



NTNU

Norwegian University of
Science and Technology

HVAC-systems

Modeling, simulation and control of HVAC-systems

Øyvind Alvsvåg

Master of Science in Engineering Cybernetics

Submission date: June 2011

Supervisor: Lars Imsland, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

TTK4900 MASTER THESIS

HVAC - systems

Modeling, simulation and control of HVAC-systems

Øyvind ALVSVÅG



Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

June 20, 2011

Abstract

It is of interest for companies to keep the annual operating cost of their buildings as low as possible. A substantial share of the annual operating costs are due to the large amount of energy needed for heating of the ventilated air and heating of the rooms inside the buildings. Also much of the electrical energy in the world today is created using fossil fuel or charcoal. This has an environmental aspect and the consumers of energy becomes more and more aware of this. Thus reducing the energy used by a buildings HVAC system can save the users for considerable expenditures and also has an environmental aspect.

To find an estimate of the energy consumption a mathematical model representing a building and its HVAC system have been made. This model has been made up of several smaller models representing each component present in the building. These models have then been implemented in `Simulink` and the response of the system has been simulated for different scenarios. From these simulations the energy consumption has been extracted and compared to each other. Thus the amount of energy saved for each scenario has been found. The models include two type of controllers to see whether or not the choice of controller design affects the energy efficiency of the system. These two controller designs are the PID controller and the MPC control scheme. Also a discretized and simplified model of the building to be used together with the MPC controller has been found using system identification. In addition to this a Kalman filter that estimates unknown states and filter out disturbances are included in the MPC control scheme.

The results from the simulations using a PID controller indicates a possible annual saving of substantial amounts. Thus this report shows that the annual energy consumption in a building can be greatly reduced by introducing simple and relatively cheap modifications to the HVAC system. The results from the simulations using the MPC scheme indicates that even more energy can be saved using this advanced control scheme. However, in order to verify this the MPC controller needs to be fine tuned and several more experiments needs to be reviewed.

Preface

This report has been written in corporation with Elektro Bodø A/S and I would like to thank John Olaf Johansen and Ole-Arne Torsvik for help regarding information about Saltstraumen elementary school and its HVAC system. I would also like to thank my supervisor Lars Imslund for his contributions. They have all been of great help to me and without their contributions this report could not have been written.

I would also like to thank my classmates Anders Willersrud, Anne Mai Ersdal, Eirik Viksmo-Slettan, Joakim Haugen, Jon Tore Hafstad, Jonas Magnussen and Tor Aksel Heirung for their inputs during my studies.

My family, especially my mom and dad, has been of great support to me throughout my studies and I am in great debt of gratitude to them.

Finally I would like to thank my fiancée Ane Breivik for her unconditional support and help.

Øyvind Alvsvåg
June 20, 2011

Contents

Abstract	i
Preface	iii
List of Figures	xii
List of Tables	xiii
1 Introduction	1
2 HVAC fundamentals	5
2.1 Basics	5
2.1.1 Ventilation and air conditioning	6
2.1.2 Heating	6
3 Modeling HVAC systems	11
3.1 Short intro to modeling	11
3.1.1 Law of conservation of mass	12
3.1.2 Law of conservation of energy	12
3.2 Modeling approaches to airflow and temperature changes in buildings	14
3.2.1 CFD modeling	14
3.2.2 Multizone network airflow modeling	14
3.3 Modeling a single-room building	17
3.3.1 Modeling the wall	17
3.3.2 Modeling the ceiling	18
3.3.3 Modeling the floor	18
3.3.4 Modeling the windows	19
3.3.5 Modeling the infiltration	19
3.3.6 Modeling how the ventilation affects the room	20
3.3.7 Modeling the radiators	20
3.3.8 Modeling how the radiators affects the room	21

CONTENTS

3.3.9	Modeling the indoor air	21
3.3.10	Connecting the elements of the room	21
3.3.11	Verification of the room model	23
3.4	Connecting adjoining rooms	25
3.4.1	Model redesign	26
3.4.2	Verification of the new design	30
3.4.3	Verification of the 3-room model	30
3.5	Modeling the entire building	32
3.6	Modeling the ventilation system	35
3.6.1	Modeling the fan	35
3.6.2	Modeling the heat recovery unit	38
3.6.3	Modeling the heating coil	41
3.6.4	Modeling the shunt connection	46
3.6.5	Modeling pipes and ducts	50
3.6.6	Connecting each element of the ventilation system	59
3.6.7	Extending the model to enable varying air volume	59
4	System identification	65
4.1	Input design	67
4.1.1	Persistently exciting	67
4.1.2	Crest factor	68
4.1.3	Choosing input signals	69
4.2	Selecting a model structure	74
4.2.1	ARX-model	76
4.2.2	ARMAX-model	78
4.2.3	OE-model	80
4.2.4	Box-Jenkins model	82
4.2.5	Process models	83
4.2.6	State-space models	84
4.2.7	Model structure selection	85
4.3	System identification procedure and results	88
4.3.1	Importing and preprocessing the data set	88
4.3.2	Model identification	88
4.3.3	Results and model verification	92
5	Automatic control	99
5.1	Control strategy	100
5.2	PID controller	101
5.2.1	Room temperature	104
5.2.2	Ventilation temperature	108
5.3	Model predictive control	112

5.3.1	Receding horizon	113
5.3.2	State space representation	114
5.3.3	Discretization of the state space	116
5.3.4	Observer	117
5.3.5	MPC, theory and implementation	124
6	Energy consumption	143
6.1	Calculation of energy consumption	143
6.1.1	Energy usage in the ventilation system	143
6.1.2	Energy usage in the radiator system	146
6.2	Energy savings	147
6.2.1	Energy savings in the ventilation system	148
6.2.2	Energy savings in the radiator circuit using a PID controller	156
6.2.3	Energy savings in the radiator circuit using MPC	159
7	Discussion	163
8	Conclusion and further works	171
	Bibliography	177
	Appendices	181
A	Calculation of electrical components in the building model	181
A.1	Calculation of thermal resistance	181
A.1.1	Outer thermal transmittance of external walls and ceiling .	181
A.1.2	Inner thermal transmittance of external walls and ceiling .	182
A.1.3	Thermal transmittance of internal walls	183
A.1.4	Thermal transmittance of the floor	183
A.1.5	Convective thermal resistance wall to air and ceiling to air	184
A.1.6	Thermal transmittance of the windows	184
A.1.7	Convective thermal resistance window to air	185
A.1.8	Thermal radiative resistance window to wall	185
A.1.9	Convective thermal resistance in the air due to ventilation	185
A.1.10	Convective thermal resistance in the air due to infiltration	186
A.1.11	Thermal radiative resistance radiator to wall	187
A.1.12	Convective thermal resistance radiator to wall	187
A.2	Calculation of heat capacities	188
A.2.1	Heat capacity of the air inside a room	188
A.2.2	Heat capacity of the wall	188
B	System identification	191

CONTENTS

B.1	Model identification results	191
C	Model predictive control	207
C.1	State space representation of the rooms	207
C.1.1	Room k133	208
C.1.2	Room k134	209
C.1.3	Room k135	210
C.1.4	Room k136	211
C.1.5	Room k143	212
C.1.6	Room k164	213
D	M-files used in the model	215
D.1	Initialization	216
D.2	Constants	218
D.3	Room values	221
D.4	Input parameters	232
D.5	Preprocessing	233
D.6	Kalman filter - embedded Matlab function	234
D.7	MPC preparations	235
D.8	Creation of MPC matrices	241
D.9	MPC function without constraints	243
D.10	MPC function with constraints	244
D.11	Feed forward	246
D.12	Energy consumption radiators	247
D.13	Energy consumption ventilation	250

List of Figures

2.1	Ventilation system, Saltstraumen elementary school	7
2.2	Heating system, Saltstraumen elementary school	9
2.3	Exploded view of the radiator connections	10
3.1	Conservation of mass.	12
3.2	Conservation of energy.	12
3.3	Electrical analogy - wall	18
3.4	Electrical analogy - floor	19
3.5	Electrical analogy - window	19
3.6	Electrical analogy - infiltration	20
3.7	Electrical analogy - room	22
3.8	Temperature development of the passive single-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C	24
3.9	Temperature development of the passive single-room model - duration: 8 days, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C	24
3.10	Temperature development of the passive single-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 10°C	25
3.11	Temperature development of the passive single-room model - duration: 8 day, outdoor temperature: varying, initial indoor temperature: 20°C, initial wall temperature: 20°C	26
3.12	Electrical analog room and walls; new design	27
3.13	Electrical analog room; new design	28
3.14	Electrical analog wall; new design	28
3.15	Electrical analog wall with windows; new design	29
3.16	3 rooms connected using the new design	29

LIST OF FIGURES

3.17	Temperature development of the new passive single-room model - duration: 8 days, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C	30
3.18	Temperature development of the new passive single-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C	31
3.19	Temperature development of the new passive 3-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C	32
3.20	Overview of Saltstraumen elementary school, main building	33
3.21	Thermal model of the building that this report is based on	34
3.22	Schematic view of the fan	35
3.23	Block diagram of the fan model	38
3.24	Schematic view of the heat recovery unit	38
3.25	Block diagram of the heat recovery unit model	41
3.26	Schematic view of the heating coil	42
3.27	Block diagram of the heating coil model	46
3.28	Schematic view of the shunt connection	46
3.29	Block diagram of the shunt connection model	51
3.30	Block diagram of the pipe model	59
3.31	Block diagram of the ventilation system of Saltstraumen elementary school	60
3.32	Simulink model of the heating coil with variable air flow.	63
3.33	Temperature development of the ventilation, water into the heating coil and water out of the heating coil using the heating coil with variable airflow. Also the control signal can be seen.	64
4.1	System identification - effect from adjoining rooms	71
4.2	System identification - model of the radiator excitation signal	72
4.3	System identification - model of the ventilation excitation signal	72
4.4	System identification - model of the outdoor excitation signal	73
4.5	System identification - model of the adjoining room excitation signal	74
4.6	System identification - excitation signals	75
4.7	ARX model structure	77
4.8	ARMAX model structure	79
4.9	OE model structure	82
4.10	Box-Jenkins model structure	83
4.11	Structure of the process model	84
4.12	Results from the identification procedure using different model types.	87
4.13	System identification toolbox GUI and the import data window.	89
4.14	System identification toolbox GUI, process models.	91

LIST OF FIGURES

4.15 Simple RC-circuit. Source: Wikipedia	93
4.16 System identification results, room k133 - radiator.	96
4.17 System identification results, room k133 - outdoor temperature.	97
4.18 System identification results, room k133 - ventilation.	98
5.1 The classic feedback loop	99
5.2 Feedforward	100
5.3 PID controller	102
5.4 PID controller with saturation	103
5.5 PID controller with saturation and anti wind-up	104
5.6 Model of the room with radiator model and controller	106
5.7 Exploded view of the radiator and controller block in Figure 5.6	106
5.8 Fixed room temperature with controller: Set temperature: 18°C. Disturbance from a varying outdoor temperature.	108
5.9 Varying room temperature with controller: Set temperature switching between 21°C and 18°C. Disturbance from a varying outdoor temperature.	109
5.10 Fixed ventilation temperature with controller: Set temperature: 18°C. Disturbance from a varying outdoor temperature.	111
5.11 Varying ventilation air temperature with controller: Set temperature switching between 18°C and 16°C. Disturbance from a varying outdoor temperature.	112
5.12 Comparison of the freq. response of the original model and the approximate model of room k133 found using bode plot	115
5.13 Luenberger observer (Magnussen 2010)	120
5.14 Kalman filter loop (Ersdal 2011)	123
5.15 MPC control scheme without feed-forward and constraints	129
5.16 MPC control scheme with feed forward	132
5.17 MPC control scheme with feed forward and constraints	136
5.18 The internal structure of the block labeled MPC in Figure 5.17	137
5.19 Fixed room temperature and estimated room temperature with MPC controller: Set temperature: 21°C. Disturbance from a varying outdoor temperature.	140
5.20 Varying room temperature and estimated room temperature with MPC controller. Daytime temperature: 21°C. Nighttime temperature: 18°C. Disturbance from a varying outdoor temperature. $H_p = H_u = 35$	141
5.21 Varying room temperature and estimated room temperature with MPC controller. Daytime temperature: 21°C. Nighttime temperature: 18°C. Disturbance from a varying outdoor temperature. $H_p = H_u = 100$	142

LIST OF FIGURES

6.1	Outdoor temperature development	148
6.2	Temperature development of air sent into the heating coil	149
6.3	Energy consumption with ventilation temperature of 18°C and 16°C	150
6.4	Energy consumption with ventilation temperature at 18°C for half of the day and 16°C the rest of the day.	151
6.5	Energy consumption with air flow at $0.857m^3/s$ and $0.7713m^3/s$.	152
6.6	Air temperature after the HRU compared with the temperature before the HRU.	154
6.7	Energy consumption with and without HRU.	154
6.8	Difference in energy consumption for the buildings radiators with a room temperature of 21°C and a room temperature of 18°C. . .	157
6.9	Energy consumption of the buildings radiators with varying room temperature.	158
6.10	Difference in energy consumption for room k133 using different controllers. Fixed set temperature.	160
6.11	A close-up of the difference in energy consumption for room k133 using different controllers. Fixed set temperature.	161
6.12	Difference in energy consumption for room k133 using different controllers. Varying set temperature.	162
B.1	System identification results, room k134 - radiator.	192
B.2	System identification results, room k134 - outdoor temperature. .	193
B.3	System identification results, room k134 - ventilation.	194
B.4	System identification results, room k135 - radiator.	195
B.5	System identification results, room k135 - outdoor temperature. .	196
B.6	System identification results, room k135 - ventilation.	197
B.7	System identification results, room k136 - radiator.	198
B.8	System identification results, room k136 - outdoor temperature. .	199
B.9	System identification results, room k136 - ventilation.	200
B.10	System identification results, room k143 - radiator.	201
B.11	System identification results, room k135 - outdoor temperature. .	202
B.12	System identification results, room k143 - ventilation.	203
B.13	System identification results, room k164 - radiator.	204
B.14	System identification results, room k164 - outdoor temperature. .	205
B.15	System identification results, room k164 - ventilation.	206

List of Tables

3.1	Electrical Analogy of thermal flow	17
3.2	Symbols used in the fan model	36
3.3	Symbols used in the heat recovery unit model	39
3.4	Symbols used in the heating coil model	42
3.5	Symbols used in the shunt connection model	47
3.6	Symbols used in the pipe model	53
4.1	Inputs, outputs and measured disturbances of the building	66
4.2	Excitation signals - pros and cons (see Ljung (2006) for further details)	70
4.3	Model structures - pros and cons (see Ljung (2006) for further details)	86
4.4	Transfer functions from the system identification process.	95
5.1	Inputs, outputs and measured disturbances of the building	101
6.1	Energy consumption of components in the ventilation system	144
6.2	Energy consumption of components in the radiator circuit	146
8.1	Energy consumption, ventilation	172
8.2	Energy consumption, radiator circuit	172

LIST OF TABLES

1

Introduction

These days people and companies becomes more and more aware of their energy consumptions. There are many reasons for this, where the most obvious may be the focus that has been on global warming lately. With the introduction of the non optional energy markings of buildings in Norway this focus has increased even more. Especially the amount of energy that is used in ventilation and heating of the building has been brought into focus. The reason for this is that ventilation and heating is one of the most energy consuming parts of the building. Thus reducing this consumption would have an economical aspect as well as the fact that the building would receive a better score when it is energy marked.

Many of todays HVAC systems in aged buildings has never been upgraded. Economical reasons might be one of the causes for this. But the fact that many people simply are unaware of the possible economical benefits that follows from an upgraded HVAC system may be the main reason. It is therefore important to enlighten those responsible for the building of the possible economical benefits of simple and inexpensive changes to the system. This is what this report tries to emphasize.

It should be noted that this report is a continuation of the report Alvsvåg (2010). Hence several of the chapters in this report are similar to the one found in Alvsvåg (2010), but with additional comments and extensions. Also since another building is used in this report the results are different from the results found in (Alvsvåg 2010).

In order to be able to make an assessment of the possible economical saving, the system itself including every component must be mathematically modeled. A mathematical model of the system makes it possible to incorporate the system into dedicated programs such as `Simulink`. `Simulink` can then run simulations

1. INTRODUCTION

of the behavior of the system over time. Every modification of the system can be simulated and then the energy consumption for every simulation can be compared. Thus the possible energy saving is obtained.

The following is a short description of the content of each chapter in this report. Chapter 2 gives the reader an introduction to HVAC systems. It describes the specific HVAC system used in this report, including the function of each component in this system. Thus the reader is introduced to HVAC systems by learning how a specific HVAC system works. Most of today's HVAC systems has a similar structure. This is the same description as given in Alvsvåg (2010).

In Chapter 3 the entire HVAC system including the building where the HVAC system is installed is mathematically modeled. First the temperature development of the building is modeled. This is done by modeling a single room. Then this model is transformed to make it easier to connect adjoining rooms. Several rooms are connected to this new model and it is verified that the model behaves naturally. Then finally every room in the building of interest is modeled and connected to each other to make up the entire building. Second the ventilation system and each of its components are modeled. The modeling of the ventilation system from Alvsvåg (2010) has been used but it has been extended with the modeling of ducts and pipes. Third and finally the room model and the ventilation model is connected to each other and the outdoor environment.

In Chapter 5 the traditional PID controller is first implemented to control the temperature of the ventilation and the temperature of the room. Then an MPC controller is implemented to control the temperature of the rooms. In order for the latter control scheme to function a discrete model of the system is needed. The model found in Chapter 3 cannot be discretized due to the involvement of `Simscape` components. Thus Chapter 4 gives an introduction to system identification and uses this to find a simplified model of the system that also can be discretized.

In a HVAC system there are two temperatures that are essential to control, namely the room temperature and the temperature of the ventilated air. Chapter 5 gives an introduction to automatic control, the PID controller and the more advanced MPC controller. The PID controllers in this Chapter are similar to the one found in Alvsvåg (2010), but since a different building and a ventilation system extended with ducts and pipes has been used the results are slightly different. The PID controller is used to control the temperature of the rooms and the ventilated air while the MPC controller is used only to control the room temperatures.

Chapter 6 states how to calculate the energy consumption of the different compo-

nents used in the HVAC system. This was also done in Alvsvåg (2010). Then the HVAC system modeled in Chapter 3 with the different automatic control schemes from Chapter 5 is simulated for different scenarios. The energy consumption for each of those scenarios are compared and the possible saving is calculated. The results of these simulations are shortly discussed.

In Chapter 7 the results of this paper are discussed more thoroughly and Chapter 8 contains a brief summary and conclusion. In addition to this Chapter 8 also contains a section where possible further work is mentioned.

1. INTRODUCTION

2

HVAC fundamentals

The objective of this chapter is to give the reader a short introduction to HVAC systems with the intent to make the following chapters easier to comprehend.

2.1 Basics

HVAC is an abbreviation for heating, ventilation and air conditioning. It is defined as the simultaneous control of temperature, humidity, radiant energy, air motion and air quality within a space for the purpose of satisfying the requirements of comfort or a process (Haines & Myers 2010). Although pressure distribution within the controlled area is not included in the definition of HVAC systems it should not be neglected as it is important to sustain an acceptable level of comfort (Haines & Myers 2010). It should be noted that temperature control is the focus of this report and hence the other mentioned factors will not be taken into account. To be able to control the above mentioned variables simultaneously many parts and different equipment must be connected and work together as one unit. The HVAC system of Saltstraumen elementary school, which this report is based on, consists of two main parts. First, the ventilation and air conditioning system; second, the heating system. This is how most of today's HVAC systems are built. The following section and Section 2.1.2 gives the reader a walk-through of this and similar systems functional description.

2. HVAC FUNDAMENTALS

2.1.1 Ventilation and air conditioning

The ventilation and air conditioning part of the HVAC system installed at Saltstraumen elementary school is shown in Figure 2.1. The following is a short functional description to give the reader an overall insight of the ventilation system used in this report. The system will be modeled and simulated in Chapter 3 using `Matlab`.

First the air is drawn from a source outside of the controlled area. Usually fresh air from the outdoor is used to this purpose and a fan is used to draw the air into the controlled area. Before the air goes through the fan it passes through a filter and a heat recovery unit. The filter makes sure that the ventilation air in the building is not contaminated. While the heat recovery unit utilizes heat from the return air to heat up the fresh air¹. It is possible to turn this unit off if the outdoor temperature should rise above a certain value. In Saltstraumen elementary school this happens at an outdoor temperature of 18°C. Then the fresh air passes through the intake fan and through a heating coil. This unit makes sure that the temperature of the air to the controlled area has a predefined temperature. This is done by letting hot water flow through the heating coil and heat up the air. The temperature of the water sent into the heating coil is controlled by regulating the opening of a shunt valve. Finally the heated fresh air continues to the controlled area where it is mixed with the rest of the air already in the room.

From the controlled area used air is drawn towards the air outlet by another fan. The inlet and outlet fans are always working at the same speed to ensure that the amount of air into the room is equal to the amount of air out of the room at all times. This ensures that mass balance of the air in the building is maintained. After the outlet fan the used air flows through a filter before entering the heat recovery unit where it is used to heat up the fresh air. Finally the used air exits the controlled area.

2.1.2 Heating

The heating part of the HVAC system installed at Saltstraumen elementary school is shown in Figure 2.2. The following is a short functional description to give the reader an overall insight of this system. As for the previous section, describing the ventilation, the heating system will also be modeled and simulated in Chapter 3 using `Matlab`.

¹The heat recovery unit is only used in regions where the mean outdoor temperature is low.

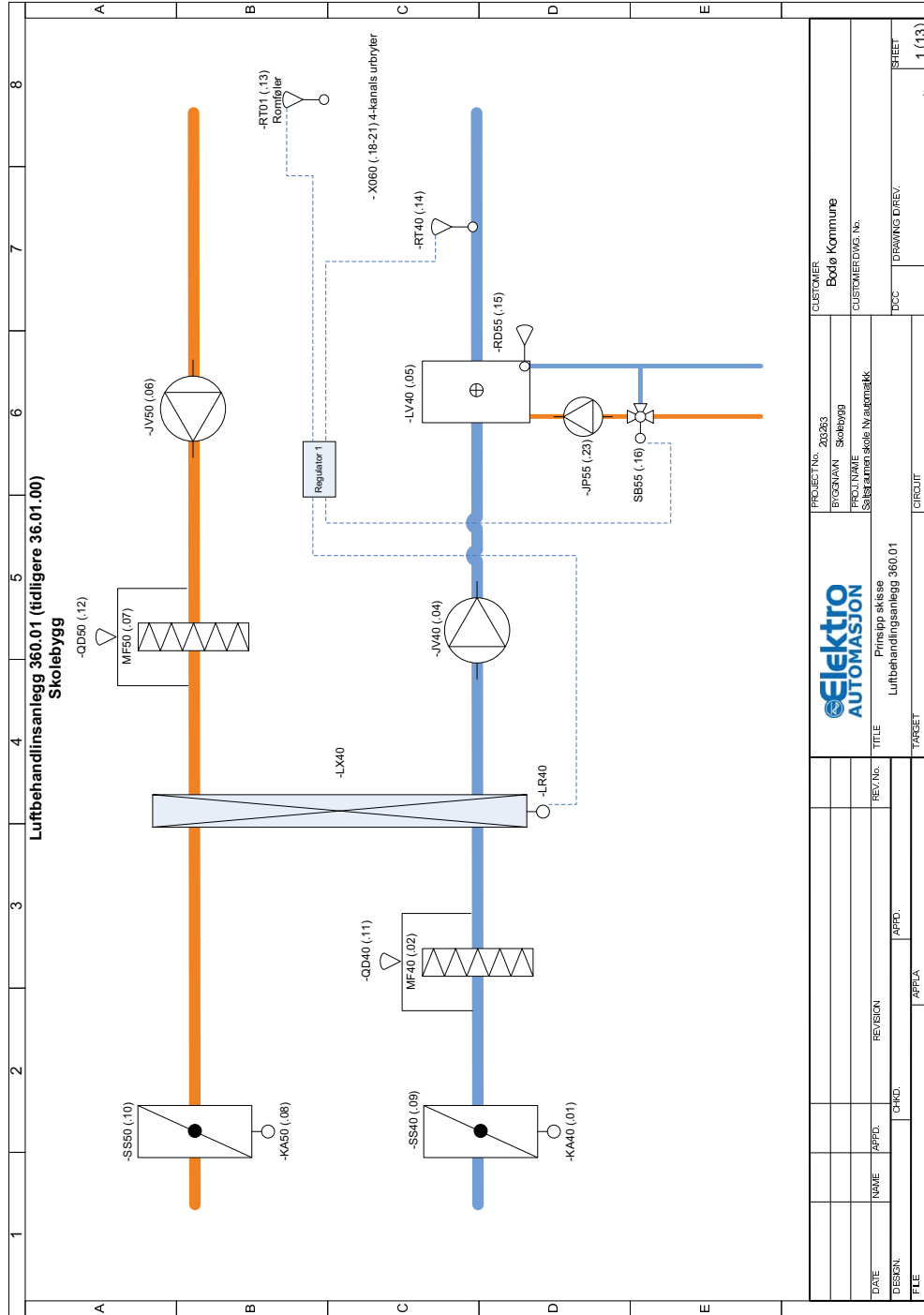


Figure 2.1: Ventilation system, Saltstraumen elementary school

2. HVAC FUNDAMENTALS

As can be seen from Figure 2.2 there are two radiator circuits in the main building of Saltstraumen elementary school. Only one of the radiator circuits will be explained in this section as they are identical in functionality. In the radiator circuit (to the left of the shunt valve in Figure 2.2) there is a constant flow of water. This flow is determined by the speed of an electrical pump that circulates the water. A shunt valve regulated by a controller designates the amount of hot water from the reservoir (“fra varmestokk”) and the amount of water that is to be recycled. The sum of these two should always add up to 100% of the fixed flow rate. The calculation of the valve position is done by the regulator which determines the valve opening based on the outdoor temperature and the water temperature inside the radiator circuit. The outdoor temperature is taken into account to compensate for cold or hot days. This means that the water sent to the radiators will be warmer when the outdoor temperature is low and vice versa.

Figure 2.2 indicates that the water is running through each radiator before returning. This is a simplification in the drawing and the actual connection of the radiators can be seen in Figure 2.3. From this figure it is clear that each radiator has its own valve controlling the flow of water through the radiator. This valve is regulated by a controller. The controller ensures that hot water is brought into the radiator, hence heating it up, when the room temperature is too low. Also, when the room temperature is higher than it should be, the controller makes sure that the valve closes, hence making sure the radiator does not receive more energy (e.g. hot water). When the room temperature is at its set point the controller ensures it stays there by compensating for disturbances causing heat loss in the room. It is also seen from this figure that the water available to each radiator is a fraction ($\frac{1}{n}$) of the total water flowing in the circuit. n is the number of radiators in the circuit. The reason for this is to ensure that each radiator has an equal amount of water available. Without this restriction the first radiator in the circuit tends to use most of the hot water, hence leaving the other radiators almost without any effect.

It should be noted that modern HVAC systems might contain additional features. Since these will not be introduced in this report they will only be briefly mentioned here. Some of today's HVAC systems has the ability to decide where to distribute the air and also the amount of air distributed to a particular room. This is called variable air volume (VAV). Some also distributes air according to the CO_2 values of the rooms. It is also popular for new buildings to use a central control and monitoring system to control and monitor the buildings ventilation, heating, lighting and other technical features installed in the building. These systems are also used to log events like alarms or the entire systems behavior over time. Another possible and popular feature for ventilation systems is to recycle the air inside the building when the building is not in use.

