: Test results for First floor with EKF.

Lastly a test for white noise using test based on number of sign changes when

$$\text{Number of sign changes} \in B(N-1, \frac{1}{2}) \tag{A.70}$$

is apporximately equal to

$$\in_{approx} \; N(\frac{N-1}{2}, \frac{N-1}{4}) \tag{A.71}$$

Table A.1 shows the distribution signchanges for each of the 11 EKF estimates compared to the real $T_{r,i}$.

Table A.1: Whiteness test results for estimated room temperatures

Room	Whiteness
0,1	0.48167539
0,2	0.5026178
0,3	0.46073298
0,4	0.5026178
0,5	0.47120419
0,6	0.51308901
0,7	0.5026178
1,1	0.48167539
1,2	0.48167539
1,3	0.46073298
1,4	0.46596859

A.7 Example for EKF and UKF

In this section a simple example of the implementation of an EKF and UKF is presented.

A system with non-linear state equations and measurement equations is used to show the use of both filters:

$$\begin{aligned} x_{k+1} =& f(x_k, u_k) = a \sin(x_k + \varphi_f) + b\, u_k \\ y_k =& h(x_k, u_k) = \sin(c\, x_k + \varphi_h) \end{aligned} \tag{A.72}$$

$$x_{k+1} = f(x_k, u_k) + w_k, \qquad w_k \overset{i.i.d.}{\sim} \mathcal{N}(0, Q_k) \tag{A.73}$$

$$y_k = h(x_k, u_k) + v_k, \qquad v_k \overset{i.i.d.}{\sim} \mathcal{N}(0, R_k) \tag{A.74}$$

and

$$x_0 \sim \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{A.75}$$

assume a, b,, c, φ_f, φ_h, Q, R, x_0, P_0 to be known.

Then the EKF can be calculated by the algorithm presented below.

Initial conditions:

$$x_0 \sim \mathcal{N}(\hat{x}_{0|-1}, P_{0|-1}) \tag{A.76}$$

$$\tag{A.77}$$

Measurement update:

$$\begin{aligned} \hat{y}_{k|k-1} &= h(\hat{x}_{k|k-1}, u_k) = \sin(c\, \hat{x}_{k|k-1} + \varphi_h) \\ \tilde{y}_{k|k-1} &= y_k - \hat{y}_{k|k-1} = \sin(c\, x_k + \varphi_h) - \sin(c\, \hat{x}_{k|k-1} + \varphi_h) \\ C_k &= \left.\frac{\partial h(x,u)}{\partial x^{\mathrm{T}}}\right|_{\hat{x}_{k|k-1}, u_k} = c\, \cos(c\, \hat{x}_{k|k-1} + \varphi_h) \\ K_k &= P_{k|k-1}\, C_k^{\mathrm{T}} \left(C_k\, P_{k|k-1}\, C_k^{\mathrm{T}} + R_k\right)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k\, \tilde{y}_{k|k-1} \\ P_{k|k} &= (I - K_k\, C_k)\; P_{k|k-1}\; (I - K_k\, C_k)^{\mathrm{T}} + K_k\, R_k\, K_k^{\mathrm{T}} \end{aligned} \tag{A.78}$$

Time update:

$$\begin{aligned} \hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, u_k) = a\, \sin(\hat{x}_{k|k} + \varphi_f) + b\, u_k \\ A_k &= \left.\frac{\partial f(x,u)}{\partial x^{\mathrm{T}}}\right|_{\hat{x}_{k|k}, u_k} = a\, \cos(\hat{x}_{k|k} + \varphi_f) \\ P_{k+1|k} &= A_k\, P_{k|k}\, A_k^{\mathrm{T}} + Q_k \end{aligned} \tag{A.79}$$

The UKF can be calculated according to Section A.3, where the unscented transform is used. This gives the following results for the system described in Equation A.72. The UKF filter has been implemented with $\lambda = 2$, $\beta = 0$. The parameters for the system set to be

$a = 1$, $b = 2$, $c = 2$, $\varphi_f = 1$, $\varphi_h = 2$, $Q = 10^2$, $R = 10^2$, $x_0 = 1$, $P_0 = 1$. Figure A.3 shows the test results from the implementation of the UKF and EKF filters.

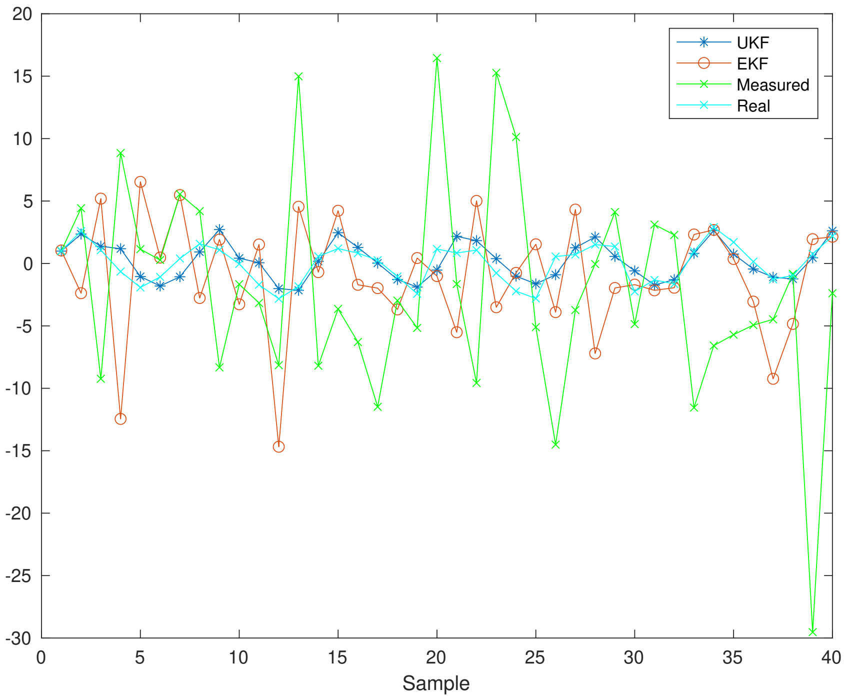

Figure A.3: This figure shows the test results of the EKF and UKF estimates compared to the real system state and the noisy system state.

The root mean square error between the UKF estimate and the real state is 1.0263. The root mean square error between the EKF estimate and the real state is 4.2734.

Therefore it can be concluded that for a highly non-linear system like Equation A.72 an UKF filter might yield better state estimates. As the lumped parameter multi-zone model described in Chapter 3 only contains non-linear parts that are bi-linear an EKF is considered sufficient for state estimates.