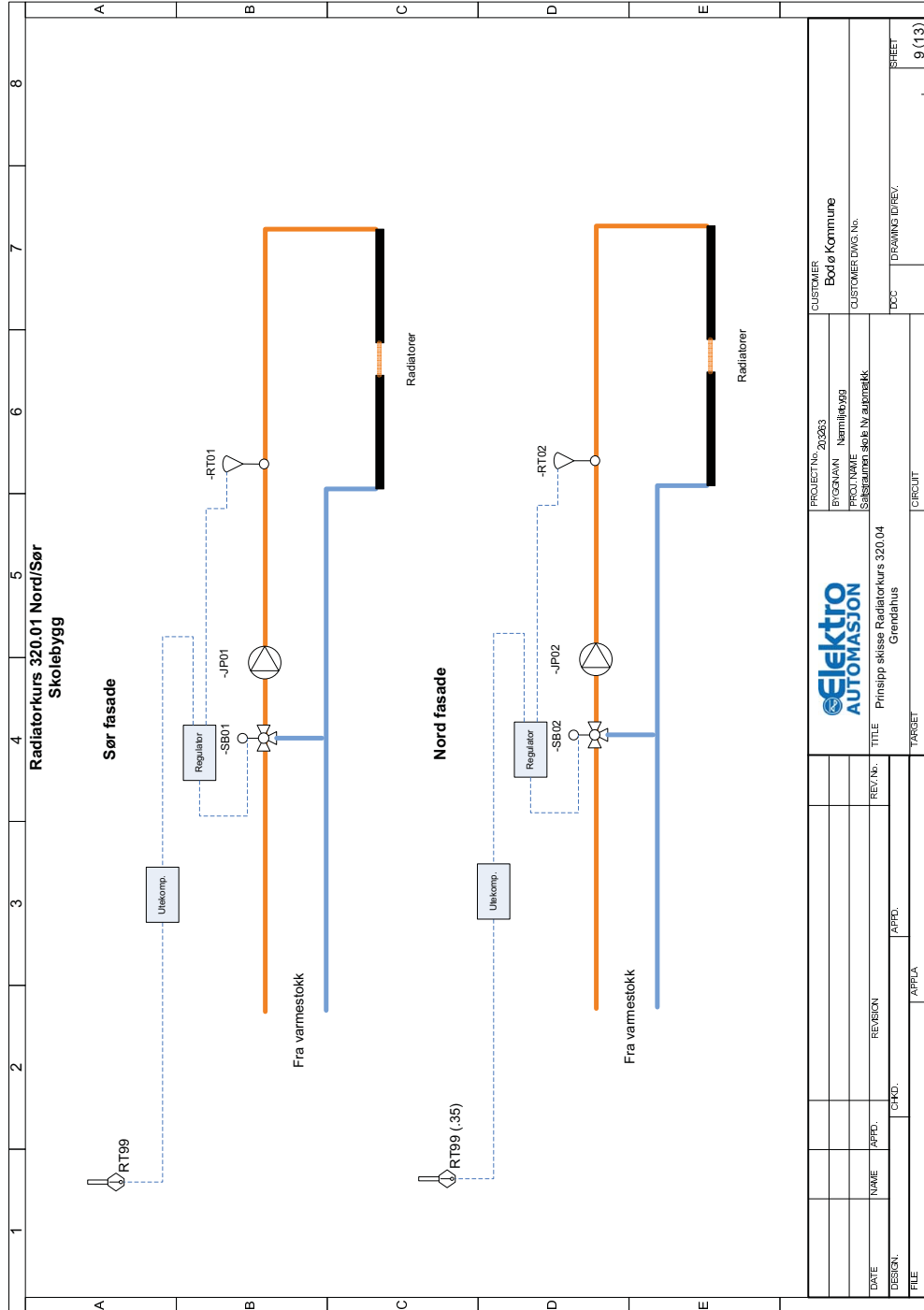


Figure 2.2: Heating system, Saltstraumen elementary school

2. HVAC FUNDAMENTALS

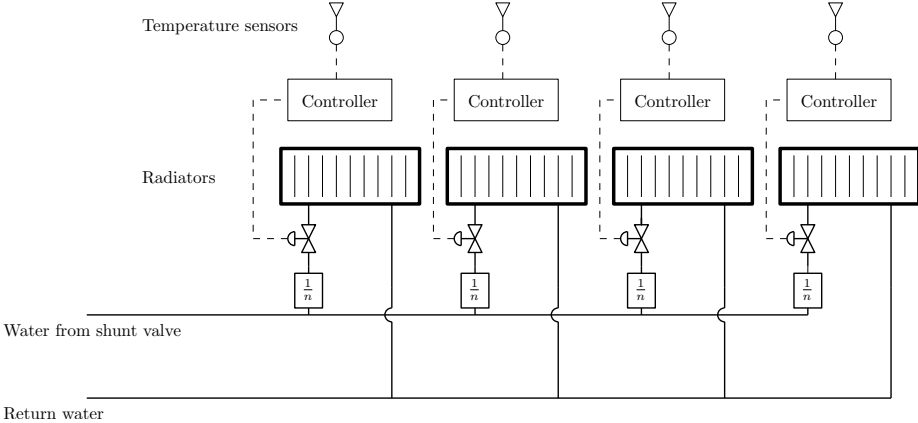


Figure 2.3: Exploded view of the radiator connections

3

Modeling HVAC systems

3.1 Short intro to modeling

Modeling is the generation of mathematical equations used to describe the response of dynamic processes (Balchen et al. 2003). Usually this is done by describing the system using well known balance laws (e.g. the law of conservation of mass). Then these equations are Laplace transformed so that the system can be described in the s-plane as a transfer function. A transfer function is a function that describes how an input signal transfers through a device (e.g. a valve or a pipe) and becomes the output signal. More about transfer functions can be read in Balchen et al. (2003). When a physical system has been modeled it can be simulated using a designated program, e.g. `Matlab`. Simulating a model is an easy and fast method to show how the model responds to inputs. For instance the response of a process excited by a step response may be simulated on a computer in a matter of seconds, while in real time it might take several hours or days before the systems settles. Thus, one of the most important reasons for using mathematical models is that it has the potential of saving a great deal of money. By running tests on a computer it is easier and faster to correct undetected errors before the system is implemented into the real system. It is also safer to run tests on the model rather than on the actual system (Balchen et al. 2003). Another important reason why mathematical modeling is important and has become as popular is that modern controllers can utilize the information from the models. This information can be used to filter away the disturbance from the control signals in order to reduce wear and tear (e.g. Kalman filter). Or it can be used to calculate estimates of future outputs in order to generate more optimal control signals (i.e. MPC) than the traditional PID-controller. Also, to be able to control certain processes, a model based on the actual physical system is necessary

3. MODELING HVAC SYSTEMS

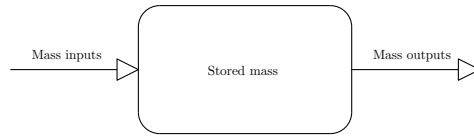


Figure 3.1: Conservation of mass.

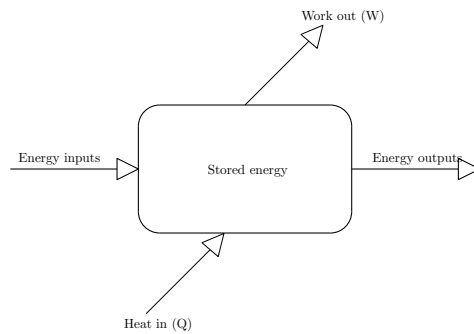


Figure 3.2: Conservation of energy.

(Balchen et al. 2003).

When modeling HVAC systems the following two laws of balance are imperative.

3.1.1 Law of conservation of mass

The law of conservation of mass states that the mass of a closed system will remain constant over time; the mass can neither be created nor be destroyed (Haines & Myers 2010). This means that mass stored in a fixed volume is only altered due to mass inputs and mass outputs. This is illustrated in Figure 3.1 and is often referred to as mass balance.

3.1.2 Law of conservation of energy

The law of conservation of energy states that the energy of a closed system will remain constant over time. As for the mass, energy can neither be created nor destroyed (Haines & Myers 2010). For that reason energy stored in a fixed volume is only altered due to energy inputs and energy outputs (Hauge & Støvneng 2009). It can however change form, e.g. kinetic energy can become thermal energy due to friction. This is illustrated in Figure 3.2.

First law of thermodynamics

The first law of thermodynamics states that *the internal energy of an isolated system is constant* (Hauge & Støvneng 2009). This is essentially the same as the law of conservation of energy.

Second law of thermodynamics

The second law of thermodynamics states that *heat can flow from a mass with higher temperature to a mass with lower temperature, but never the other way around* (Hauge & Støvneng 2009). In buildings heat loss is caused by the process of heat transfer due to conduction, convection and radiation. Heat conduction is the flow of heat through conductors (e.g a wall) when subjected to the influence of temperature differences (Lienhard IV & Lienhard V 2008). This is described by Fourier's equation (Hauge & Støvneng 2009)

$$\rho C_p \frac{dt}{dT} = k \left[\frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} + \frac{\partial^2 t}{\partial z^2} \right] \quad (3.1)$$

where

T = time (s)

t = temperature ($^{\circ}C$)

k = thermal conductivity ($\frac{W}{m \cdot K}$)

ρ = density ($\frac{kg}{m^3}$)

C_p = specific heat capacity ($\frac{kJ}{kg \cdot K}$)

x, y, z = coordinates in space or dimensions (m)

Heat convection, heat transfer due to movement in the air (Hauge & Støvneng 2009), can be neglected in material densities above $20kg/m^3$ (Rockwool (n.d.) n.d.). Hence the modeling described in this chapter is done with the assumption that there is no heat convection inside the structures of a building (e.g. a wall). Though convection exist where the surface of the construction is in contact with a substance of density below $20kg/m^3$ (e.g. air). Radiation contributes to the total energy or heat transference because of the fact that when radiation waves hits an object some of the radiation is absorbed by the object (Hauge & Støvneng 2009). The rest of the radiation waves or energy is reflected or travels through the object.

3.2 Modeling approaches to airflow and temperature changes in buildings

Exact modeling of a room is a complicated and time consuming task and it requires extended knowledge of every physical part of the room. Simplifications is therefore widely used in order to arrive at a simplified but precise enough model of the room without being too time-consuming. Two common modeling approaches for calculating indoor airflow and contaminant concentrations in buildings are computational fluid dynamics (CFD) and multi-zone network airflow modeling (ASHRAE 2009).

3.2.1 CFD modeling

CFD modeling is commonly used for a macroscopic view of HVAC-components, i.e. modeling of a single room in a building. This modeling approach solves Navier-Stokes equations (ASHRAE 2009) and resembles the physical reality very closely. However this kind of modeling requires a significant amount of time and is therefore not a good approach when modeling a whole building with many rooms and hallways.

3.2.2 Multizone network airflow modeling

This modeling approach is less time consuming and it makes whole-building modeling possible over time as long as a year, but it is far less detailed (e.g. no internal-room airflow details) (ASHRAE 2009).

In this paper a building with many rooms of different size, shape and disturbances is to be modeled and simulated. This would make it very time consuming to use CFD as a modeling approach, hence modeling of the building in this paper will be done using multi-zone network airflow modeling. It is assumed a constant mass in each room and the flow which are to be considered is heat flow into and out of each of the rooms by ventilation, infiltration, heat transfer through materials and radiation. The following section introduces a useful analogy between electrical and thermal conduction.

3.2 Modeling approaches to airflow and temperature changes in buildings

Analogy between thermal and electrical conduction

When Georg Ohm developed his law for electrical circuits he was inspired by Fourier's law of heat conduction (Narasimhan 1999). Ohm's law predicts the flow of electrical charge (i.e. current) in electrical conductors when subjected to the influence of voltage differences (Nilsson & Riedel 2005). This is equivalent to Fourier's law which predicts the flow of heat in heat conductors when subjected to the influence of temperature differences (Lienhard IV & Lienhard V 2008). When modeling heat conduction in a building it is possible to use this analogy and thereby use electrical circuits to model the thermal flow through the walls, windows, ducts, and doors of buildings. The remainder of this section proves this.

Ohm's law is expressed as

$$I = \frac{\Delta V}{R_e} \quad (3.2)$$

where I is the current flowing through an element, ΔV is the voltage across the element and R_e is the electrical resistance across the element. According to Haines & Myers (2010) heat conduction (Fourier's law) can be written as

$$q = \frac{\Delta T}{R_t} \quad (3.3)$$

where q is the rate of heat conduction, ΔT is the temperature difference and R_t is the thermal resistance. A closer look at Equation 3.2 and 3.3 reveals that both Ohm's law and Fourier's law states how energy is transferred through matter by conduction. Electrical current, I , is defined as the rate of charge flow (Nilsson & Riedel 2005), this is equivalent to Fourier's flow, q , the rate of heat conduction. Ohm describes the voltage, ΔV over an element as the energy per unit charge created by separation (Nilsson & Riedel 2005). This is equivalent to Fourier's temperature difference, ΔT , which also is a measurement of the energy difference over an element (e.g. a wall). Finally the resistance in Ohm's law, R_e , is defined as the capacity of materials to impede the flow of electrical charge, current (Nilsson & Riedel 2005). Similarly Fourier's resistance, R_t can be defined as

$$R_t = \frac{1}{U \cdot A} \quad (3.4)$$

3. MODELING HVAC SYSTEMS

where U is the overall heat transfer coefficient and A is the area, e.g. area of a wall (Novakovic 1995, Haines & Myers 2010). U is given as the reciprocal of R , which is a measure of thermal resistance in the given construction given as

$$R = \frac{d}{\lambda} \quad (3.5)$$

where λ is the heat transfer coefficient and d is the thickness of the element, e.g. the wall (Thue 2006). R_t can now be written as

$$R_t = \frac{R}{A} \quad (3.6)$$

$$= \frac{d}{\lambda \cdot A} \quad (3.7)$$

This shows that both the resistance from Ohm's law and the resistance from Fourier's law can be defined as the capacity of materials to impede flow of energy through the material. Thus every elements of Fourier's law has been described by a similar electrical element found in Ohm's law.

Both solids and gases, like every other matters in the universe, consist of a mass and hence it has the ability to store energy (e.g. heat), this is also the characteristic of the capacitor. Thus, this electrical element can be used to simulate the accumulation of energy stored in the air, walls, roofs and ceilings.

When modeling heat conduction in buildings the electrical analogy just described is a popular approach. It makes the modeling both intuitive and easy to understand. Table 3.1 is a summary of the electrical analogy defined in this section.

The modeling and simulation in this thesis is done using **Matlab** and **Simulink** together with the **Simscape** extension which provides electrical elements and structure. The use of this extension makes it possible to model an entire building using the electrical analogy from the previous section directly. Hence there is no need to transform each resistor and capacitor into a transfer function as would be needed if using **Simulink** without the **Simscape** extension. Thus by using **Simscape** the models of the room elements found in the next sections can be implemented directly. It is also easier to physically understand the model of a building when it consists of a network of electrical elements instead of a network of transfer functions. The disadvantage is that it is not possible to obtain discrete models from **Simscape**. To be able to obtain this in **Matlab** the model needs to be defined by transfer functions. The MPC scheme introduced later in this report depends on

3.3 Modeling a single-room building

Table 3.1: Electrical Analogy of thermal flow

Quantity	Thermal System	Electrical System
Potential	Temperature difference (ΔT)	Voltage (U)
Flow	Heat flux (q)	Current (I)
Resistance	$\frac{\text{Potential}}{\text{Flow}}$ ($R = \frac{\Delta T}{q}$)	Resistance ($R = \frac{U}{I}$)
Capacitance	$\frac{\text{Storage}}{\text{Potential}}$ ($C = \frac{\int q dt}{\Delta T}$)	Capacitor ($C = \frac{\int I dt}{V}$)

a discrete representation of the system. —endre setning i parantes—(In order to find such a representation system identification will be used).

3.3 Modeling a single-room building

In this section a square building with a single room is modeled to give the reader a basic understanding of the electrical equivalent modeling approach. When modeling the temperature changes in a room there are numerous variables to consider. Heat conduction through walls, floor and ceiling, heat storage in materials, solar radiation, heat transferred through the ventilation and heat contribution from radiators are the most significant.

When modeling a room or a building the ceiling and the roof can be modeled in the exact same manner as the walls, in fact all the walls, ceilings and roofs are considered as one large wall in Novakovic (1995). Using this approach the model uses fewer electrical components to make up the whole room. The drawback is that it is difficult to connect adjoining rooms (see Section 3.4). Thus the model of a room in this report will distinguish between each wall, the ceiling and the roof. Calculation of the resistors and capacitors used in the modeling are explained in Appendix A.

3.3.1 Modeling the wall

Using the electrical analogy introduced in Section 3.2.2, the wall in a building (excluding the windows) can be thought of as a resistance that tries to withstand the transfer of heat from the outdoor to the inside of the wall. This resistance is then connected with a capacitor that presents the heat storage in the wall, and another resistor which represent the heat resistance from the inside of the wall to the indoor air near the wall. Finally, a third resistor is connected to the electrical

3. MODELING HVAC SYSTEMS

network, this resistor represents heat convection between air close to the wall and the rest of the air. Since temperature is equivalent to voltage, the capacitance is set with an initial voltage equal to the initial wall temperature. When these electrical elements are connected as described (see Figure 3.3), the variation in voltage resembles the variation in temperature across the wall.

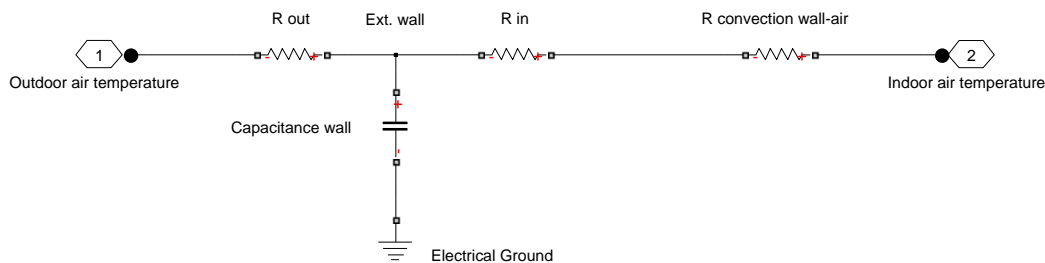


Figure 3.3: Electrical analogy - wall

3.3.2 Modeling the ceiling

As mentioned earlier, modeling the ceiling of a room can be done in a similar manner as for the walls. Thus there are essentially no difference between the walls and the ceiling other than the resistors and the capacitor values. Note that if the building consist of more than one floor the ceiling might not be connected to the outdoor temperature but to the temperature of the floor above. The electrical assembly of the ceiling is thus equal to the electrical assembly of the wall shown in Figure 3.3.

3.3.3 Modeling the floor

Modeling the floor is also similar to the walls. The difference is that the floor is not in contact with the outdoor temperature but with the temperature of the ground or the floor below. In this report the floor model has been simplified to a resistor, representing the thermal resistance of the floor, in series with a capacitor, representing the thermal mass of the floor, which in turn is connected to electrical ground (i.e. 0 °C). This is possible since there is only one floor in the building. Otherwise the floor needs to be modeled in the same way as the walls and the ceiling.

3.3 Modeling a single-room building

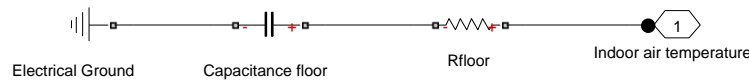


Figure 3.4: Electrical analogy - floor

3.3.4 Modeling the windows

When modeling a window it is assumed that the window itself is massless, that is, it does not accumulate any heat. Because of this simplification the electrical analogy of the window does not include any capacitors. Hence the window model consists of two resistors; one that represents the resistance of heat conduction through the window, and one that represents the heat convection between the indoor air close to the window and the rest of the air in the room (Novakovic 1995). The electrical heat transfer model of a window is shown in Figure 3.5.



Figure 3.5: Electrical analogy - window

3.3.5 Modeling the infiltration

There are no buildings that are completely air tight. Windows leak air around the frames and gaskets even when they are fixed in place. This is also the case with doors. Human traffic increases the air movement and thus also the infiltration of air in and out of a room or the entire building. Although walls seems to be tight there have been measurements indicating air flow even through masonry walls (Haines & Myers 2010). Infiltration of air in buildings are due to pressure differences on each side of the element (e.g. a wall, a window or a door). This pressure difference forces air in and out of buildings. Infiltration in buildings is hard to calculate but Novakovic (1995) operates with an air change rate of 0.1-0.3 for the infiltration.

In this model, heat transfer due to infiltration between adjoining room is ignored due to the fact that the temperature in every room is almost identical. Hence the model only considers infiltration through external walls. According to Novakovic (1995), modeling of the infiltration is done with a single resistor as shown in Figure 3.6.

3. MODELING HVAC SYSTEMS

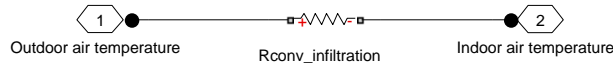


Figure 3.6: Electrical analogy - infiltration

3.3.6 Modeling how the ventilation affects the room

Ventilation contributes to the heat in the building by circulation of air. In this model the ventilation is modeled as a voltage source connected to a resistor. This voltage source represents the temperature of the ventilated air. The value of the resistor is calculated with the knowledge of the amount of air passing in and out of the room through the ventilation system (see Appendix A.1). The infiltration model, Figure 3.6 is equivalent to the ventilation model although with a different resistor value. Also the connected temperature is not the outdoor temperature, but the temperature of the ventilated air.

3.3.7 Modeling the radiators

Radiators actually behaves like a heat exchanger. Water from the radiator circuit enters the radiator and thus the radiator heats up. This heating process takes a while because of the radiators ability to store energy (i.e. heat). Once the radiator heats up the heat from it is transfered to the air by natural heat conduction and convection. Modeling this using physical laws is a complicated and time consuming task. Due to this the radiator in this report is modeled simply as a first order transfer function that acts similar to the way a real radiator would do. Assuming that the efficiency of the radiator is 100 % gives the following transfer function for the radiator:

$$H(s) = \frac{1}{Ts + 1} \quad (3.8)$$

In this equation T is the time constant. The value of this determines the time it takes for the output signal to be about 63% of the input signal. With this in mind T is chosen to be 100. Meaning that it takes 100 seconds until the temperature of the radiator is approximately 63% of the water temperature sent to the radiator. It should be noted that usually the water temperature to the radiator is fixed while the flow of water through the radiator changes. This was explained in Section 2.1.2. Although this is usually the case, the effect is the same as if the water temperature was varying with constant flow. Exploiting this fact makes

the modeling in this report easier since only information about the temperature is available.

3.3.8 Modeling how the radiators affects the room

From the previous section the temperature of the radiator is known. By knowing the temperature of the radiators, the effect on the room caused by them can be modeled in the same way as the ventilation. The only difference is that the resistors are not connected to the outdoor temperature as the ventilation is, but to another voltage source which represent the temperature of the radiator. The infiltration model, Figure 3.6, is equivalent to the radiator model although with a different resistor value (see Appendix A) and with the radiator temperature instead of the outdoor temperature.

3.3.9 Modeling the indoor air

Indoor air temperature is modeled simply as a capacitor which represent the heat capacity of the air in the room. The initial voltage of this capacitor represents the initial temperature of the room. See Appendix A for the calculation of this.

3.3.10 Connecting the elements of the room

When modeling an entire room each of the preceding building elements present in that particular room needs to be connected. This is done by simply connecting the port called “indoor air temperature” to the same port on some other building element. Although this leads to a good model of a room it does not include radiation through the windows and into the wall and ceiling, nor does it include radiation from the radiators to the wall and ceiling. To implement this, resistors representing radiation is connected between the wall/ceiling and in between the two resistors representing the window. The same thing is also done for the radiator. The room model is now a better representation of the real world, thus this implementation should not be neglected. There will of course also be radiation to the floor, but to ensure that the building model does not become too complex, this is neglected. Also the wall with the windows are not expected to receive as much radiation energy as the other walls. This is not taken into account when modeling and thus it will, to some extent, balance the fact that the floor is not taken into account. The resulting electronic circuit, that models the entire room, is shown in Figure 3.7.

3. MODELING HVAC SYSTEMS

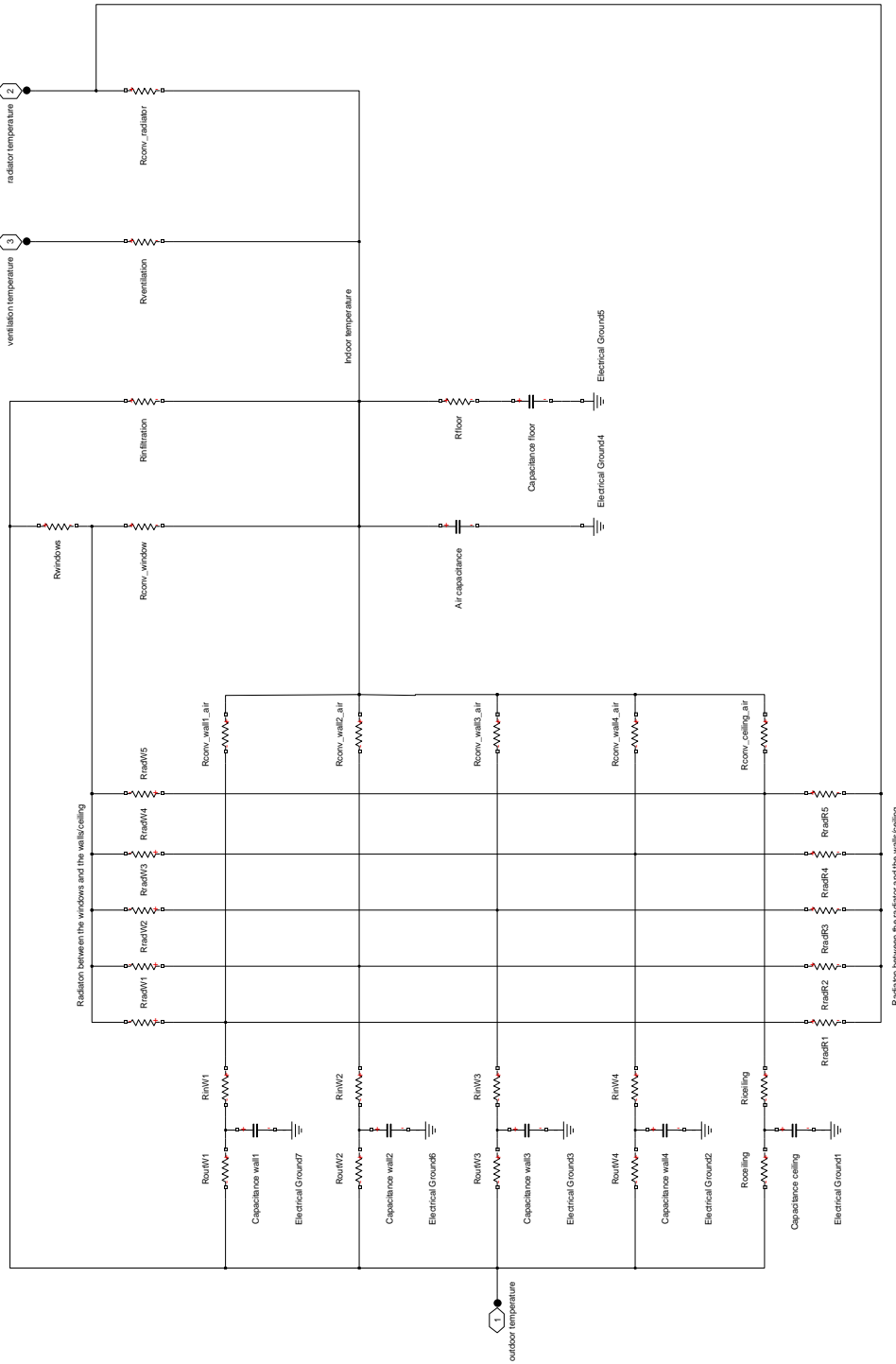


Figure 3.7: Electrical analogy - room

Thus it has been shown how to model a single-room building using `Simscape`. Each of the electrical elements (e.g. resistors and capacitors) are based on simple physical equations as briefly mentioned earlier. See A for details.

3.3.11 Verification of the room model

To verify that the room model behaves as expected some simulations of the model is done to show the temperature development of the room based on various outdoor temperatures. These simulations are based on a passive room, meaning that there are no active elements (e.g. a radiator or ventilation) that affects the temperature development in the room. Thus the temperature development is only due to initial temperature differences between the outdoor air and the indoor air. Temperature changes in the room due to radiators and ventilation will be introduced later in the report. Figure 3.8 shows the temperature development using 5°C as outdoor temperature and 20°C as initial indoor temperature. Together with an initial temperature of 20°C for the walls, ceiling and roof, and with the ventilation turned off, the model was simulated for a duration of 1 day. The figure shows that the room temperature falls towards the outdoor temperature. Simulating over a longer period of time verifies that the temperature drop actually stops when the indoor temperature is equal to the outdoor temperature (Figure 3.9). This response makes sense since the room itself does not produce any heat. Thus the indoor temperature should fall towards the outdoor temperature. As can be seen from the plot in Figure 3.9 the temperature falls rapidly during the first hours before slowing down as the indoor temperature approaches the outdoor temperature. This is as expected according to the second law of thermodynamics (see Equation 3.1).

Figure 3.10 shows the same as Figure 3.8 with one exception; the initial temperature of the walls, ceiling and roof is now 10°C instead of 20°C . As can be seen from the plot in this figure the temperature drop is a lot steeper during the first hour. This type of temperature response is as expected due to the fact that the thermal mass of the walls, ceiling and floor are considerable larger than the thermal mass of the air inside of the building. Hence the air rapidly reaches the temperature of the wall and then follows this temperature until both temperatures are equal to the outdoor temperature.

Finally the model is tested with a varying outdoor temperature. Now the outdoor temperature is a sinus curve with an amplitude of 10°C and a frequency of one per day plus a constant temperature of 5°C . This leads to a temperature varying between -5°C and 15°C during 24 hours. The resulting indoor temperature of this type of external temperature is shown in Figure 3.11. By comparing

3. MODELING HVAC SYSTEMS

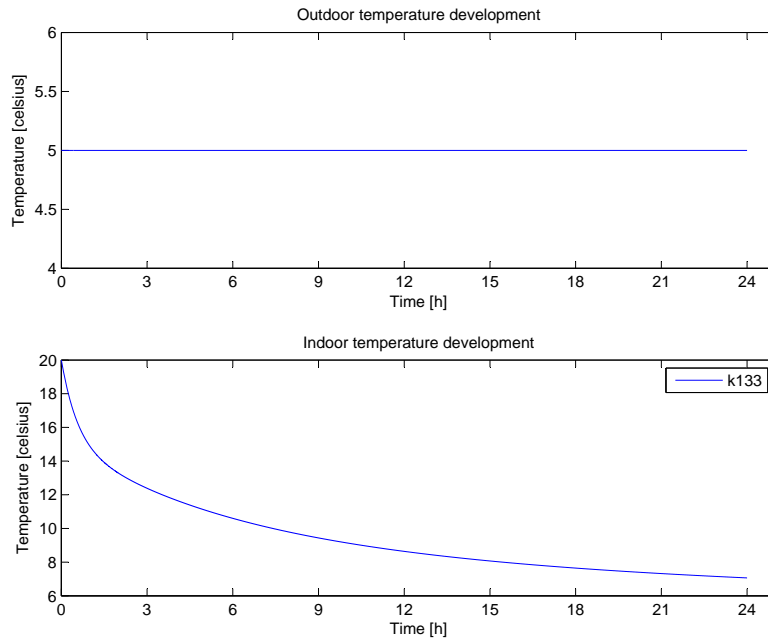


Figure 3.8: Temperature development of the passive single-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C

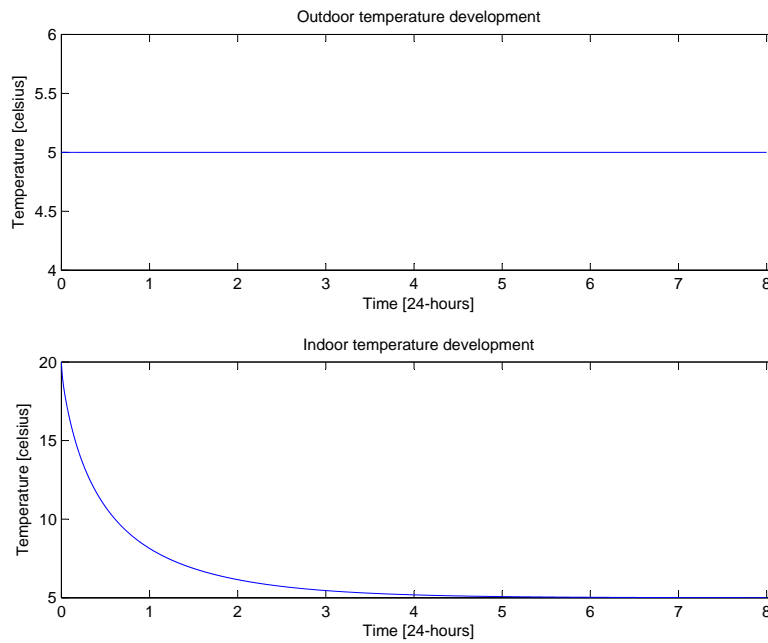


Figure 3.9: Temperature development of the passive single-room model - duration: 8 days, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C

3.4 Connecting adjoining rooms

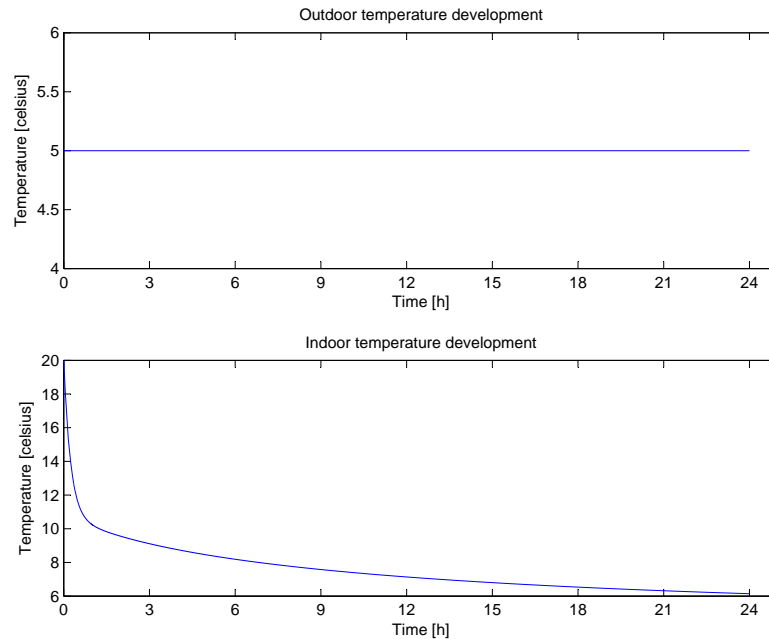


Figure 3.10: Temperature development of the passive single-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 10°C

this temperature response to the external temperature it is easy to see that the indoor temperature descends towards the outdoor temperature and then follows the external temperature variation, although with a lag. This lag is present because of the large thermal mass in the walls. Thermal mass stores energy in the sense of heat and thus needs a certain time to cool down or heat up. This is also the reason why the indoor temperature never reaches the extremes of the varying temperature (i.e. -5°C or 15°C).

Together these plots shows that the thermal model of the room is responding to outdoor temperature changes as one would expect from real life experiences. Thus it has been shown that the model is valid, meaning accurate enough, and that it can be used to simulate temperature development in buildings.

3.4 Connecting adjoining rooms

This section extends the single-room model by connecting several adjoining rooms. By doing this one can model an entire building consisting of several rooms, hallways and floors. The extended model in this report will only have one floor, but

3. MODELING HVAC SYSTEMS

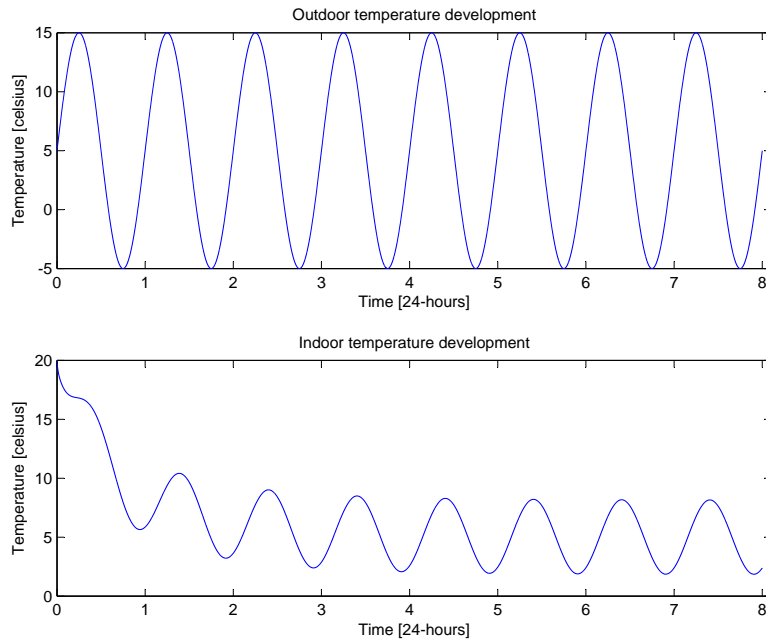


Figure 3.11: Temperature development of the passive single-room model - duration: 8 day, outdoor temperature: varying, initial indoor temperature: 20°C, initial wall temperature: 20°C

connecting several floors to the model is done exactly the same way as for the walls connecting adjoining rooms.

3.4.1 Model redesign

When creating a model of a building consisting of several rooms one could be tempted to simply copy the design of the room model constructed in the previous section. The drawback of this is that it leads to a lot of lines crossing each other, especially when many rooms are interconnected. Another and more severe drawback is that the walls connected to another room is also present in the adjoining room. This leads to a “double wall” or a wall twice as thick as it should be, thus the resistors of those walls must be altered as must the capacitors.

To make the model more intuitive, without too many lines crossing each other, thus making the model easier to read, and without the need of altering the resistors and the capacitors, a new model design is suggested. This design is equivalent to the single-room design but with the walls, including windows, separated from the rest of the room (i.e. a model for each of the walls and a model for the rest of the room). Thus making it easier to connect adjoining rooms, see Figure 3.12. The

3.4 Connecting adjoining rooms

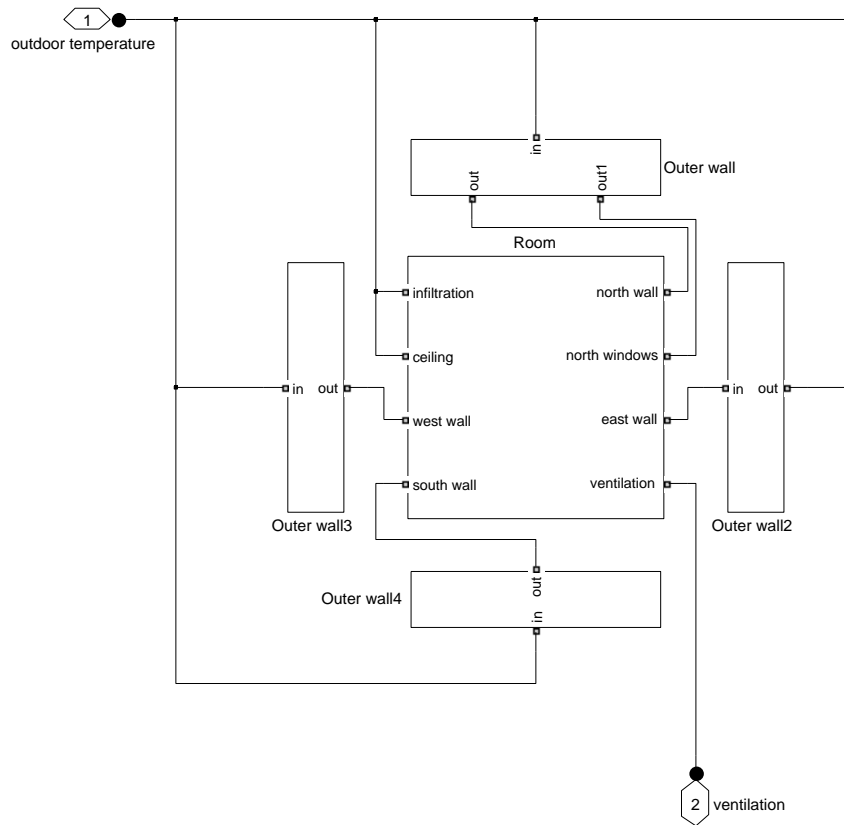


Figure 3.12: Electrical analog room and walls; new design

model of the room is now shown in Figure 3.13 and the model of the walls without windows is seen in Figure 3.14. Finally the model of the wall with windows is shown in Figure 3.15.

When redesigning the model there are a few things one needs to be aware of to avoid that the redesigning becomes redundant. First; when extracting the walls one should not extract the resistor representing heat convection between the wall and the indoor air. This would lead to many connective lines between the wall and the room. There is nothing wrong in doing it so, but it becomes difficult to follow the lines and thus the model becomes difficult to grasp. Secondly; as for the same reason as above, the resistor representing heat convection between the window and the indoor air should not be extracted from the room into the wall. Finally; since this is a building with only one floor the resistors and the capacitor representing the ceiling should not be extracted at all. This would only lead to yet another block in the main structure and even more crossing lines, hence the ceiling is included inside the room model. This means that only the walls is extracted from the room model. Altogether this makes up a well organized model

3. MODELING HVAC SYSTEMS

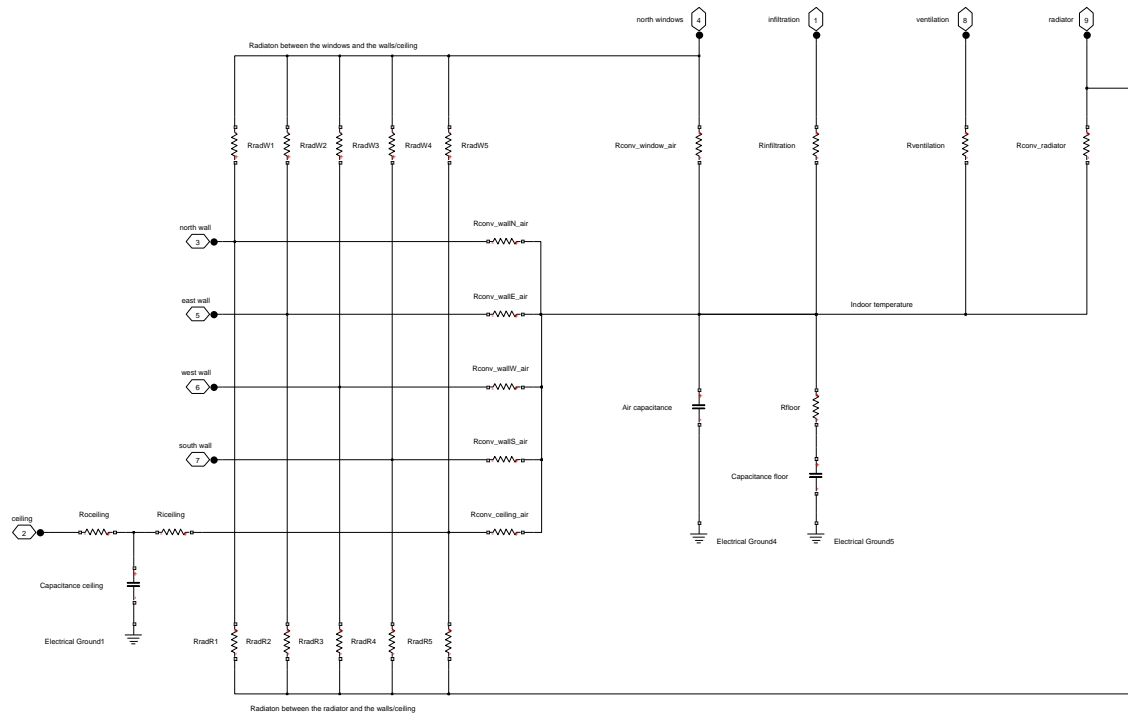


Figure 3.13: Electrical analog room; new design

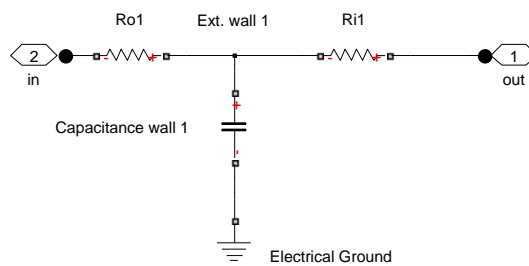


Figure 3.14: Electrical analog wall; new design

3.4 Connecting adjoining rooms

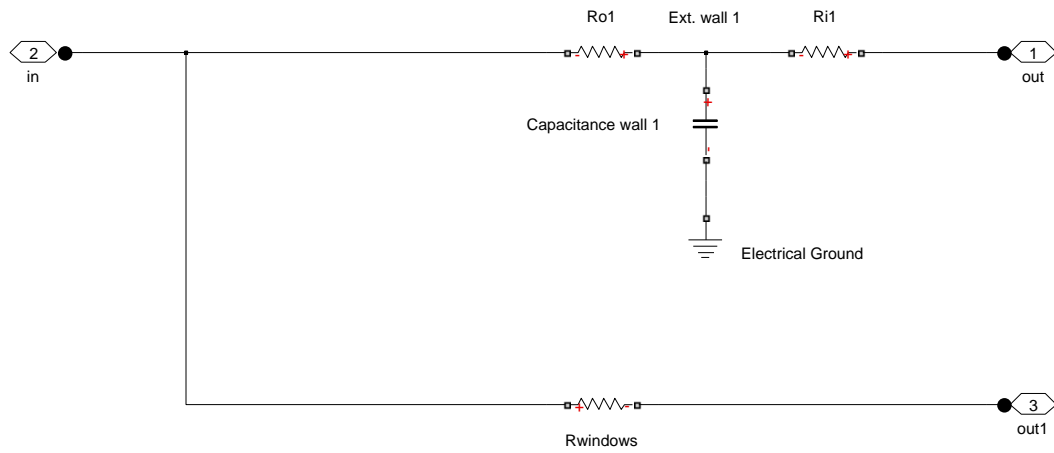


Figure 3.15: Electrical analog wall with windows; new design

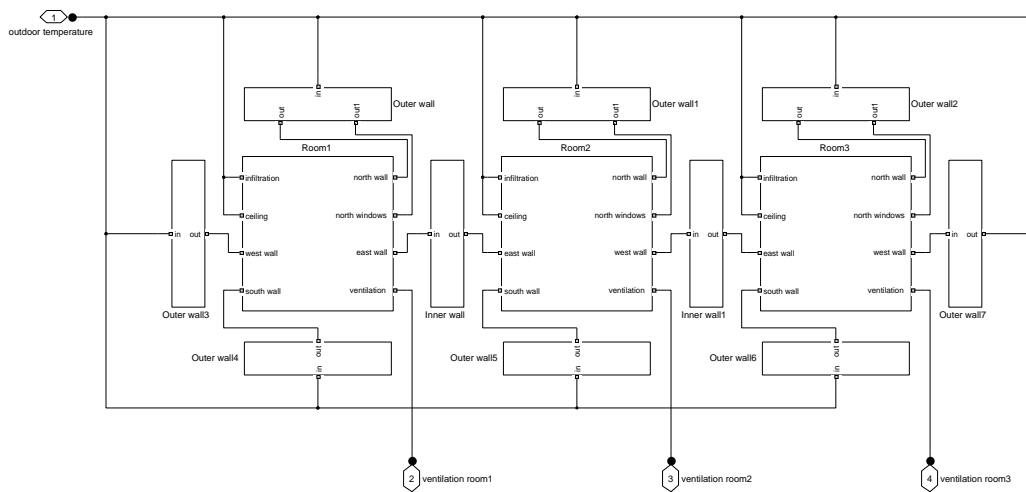


Figure 3.16: 3 rooms connected using the new design

with few connective lines crossing each other. If modeling a building with more than one floor the ceilings and floors should also be extracted from the model. Note that in this case the ceiling of floor one is the same as the floor of floor two (i.e. equal to a wall shared by two adjoining rooms).

With the new design it is easy to connect adjacent rooms and, if applicable, adjacent floors. In Figure 3.16 this is done for a building consisting of 3 rooms and one floor.

3. MODELING HVAC SYSTEMS

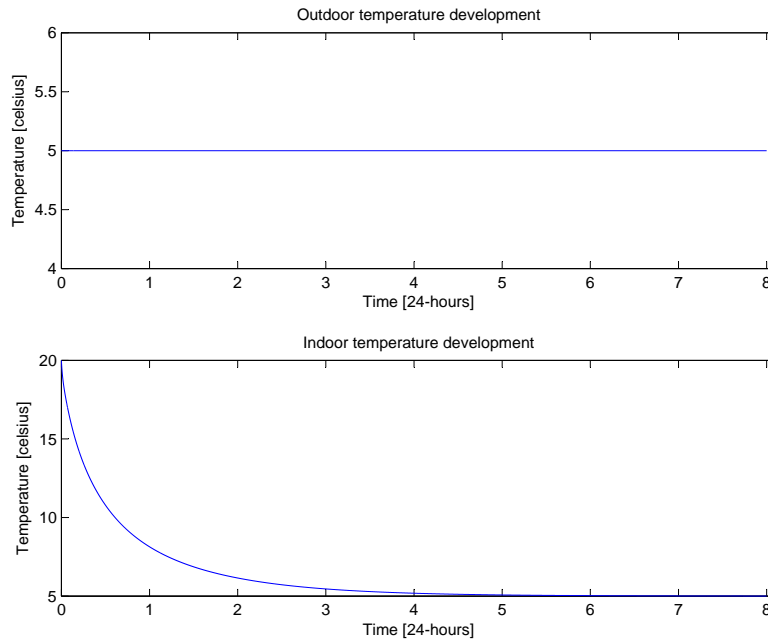


Figure 3.17: Temperature development of the new passive single-room model - duration: 8 days, outdoor temperature: 5°C , initial indoor temperature: 20°C , initial wall temperature: 20°C

3.4.2 Verification of the new design

Comparing simulations of the single passive room equal to the one in Section 3.3.10, but using the new design (see Figure 3.12), shows that both designs responds exactly the same way. In this report the two designs are compared by simulating the temperature development in the room over 8 days with an constant outdoor temperature of 5°C and an initial temperature in the room, the ceiling, the floor and the walls of 20°C . Figure 3.9 shows the indoor temperature response in the previous model while 3.17 shows the same for the new model. When comparing these figures it is easy to see that they are equal. Simulating over different periods of time and with different outdoor temperature and initial temperatures gives the same results as for the previous room model. This verifies that the two model designs are equal to each other.

3.4.3 Verification of the 3-room model

The new 3-room model (Figure 3.16) was simulated over a period of 24-hour with a constant outdoor temperature as was the new single-room building model

3.4 Connecting adjoining rooms

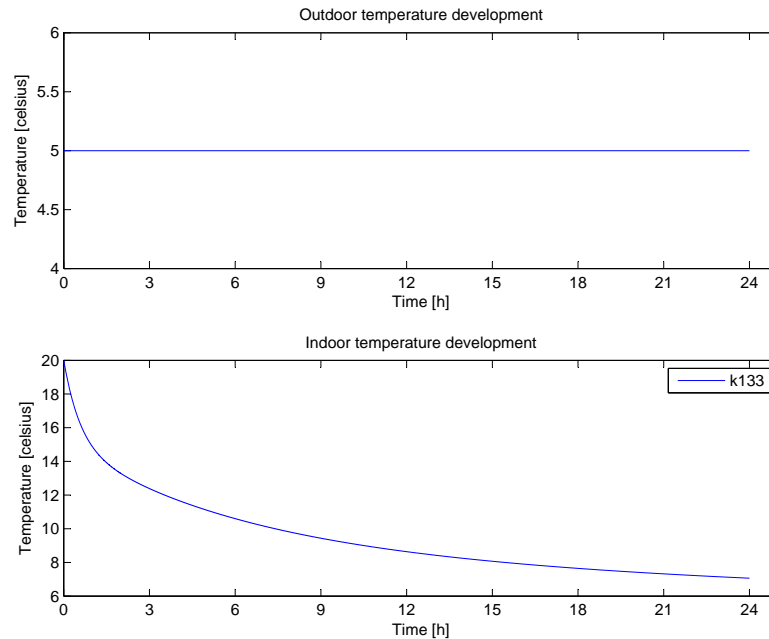


Figure 3.18: Temperature development of the new passive single-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C

(Figure 3.12). The results of these simulations are shown in Figure 3.18 and 3.19. If one compares the response of the single-room model with the response of the 3-room model it shows that the latter responds a little bit slower. The reason for this is as follows: Although the volume of the building is tripled the area of the outdoor walls is not tripled. Hence the effect from the outdoor temperature is not tripled. Also there are internal walls which makes a significant contribution to the thermal mass stored inside the building. Hence this slower response in temperature development is as expected. Figure 3.19 also shows that the room in the middle (i.e. k134) has the slowest reaction to outdoor temperature. The reason for this is that this room is protected from the outdoor temperature by the other rooms on two of its sides. Together this shows that the thermal model with several rooms connected to each other is valid.

Altogether this shows that the thermal model of the room behaves satisfactory and responds as one would expect from real world experience. Having proved this there is a simple but time consuming task to implement several more rooms. This will be done in the next section.

3. MODELING HVAC SYSTEMS

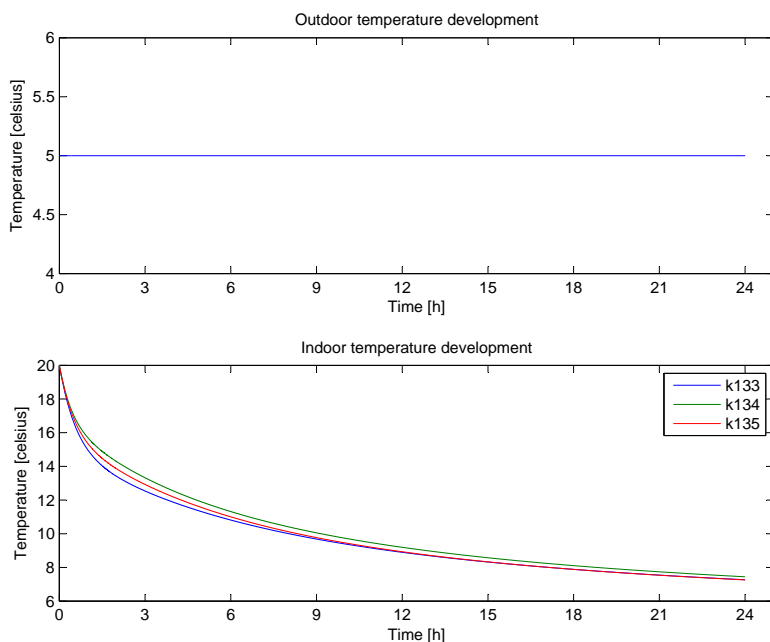


Figure 3.19: Temperature development of the new passive 3-room model - duration: 1 day, outdoor temperature: 5°C, initial indoor temperature: 20°C, initial wall temperature: 20°C

3.5 Modeling the entire building

By exploiting the new model structure a model of the entire building may be created. In this report the entire building consists of 6 rooms, namely room k133, k134, k135, k136, k143 and k164 from Saltstraumen elementary school. A technical drawing of the school is shown in Figure 3.20 to give the reader a view of how the modeled rooms actually looks like. A PDF of this is also included on the attached CD. From this figure it is clear that the rooms included in this report originally are connected to other rooms which has been removed. These walls are considered to be external walls without windows. The m-file `room_values.m` found on the attached CD contains all the values needed for the modeling and simulation of each of the rooms. This includes the room dimensions, computation of every resistors and every capacitor which is used in the model. Appendix A contains the equations used in this file. The thermal model of the 6 rooms included in this report is shown in Figure 3.21. It is easy to recognize each room of the model (Figure 3.21) by comparing it to the 6 rooms with the same name in the technical drawing found in Figure 3.20.

3.5 Modeling the entire building

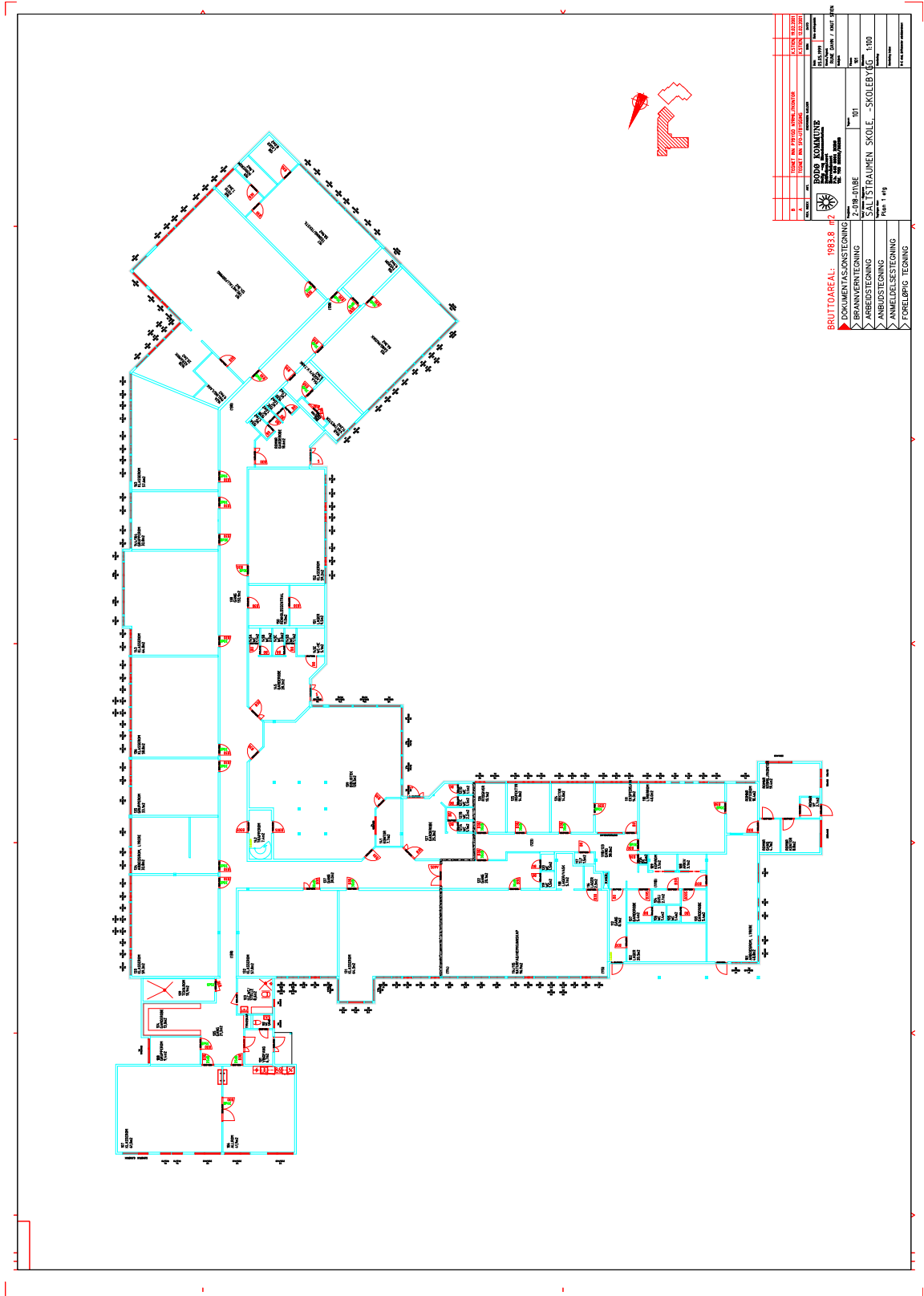


Figure 3.20: Overview of Saltstraumen elementary school, main building

3. MODELING HVAC SYSTEMS

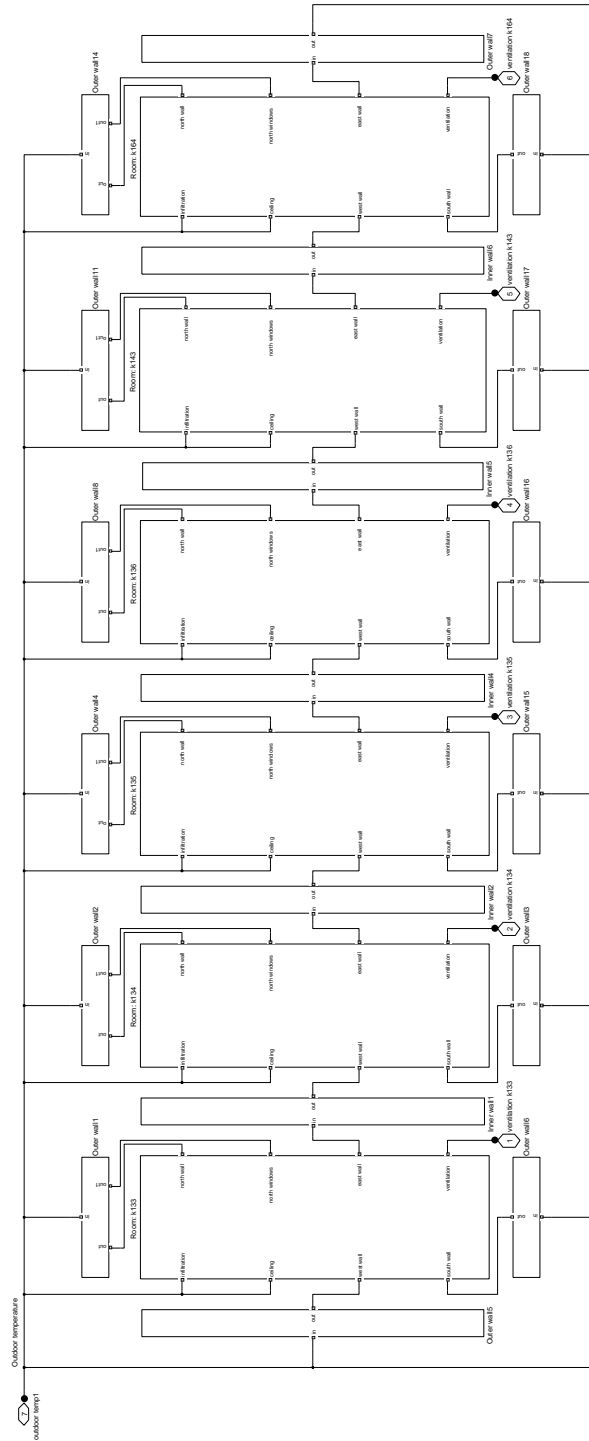


Figure 3.21: Thermal model of the building that this report is based on

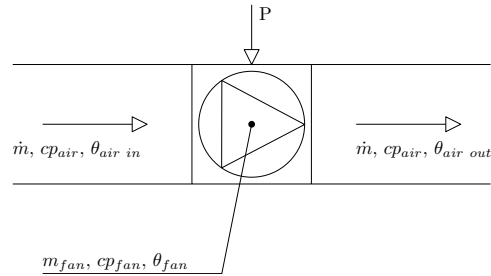


Figure 3.22: Schematic view of the fan

3.6 Modeling the ventilation system

When modeling the ventilation system each element will be modeled according to the law of energy balance (heat balance) explained in Section 3.1.2 and mass balance explained in Section 3.1.1. Some simplifications or assumptions have been made in each of the models in order to keep them as simple as possible, but still making sure that they are a good approximation to the real world. Each of the models represented in this section are based on the models found in Novakovic (1995).

3.6.1 Modeling the fan

When modeling the air temperature in the ventilation system it is important to take the fans into account. This is because each fan normally contributes to an increase of the air temperature by a magnitude of 1-2 K. The exact temperature rise depends on the increase of pressure over the fan and whether or not the engine of the fan is placed inside or outside the duct. This model however, is a model of both the fan and its motor, meaning that the fan and the motor can be described by a common time constant. Every symbol used in this model are listed in Table 3.2. A schematic view of the fan can be seen in Figure 3.22.

Simplifications and assumptions

- Constant airflow
- Fan temperature is equal to air temperature out from the fan ($\theta_{fan} = \theta_{ao}$)

3. MODELING HVAC SYSTEMS

Table 3.2: Symbols used in the fan model

Symbol	Description	Unit	Remarks
m_{fan}	Mass, fan	kg	
cp_{fan}	Specific heat capacity, fan	$\frac{J}{kg \cdot K}$	
θ_{fan}	Temperature, fan	$^{\circ}C$	
$\Delta\theta_{fan}$	Static temperature change through fan	$^{\circ}C$	
\dot{m}_{air}	Air flow through the fan	$\frac{kg}{s}$	
cp_{air}	Specific heat capacity, air	$\frac{J}{kg \cdot K}$	
P	Power, fan	W	$P = \dot{m}_{air} \cdot cp_{air} \cdot \Delta\theta_{fan}$
θ_{ai}	Temperature, air into fan	$^{\circ}C$	
θ_{ao}	Temperature, air from fan	$^{\circ}C$	

Mathematical model

The mathematical model is based on the assumption that the air flow through the fan is constant. Thus, only the law of conservation of energy (see Section 3.1.2) is needed to find a mathematical description of the temperature changes through the fan. From this law the following equation can be found:

$$\frac{\partial(m_{fan} \cdot cp_{fan} \cdot \theta_{ao})}{\partial t} = \dot{m}_{air} \cdot cp_{air} \cdot (\theta_{ai} - \theta_{ao}) + P \quad (3.9)$$

As previously mentioned the temperature increase over the fan is from 1-2 K in statical conditions. It is possible to define the power of the fan by the following equation in statical conditions.

$$P = \dot{m}_{air} \cdot cp_{air} \cdot (\theta_{ai} - \theta_{ao})_{statical} = \dot{m}_{air} \cdot cp_{air} \cdot \Delta\theta_{fan} \quad (3.10)$$

Combining this equation with Equation 3.9, gives the following equation for temperature development across the fan.

$$T_{fan} \cdot \frac{\partial\theta_{ao}}{\partial t} + \theta_{ao} = \theta_{ai} + \Delta\theta_{fan} \quad (3.11)$$

where

3.6 Modeling the ventilation system

$$T_{fan} = \frac{m_{fan} \cdot cp_{fan}}{\dot{m}_{air} \cdot cp_{air}} \quad (3.12)$$

According to Novakovic (1995) a typical value for T_{fan} is 30 seconds.

Transfer function

Laplace transformation of Equation 3.11 gives

$$T_{fan} \cdot \theta_{ao}(s) \cdot s + \theta_{ao}(s) = \theta_{ai}(s) + \Delta\theta_{fan}(s) \quad (3.13)$$

See Balchen et al. (2003) for a short introduction to transfer functions. Solving the above equation with respect to θ_{ao} results in the following

$$\theta_{ao}(s) = H_{aa}(s) \cdot \theta_{ai}(s) + H_{fan}(s) \cdot \Delta\theta_{fan}(s) \quad (3.14)$$

where $H_{aa}(s)$ and $H_{fan}(s)$ are given by

$$H_{aa}(s) = \frac{\theta_{ao}(s)}{\theta_{ai}(s)} = \frac{1}{T_{fan} \cdot s + 1} \quad (3.15)$$

$$H_{fan}(s) = \frac{\theta_{ao}(s)}{\Delta\theta_{fan}(s)} = \frac{1}{T_{fan} \cdot s + 1} \quad (3.16)$$

The block diagram of this model is seen in Figure 3.23. This can easily be implemented into **Simulink** using the block "transfer function with initial states". The reason for using this block is that it gives the system developer the opportunity to set initial temperature of the air passing through the fan.

3. MODELING HVAC SYSTEMS

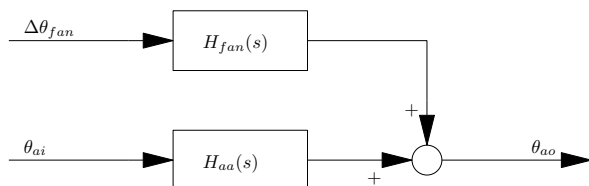


Figure 3.23: Block diagram of the fan model

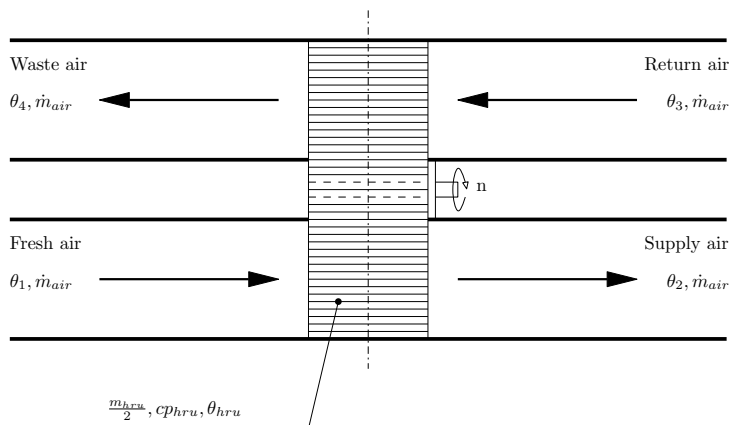


Figure 3.24: Schematic view of the heat recovery unit

3.6.2 Modeling the heat recovery unit

This section describes a heat recovery unit with a rotating heat exchanger and is based on the heat recovery unit found in Novakovic (1995). However, this model neglects humidity whereas Novakovic (1995) does not. The reason for doing so is that the focus in this report is temperature changes and the affection on temperature based on humidity is barely noticeable. Figure 3.24 is a schematic view of the heat recovery unit.

Simplifications and assumptions

- Constant airflow. Supply and return airflow are equal.
- Constant rotational speed of the rotating heat exchanger.
- Uniform temperature in each half of the metal in the heat recovery unit.
- Supply temperature is equal to the metal temperature in the warm half of the heat recovery unit ($\theta_2 = \theta_{hru}$).

3.6 Modeling the ventilation system

Table 3.3: Symbols used in the heat recovery unit model

Symbol	Description	Unit	Remarks
m_{hru}	Mass, heat recovery unit (hru)	kg	
cp_{hru}	Specific heat capacity, heat recovery unit	$\frac{J}{kg \cdot K}$	
θ_{hru}	Temperature, heat recovery unit	$^{\circ}C$	
\dot{m}_{air}	Air flow in supply and waste ducts	$\frac{kg}{s}$	
cp_{air}	Specific heat capacity, air	$\frac{J}{kg \cdot K}$	Al or Al/Cu
θ_1	Temperature, fresh air	$^{\circ}C$	
θ_2	Temperature, supply air	$^{\circ}C$	
θ_3	Temperature, return air	$^{\circ}C$	
θ_4	Temperature, waste air	$^{\circ}C$	
n	Rotational speed of the rotating heat exchanger	$^{\circ}C$	

It should be noted that uniform temperature in each half of the metal in the heat recovery unit is a major simplification and not how it is in the real world. In reality the air temperature would slowly descend from where it enters the cooling side of the heat recovery unit and until it exits the cooling side. The opposite would happen on the heating side of the heat recovery unit. But in order to keep the model as simple as possible the unit is divided into two parts where the temperature is fixed. The mathematical solution is approximately the same.

Mathematical model

The mathematical model is based on the assumption that the air flow through the heat recovery unit is constant. Thus, only the law of conservation of energy (see Section 3.1.2) is needed to find a mathematical description of the temperature changes through the heat recovery unit. From this law the following equation can be found:

$$\frac{\partial(m_{hru} \cdot cp_{hru} \cdot \theta_2)}{\partial t} = \dot{m}_{air} \cdot cp_{air} \cdot (\theta_1 - \theta_2) + \dot{m}_{air} \cdot cp_{air} \cdot (\theta_3 - \theta_4) \quad (3.17)$$

A heat recovery unit consist of several types of metal. Usually this is a combination of aluminum and copper or only copper. In order to find the specific heat capacity of the heat recovery unit, one needs to know the share of aluminum and the share of copper in the unit.

Using the exact same method as for the fan, rearranging Equation 3.17 gives

3. MODELING HVAC SYSTEMS

$$T_{hru} \cdot \frac{\partial \theta_2}{\partial t} + \theta_2 = \theta_1 + \theta_3 - \theta_4 \quad (3.18)$$

where

$$T_{hru} = \frac{m_{hru} \cdot cp_{hru}}{2 \cdot \dot{m}_{air} \cdot cp_{air}} \quad (3.19)$$

Introducing the temperature efficiency (ϵ_θ):

$$\epsilon_\theta = \frac{\theta_2 - \theta_1}{\theta_3 - \theta_1} = \frac{\theta_3 - \theta_4}{\theta_3 - \theta_1} \quad (3.20)$$

Solving for θ_4 and combining this with Equation 3.18 gives

$$T_{hru} \cdot \frac{\partial \theta_2}{\partial t} + \theta_2 = (1 - \epsilon_\theta) \cdot \theta_1 + \epsilon_\theta \cdot \theta_3 \quad (3.21)$$

It should be noted that the equation for temperature efficiency is valid only when the airflow is constant.

Transfer function

Laplace transformation of Equation 3.21 gives

$$T_{hru} \cdot \theta_2(s) \cdot s + \theta_2(s) = (1 - \epsilon_\theta) \cdot \theta_1(s) + \epsilon_\theta \cdot \theta_3(s) \quad (3.22)$$

Solving this equation with respect to θ_2 results in the following

$$\theta_2(s) = H_{\theta,12}(s) \cdot \theta_1(s) + H_{\theta,32}(s) \cdot \theta_3(s) \quad (3.23)$$

where $H_{\theta,12}(s)$ and $H_{\theta,32}(s)$ are given by

3.6 Modeling the ventilation system

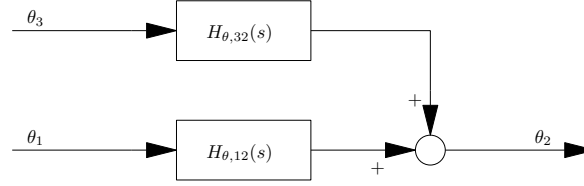


Figure 3.25: Block diagram of the heat recovery unit model

$$H_{\theta,12}(s) = \frac{\theta_2(s)}{\theta_1(s)} = \frac{1 - \epsilon_\theta}{T_{hru} \cdot s + 1} \quad (3.24)$$

$$H_{\theta,32}(s) = \frac{\theta_2(s)}{\theta_3(s)} = \frac{\epsilon_\theta}{T_{hru} \cdot s + 1} \quad (3.25)$$

The block diagram of this model is seen in Figure 3.25

3.6.3 Modeling the heating coil

This model describes how the temperature changes in the water and the air when passing through the heating coil. Figure 3.26 is a schematic view of the heating coil.

Simplifications and assumptions

- Constant airflow through the heating coil ($\dot{m}_{air} = constant$).
- Constant flow of water through the heating coil ($\dot{m}_w = constant$).
- No heat loss to the surrounding elements.
- Direct connection between metal temperature (θ_{hc}) of the heating coil and air temperature into the heating coil (θ_{ai}).
- Metal temperature of the heating coil is equal to the water temperature from the heating coil ($\theta_b = \theta_{wo}$).

Under the assumption that the flow of water and air in the heating coil are constant it is possible to express the heat transfer coefficients as a function of the heating coils nominal working conditions. That is, those conditions that are present when both the water and the air temperature are given as design values. By doing so it is not necessary to calculate the heating coils u-factor.

3. MODELING HVAC SYSTEMS

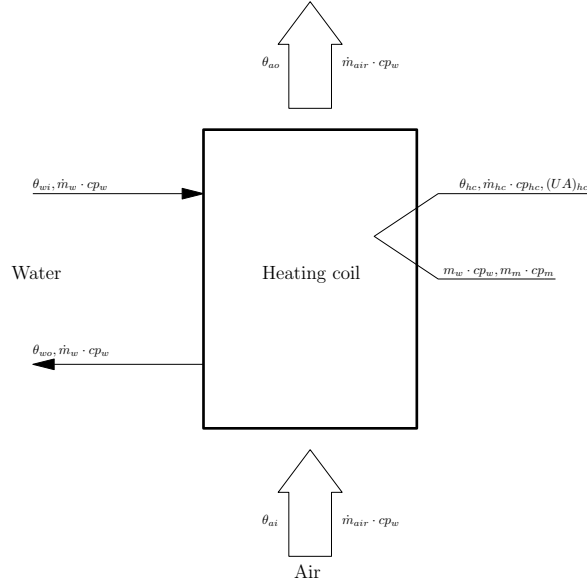


Figure 3.26: Schematic view of the heating coil

Table 3.4: Symbols used in the heating coil model

Symbol	Description	Unit	Remarks
θ_{wi}	Temperature, water in	$^{\circ}\text{C}$	
θ_{wo}	Temperature, water out	$^{\circ}\text{C}$	
θ_{hc}	Temperature, heating coil	$^{\circ}\text{C}$	$\theta_{hc} = \theta_{wo}$
θ_{ai}	Temperature, air in	$^{\circ}\text{C}$	
θ_{ao}	Temperature, air out	$^{\circ}\text{C}$	
\dot{m}_w	Mass flow, water	$\frac{\text{kg}}{\text{s}}$	
\dot{m}_{air}	Mass flow, air	$\frac{\text{kg}}{\text{s}}$	
m_w	Mass of water in the heating coil	kg	
m_m	Mass of metal in the heating coil	kg	
m_b	Total mass of the heating coil (incl. water)	kg	$m_{hc} = m_w + m_m$
$(UA)_{hc}$	Resulting heat transmission number	$\frac{\text{W}}{\text{K}}$	
cp_w	Specific heat capacity, water	$\frac{\text{J}}{\text{kg}\cdot\text{K}}$	
cp_{air}	Specific heat capacity, air	$\frac{\text{J}}{\text{kg}\cdot\text{K}}$	
cp_m	Specific heat capacity, metal	$\frac{\text{J}}{\text{kg}\cdot\text{K}}$	
cp_{hc}	Specific heat capacity, heating coil	$\frac{\text{J}}{\text{kg}\cdot\text{K}}$	$cp_{hc} = \frac{cp_w \cdot m_w + cp_m \cdot m_m}{m_{hc}}$

Mathematical model

The mathematical model is based on the assumption that both the air flow and the water flow through the heating coil is constant. Thus, only the law of conservation of energy (see Section 3.1.2) is needed to find a mathematical description of the temperature changes through the heating coil.

The following equation is a dynamic heat balance for the water in the heating coil.

$$\frac{\partial(m_{hc} \cdot cp_{hc} \cdot \theta_{hc})}{\partial t} = \dot{m}_w \cdot cp_w \cdot (\theta_{wi} - \theta_{wo}) - (UA)_{hc} \cdot (\theta_{wo} - \theta_{ai}) \quad (3.26)$$

Due to the fact that both m_{hc} and cp_{hc} are constants and $\theta_{hc} = \theta_{wo}$, this equation can be written as

$$m_{hc} \cdot cp_{hc} \cdot \frac{\partial \theta_{wo}}{\partial t} = \dot{m}_w \cdot cp_w \cdot (\theta_{wi} - \theta_{wo}) - (UA)_{hc} \cdot (\theta_{wo} - \theta_{ai}) \quad (3.27)$$

where the resulting heat transmission number $(UA)_{hc}$, found by multiplying the U factor by the area of the object of interest, is unknown. The expression for this will be derived now.

Static heat balance for the air in the heating coil can be written as

$$m_{air} \cdot cp_{air} \cdot (\theta_{ao} - \theta_{ai}) = (UA)_{hc} \cdot (\theta_{wo} - \theta_{ai}) \quad (3.28)$$

Introducing relative water cooling (ϵ_w) and relative air heating (ϵ_{air})

$$\epsilon_w = \frac{\theta_{wi} - \theta_{wo}}{\theta_{wi} - \theta_{ai}} \quad (3.29)$$

$$\epsilon_{air} = \frac{\theta_{ao} - \theta_{ai}}{\theta_{wi} - \theta_{ai}} \quad (3.30)$$

3. MODELING HVAC SYSTEMS

Both ϵ_w and ϵ_{air} are calculated during nominal conditions. These will be constant as long as both the air flow and the water flow through the heating coil remain constant. Combining Equation 3.28-3.30 gives an expression for $(UA)_{hc}$

$$(UA)_{hc} = \frac{\epsilon_{air}}{1 - \epsilon_w} \cdot \dot{m}_{air} \cdot cp_{air} \quad (3.31)$$

Another expression for $(UA)_{hc}$ is possible since the static heat balance of the air is equal to the static heat balance of the water. This gives the following expression for $(UA)_{hc}$

$$(UA)_{hc} = \frac{\epsilon_w}{1 - \epsilon_w} \cdot \dot{m}_w \cdot cp_w \quad (3.32)$$

This equation can then be inserted into Equation 3.27 and it leads to the following equation for the dynamic heat balance of the water in the heating coil:

$$T_{hc} \frac{\partial \theta_{wo}}{\partial t} + \theta_{wo} = (1 - \epsilon_w) \cdot \theta_{wi} + \epsilon_w \cdot \theta_{ai} \quad (3.33)$$

where

$$T_{hc} = \frac{m_{hc} \cdot cp_{hc}}{\dot{m}_w \cdot cp_w} \cdot (1 - \epsilon_w) \quad (3.34)$$

By combining Equation 3.29 and Equation 3.30 the following relation can be derived

$$\theta_{wo} = \frac{1 - \epsilon_w}{\epsilon_{air}} \cdot \theta_{ao} + \frac{1 - \epsilon_w - \epsilon_{air}}{\epsilon_{air}} \cdot \theta_{ai} \quad (3.35)$$

This equation can be used in combination with Equation 3.33 to end up with the following equation for the heat balance of the air in the heating coil:

$$T_{hc} \frac{\partial \theta_{ao}}{\partial t} + \theta_{ao} = \epsilon_{air} \cdot \theta_{wi} + T_{hcll} \cdot \frac{\partial \theta_{ai}}{\partial t} + (1 - \epsilon_{air}) \cdot \theta_{ai} \quad (3.36)$$

3.6 Modeling the ventilation system

Transfer function

Laplace transformation of Equation 3.33 gives

$$T_{hc} \cdot \theta_{wo}(s) \cdot s + \theta_{wo}(s) = (1 - \epsilon_w) \cdot \theta_{wi}(s) + \epsilon_w \cdot \theta_{ai}(s) \quad (3.37)$$

Laplace transformation of Equation 3.36 gives

$$T_{hc} \cdot \theta_{ao}(s) \cdot s + \theta_{ao}(s) = \epsilon_{air} \cdot \theta_{wi}(s) + T_{hcll} \cdot \theta_{ai}(s) \cdot s + (1 - \epsilon_{air}) \cdot \theta_{ai}(s) \quad (3.38)$$

Equation 3.37 and 3.38 can be written as four transfer functions. First, a transfer function from incoming water temperature to outgoing water temperature:

$$H_{ww}(s) = \frac{\theta_{wo}(s)}{\theta_{wi}(s)} = \frac{1 - \epsilon_w}{T_{hc} \cdot s + 1} \quad (3.39)$$

Second, a transfer function from incoming air temperature to outgoing water temperature:

$$H_{aw}(s) = \frac{\theta_{wo}(s)}{\theta_{ai}(s)} = \frac{\epsilon_w}{T_{hc} \cdot s + 1} \quad (3.40)$$

Third, a transfer function from incoming water temperature to outgoing air temperature:

$$H_{wa}(s) = \frac{\theta_{ao}(s)}{\theta_{wi}(s)} = \frac{\epsilon_{air}}{T_{hc} \cdot s + 1} \quad (3.41)$$

And finally, a transfer function from incoming air temperature to outgoing air temperature:

$$H_{aa}(s) = \frac{\theta_{ao}(s)}{\theta_{ai}(s)} = \frac{(1 - \epsilon_{air}) + T_{hcll} \cdot s}{T_{hc} \cdot s + 1} \quad (3.42)$$

The block diagram of these four transfer functions can be seen in Figure 3.27.

3. MODELING HVAC SYSTEMS

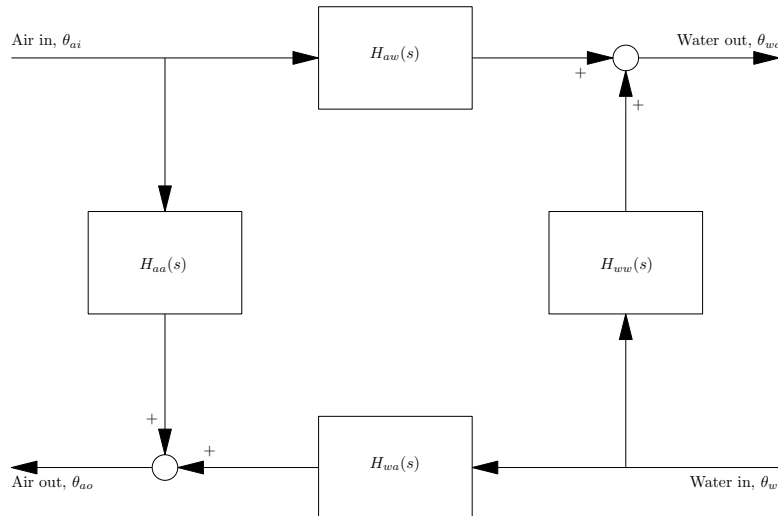


Figure 3.27: Block diagram of the heating coil model

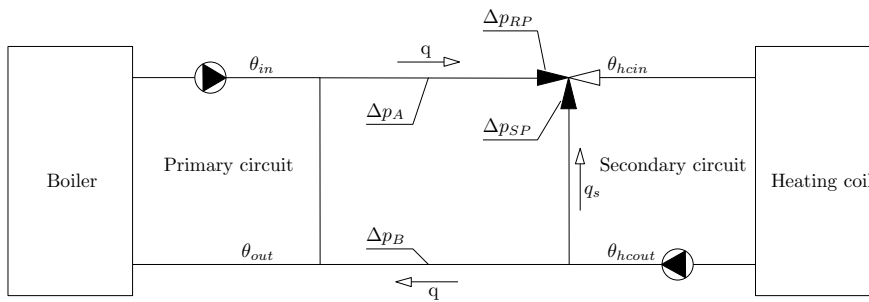


Figure 3.28: Schematic view of the shunt connection

3.6.4 Modeling the shunt connection

This model is based on static pressure and heat balance. Figure 3.28 is a schematic view of the shunt connection. As can be seen from this figure the shunt connection consists of a three way valve and ancillary pipe coupling. This four way shunt connection is called Norwegian coupling (Norsk kobling).

Simplifications and assumptions

- Constant pressure drop, and hence a constant amount of water ($q = 1$) in the secondary circuit (heating coil circuit).
- Turbulent flow.
- No heat loss to the surrounding elements in the valve.

3.6 Modeling the ventilation system

Table 3.5: Symbols used in the shunt connection model

Symbol	Description	Unit	Remarks
θ_{in}	Water temperature from the primary circuit	$^{\circ}\text{C}$	
θ_{out}	Water temperature to the primary circuit	$^{\circ}\text{C}$	
θ_{hci}	Water temperature to the heating coil	$^{\circ}\text{C}$	
θ_{hco}	Water temperature from the heating coil	$^{\circ}\text{C}$	
Δp_{RP}	Pressure drop over the regulate port (100% open)	Pa	
Δp_{SP}	Pressure drop over the shunt port (100% open)	Pa	
Δp_{II}	Pressure drop over the pipe lengths A and B	Pa	$\Delta p_{II} = \Delta p_A + \Delta p_B$
Δp_{III}	Pressure drop over the shunt port (100% open)	Pa	
z	Relative stem position (regulate port)		$0 \leq z = \frac{Z}{Z_{max}} \leq 1$
q	Relative water flow (regulate port)		$0 \leq q = \frac{Q}{Q_{max}} \leq 1$
q_s	Relative water flow (shunt port)		$q_s = 1 - q$

With the first assumption it is possible to ignore resistance in the secondary circuit and only calculate the resistance in those pipes where the water quantity varies. This is due to the fact that the resistance in pipes with constant amount of water will affect every flow directions with an equal amount. The second simplification indicates the following relationship between the pressure drop and the amount of water:

$$\Delta p = K \cdot \dot{Q}^2 \quad (3.43)$$

This assumption is valid in almost all cases. It should be noted that if laminar flow should occur, then the relationship between pressure and water will be linear. However, in this model it is assumed turbulent flow at all times.

Mathematical model

The following mathematical model is based on the assumption that there is a static pressure and heat balance in the shunt connection. The heat balance is only considered over the three way valve, while the pressure balance is calculated using the entire shunt connection. Modeling of the tree way valve is done by modeling two separate two way valves. The pressure model is used as a connection between the valve gain and the temperature response on the water from the valve.

Static pressure balance gives

3. MODELING HVAC SYSTEMS

$$\Delta p_{RPz} + \Delta p_{IIz} = \Delta p_{SPz} + \Delta p_{IIIz} = \text{constant} \quad (3.44)$$

The pressure drops in the above equation are all marked with z since they vary with the valve position. Introducing the pressure drop when the valve opening is at 100%. This value is then scaled with respect to relative water amount and valve characteristic. The amount of water through the regulate port as a function of the performance number k_{ws} and the pressure drop is given as:

$$\dot{Q}_{RP} = k_{vs} \cdot f_{RP}(z) \cdot \sqrt{\Delta p_{RPz}} = \dot{Q}_{max} \cdot f_{RP}(z) \cdot \sqrt{\frac{\Delta p_{RPz}}{\Delta p_{RP}}} \quad (3.45)$$

where f_{RP} is the valve characteristic of the regulate port. This equation can be solved with respect to the varying pressure, and this leads to the following equation for the pressure drop over the regulate port valve:

$$\Delta p_{RPz} = \frac{q^2}{f_{RP}(z)^2} \cdot \Delta p_{RP} \quad (3.46)$$

The exact same is valid for the shunt port valve. This is done by substituting Δp_{RPz} by Δp_{SPz} , $f_{RP}(z)$ by $f_{SP}(z)$ and Δp_{RP} by Δp_{SP} in the above equation.

If the pressure drops are scaled with respect to the amount of water and the valve characteristic, the following equation can be obtained

$$\left(\frac{\Delta p_{RP}}{f_{RP}(z)^2} + \Delta p_{II} \right) \cdot q^2 = \left(\frac{\Delta p_{SP}}{f_{SP}(z)^2} + \Delta p_{III} \right) \cdot q_s^2 \quad (3.47)$$

It is possible to write $q_s = 1 - q$ since the water flow in the secondary circuit is constant. If this is inserted in Equation 3.47 it is possible to write the equation for relative amount of water through the regulate port as

$$q = \frac{1}{1 + \sqrt{A}} \quad (3.48)$$

3.6 Modeling the ventilation system

where

$$A = \frac{\frac{\Delta p_{RP}}{f_{RP}(z)^2} + \Delta p_{II}}{\Delta p_{SP}} f_{SP}(z)^2 + \Delta p_{III} \quad (3.49)$$

It is worth noting that both A and q are functions of the valve position z . This equation shows the relation between relative stem position and the relative amount of water from the valve. This is known as the valves operation characteristic. The relation is nonlinear, but the degree of nonlinearity depends on the valve characteristic. What this means is that the amount of water from the valve is not proportional to the valve opening.

There exists a number of different valve characteristic. The most common are listed below:

Linear:

$$f_{RP}(z) = \left(1 - \frac{1}{R_{RP}}\right) \cdot z + \frac{1}{R_{RP}} \quad (3.50)$$

Quadratic:

$$f_{RP}(z) = \left(1 - \frac{1}{R_{RP}}\right) \cdot z^2 + \frac{1}{R_{RP}} \quad (3.51)$$

Exponential:

$$f_{RP}(z) = R_{RP}^{z-1} \quad (3.52)$$

Complementary to exponential:

$$f_{RP}(z) = \frac{1}{R_{RP}} + 1 - R_{RP}^{-z} \quad (3.53)$$

In Equation 3.50-3.53 R_{RP} is the theoretical regulating relation ($R = \frac{Q_{max}}{Q_{min}}$). This describes the relationship between volumetric flow of water when $z = 1$

3. MODELING HVAC SYSTEMS

(Q_{max}) and a fictive leakage (Q_{min}) through the port when $z = 0$. Q_{min} is the smallest controllable amount possible. R_{RP} is only used to be able to describe the characteristics mathematically. Equation 3.50-3.53 is only valid for the regulate port of the tree way valve. In order to find the valve characteristics for the shunt port one needs to substitute z with $z - 1$ in the mentioned equations. In addition to this R_{RP} must be replaced with R_{SP} .

The heat balance for the water in the three way valve can be written as (see Figure 3.28):

$$\theta_{out} = (1 - q) \cdot \theta_{in} + q \cdot \theta_{hco} \quad (3.54)$$

$$\theta_{hci} = q \cdot \theta_{in} + (1 - q) \cdot \theta_{hco} \quad (3.55)$$

Due to the fact that the relative water amount q is a function of the valve position z , where $z = z(t)$, Equation 3.54 and 3.55 are nonlinear.

Transfer function

Nonlinear models are normally linearized so that they can be described by a transfer function and thus are easier to handle for simulation programs. Although this is normally the case, it will not be done for the shunt connection in this report since `Matlab` is capable of handling such a nonlinear system. The advantage of using a nonlinear model instead of the linearized transfer function is that it gives a better approximation of the actual system. It is also valid for every valve position and not only for values near the operating point. The nonlinear block diagram of the shunt connection (Equation 3.55, 3.48 and 3.49) can be seen in Figure 3.29. In this model an exponential valve characteristic for the regulate port and a linear valve characteristic for the shunt port has been chosen.

3.6.5 Modeling pipes and ducts

This section is a mathematical description of how the temperature of the ventilated air and the temperature of the water through the pipes connecting each component of the ventilation system change during transfer. The mathematical model for pipes and ducts is identical, differing only in physical measurements

3. MODELING HVAC SYSTEMS

of the pipe or duct and the type of fluid flowing through it. Thus only the pipe model will be deduced in this section. If modeling a duct there are two things that needs to be altered in the equations found below; the fluid properties needs to be changed to air instead of water and the dimensions of the pipe needs to be changed to the dimensions of the duct. This model takes into account the transportation delay through the pipe and it also describes the loss or gain of energy to the water due to the temperature surrounding the pipe.

Simplifications and assumptions

- Constant flow of water.
- Constant fluid temperature in radial direction.
- Constant temperature of the pipe material in radial direction.
- The surrounding temperature is constant in x-direction.

The first of these assumptions causes the coefficient of surface conductance due to convection to be approximately constant. In reality this value will vary slightly depending on the temperature. The reason for this is that the viscosity, and hence the Prandtl number, varies with the temperature. This approximation is however important in order to end up with a linear model and thus avoid linearization.

Mathematical model

Observing a thin slice of the pipe with infinitesimal length dx . The energy balance of the fluid inside the pipe can be written as shown in 3.56.

$$\begin{aligned} \frac{\partial(A_w \cdot dx \cdot \rho_w \cdot cp_w \cdot \theta_w)}{\partial t} &= \dot{Q}_w \cdot \rho_w \cdot cp_w \cdot \theta_w \\ -\dot{Q}_w \cdot \rho_w \cdot cp_w \cdot (\theta_w + \frac{\partial\theta_w}{\partial x} \cdot dx) - \pi \cdot D_i \cdot dx \cdot h_i \cdot (\theta_w - \theta_p) & \end{aligned} \quad (3.56)$$

where A_w is the cross-section area of the fluid in the pipe. For a circular pipe the equation for this is:

$$A_w = \frac{\pi}{4} \cdot D_i^2 \quad (3.57)$$

3.6 Modeling the ventilation system

Table 3.6: Symbols used in the pipe model

Symbol	Description	Unit	Remarks
L	Length of pipe/duct	m	
x	Linear measure	m	
D_i	Inner diameter of the pipe	m	
D_o	Outer diameter of the pipe	m	
\dot{Q}_w	Volume flow	$\frac{m^3}{s}$	
θ_w	Water temperature	$^{\circ}C$	$\theta_w(x, t)$
θ_{wi}	Water temperature into the pipe	$^{\circ}C$	$\theta_{wi} = \theta_w(0, t)$
θ_{wo}	Water temperature out of the pipe	$^{\circ}C$	$\theta_{wo} = \theta_w(L, t)$
θ_p	Pipe temperature	$^{\circ}C$	$\theta_p(x, t)$
θ_s	Surrounding temperature	$^{\circ}C$	$\theta_s(t) \neq \theta_s(x, t)$
ρ_w	Water density	$\frac{kg}{m^3}$	
ρ_p	Pipe density	$\frac{kg}{m^3}$	
cp_w	Specific heat capacity water	$\frac{J}{kg \cdot K}$	
cp_p	Specific heat capacity pipe	$\frac{J}{kg \cdot K}$	
h_i	Inner convective heat transmission	$\frac{W}{m^2 \cdot K}$	
U	U-value of the pipe including insulation and radiation	$\frac{W}{m^2 \cdot K}$	

Equation 3.57 inserted into Equation 3.56 gives the following equation:

$$\frac{\partial(\frac{\pi}{4} \cdot D_i^2 \cdot dx \cdot \rho_w \cdot cp_w \cdot \theta_w)}{\partial t} = \dot{Q}_w \cdot \rho_w \cdot cp_w \cdot \theta_w - \dot{Q}_w \cdot \rho_w \cdot cp_w \cdot (\theta_w + \frac{\partial \theta_w}{\partial x} \cdot dx) - \pi \cdot D_i \cdot dx \cdot h_i \cdot (\theta_w - \theta_p) \quad (3.58)$$

Rearranging this equation gives

$$T_1 \cdot \frac{\partial \theta_w}{\partial t} + T_1 \cdot v \cdot \frac{\partial \theta_w}{\partial x} + \theta_w = \theta_p \quad (3.59)$$

where the speed of the fluid in x-direction, v , is

$$v = \frac{4 \cdot \dot{Q}_w}{\pi \cdot D_w^2} \quad (3.60)$$

and T_1 is defined as

3. MODELING HVAC SYSTEMS

$$T_1 = \frac{D_i \cdot \rho_w \cdot cp_w}{4 \cdot h_i} \quad (3.61)$$

Energy balance for the pipe wall can be written as

$$\frac{\partial(A_p \cdot dx \cdot \rho_p \cdot cp_p \cdot \theta_p)}{\partial t} = \pi \cdot D_i \cdot dx \cdot h_i \cdot (\theta_w - \theta_p) - \pi \cdot D_o \cdot dx \cdot U \cdot (\theta_p - \theta_s) \quad (3.62)$$

where A_p is the cross-section area of the pipe wall. For a circular pipe the equation for this is:

$$A_p = \frac{\pi}{4} \cdot (D_o^2 - D_i^2) \quad (3.63)$$

An approximation for the U-value of a circular pipe can be calculated using the following equation:

$$U = \frac{1}{\frac{D_o}{2 \cdot \lambda_{insulation}} \cdot \ln\left(\frac{D_o + 2 \cdot \delta}{D_o}\right) + \frac{D_o}{D_o + 2 \cdot \delta} \cdot \frac{1}{h_o}} \quad (3.64)$$

where $\lambda_{insulation}$ is the coefficient of thermal conductivity for the insulation material. δ is the insulation thickness. The heat transmission coefficient on the pipe, h_o , includes both convection and radiation. The inner convective heat transmission (from water to pipe wall) is included in the first term of Equation 3.62. Hence it is not included in the U-value.

As with Equation 3.58 insertion of A_p into Equation 3.62 and rearranging gives

$$T_2 \cdot \frac{\partial \theta_p}{\partial t} + \theta_p = K_1 \cdot \theta_w + K_2 \cdot \theta_s \quad (3.65)$$

where

$$K_1 = \frac{h_i \cdot D_i}{h_i \cdot D_i + U \cdot D_o} \quad (3.66)$$

3.6 Modeling the ventilation system

$$K_2 = \frac{U \cdot D_o}{h_i \cdot D_i + U \cdot D_o} \quad (3.67)$$

$$K_1 + K_2 = 1 \quad (3.68)$$

$$T_2 = \frac{(D_o^2 - D_i^2) \cdot \rho_p \cdot c p_p}{4 \cdot (h_i \cdot D_i + U \cdot D_o)} \quad (3.69)$$

Transfer function

Laplace transformation of Equation 3.59 gives the following result:

$$T_1 \cdot s \cdot \theta_w(x, s) + T_1 \cdot v \cdot \frac{\partial \theta_w(x, s)}{\partial x} + \theta_w(x, s) = \theta_p(x, s) \quad (3.70)$$

While Laplace transformation of Equation 3.65 gives

$$T_2 \cdot s \cdot \theta_p(x, s) + \theta_p(x, s) = K_1 \cdot \theta_w(x, s) + K_2 \cdot \theta_s(s) \quad (3.71)$$

Equation 3.71 can be rearranged into the following expression for θ_p :

$$\theta_p(x, s) = \frac{K_1}{T_2 \cdot s + 1} \cdot \theta_w(x, s) + \frac{K_2}{T_2 \cdot s + 1} \cdot \theta_s(s) \quad (3.72)$$

Putting this equation into Equation 3.70 and rearranging results in the following equation

$$\frac{\partial \theta_w(x, s)}{\partial x} + \alpha(s) \cdot \theta_w(x, s) = \beta(s) \cdot \theta_s(s) \quad (3.73)$$

where

3. MODELING HVAC SYSTEMS

$$\alpha(s) = \frac{1}{T_1 \cdot v} \cdot \left(T_1 \cdot s + 1 - \frac{K_1}{T_2 \cdot s + 1} \right) \quad (3.74)$$

$$\beta(s) = \frac{1}{T_1 \cdot v} \cdot \frac{K_2}{T_2 \cdot s + 1} \quad (3.75)$$

Solving the first order differential equation above (Equation 3.73) and inserting the following limits: $\theta_w(0, s) = \theta_{wi}(s)$ and $\theta_w(L, s) = \theta_{wo}(s)$ gives:

$$\theta_{wo}(s) = \theta_{wi}(s) \cdot e^{-\alpha(s) \cdot L} + \frac{\beta(s)}{\alpha(s)} \cdot (1 - e^{-\alpha(s) \cdot L}) \cdot \theta_s(s) \quad (3.76)$$

From this equation we have two transfer functions. One from the water temperature entering the pipe to the water temperature exiting the pipe:

$$H_p(s) = \frac{\theta_{wo}(s)}{\theta_{wi}(s)} = e^{-\alpha(s) \cdot L} \quad (3.77)$$

And one from the surrounding temperature to the water temperature out of the pipe:

$$H_s(s) = \frac{\theta_{wo}(s)}{\theta_s(s)} = \frac{\beta(s)}{\alpha(s)} \cdot (1 - e^{-\alpha(s) \cdot L}) \quad (3.78)$$

where $\alpha(s)$ and $\beta(s)$ is as given in Equation 3.74 and 3.75.

It is possible to simplify Equation 3.77 and 3.78. In order to do this these equations needs to be rewritten in such a way that it is possible to make use of Padé approximation. Starting with Equation 3.77. This equation can be written as

$$H_p(s) = e^{-\tau_p \cdot s} \cdot e^{-m_p \cdot \frac{T_2 \cdot s}{T_2 \cdot s + 1}} \cdot e^{-n_p} \quad (3.79)$$

where

3.6 Modeling the ventilation system

$$\tau_p = \frac{L}{v} = \frac{\pi \cdot D_i^2 \cdot L}{4 \cdot \dot{Q}_w} \quad (3.80)$$

$$m_p = \frac{L}{v} \cdot \frac{K_1}{T_1} = \frac{h_i^2 \cdot \pi \cdot D_i \cdot L}{\dot{Q}_w \cdot \rho_w \cdot c_{p_w} \cdot \left(h_i + U \cdot \frac{D_o}{D_i} \right)} \quad (3.81)$$

$$n_p = \frac{L}{v} \cdot \frac{K_2}{T_1} = \frac{h_i \cdot U \cdot \pi \cdot D_o \cdot L}{\dot{Q}_w \cdot \rho_w \cdot c_{p_w} \cdot \left(h_i + U \cdot \frac{D_o}{D_i} \right)} \quad (3.82)$$

τ_p can be regarded as the transportation delay of water through the pipe. Using Padè approximation it is possible to write the non-rational transfer function, Equation 3.79, as

$$H_p(s) \approx K_p \cdot \frac{1 + T_{p1} \cdot s}{1 + T_{p2} \cdot s} \cdot e^{-\tau_p \cdot s} \quad (3.83)$$

where

$$K_p = e^{-n_p} \quad (3.84)$$

$$T_{p1} = e^{\frac{-m_p}{2}} \cdot T_2 \quad (3.85)$$

$$T_{p2} = e^{\frac{m_p}{2}} \cdot T_2 \quad (3.86)$$

T_2 and m_p is defined earlier in Equation 3.69 and 3.81. K_p can be regarded as the gain through the pipe. This value is less than 1, hence the water in the pipe experience some loss of heat during transfer. Setting the U-value of Equation 3.81 and 3.82 to zero leads to a system without heat loss.

3. MODELING HVAC SYSTEMS

Equation 3.78 can be written as

$$H_s(s) = \frac{\beta(s)}{\alpha(s)} \cdot \left(1 - e^{-\tau_p \cdot s} \cdot e^{-m_p \cdot \frac{T_2 \cdot s}{T_2 \cdot s + 1}} \cdot e^{-n_p} \right) \quad (3.87)$$

where T_2 is given by Equation 3.69, m_p by Equation 3.81, n_p by Equation 3.82 and τ_p is given by Equation 3.80. The transfer function (Equation 3.87) can using Padè approximation be written as:

$$H_s(s) \approx \frac{\beta(s)}{\alpha(s)} \cdot \left(1 - K_p \cdot \frac{1 + T_{p1} \cdot s}{1 + T_{p2} \cdot s} \cdot e^{-\tau_p \cdot s} \right) \quad (3.88)$$

The relation between α and β can, using Equation 3.68, be written as:

$$\frac{\beta}{\alpha} = \frac{K_2}{T_1 \cdot T_2 \cdot s^2 + (T_1 + T_2) \cdot s + K_2} \quad (3.89)$$

This means that the transfer function given in Equation 3.88 is of third order. A reasonable assumption is to assume that the variation in the surrounding temperature is of low frequency (i.e. θ_s varies slowly with a frequency lower than 1) it is then possible to write the transfer function as

$$H_s(s) \approx \frac{1 + T_{p2} \cdot s - K_p \cdot (1 + t_{p1} \cdot s) \cdot e^{-\tau_p \cdot s}}{T_o \cdot s + 1} \quad (3.90)$$

where

$$T_o \approx \frac{T_1 + T_2 + K_2 \cdot T_{p2}}{K_2} \quad (3.91)$$

This is possible since higher order terms (s^3 and s^2) will be small compared to the first order terms (s). Another simplification is possible by assuming that there is a constant temperature surrounding the pipes/ducts. Using the finite value theorem (see Balchen et al. (2003)) on Equation 3.88 gives the following static transfer function:

3.6 Modeling the ventilation system

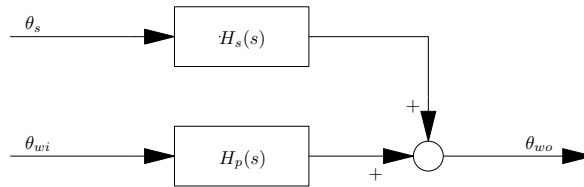


Figure 3.30: Block diagram of the pipe model

$$H_s(s) = 1 - K_p \quad (3.92)$$

Finally Equation 3.83 and Equation 3.92 can be implemented in **Simulink** as the pipe model. The block diagram of these equations can be seen in Figure 3.30.

3.6.6 Connecting each element of the ventilation system

When every component of a ventilation system has been modeled each element can be connected to each other. This connected system makes up the complete ventilation system of the building. Figure 3.31 shows how the ventilation system in this report is represented in **Simulink**. As can be seen from this figure once one has established the transfer functions for every component it is a fairly easy task to connect them to one another. In Figure 3.31 there is also a controller connected to the ventilation system. Chapter 5 gives an introduction to controllers and states the function of the controller seen in this figure.

3.6.7 Extending the model to enable varying air volume

The model represented above is not valid for varying air volumes. Attempts have been made to extend the model to make it valid also for varying air volumes. The following section is a summary of these attempts and its result.

The models of the components listed below all depends on a constant air volume:

- The fan
- The heating coil
- The heat recovery unit
- The pipes and ducts

3. MODELING HVAC SYSTEMS

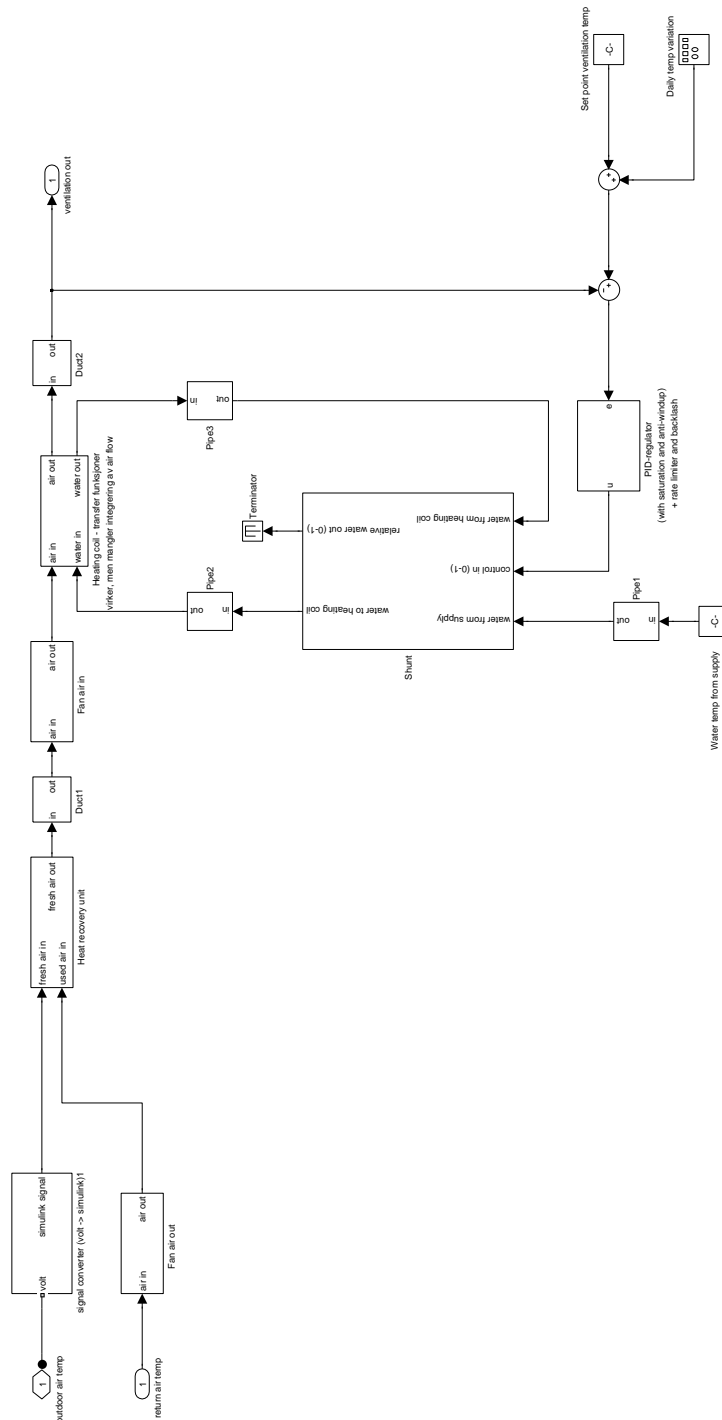


Figure 3.31: Block diagram of the ventilation system of Saltstraumen elementary school

3.6 Modeling the ventilation system

Thus each of these components needs to be remodeled in order for the system to be valid for varying airflow. The heating coil was chosen as the first component to remodel because of its importance in the system.

Remodeling the heating coil

The differential equations for the heating coil, Equation 3.26 and 3.28, given in Section 3.6.3 are restated here for the convenience of the reader.

$$m_{hc} \cdot cp_{hc} \cdot \frac{\partial \theta_{wo}}{\partial t} = \dot{m}_w \cdot cp_w \cdot (\theta_{wi} - \theta_{wo}) - (UA)_{hc} \cdot (\theta_{wo} - \theta_{ai}) \quad (3.93)$$

$$m_{air} \cdot cp_{air} \cdot (\theta_{ao} - \theta_{ai}) = (UA)_{hc} \cdot (\theta_{wo} - \theta_{ai}) \quad (3.94)$$

With varying air flow the assumption of a constant air flow does not hold anymore. This means that \dot{m}_{air} (included in $(UA)_{hc}$) is varying over time and should be written as $\dot{m}_{air}(t)$. Note that \dot{m}_w is still constant due to the design of the shunt connection. Since $\dot{m}_{air}(t)$ is now varying with time the differential equation for air, Equation 3.94, is not valid anymore and needs to be rewritten into the same form as Equation 3.93. This leads to the following differential equation

$$m_{hc} \cdot cp_{hc} \cdot \frac{\partial \theta_{ao}}{\partial t} = \dot{m}_{air}(t) \cdot cp_{air} \cdot (\theta_{ai} - \theta_{ao}) + (UA)_{hc} \cdot (\theta_{wo} - \theta_{ai}) \quad (3.95)$$

Equation 3.93 and 3.95 describes the change of temperatures in the water and the air during its transit across the heating coil. These equations are in contradiction to the one found in Section 3.6.3 valid for varying air velocity. It should be noted that since $\dot{m}_{air}(t)$ varies with time it is not possible to represent the new heating coil model using transfer functions. Thus the equations must be implemented directly in `Matlab`. The shunt connection was also implemented using this approach.

Next the unknown value UA_{hc} needs to be found. There are several ways to define this value. In Section 3.6.3 UA_{hc} is defined using relative water cooling, ϵ_w , and relative air heating, ϵ_{air} , as

$$(UA)_{hc} = \frac{\epsilon_{air}}{1 - \epsilon_w} \cdot \dot{m}_{air} \cdot cp_{air} \quad (3.96)$$

3. MODELING HVAC SYSTEMS

or

$$(UA)_{hc} = \frac{\epsilon_w}{1 - \epsilon_w} \cdot \dot{m}_w \cdot cp_w \quad (3.97)$$

It should be noted that both ϵ_{air} and ϵ_w are constant only as long as both the air flow and the water flow through the heating coil remains constant. This is not the case anymore since the air flow is set to vary. However it is expected that they will only vary with a small fraction due to the varying air volume through the heating coil and hence they are still considered to be constant although this is not entirely correct. Thus the only variable from these equations are \dot{m}_{air} . Equation 3.96 includes this and hence this Equation should be used to model the heat transfer between the air and the water. Combining this with Equation 3.93 and Equation 3.95 results in the **Simulink** model seen in Figure 3.32.

The reader is advised to read Section 5.2.2 in order to better understand the results presented below as it is found using the controller presented in that section. Simulating the ventilation system using this heating coil and otherwise holding the parameters equal to the one in Example 5.3 results in the temperature development seen in Figure 3.33. This temperature development is clearly not as desired or physical possible as it barely deviates from the air temperature into the heating coil (see Figure 6.6) even though the valve opening is 100%. This result should actually be the same as the one found in Section 5.2.2 Example 5.3 had the heating coil functioned properly. Clearly it is not and hence one can make the conclusion that the heating coil from this section does not function as intended.

It should be noted that another and common way of defining the heat transmission, defined as $Q = (UA)_{hc}(\theta_{wo} - \theta_{ai})$ above, is to define it as $(UA)_{hc}LMTD$ where $LMTD$ is the log mean temperature difference. This is defined as

$$LMTD = \frac{\Delta T_{out} - \Delta T_{in}}{\ln\left(\frac{\Delta T_{out}}{\Delta T_{in}}\right)} \quad (3.98)$$

Using this definition for the heat transfer does not lead to any better results than with the use of the old definition. After several attempts of modeling a heating coil valid for varying airflow by introducing many different models, of which the one presented above is the most promising, it was decided due to time constraints to not implement a model that includes this property.

3.6 Modeling the ventilation system

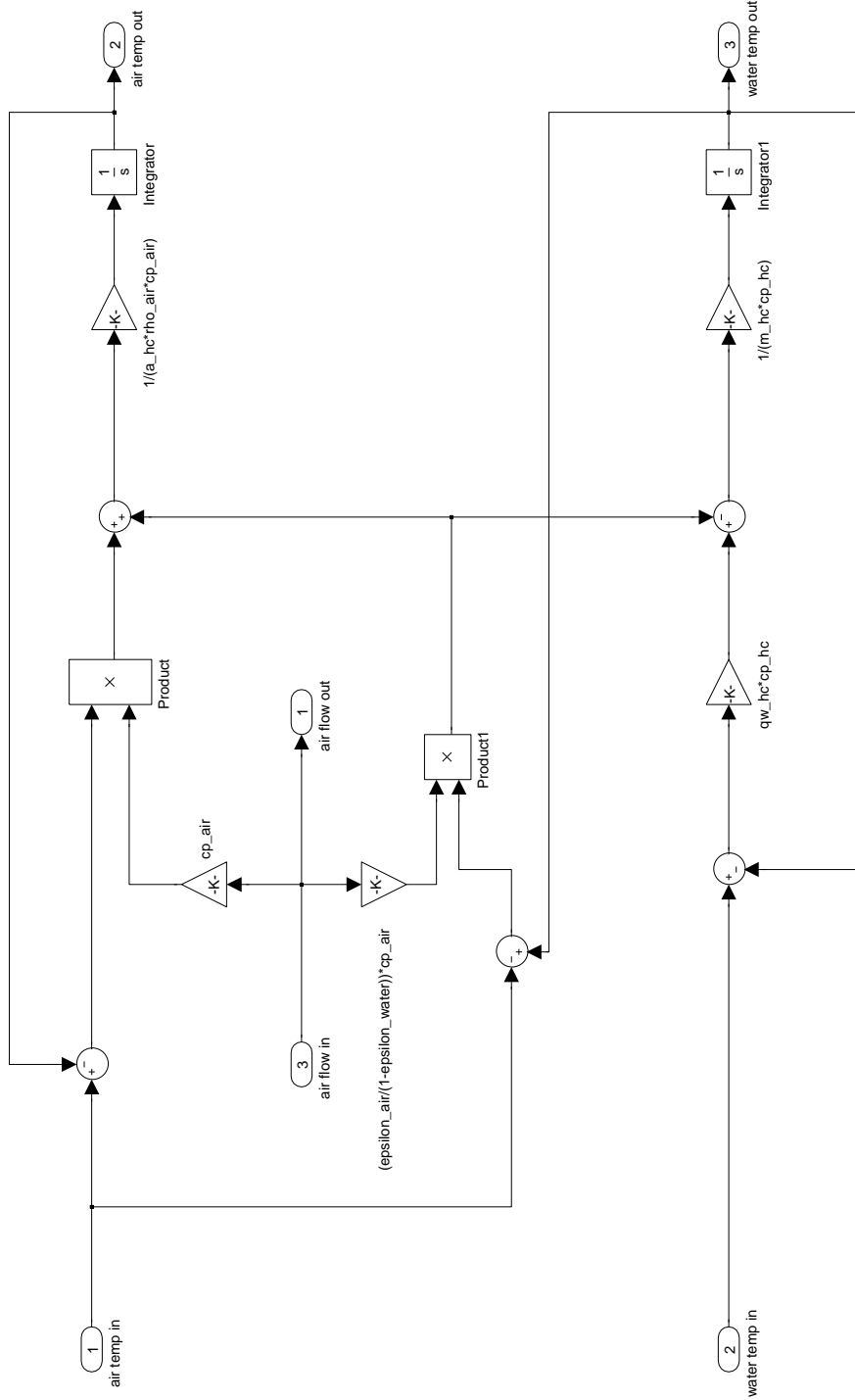


Figure 3.32: Simulink model of the heating coil with variable air flow.

3. MODELING HVAC SYSTEMS

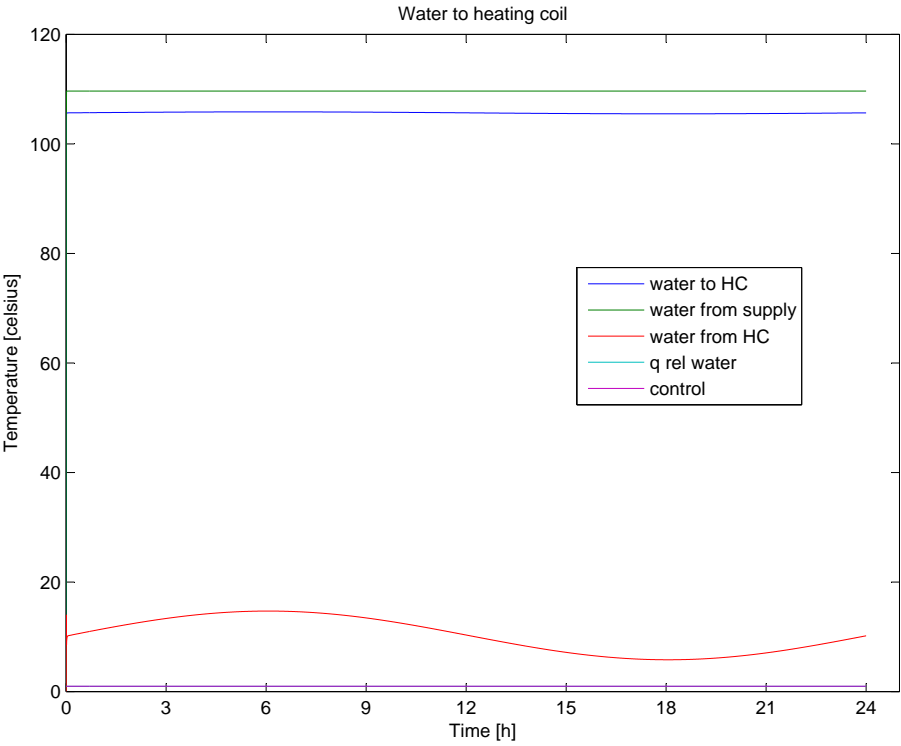


Figure 3.33: Temperature development of the ventilation, water into the heating coil and water out of the heating coil using the heating coil with variable airflow. Also the control signal can be seen.

4

System identification

In Section 5.3 the MPC control scheme is used to control the indoor temperature. In order for this scheme to function the model presented in the previous chapter needs to be discretized. It turns out that this is not yet possible for models made in `Simscape`. Thus a model that can be discretized and that behaves similar to the model from Chapter 3 needs to be found. This is possible using system identification on the existing model since the output, y , and the input, u , is known. Also the disturbance from the outdoor climate, the ventilation system, and the adjacent rooms, w , is known.

System identification can be divided into the following three basic steps:

1. Creation of a data set to be used for model identification.
2. Choosing a model structure, i.e. a set of candidate models.
3. Find the model within the set of models that best represents the system using a selection rule, e.g. the least squares selection rule or the instrumental-variable method.

Although these three steps seems to be fairly straightforward care must be taken in each of these steps in order to end up with a model capable of representing the real system in a good way. An example may be the data set from step one. This data set needs to possess certain characteristics in order to contain enough information for the identification procedure in step three to find a good model from the set of models. In Section 4.1 the characteristics and the choice of the input design, i.e. the data set, is elaborated. Section 4.2 elaborates the choice of model structure. The third step, choosing the best model within the set of models, is handled almost automatically by the System identification toolbox in `Matlab`. Almost automatically means that the user must decide whether to

4. SYSTEM IDENTIFICATION

Table 4.1: Inputs, outputs and measured disturbances of the building

Inputs (MV)	Measured disturbances (DV)	Outputs (CV)
Radiator gain	Outdoor temperature Ventilation temperature Temperature in adjacent rooms	Room temperature

use the least squares selection rule or the instrumental-variable method or any other estimation method, the rest of the estimation procedure is handled by the toolbox.

A basic rule-of-thumb is that model fitting becomes harder with more outputs and simpler with more inputs. This is due to the complexity that follows from a MIMO-system, i.e. the coupling between the different inputs and outputs (Ljung 2006). Since the above mentioned disturbances are known it is possible to regard them as extra inputs to the system. According to the rule-of-thumb just mentioned this makes the system identification problem easier. The system to be identified is clearly a MIMO system since it consists of several rooms with one output each, namely the indoor temperature, and several inputs, namely the radiator gain, the ventilation temperature, the outdoor temperature and the temperature of the adjacent rooms. Since system identification for systems with several outputs are much more difficult to solve than systems with only one output it is best to avoid this if possible. A simple but effective way to achieve this is to divide the identification problem into several identification problems that are easier to solve, e.g. has fewer outputs. This can easily be done by splitting the problem of identifying the entire building into smaller problems consisting of identifying one room at a time. When each of these systems have been identified they can be connected to make up the entire system.

Table 4.1 gives an overview of the inputs, measured disturbances and outputs of the system to be identified. That is the MV, DV and CV of the system. Note that the measured disturbances are also considered to be inputs and there are no unmeasured disturbances. From a practical point of view this might not be realistic since there normally are disturbances entering the system due to measurement errors. However this has not been taken into account during the system identification procedure.

4.1 Input design

Now that the input, output and disturbance parameters has been defined, a input signal that can be used for the system identification needs to be constructed. This input signal must, in order to be able to identify a model that acts the same way as the real system, be informative. Informative means that the data set (i.e. the input signal) allows discrimination between any two different models in a set of models (Ljung 2006). This means that the input should be persistently exciting (p.e.) of a certain order. I.e. that it contains sufficiently many distinct frequencies (Ljung 2006). Thus there exists many different signals that are sufficient and every one of those will result in a model that behaves similar to the real system. Hence the choice of input signal plays an important role in the validity of the obtained model and it should not be chosen arbitrarily, but in such a way that the identified model holds the same features as the real system. The most important characteristics the input signal must exhibit are listed below.

- The input to the system under test should excite the system. This excitation is dependent on the spectrum of the input signal and not the actual waveform. Hence the frequency range of the input should be chosen so that most of its energy is within the frequency ranges important to the system (see Section 4.1.1 for more details).
- Input value levels should exercise the entire range of variation.
- The input signal should deliver as much input power into the system as possible. This property is defined by the crest factor (see Section 4.1.2). The smaller the crest factor is, the larger the total energy into the system is.
- For MIMO systems the different input signals must be uncorrelated with each-other in order to allow the system identification method to discriminate how the input signals affects the system response.

4.1.1 Persistently exciting

An input signal is said to be p.e. of order n if the spectrum, $\Phi_u(\omega)$ is different from zero on at least n points in the interval $-\pi < \omega \leq \pi$. For an open loop experiment to be informative it has to be p.e. of an order equal to the number of model parameters to be found (Ljung 2006). A brief example of this can be seen

4. SYSTEM IDENTIFICATION

in the Example 4.1

Example 4.1 Persistently exciting

Given the following model structure

$$G(q, \theta) = \frac{B(q, \theta)}{F(q, \theta)} = \frac{q^{-n_k}(b_1 + b_2q^{-1} + \dots + b_{n_b}q^{-n_b+1})}{1 + f_1q^{-1} + \dots + f_{n_f}q^{-n_f}} \quad (4.1)$$

An open loop experiment with an input that is p.e. of order $n_b + n_f$ is informative enough to identify this model.

4.1.2 Crest factor

The crest factor is according to Ljung (2006) defined as follows for a zero mean signal

$$C_r^2 = \frac{\max_t u^2(t)}{\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N u^2(t)} \quad (4.2)$$

where u is the signal amplitude.

The lower bound of C_r is 1, this is achieved with a binary, symmetric signal. Ergo a binary signal has an advantage over other input signals when it comes to the amount of power sent into the system. However the reader should be aware of the fact that a binary signal does not have the ability to detect non-linearities in the system. However, using the existing knowledge of the building model (see Section 3.3), it is safe to conclude that the model to be identified is a linear system. As a small crest factor means a high amount of energy delivered into the system it also means an enhanced signal-to-noise ratio. This is clearly preferable.

The following is a list of commonly chosen input signals for system identification:

- Filtered Gaussian white noise.

- Random binary signal.
- Pseudo-random binary signal (PRBS).
- Multi-sines.
- Chirp signals or swept sinusoids.

Although each of the above mentioned signals commonly are used for system identification one of them might be better to use than the others. This depends on the system to be identified. In Table 4.2 some of the pros and cons for each of these signals are mentioned. For further details regarding these signals see Ljung (2006).

4.1.3 Choosing input signals

Taking into account the above mentioned properties of which a signal used for system identification should exhibit, including the pros and cons of common system identification signals, signals suitable for each of the inputs to the model identification will now be constructed. Since the system to be identified is an already existing model in `Simulink` the variables of the process that normally cannot be arbitrarily chosen, such as the outdoor air temperature, can in this case be chosen so that it simplifies the identification of the process. Due to this each of the input signals can be chosen to be of such a nature that it makes the system identification easier. This means that the input signals used for system identification might have an unrealistic behavior from a physical point of view, and instead is constructed in such a way that it excites the system and hence makes the system identification process easier. Although this is possible it should be avoided to make the system identification as close to real life experiments as possible.

For real systems it is not possible to arbitrarily select each of the excitation signals (e.g. the outdoor temperature). Thus the excitation signals in this report are chosen in such a way that it might resemble a real experiment. The drawback of this is that some of the excitation signals are overrun by other and more powerful excitation signals. Also it might be impossible to ensure that the excitation signals are uncorrelated of each other. In this report this is very much so. E.g. the excitation signal from the radiator greatly affects the indoor temperature while the outdoor temperature and the ventilation temperature has little effect on the indoor temperature compared to the effect from the radiator. The temperature effect on the room caused by the adjoining rooms are so small that even with an excitation signal with relatively high amplitude it has close to no effect on

4. SYSTEM IDENTIFICATION

Table 4.2: Excitation signals - pros and cons (see Ljung (2006) for further details)

Signal type	Pros	Cons
Filtered Gaussian white noise	Simple choice. Almost any signal spectrum can be achieved. Transient effects can be eliminated.	Not necessarily good second order properties.
Random binary signal	Ideal crest factor.	Not full control of shaping the spectrum. Not necessarily good second order properties.
Pseudo-random binary signal	Ideal crest factor. Second order properties secured. The covariance matrix has a special pattern.	Not straightforward to generate mutually uncorrelated sequences. Limited choice of experiment length.
Multi-sines	Easy to place the signal power at desired frequencies.	Care must be taken when combining sinusoids to control the crest factor.
Chirp signals	Low crest factor ($\sqrt{2}$)	Power contributions outside the frequency band.

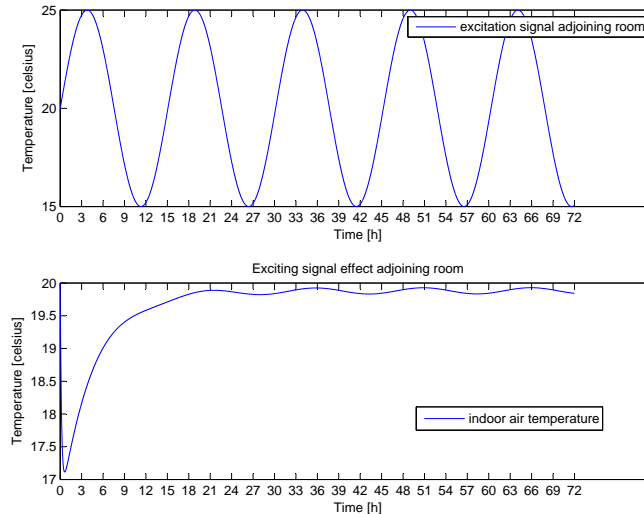


Figure 4.1: System identification - effect from adjoining rooms

the room temperature. An example of this can be seen in Figure 4.1. It can also be assumed that the temperatures of the adjoining rooms are not differing more than a couple of degrees from the temperature in the room to be identified. Thus this part of the original model has been neglected and no model has been identified for the connection between the rooms. This means that the rooms can be thought of as separate systems during the identification procedure.

Excitation signal - radiator

The radiator temperature gain is expected to vary between the current room temperature and 70°C since this is the minimum and maximum temperature for that component. This is the only manipulated variable in the room and hence the signal used for system identification can be chosen to vary arbitrarily within its limits. Note that the ventilation temperature also is a possible manipulated variable but is not used so in this report. It is instead seen upon as a measured disturbance. The excitation signal for the radiator is thus modeled as filtered Gaussian white noise that varies between the maximum and minimum values. Using this kind of signal and by the proper choice of filter almost any signal spectrum can be achieved (Ljung 2006). The radiator temperature can be assumed to be relatively slow varying. Thus a band pass filter is chosen to filter out high frequencies and very low frequencies. Finally the signal has been scaled in order to achieve the desired performance. Figure 4.2 shows the building blocks used to generate the excitation signal for the radiator in **Simulink** while Figure 4.6 shows the excitation signal itself, i.e. the output from Figure 4.2.

4. SYSTEM IDENTIFICATION

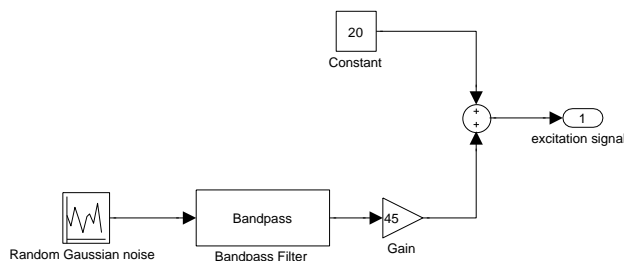


Figure 4.2: System identification - model of the radiator excitation signal

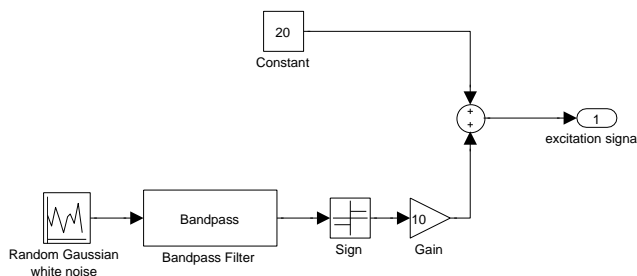


Figure 4.3: System identification - model of the ventilation excitation signal

Excitation signal - ventilation

The temperature of the ventilation can be assumed to be constant in normal operating conditions. But in order to find its effect on the room to be identified it cannot be fixed during the identification process. In other words, a fixed temperature for the ventilation is not p.e. enough. To deal with this problem the temperature of the ventilation have been modeled as a random binary signal (a signal which assumes only two values). This has been done using white, zero mean Gaussian noise which has been filtered through an appropriate linear band-pass filter. Note that another frequency band than for the other signals should be used to ensure that the different identification signals are uncorrelated of each other. Then the sign of the output from the filter has been taken, and finally this has been amplified in order for the signal to make a contribution on the output, i.e. the indoor temperature. This is one of the most common excitation signals used for system identification and the **Simulink** model used to create such a signal can be seen in Figure 4.3. One reason for this is that its crest factor is ideal, i.e. 1. Unfortunately the signal spectrum does change due to the sign operation, however it has a high signal-to-noise ratio in low frequency band as long as its period is set high enough (this is done in the filtering). Due to this it is a signal suited for use in slow physical systems and processes such as this. The generated excitation signal for the ventilation can be seen in Figure 4.6.

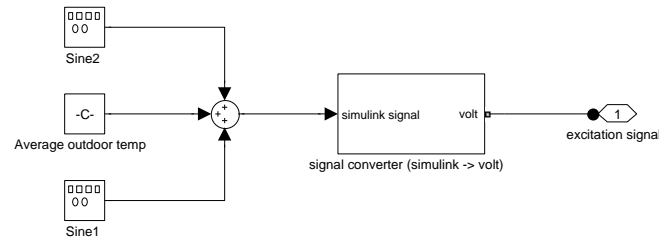


Figure 4.4: System identification - model of the outdoor excitation signal

Excitation signal - outdoor temperature

The outdoor temperature is modeled as a slowly varying sine wave where the temperature is high during the day and low during the night. A sinusoid gives rise to two spectral lines (Ljung 2006). This means that a sine wave is persistently exciting of order two. Another sine wave on top of an existing one and with another frequency give rise to two additional spectral lines. Thus such a signal would be persistently exciting of order 4. Yet another sine wave with yet another frequency would result in a signal that would be persistently exciting of order 6 and so on. If one of the frequencies are equal to 0 or π the order would drop by 1. This is due to the fact that these frequencies only results in one spectral line.

For the identification process a sine wave with a different frequency will be added to the existing sine wave that resembles the outdoor temperature. Since the generated excitation signal is built up by two sine waves it is persistently exciting of order 4. This means that it is possible to identify up to 4 parameters using this input signal. As will be seen later the model we are identifying is of first order with a time delay, meaning that there are 3 parameters to identify (K_p , T_p and T_d). This proves that the excitation signal is informative enough. The generated excitation signal for the ventilation can be seen in Figure 4.6 while the **Simulink** model that generates this signal is shown in Figure 4.4.

Excitation signal - adjoining rooms

As previously shown the adjoining rooms have very little to no effect on the room to be identified, assuming that the adjoining room temperature is close to the same as the temperature of the room to be identified. This effect is thus neglected in this report. However, in other buildings this might not be the case. There might for instance be adjoining rooms which differ significantly in temperature (e.g. one of the rooms is used as a refrigeration room). For this reason a possible excitation signal that can be used for the identification of the adjacent rooms effect on the

4. SYSTEM IDENTIFICATION

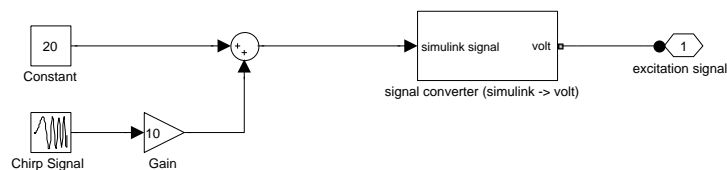


Figure 4.5: System identification - model of the adjoining room excitation signal

room is included here.

Usually the temperature of the adjoining rooms is assumed to be fairly constant. As for the ventilation, this cannot be so during the identification process and hence it has been altered during this process in order to achieve a signal that is exciting enough to capture the effects it has on the room. In addition to this it needs to be uncorrelated with the other signals and hence a chirp signal has been chosen. A chirp signal is a sine wave where the frequency continuously changes over time. Since the system to be identified is very slow the chirp signal is chosen with an initial frequency of 0.00001 Hz. The frequency at the end of the chirp signal is chosen to be 0.0001 Hz. The generated excitation signal for the adjoining room(s) can be seen in Figure 4.6 while the Simulink model that generates this signal is shown in Figure 4.5.

4.2 Selecting a model structure

When identifying a system and presenting it as a mathematical model there are several model structures to choose from. The model structures can be divided into two main categories - linear and nonlinear. Since the system in this report is clearly linear and time-invariant, only linear model structures will be presented. This section is an introduction to some of the most common linear model structures and their pros and cons. Every model structure presented in this section are parametric, meaning that the models are described in terms of differential equations and transfer functions. Parametric model structures are also known as black-box models (Instruments n.d.). If however parts of the model is known it is called a grey-box model. An example of this might be that there are time delays in the model which are known based on a priori knowledge of the system. Such an a priori knowledge of the system should be exploited as it makes the system easier to identify.

According to Ljung (2006) a linear system can be described using the general-linear polynomial model or the general-linear model as seen in Equation 4.3.

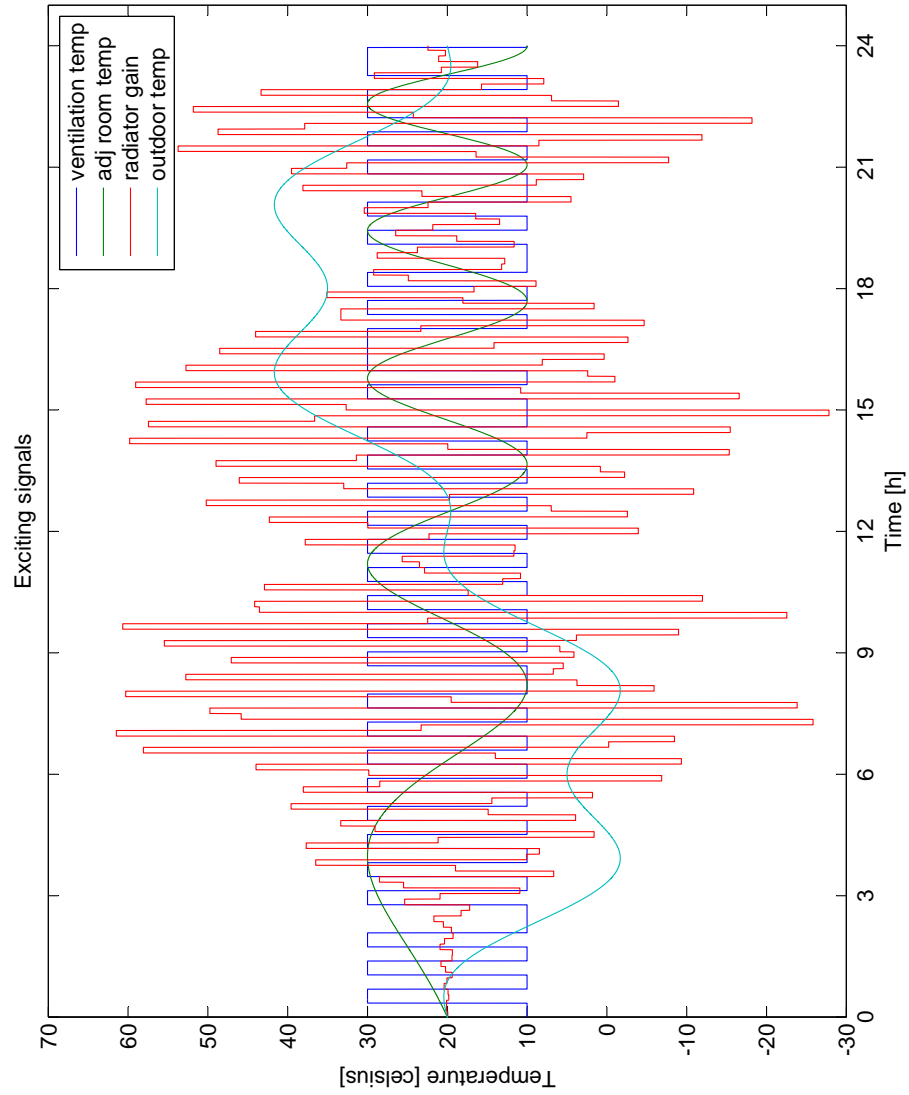


Figure 4.6: System identification - excitation signals

4. SYSTEM IDENTIFICATION

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (4.3)$$

where $u(t)$ is the input and $y(t)$ is the output of the system. $e(t)$ is the disturbance of the system. $G(q, \theta)$ is the transfer function of the deterministic part of the system while $H(q, \theta)$ is the transfer function of the stochastic part of the system.

The one-step-ahead prediction $\hat{y}(t|\theta)$ for Equation 4.3 is according to Ljung (2006) defined as:

$$\hat{y}(t|\theta) = H^{-1}(q, \theta)G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)]y(t) \quad (4.4)$$

The following description of different model structures are mainly based on theory found in Ljung (2006).

4.2.1 ARX-model

The simplest input-output relationship is obtained by describing it as a linear difference equation on the following form:

$$\begin{aligned} & y(t) + a_1y(t-1) + a_2y(t-2) + \dots + a_{n_a}y(t-n_a) \\ & = b_1u(t-1) + b_2u(t-2) + \dots + b_{n_b}u(t-n_b) + e(t) \end{aligned} \quad (4.5)$$

In this case the adjustable parameters are

$$\theta = [a_1 \quad a_2 \cdots a_{n_a} \quad b_1 \quad b_2 \cdots b_{n_b}]^T \quad (4.6)$$

These are the parameters to be estimated so that the system model behaves similar to the measurements from the real system. Equation 4.5 corresponds to Equation 4.3 with

$$G(q, \theta) = \frac{B(q)}{A(q)} \quad (4.7)$$

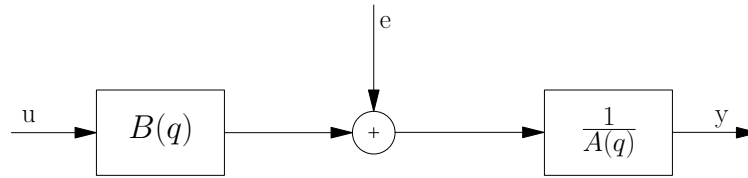


Figure 4.7: ARX model structure

and

$$H(q, \theta) = \frac{1}{A(q)} \quad (4.8)$$

where

$$A(q) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad (4.9)$$

and

$$B(q) = b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad (4.10)$$

Equation 4.5 is called an ARX-model, where AR refers to the autoregressive part $A(q)y(t)$ and X to the extra input $B(q)u(t)$. Figure 4.7 shows the signal flow of the ARX model. From this figure it can be seen that the disturbance is a part of the system dynamics, i.e. the transfer function of the deterministic part of the system, $G(q, \theta)$ and the transfer function of the stochastic part of the system, $H(q, \theta)$ have the same set of poles. This is not natural from a physical point of view, however a good signal-to-noise ratio reduces this disadvantage. Even with the disadvantage just mentioned the ARX-model has a very important property that makes it a prime choice in many applications: The predictor defines a linear regression (Ljung 2006). This will now be explained Inserting Equation 4.7 and 4.8 into Equation 4.4 gives

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t) \quad (4.11)$$

Now, introducing the vector

4. SYSTEM IDENTIFICATION

$$\varphi = [-y(t-1) \cdots -y(t-n_a) \quad u(t-1) \cdots u(t-n_b)]^T \quad (4.12)$$

makes it possible to write Equation 4.11 as

$$\hat{y}(t|\theta) = \theta^T \varphi(t) = \varphi^T(t)\theta \quad (4.13)$$

The predictor is a scalar product between the known data vector $\varphi(t)$ and the parameter vector θ . Such a model is called a linear regression which is stated above as one of the ARX model structure benefits. It exists powerful and simple estimation methods to determine θ in such cases.

4.2.2 ARMAX-model

The disadvantage with the simple ARX-model is the lack of adequate freedom in describing the properties of the disturbance term. With the ARMAX-model, flexibility in the disturbance model is achieved by describing the equation error as a moving average (MA) of white noise. This results in the following model.

$$y(t) + a_1y(t-1) + a_2y(t-2) + \cdots + a_{n_a}y(t-n_a) = b_1u(t-1) + b_2u(t-2) + \cdots + b_{n_b}u(t-n_b) + e(t) + c_1e(t-1) + c_2e(t-2) + \cdots + c_{n_c}e(t-n_c) \quad (4.14)$$

With

$$C(q) = 1 + c_1q^{-1} + \cdots + c_{n_c}q^{-n_c} \quad (4.15)$$

and $A(q)$ and $B(q)$ defined as in Section 4.2.1 the model can be written as

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (4.16)$$

which corresponds to Equation 4.3 with

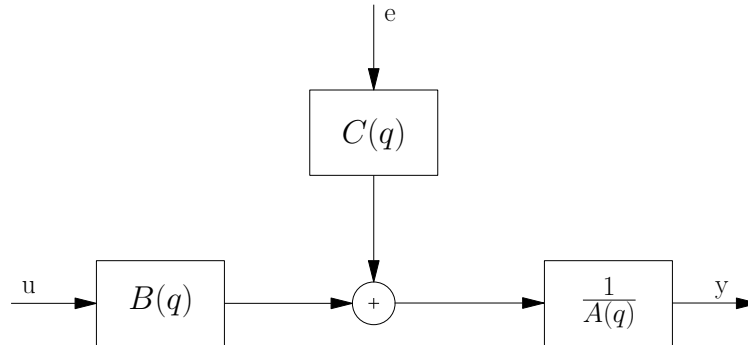


Figure 4.8: ARMAX model structure

$$G(q, \theta) = \frac{B(q)}{A(q)} \quad (4.17)$$

and

$$H(q, \theta) = \frac{C(q)}{A(q)} \quad (4.18)$$

This model structure can be seen in Figure 4.8. θ is now defined as

$$\theta = [a_1 \ a_2 \ \cdots \ a_{n_a} \ b_1 \ b_2 \ \cdots \ b_{n_b} \ c_1 \ c_2 \ \cdots \ c_{n_c}]^T \quad (4.19)$$

As for the ARX-model the predictor for Equation 4.16 is obtained by inserting Equation 4.17 and 4.18 into Equation 4.4. This gives

$$\hat{y}(t|\theta) = \frac{B(q)}{C(q)}u(t) + \left[1 - \frac{A(q)}{C(q)}\right]y(t) \quad (4.20)$$

which can be rewritten in formal analogy with Equation 4.13. To achieve this Equation 4.20 is written as

$$C(q)\hat{y}(t|\theta) = B(q)u(t) + [C(q) - A(q)]y(t) \quad (4.21)$$

4. SYSTEM IDENTIFICATION

then $[1 - C(q)]\hat{y}(t|\theta)$ is added to both sides of this equation. This gives

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t) + [C(q) - 1][y(t) - \hat{y}(t|\theta)] \quad (4.22)$$

By introducing the prediction error

$$\eta(t, \theta) = y(t) - \hat{y}(t|\theta) \quad (4.23)$$

and extending the vector $\varphi(t, \theta)$ from the ARX-model into the following

$$\varphi(t, \theta) = [-y(t-1) \cdots -y(t-n_a) \quad u(t-1) \cdots u(t-n_b) \quad \eta(t-1, \theta) \cdots \eta(t-n_c, \theta)]^T \quad (4.24)$$

Equation 4.22 can be written as

$$\hat{y}(t|\theta) = \varphi^T(t, \theta)\theta \quad (4.25)$$

This is very similar to the linear regression found for the ARX-model (see Equation 4.13). However it is not a linear regression due to the nonlinear effects of θ in $\varphi(t, \theta)$. It is thus called a pseudo-linear regression. ARMAX-models are useful when there are dominating disturbances that enters early in the process, e.g. a wind gust affecting the input of an aircraft or a ship.

Both of the above models has the polynomial A as a common factor in the denominators. This might be unrealistic since the deterministic and stochastic part of the system does not necessarily share the same set of poles. Another approach where the disturbance is excluded from the system dynamic should thus be considered. The OE-model and the Box-Jenkins model are such models.

4.2.3 OE-model

The Output-Error model (OE-model) describes the system dynamics separately. No parameters are used to model the disturbance. The basic idea is to assume that the relationship between the undisturbed output, w , can be written as a linear

4.2 Selecting a model structure

difference equation. Also the disturbance is considered to be white measurement noise. Then it is possible to write the model as

$$w(t) + f_1 w(t-1) + f_2 w(t-2) + \cdots + f_{n_f} w(t-n_f) = b_1 u(t-1) + b_2 u(t-2) + \cdots + b_{n_b} u(t-n_b) \quad (4.26)$$

$$y(t) = w(t) + e(t) \quad (4.27)$$

With

$$F(q) = 1 + f_1 q^{-1} + \cdots + f_{n_f} q^{-n_f} \quad (4.28)$$

and the polynomial $B(q)$ as before, the model can be written as

$$y(t) = \frac{B(q)}{F(q)} u(t) + e(t) \quad (4.29)$$

this corresponds to Equation 4.3 with

$$G(q, \theta) = \frac{B(q)}{F(q)} \quad (4.30)$$

and

$$H(q, \theta) = 1 \quad (4.31)$$

Figure 4.9 depicts this model structure. In this case the parameter vector to be estimated is

$$\theta = [b_1 \quad b_2 \cdots b_{n_b} \quad f_1 \quad f_2 \cdots f_{n_f}]^T \quad (4.32)$$

4. SYSTEM IDENTIFICATION

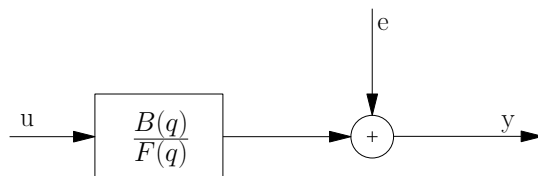


Figure 4.9: OE model structure

Inserting Equation 4.30 and 4.31 into Equation 4.4 gives

$$\hat{y}(t|\theta) = \frac{B(q)}{F(q)}u(t) = w(t, \theta) \quad (4.33)$$

To write this in formal agreement with the ARMAX-model predictor (see Equation 4.25) the following vector is used

$$\varphi(t, \theta) = [u(t-1) \cdots u(t-n_b) \quad w(t-1, \theta) \cdots w(t-n_f, \theta)]^T \quad (4.34)$$

4.2.4 Box-Jenkins model

A natural extension of the OE-model is to also model the properties of the output error. This gives a model where the disturbance properties is taken into account, but in contrast to the ARX or the ARMAX model, the disturbance is completely separated from the system dynamics.

As for the ARMAX-model the properties of the output error could be described as an ARMA model. This gives the following model structure for the Box-Jenkins model

$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (4.35)$$

this corresponds to Equation 4.3 with

$$G(q, \theta) = \frac{B(q)}{F(q)} \quad (4.36)$$

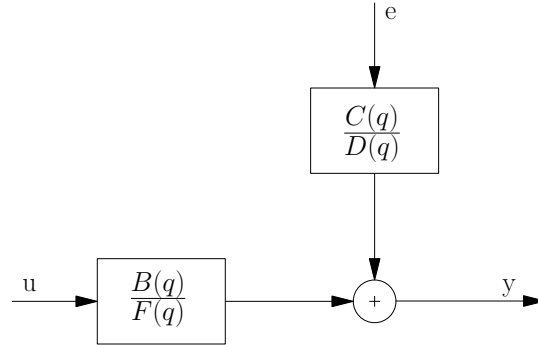


Figure 4.10: Box-Jenkins model structure

and

$$H(q, \theta) = \frac{C(q)}{D(q)} \quad (4.37)$$

Figure 4.10 is a graphical representation of this model structure. In this case the parameter vector to be estimated is

$$\theta = [b_1 \ b_2 \cdots b_{n_b} \ c_1 \ c_2 \cdots c_{n_c} \ d_1 \ d_2 \cdots d_{n_d} \ f_1 \ f_2 \cdots f_{n_f}]^T \quad (4.38)$$

Inserting Equation 4.36 and 4.37 into Equation 4.4 gives

$$\hat{y}(t|\theta) = \frac{D(q)B(q)}{C(q)F(q)}u(t) + \frac{C(q) - D(q)}{C(q)}y(t) \quad (4.39)$$

This model is useful if there are disturbances entering late in the process. E.g. disturbances due to measurement noise on the output.

4.2.5 Process models

Process models or transfer function models are used to describe the deterministic part of the system, thus neglecting the stochastic part, i.e. the disturbance. This type of model is ideal for identifying physical systems. The primary argument for using such a model structure is that they provide estimation of delays and

4. SYSTEM IDENTIFICATION

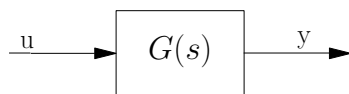


Figure 4.11: Structure of the process model

the coefficients of the identified model has a physical interpretation (Ljung n.d.). Either single-input, single-output (SISO) or multiple-input, single-output (MISO) systems can be identified by this model structure. For multiple-input, multiple-output (MIMO) state-space models should be used (Instruments n.d.).

The use of process models generates an estimated transfer function for each input, e.g. a process with four inputs and one output will generate four transfer functions. These transfer functions will then describe how each of the inputs affects the output using static gain, a time delay before the system output responds to the input, and characteristic time constants associated with poles and zeros. The model structure can be written as

$$y(t) = G(s)u(t) \quad (4.40)$$

where $G(s)$ represents the transfer function to be estimated and $y(t)$ and $u(t)$ are respectively the input(s) and output of the system. Note that the structure of the transfer function for each input may be selected independently, e.g. an input to an output might be described by a first order transfer function while another input to the same or another output might be described by a transfer function of another order. Figure 4.11 depicts this model structure for a SISO system.

4.2.6 State-space models

A state-space model represents the relationship between input, output and noise as a system of first order differential equations using a state vector, $x(t)$ (Ljung 2006). This type of representation is very useful since insight into the physical mechanisms of the system can easily be incorporated into such a model. This type of model should be used if the system to be identified is a MIMO physical system.

4.2.7 Model structure selection

As briefly seen in the presentation of each of the model structures above there are several considerations to be taken when determining which model structure to choose. Table 4.3 can be used as a reference or a guidance to help in the decision making.

Since all disturbances that affect the system to be identified (e.g. the outdoor temperature) are measured and hence are treated as inputs, a model structure which does not include estimation of the disturbance model is chosen. This excludes the ARX, ARMAX and Box-Jenkins models. To choose between the different models that fulfill the above criterion some additional criteria must be stated. The system to be identified is clearly a physical one, and as previously stated a linear one, since it basically is a network consisting only of resistors and capacitors. Process models and state-space models are ideal to use when identifying such systems. Also the models created using these approaches have coefficients that have a physical interpretation. This makes it easier for the user to verify whether or not the model found in the identification process is physically likely. Finally the system to be identified are as previously stated MISO systems. Due to this and the other properties mentioned above process models have been chosen when identifying a simple model that can be discretized but also behaves as identical to the original system as possible.

It should be noted that other types of models such as ARX, ARMAX, OE and Box-Jenkins also have been tested for identification of the system. These models were however, and as expected from the previous discussion, found to be less accurate than the process model approach. However it should be noted that the identification of these model types are solved using more efficient algorithms and hence the possibility of using such model types should not be neglected. Figure 4.12 shows the results from the system identification when ARX, ARMAX, OE, Box-Jenkins and process models respectively are used to identify room k133 in the building. As seen from this figure a first order process model has a 93.58 % match and is clearly the best selection.

It should be noted that these results have been found while every excitation signal is active. In Section 4.3 where the identification procedure is elaborated none of the excitation signals have been active at the same time. This has been done to ensure that the transfer functions from an input to the output is an even more accurate description of the original system. This will be further elaborated in Section 4.3.3.

Table 4.3: Model structures - pros and cons (see Ljung (2006) for further details)

Model structure	Pros	Cons
ARX-models	Simple input-output relation. Linear regression. Preferable when the model order is high.	Disturbances are part of the system dynamics. Little freedom to describe the disturbance.
ARMAX-models	Flexible disturbance model. Useful when there are dominating disturbances entering early in the process.	Disturbances are part of the system dynamics. Not a linear regression.
OE-models	The system dynamics and the disturbance are described separately.	Not a linear regression. No disturbance model.
Box-Jenkins models	The system dynamics and the disturbance are described separately. Useful to identify processes where the disturbance enters late in the process.	Not a linear regression.
Process models	Ideal for identifying physical systems. Provides estimation of process delays. The identified coefficients has a physical interpretation.	No disturbance model. Maximum model order is 3.
State-space models	Represents the relationship between input, output and noise as a system of first order differential equations. Easy to incorporate insight into physical mechanisms of the system.	

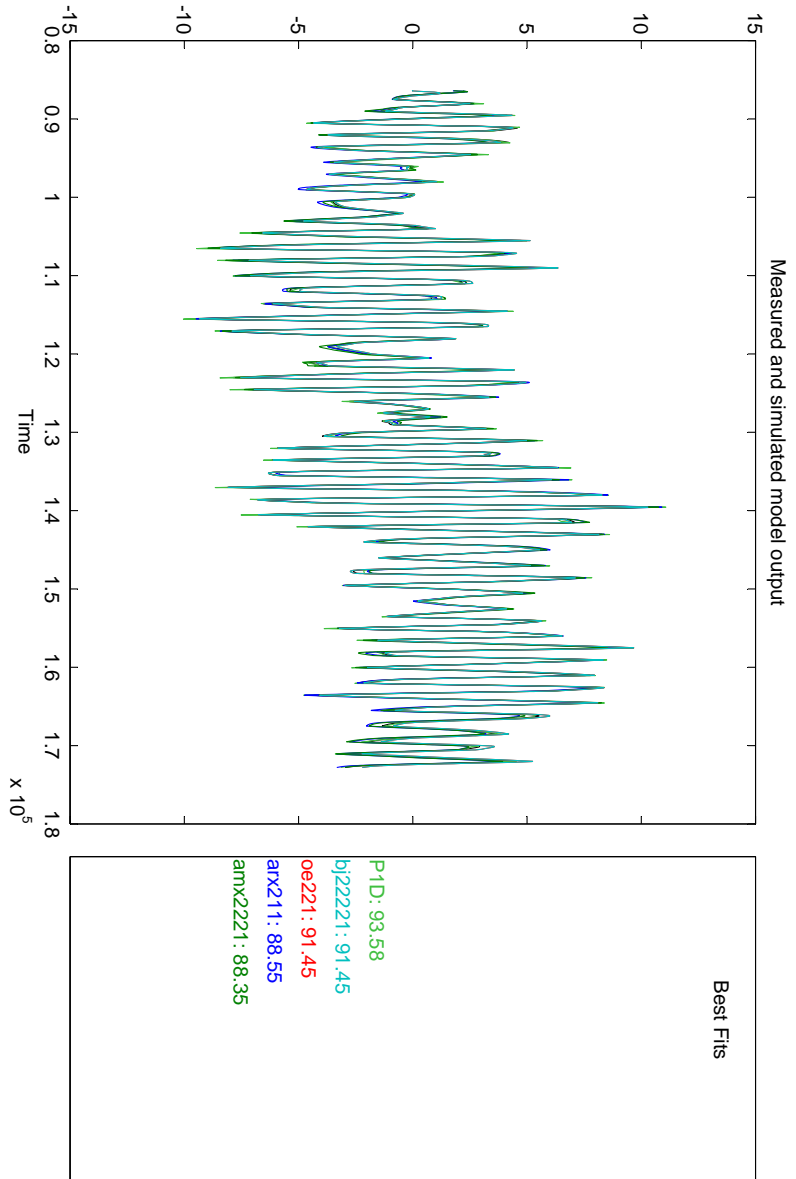


Figure 4.12: Results from the identification procedure using different model types.

4. SYSTEM IDENTIFICATION

4.3 System identification procedure and results

In this report the system has been identified using Matlab's System identification toolbox. This section describes this procedure and the results following from this.

4.3.1 Importing and preprocessing the data set

The first step in system identification is to import the logged input and output of the system into the System identification toolbox GUI. This is called the data set and is done using the following steps. First, open the System identification toolbox by typing the command `ident` in Matlab's command window. Then click the drop box displaying *Import data* and select *Time domain data* since this is the data type that has been stored in this report. Type the names of the input and output signals from the data set and select the sampling interval equal to the time step used when storing the input/output data. If these sampling intervals differs from each other the system identification toolbox will not be able to find a model that represents the system precisely even though it might seem so from the results. See Figure 4.13 for a graphical view of the procedure just described.

The next step is preprocessing of the input and output signal. A quick and efficient way of doing this is to press the drop box named *Preprocess* in the GUI and select *Quick start*. What this does to the input and output signal is the following. First the means from each input and output is removed, meaning that the data now has a zero mean value. This is done since the physical values of the input and output usually does not matter to the identification procedure. However if the physical values matter to the identification process this step should be omitted. Second the dataset, i.e. the input and the output, is divided into two sets. A set used for model identification and a set used to verify the model found in the identification process. Usually this is done simply by choosing the first half of the data set for identification and the second half for verification. It is important that the data set used for identification is different from the one used for verification. Otherwise it is not possible to say that the model is correct, meaning that it is a good representation of the dynamics of the real system. Having done this the data set is ready to be used for system identification.

4.3.2 Model identification

With the preprocessed data set it is time to start the model identification procedure. This is a procedure that can be divided into the following steps:

4.3 System identification procedure and results

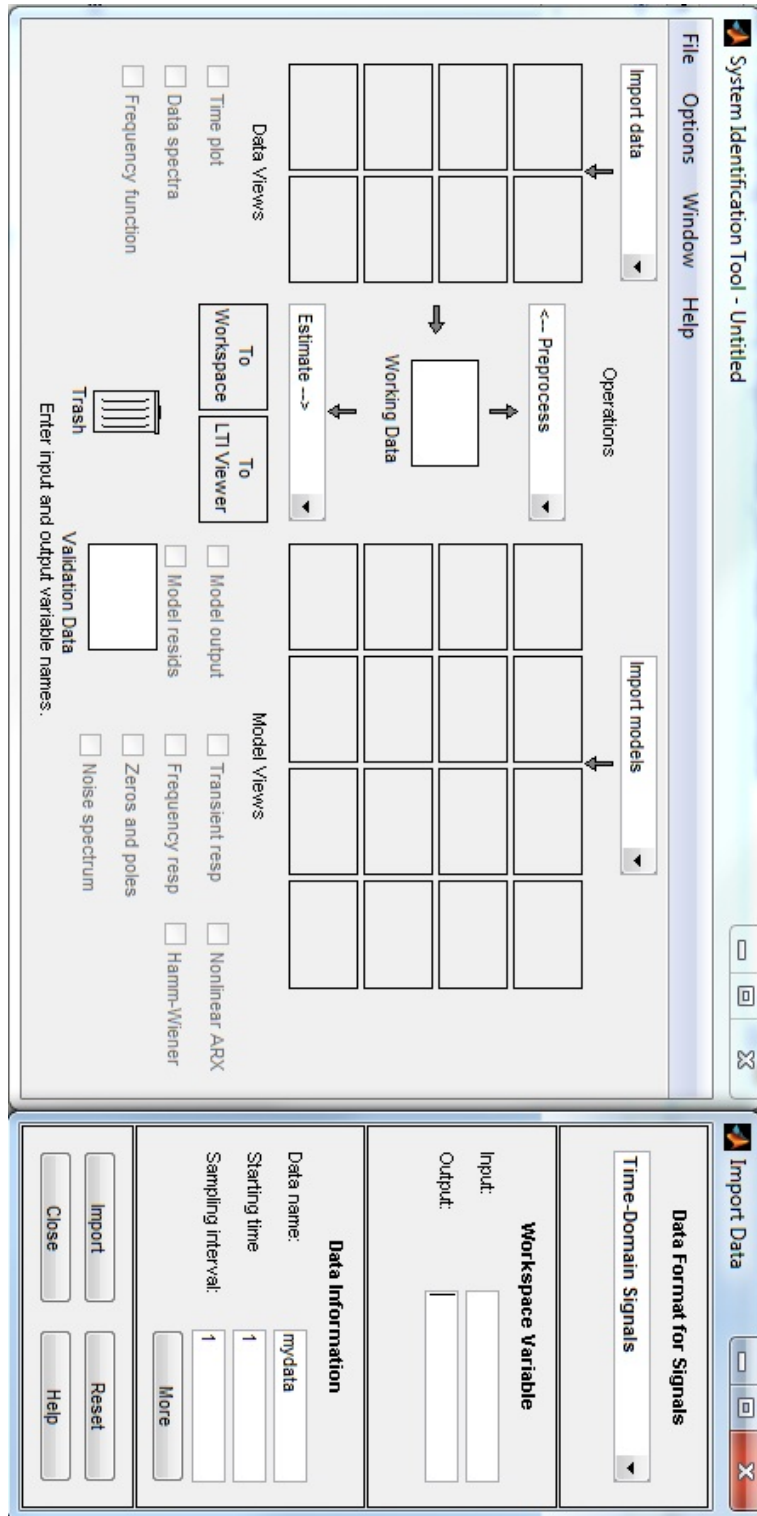


Figure 4.13: System identification toolbox GUI and the import data window.

4. SYSTEM IDENTIFICATION

1. Choosing a model structure (e.g. ARX, ARMAX, Box-Jenkins, process models).
2. Choosing a model order.
3. Choosing which estimation method to use (e.g. least square, instrumental-variable, Gauss-Newton, Levenberg-Marquardt)
4. Start the identification procedure.
5. Check the results and verify them.

For guidance regarding the first step the reader is referred to Section 4.2. In the GUI press the drop box named *Estimate* and choose the desired model structure. The best way to choose a model order (step two) if there are no information about this before the identification procedure starts, is to start with low order models and see if they are able to reflect the dynamics of the system. If not, try a model of a higher order until the model reflects the system dynamics of the real system close enough. Step three above, finding the estimation method that works best, depends on the system to be identified. Often the system to be identified is unknown. This means that in order to find the best estimation method for a certain system the best way is often trial and error. In the GUI it is possible to leave this choice to `Matlab` by selecting this option to *auto*. This is the default choice but not necessarily the best. The next step is straight forward, just press the button called *Estimate* in the GUI and the estimated model with the preferences set in the previous steps are being calculated. Figure 4.14 shows an example of how the GUI window looks like when estimation of process models have been chosen. When the model has been estimated it can be seen in the placeholders on the right hand side of the System identification toolbox GUI.

The next step is to verify the estimated model. While the first part of the data set was used for model estimation the second part is used for validation of the model. By selecting the check box called *Model output* in the GUI a new window appears where the output of the estimated model is plotted together with the output of the real system. At the right hand side of this plot the match between these plots can be seen in terms of percentage. An example of this was shown earlier in Figure 4.12. Ideally this result is 100 %.

Usually there will not be a 100 % fit between the estimated model and the real system. The part of the system data that could not be described by the model is called the residuals (Ljung 2006). These bear important information about the quality of the estimated model. According to Ljung (2006) the cross correlation between residuals and input of a good model does not go significantly outside its confidence region. If this is true it means that the basic properties of the

4.3 System identification procedure and results

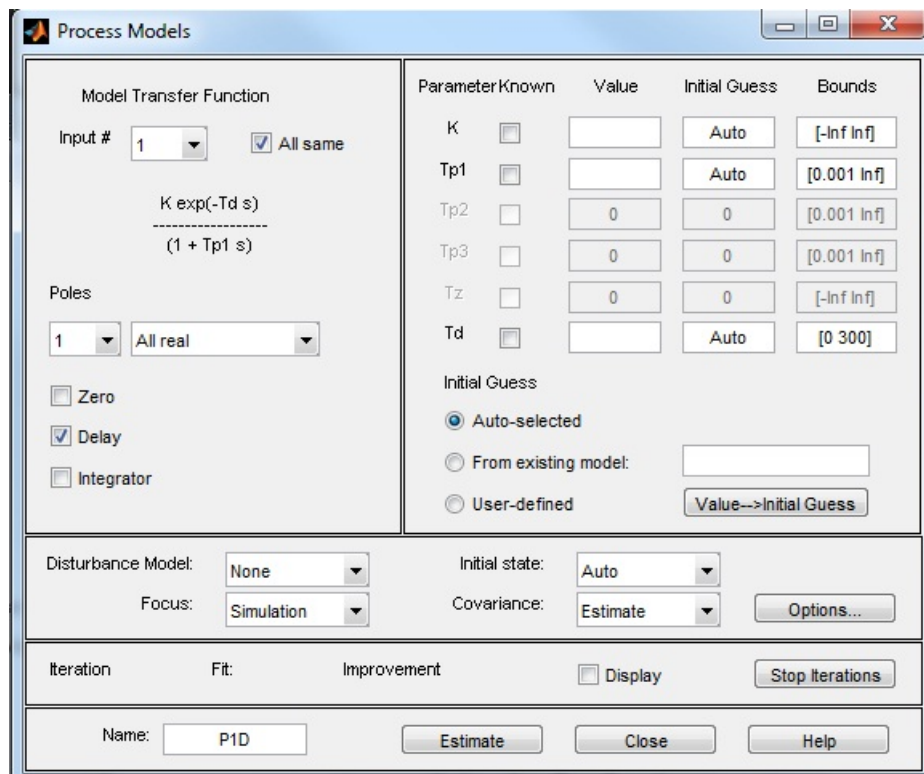


Figure 4.14: System identification toolbox GUI, process models.

4. SYSTEM IDENTIFICATION

system has been captured by the estimated model. The autocorrelation of the residuals indicates whether or not the error model is accurate. Since process models by default does not estimate the error model and there are no unknown disturbances entering the original model, the autocorrelation residual will not be used for verification of the model. To see the residual analysis in the GUI check the box named *Model resids*. A new window with the plots of the autocorrelation and the cross correlation will now appear. Figure 4.16 shows how this looks for the radiator gain in room k133. From the cross correlation plot in this figure it is clear that the estimated model resembles the response of the room to inputs closely.

4.3.3 Results and model verification

As mentioned in the start of this chapter the identification of a model that resembles the buildings response to inputs has been divided into a problem of identifying one room at a time. It has also been argued that process models should be used when identifying an estimated model of the original building. In addition to this the effect from adjoining rooms are considered to be negligible. With this in mind process models have been chosen to identify the buildings response to disturbances and inputs. Since some of the excitation signals affects the room greatly and others do not have the same effect it is difficult to estimate a good model for each of the input signals by having every excitation signals activated at the same time. Thus only one excitation signal have been active and the others are set to zero so that the System identification toolbox only needs to estimate the transfer function from the active input to the output. This procedure must then be repeated for each input. I.e. instead of identifying a single MISO system three SISO systems are identified for each room.

The results from the identification procedure for room k133 can be seen in Figure 4.16 up to and including Figure 4.18. Also the other rooms have been identified in the same manner and the results from this can be seen in Appendix B. The transfer functions following from these results have been found to be either of first or second order including time delays as seen respectively in Equation 4.41 and Equation 4.42.

$$G(s) = \frac{Kp}{T_p s + 1} e^{-T_d s} \quad (4.41)$$

4.3 System identification procedure and results

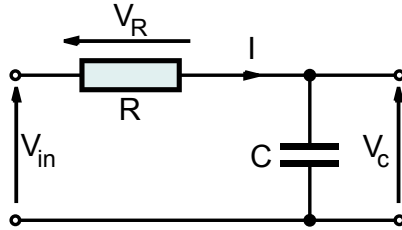


Figure 4.15: Simple RC-circuit. Source: Wikipedia

$$G(s) = \frac{Kp}{(T_{p1}s + 1)(T_{p2}s + 1)} e^{-T_d s} \quad (4.42)$$

The transfer functions found for each room can be seen in Table 4.4. Note that the transfer function between the ventilation and the room is of second order. This is due to the fact that the system dynamics was not captured good enough with a first order transfer function (i.e. the residuals was not within the confidence region). Using the rule of thumb found in Ljung (2006); “A slowly varying cross correlation function outside the confidence region is an indication of too few poles, while sharper peaks indicate too few zeros or wrong delays.”; the order of the transfer function was increased to bring the residual inside the confidence region. Also note that $e^{-T_d s} = 1$ if the time delay is zero. Due to existing knowledge of the original system the time delays between each input and the output of the system is known. The reason for this will now be elaborated.

The original building model consists only of resistors and capacitors. It can be shown that the transfer functions generated by any RC-circuit does not include a time delay. For example the simple RC-circuit in Figure 4.15 can be shown to have the following transfer functions

$$H_R(s) = \frac{V_R(s)}{V_{in}(s)} = \frac{RCs}{1 + RCs} \quad (4.43)$$

$$H_C(s) = \frac{V_C(s)}{V_{in}(s)} = \frac{1}{1 + RCs} \quad (4.44)$$

$$(4.45)$$

Thus no time delay should be present for the transfer function between the radiator gain and the room, and the transfer function between the outdoor temperature and the room. The transfer function between the ventilation temperature and

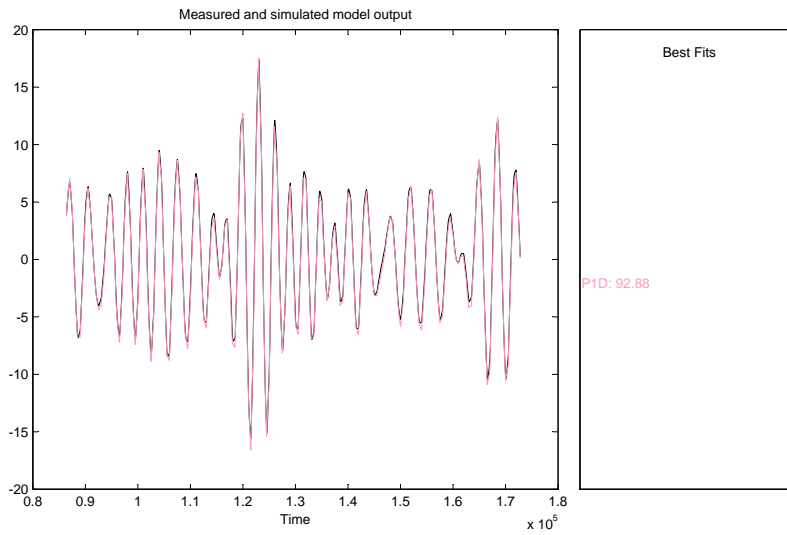
4. SYSTEM IDENTIFICATION

the room temperature does however have a time delay. This can be calculated due to the existing knowledge of the length and the diameter of the ducts and the flow of air through these. Thus the time delays are known prior to the system identification process. Hence this can and should be set as known parameters before identifying the other parameters of the process models. Again the reader is referred to Table 4.4 to see the results from the identification procedure.

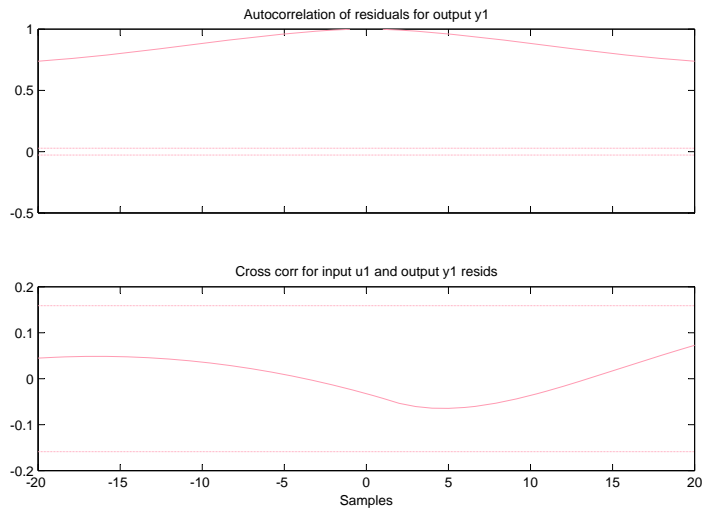
Table 4.4: Transfer functions from the system identification process.

	Radiator gain - room	Outdoor temperature - room	Ventilation - room
k133	$G(s) = \frac{0.73171}{1013.5s+1}$	$G(s) = \frac{0.134}{3658s+1}$	$G(s) = \frac{0.14385}{(14872s+1)(795.83s+1)}e^{-13.396s}$
k134	$G(s) = \frac{0.65748}{994.89s+1}$	$G(s) = \frac{0.13735}{3851.95s+1}$	$G(s) = \frac{0.16342}{(16077s+1)(999.19s+1)}e^{-13.396s}$
k135	$G(s) = \frac{0.6535}{985.05+1}$	$G(s) = \frac{0.13713}{3834.2s+1}$	$G(s) = \frac{0.16348}{(16084s+1)(999.06s+1)}e^{-13.396s}$
k136	$G(s) = \frac{0.72889}{1003.8s+1}$	$G(s) = \frac{0.12995}{3415.4s+1}$	$G(s) = \frac{0.15832}{(15509s+1)(939.29s+1)}e^{-13.396s}$
k143	$G(s) = \frac{0.69344}{1362s+1}$	$G(s) = \frac{0.1737}{3119.9s+1}$	$G(s) = \frac{0.18767}{(15543s+1)(1090.2s+1)}e^{-13.396s}$
k164	$G(s) = \frac{0.67087}{1020.3s+1}$	$G(s) = \frac{0.16706}{3517.6s+1}$	$G(s) = \frac{0.16138}{(15645s+1)(957.58s+1)}e^{-13.396s}$

4. SYSTEM IDENTIFICATION



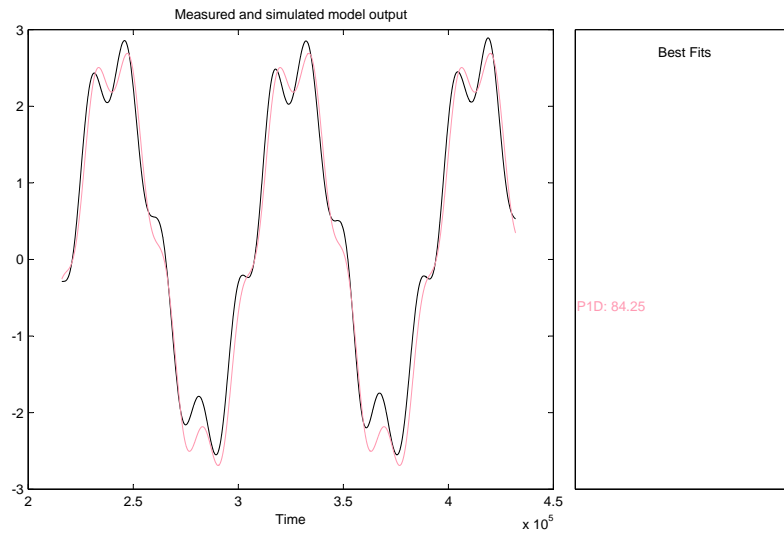
(a) Match



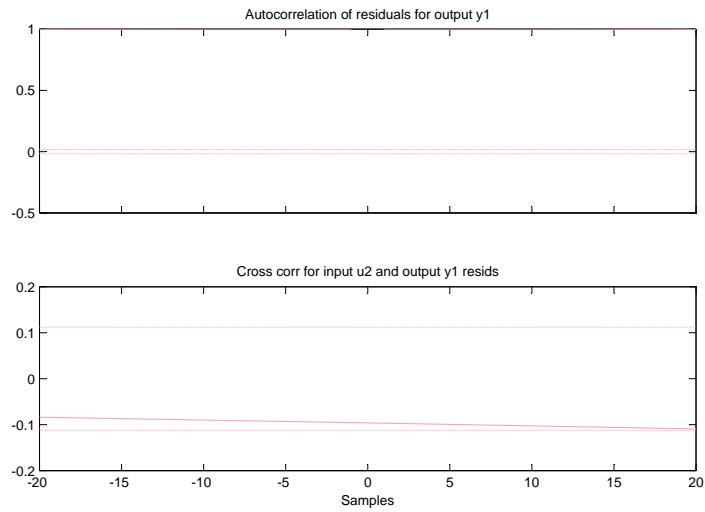
(b) Residual

Figure 4.16: System identification results, room k133 - radiator.

4.3 System identification procedure and results



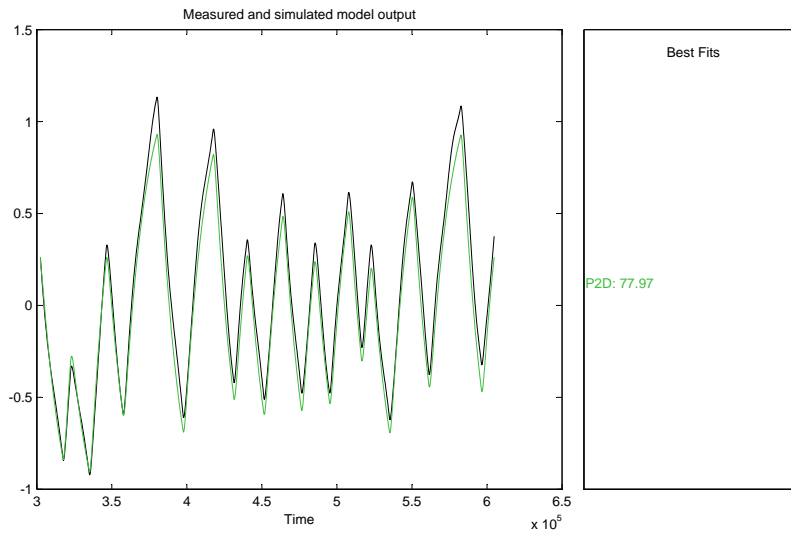
(a) Match



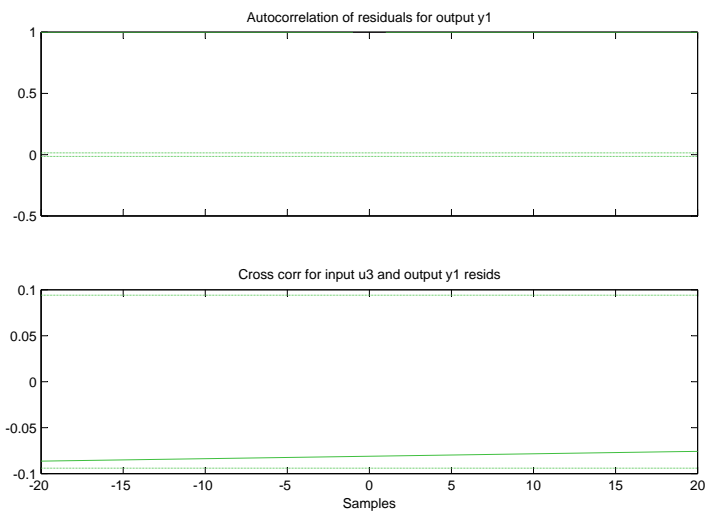
(b) Residual

Figure 4.17: System identification results, room k133 - outdoor temperature.

4. SYSTEM IDENTIFICATION



(a) Match



(b) Residual

Figure 4.18: System identification results, room k133 - ventilation.

5

Automatic control

Automatic control, meaning that the system reacts as desired without human interaction, is achieved by using a controller. This controller is placed in front of the system to be controlled and feeds the control element (e.g. a valve, rudder or motor) in the system with a gain signal. The control element then reacts to this signal and the system output changes accordingly. The output of the system are the variables to be controlled (e.g. temperature or position). In order to achieve automatic control the difference between the actual output and the desired output must be measured and feed into the controller. This difference is called the error. In the controller the error is multiplied by a factor given by some parameters. This gives the gain of the controller mentioned earlier and is used to force the error to zero. The method just described is called a feedback loop and is shown in Figure 5.1.

It should be noted that information about the disturbance that acts on the system (e.g. outdoor temperature or wind and its direction) can be exploited to make a better controller. The use of such information is called a feed forward connection (Balchen et al. 2003) and can be seen in Figure 5.2. The benefit of such a connection compared to a feedback loop is that the controller can act to this

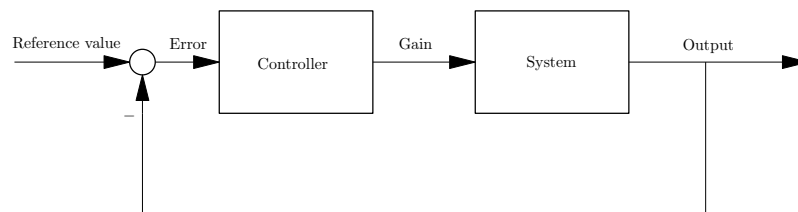


Figure 5.1: The classic feedback loop

5. AUTOMATIC CONTROL

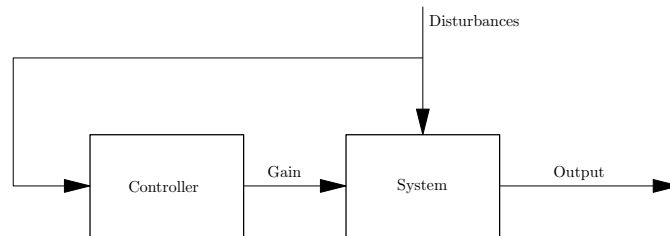


Figure 5.2: Feedforward

disturbance immediately. It does not need to wait for the disturbance to appear on the output as the feedback loop needs to. In slow systems where it might take a considerable amount of time before the output changes due to a disturbance this might be imperative. Usually a feed forward connection is used in combination with a feedback loop. Hence the feed forward connection cancel out most of the disturbance while the feedback loop takes care of the rest (i.e. drives the error to zero).

Feed forward loops are often used in HVAC systems since such systems usually are very slow. However in this report it is only included in the MPC scheme presented in Section 5.3.

5.1 Control strategy

In this report two control strategies have been implemented, namely the traditional PID control strategy and the more advanced model predictive control (MPC) strategy. The PID control strategy is used to control the ventilation temperature. For the room temperature first the PID control strategy are used, then it is substituted with the MPC controller. The reason for doing so is to clarify whether or not this advanced control scheme is able to regulate the system to its desired value by using less energy than the basic PID controller.

The temperature of the ventilation and the temperature of the room are controlled independently and hence they can be considered as two separate systems even though they really are connected to each other via the ventilation temperature. The ventilation temperature and the room temperature are the controlled variables (CVs) of the systems. There is also disturbances acting on the system. For the ventilation system the varying outdoor temperature is the only disturbance, but for the indoor temperature there are two additional disturbance acting on the system. This is the temperature of the ventilated air and the temperature from the adjoining room(s). These three temperatures are the disturbance

Table 5.1: Inputs, outputs and measured disturbances of the building

Manipulated variables (MV)	Disturbance variables (DV)	Controlled variables (CV)
Radiator gain	Outdoor temperature	Room temperature
Ventilation gain	Ventilation temperature	Ventilation temperature
	Temperature in adjacent rooms	

variables (DVs) acting on the systems. Once again it is noted that the model of the building found in Chapter 4 has neglected the disturbance from the adjoining rooms due to its negligible effect on the room as long as its temperature is near the rooms temperature. In order to stabilize the CVs at the desired set point the controller needs to be able to rule over some variables. These variables are the so called manipulated variables (MVs). For the ventilation system this variable is the valve opening of the shunt connection while the water temperature sent into the radiators is the manipulated variable for the rooms. Table 5.1 shows an overview of the MVs, CVs and the DVs of the system.

Next in this chapter the PID controller will be presented with examples that shows how this controller is able to hold the indoor temperature and the ventilation temperature at the desired values. Then the MPC control scheme will be presented with examples that shows how this controller holds the desired temperature of the room.

5.2 PID controller

To be able to produce a control signal (i.e. a gain) a controller is needed. The most common controller is the PID controller which is widely used in industrial control systems. The input of the PID controller is the previously mentioned error. The control signal of the PID controller is calculated using the error and three gains; the proportional gain, the integral gain and the derivative gain. The proportional gain is a simple multiplication of the error by a factor K_p (Underwood 1999). This determines the reaction to the current error. The integral gain is a multiplication of the sum of recent errors by a factor K_i (Underwood 1999). This determines the reaction to the sum of recent errors and its purpose is to avoid stationary deviation from the set point (infinite gain when the frequency equals zero, $\omega = 0$) (Balchen et al. 2003). The derivative gain is a multiplication of the rate of which the error has been changing by a factor K_d (Underwood 1999). This determines the reaction to the rate of which the error has been changing.

In most HVAC systems it is not necessary to use the derivative part of the PID.

5. AUTOMATIC CONTROL

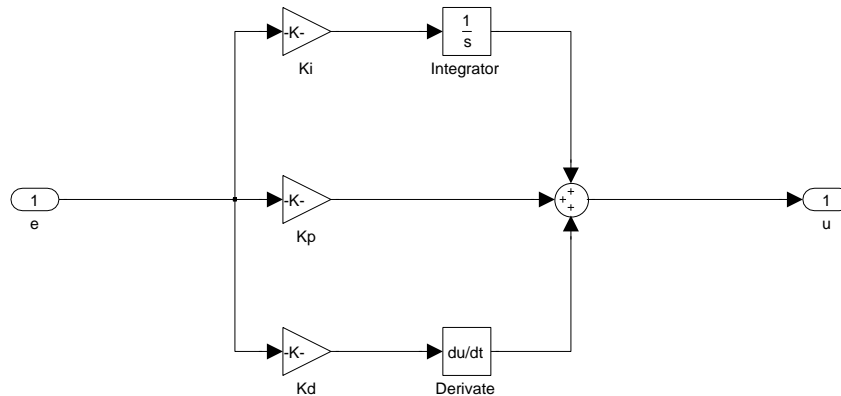


Figure 5.3: PID controller

This is neither common in general process control. Hence in this report only the proportional and integral part of the PID-controller will be used (i.e. a PI controller). The structure of a basic PID controller is shown in Figure 5.3. To get a PI-controller the gain K_d is set to zero.

Saturation and anti windup

The PID controller explained in the previous section works well, but some changes must be made so that it does not violate restrictions in the control element. A common example of such a restriction is that the rate of change in velocity of a motor (e.g. a robot arm) is restricted to some maximum value. Another example, which is more relevant to this report, is a radiator. The temperature of the radiator has a lower and an upper saturation. It cannot be warmer than the water temperature within the radiator circuit and it cannot be colder than the air where it is placed. It is important to take all restrictions into account when modeling a system. If this is not done, the model might seem to work well when simulating it, but it might not work at all when implementing it in a real system. This usually happens when the controller tries to force the control element to do something it is not capable of doing.

To make sure the controller does not violate restrictions in the control element a saturation element is used in Simulink. The saturation block has an upper and a lower bound that inhibits the controller to violate the maximum or minimum value of the control element. Thus the controller can only send signals that the control element is capable of handling. Figure 5.4 shows how the PID controller including the saturation element looks like. Although not obvious there is one major drawback with this implementation if the controller contains a integral

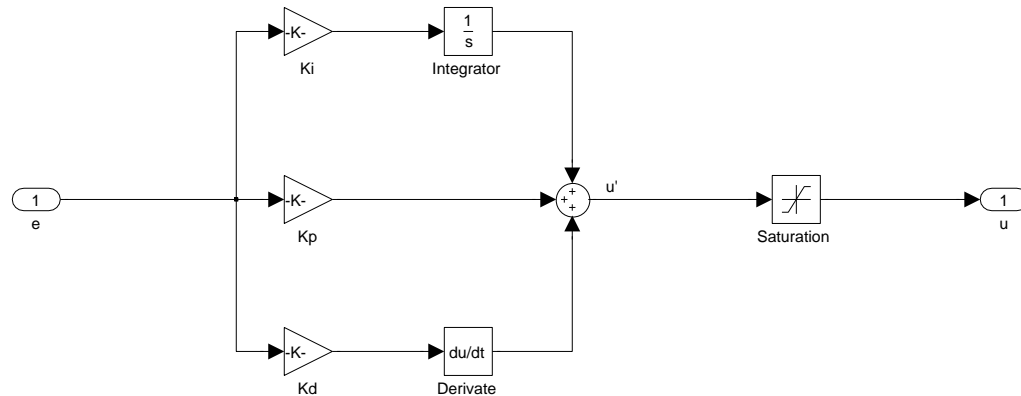


Figure 5.4: PID controller with saturation

term. This will now be discussed further.

If the PID control structure in Figure 5.4 stays within the upper and lower limits of the saturation block the controller will work as intended. However, if the controller becomes saturated the integrator will continue to build up its value for as long as the sign of the error stays the same (positive or negative). Thus when the controller tries to get out of saturation (the desired process value has been reached) it will not manage to do so because of the large accumulated value of the integrator. The integrator value will start to decrease only when the sign of the error changes (i.e. positive error becomes negative error and vice versa). This behavior is called integrator wind-up (Balchen et al. 2003). Integrator wind-up can cause the system to overshoot the desired value up to several times before settling down. It may even cause the system to become unstable (Maciejowski 2002, Skogestad & Postlethwaite 2007).

Since the control elements in this report has some restrictions a saturation block is needed in the controller. Based on the previous discussion it should also have an anti wind-up algorithm to cope with the wind-up of the integrator when in saturation. There are several ways to ensure this. The anti wind-up scheme used in this report is shown in Figure 5.5 and the following is an explanation of it. The controller calculates a desired gain, u' . This gain will pass through the saturation element and the output of this is the actual gain sent to the control element (e.g. actuator or radiator), u . When these two values are equal (i.e. there is no saturation) the anti wind-up scheme will be inactive since these signals cancel each other out. In saturation the integrator would cause the value of u' to become larger and larger. Now u' would be larger than u . The difference between these values are then multiplied with a gain, K_a , before it is added to the signal in front of the integrator. The value of K_a is usually set equal to K_i . This ensures that

5. AUTOMATIC CONTROL

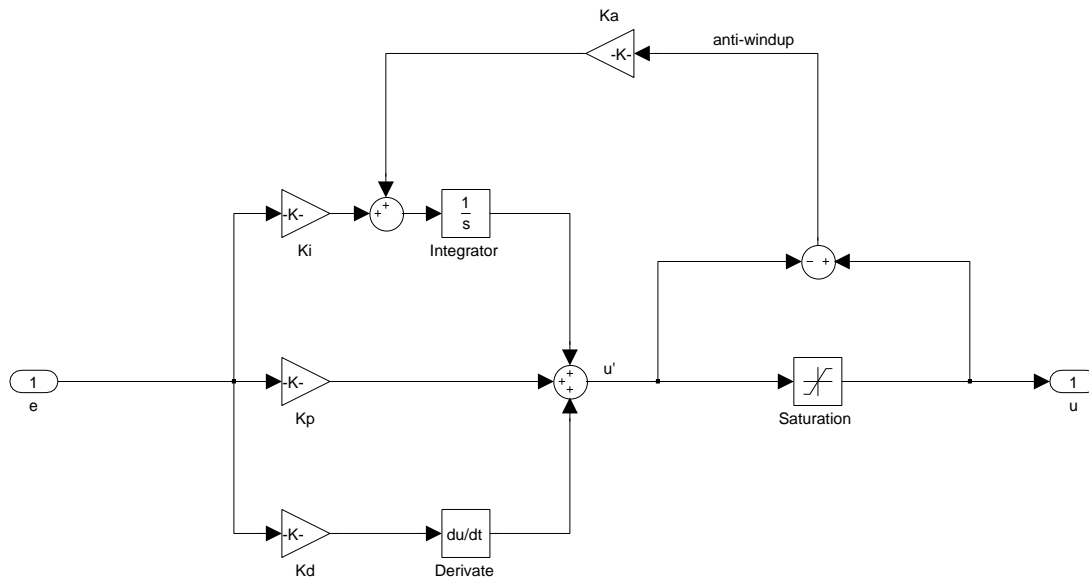


Figure 5.5: PID controller with saturation and anti wind-up

the integrator “turns off” when the system is in saturation.

In this report there are two variables to be controlled, namely the indoor temperature and the ventilation temperature. The specific method used to control these variables are explained in the following sections of this chapter.

5.2.1 Room temperature

Automatic control of the temperature in a room or a building is usually achieved by sending control signals from a controller to radiators in the specific room or building. This control signal is based on the error between the actual room temperature and the desired room temperature. Hence it is a classical feedback loop equal to the one explained in the introduction of this chapter. It should be noted that there are some new buildings that use the ventilation to heat up the building, but this is normally not the case. In this report the heating will be done by the radiators placed in the different rooms in the school building. This means that there will be a feedback loop in each of the rooms consisting of a controller and the system (i.e. the room as modeled in Chapter 3 and the radiators).

Radiator controller and its parameters

The radiators in each room is controlled to ensure a stable indoor temperature. This can be achieved by using the same controller structure as shown in Figure 5.5. To implement this controller the room model shown in Figure 3.13 is altered to the one seen in Figure 5.6. This figure has a block called “Radiator with temperature controller”. As the name suggest this is where the radiator model and the controller is placed. An exploded view of this block is shown in Figure 5.7. The observant reader might notice that the controller has an additional input, namely the indoor temperature. This is only used in the saturation block to make sure the radiator stays within its limits (e.g. the radiator cannot be colder than the temperature of the room). The radiator now has the dynamic behavior explained in Section 3.3.7 instead of a fixed temperature. The parameters of the PID-controller used must be set so that the radiators are able to hold a desired room temperature. In addition to this they must be set so that one ensures that the room temperature and the radiator temperature does not fluctuates extensively before settling down after a sudden disturbance or a change of set temperature. There exists several systematic methods for finding controller parameters that will work satisfying. Ziegler-Nichols method is an example of this (see Balchen et al. (2003) for more about this method). However the parameters found for this controller was found on the basis of trial and error. With this in mind the following parameters where chosen for this controller:

Kp_{room} 50

Ki_{room} 0.05

Kd_{room} 0

Example 5.1 and 5.2 verifies that the controller works as intended (i.e. it controls the temperature in the room).

Example 5.1 Constant room temperature

To control the radiators so that they ensures that the room holds a fixed desired temperature a control signal is sent to the radiators by the controller from Section 5.2.1. This signal will adjust the gain of the radiator. The radiator would either heat up, hold current gain or turn off. If the room is to cold the control signal would force the radiator to heat up. The opposite would happen if the room is to warm, the radiator will then turn off and wait for the temperature to drop. When the desired temperature is reached,

5. AUTOMATIC CONTROL

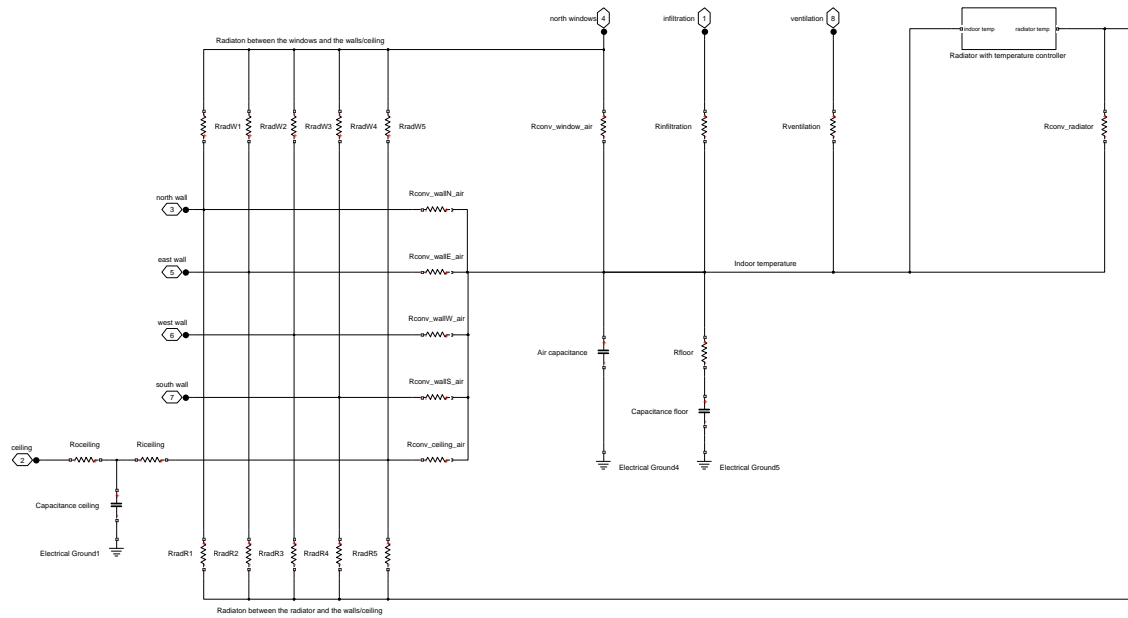


Figure 5.6: Model of the room with radiator model and controller

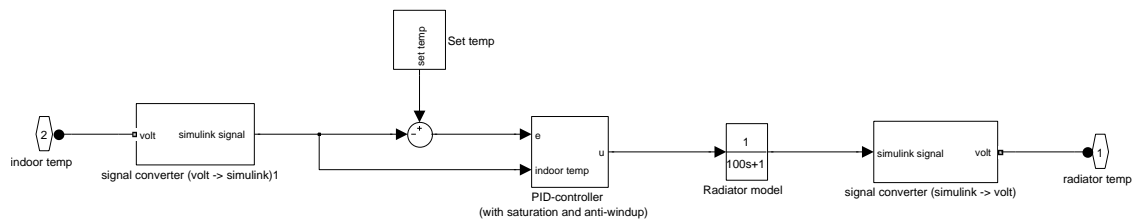


Figure 5.7: Exploded view of the radiator and controller block in Figure 5.6

the control signal will produce a gain in the radiator equal to the amount of energy needed to hold the temperature at the desired temperature. This would hold the temperature at the desired set point.

When the PI controller is used on radiators one needs to make sure that the control signal does not violate certain maximum and minimum values. It needs to stay within the limits of the radiator. In our model the limits are as listed below.

Maximum 70°C

Minimum Current room temperature

Simulation results of the building with the intent to hold a constant indoor temperature of 18°C considering a disturbance from the varying outdoor temperature can be seen in Figure 5.8. It should be noted that this figure only shows the temperature of three of the rooms, namely k133, k134 and k135. The temperature development in the other rooms are very similar, hence plotting them all in one figure will only make each room temperature difficult to follow. From this figure it is seen that the temperature rapidly approaches the given set temperature. It is also clear from the figure that the room temperature stays at the given set point even though disturbances due to a varying outdoor temperature acts on the system.

Example 5.2 Varying room temperature

Varying room temperature has become a normal and daily routine in the majority of today's buildings. Even most of today's personal residences operate with a varying reference temperature by reducing the temperature by night. To be able to achieve this the reference temperature (i.e. the set point) must be changed by some algorithm. The set point could change, say for instance from 21°C to 18°C at 16:30 in the afternoon in a school building. At this time it is expected that everyone has left the building and it is therefore not necessary to maintain a comfortable temperature in the building. The focus of the controller then shifts from comfortable to economic. In the morning, at some time before the students and teachers arrive, the set point changes back to 21°C (comfortable mode). This ensures that the temperature is back to a comfortable level when the students and teachers return. The algorithm just

5. AUTOMATIC CONTROL

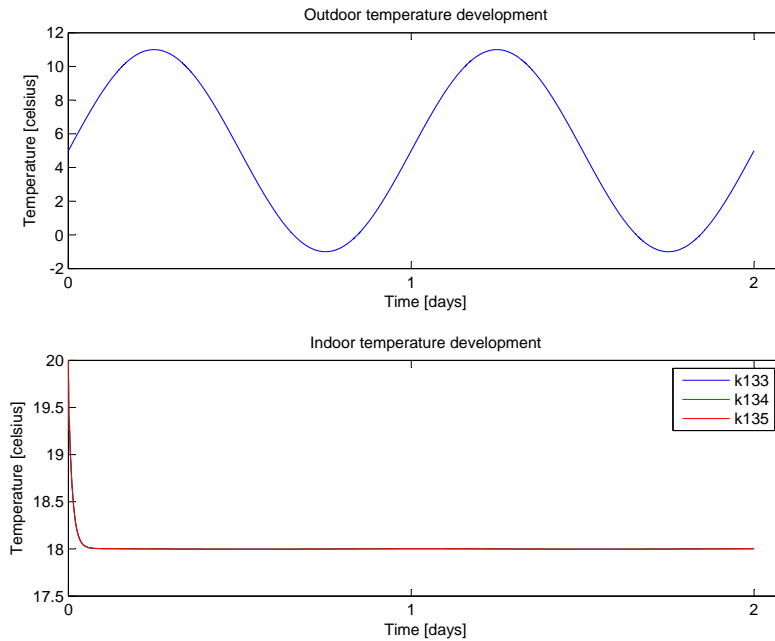


Figure 5.8: Fixed room temperature with controller: Set temperature: 18°C. Disturbance from a varying outdoor temperature.

described can be achieved by a simple if-else loop in Simulink. If one needs to make sure the set point does not change during the weekends a new if-else loop is constructed with the previous if-else loop inside of it. This should ensure that the set point stays in economic mode during the weekend but works as previously told from Monday until Friday. The exact same controller as explained in Section 5.2.1 is also used in this example.

Simulation results of the ventilation system with the intent to hold a varying indoor temperature of 21°C during the day and 18°C during the night, with a disturbance from a varying outdoor temperature can be seen in Figure 5.9. This figure clearly shows that the controller is able to control the temperature although varying set temperatures is used.

5.2.2 Ventilation temperature

The temperature of the ventilated air is also regulated to a set temperature by a controller. This controller receives information about the difference between the

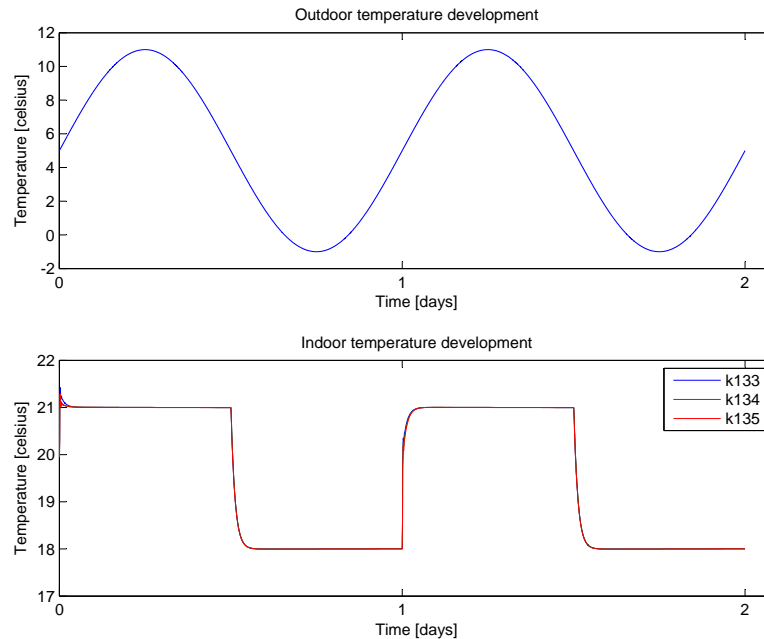


Figure 5.9: Varying room temperature with controller: Set temperature switching between 21°C and 18°C. Disturbance from a varying outdoor temperature.

current ventilation temperature and the set point temperature. Based on this information it sends control signals to a shunt valve which then opens or closes to a certain percent of maximum opening. This regulates the water temperature sent to the heating coil which then again affects the ventilation temperature. As with the control of the room temperature this is also a classic feedback loop similar to the one seen in Figure 5.1. Although the system is now consisting of the shunt valve together with the heating coil.

Ventilation controller and its parameters

When the shunt connection in the ventilation system is to be controlled, hence ensuring a stable indoor temperature, one can use the same controller structure as shown in Figure 5.5. The implementation of this is shown in Figure 3.31. It should be noted that the PID controller block used to control the ventilation temperature also contains a rate limiter and a backlash block. The rate limiter ensures that the controlled valve cannot be changed arbitrarily fast and the backlash block introduces some backlash to the valve. This makes the modeled valve behave more similar to a real valve. In order to make the same controller structure as used to control the temperature of the room behave satisfying also for the ventilation system the controller parameters must be changed. The reason for

5. AUTOMATIC CONTROL

this is the fact that the system to be controlled is entirely different and hence reacts dissimilar to inputs. Using the same method as for the room controller, namely trial and error, the following parameters were chosen for the ventilation controller:

Kp_{hc} 0.2

Ki_{hc} 0.02

Kd_{hc} 0

Note that the proportional gain is set relatively low. The reason for this is the presence of time delays in the ventilation system due to the pipes and ducts. Choosing the gain to be higher causes the system to oscillate a while before settling down at the set temperature. If an even higher gain is chosen the controller loop becomes unstable. Example 5.3 and 5.4 verifies that the controller works as intended (i.e. it controls the temperature of the ventilation air).

Example 5.3 Constant ventilation temperature

To control the ventilation temperature to a fixed and preset value a control signal is sent to the shunt valve by the controller from Section 5.2.2. This signal will adjust the opening of the shunt valve. The valve would either open up, hold current gain or close. If the ventilated air is too cold the control signal will try to open the shunt valve more. The opposite would happen if the air is too hot, the valve will then reduce the valve opening. When the desired temperature is reached, the control signal will continue to produce a gain signal to the valve so that the ventilated air continuously returns to the desired temperature regardless of disturbances that might occur.

When the PI controller is used on valves one needs to make sure that the control signal does not violate certain maximum and minimum values. The most obvious is that the valve cannot be more than 100 % open, nor less than 0 % open. Another restriction for valves is that the valve opening has a maximum rate of change. It is physical impossible for the valve to exceed these limitations. The model in this report does not take into account the rate of change limits of the valve. Hence the valve in this report can move as quickly as desired. The limits of the shunt valve are as listed below.

Maximum opening 1 (100 %)

Minimum opening 0 (0 %)

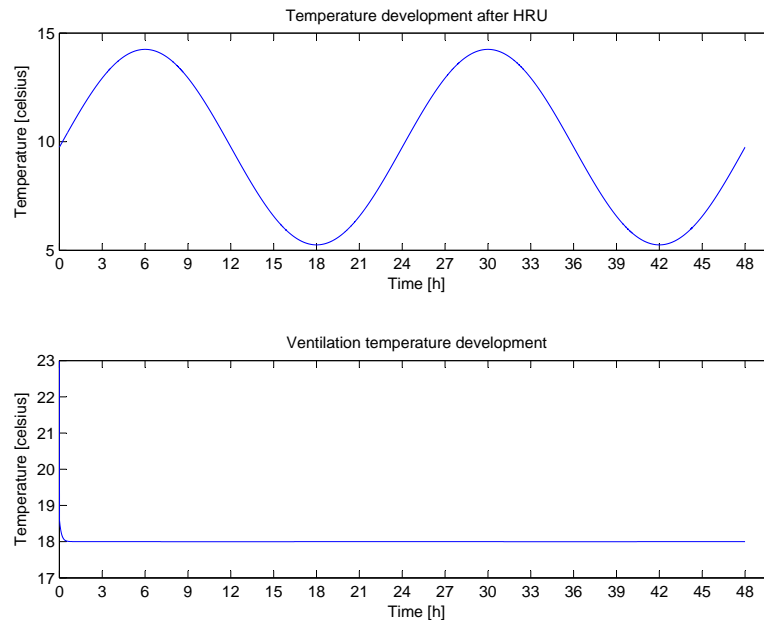


Figure 5.10: Fixed ventilation temperature with controller: Set temperature: 18°C. Disturbance from a varying outdoor temperature.

Simulation results of the ventilation system with the intent to hold a constant temperature of the ventilated air of 18°C considering a disturbance from the varying outdoor temperature can be seen in Figure 5.10. From this figure it is seen that the temperature rapidly approaches the given set temperature. It is also clear that the ventilation temperature stays at the given set point even though disturbances due to a varying outdoor temperature acts on the system.

Example 5.4 Varying ventilation temperature

Varying ventilation temperature might be used in some buildings. In Example 5.2 an algorithm to produce a varying set point was explained. A similar algorithm is used in this example also. Although now the set point switches between 19°C and 17°C. The exact same controller as explained in Section 5.2.2 is will be used in this example.

Simulation results of Saltstraumen elementary school with the intent to hold a ventilation temperature that switches between 19°C and 17°C, with a distur-

5. AUTOMATIC CONTROL

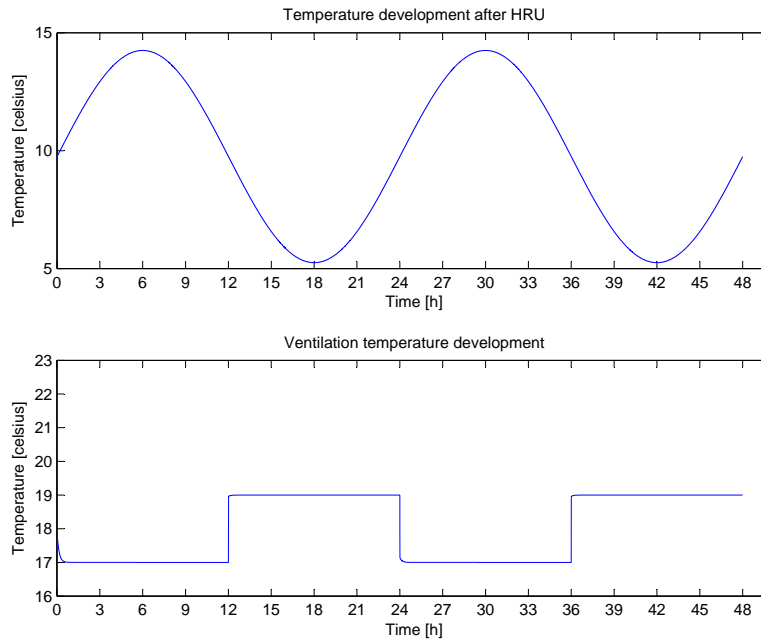


Figure 5.11: Varying ventilation air temperature with controller: Set temperature switching between 18°C and 16°C. Disturbance from a varying outdoor temperature.

bance from a varying outdoor temperature is seen in Figure 5.11. This figure clearly shows that the controller is able to control the temperature although varying set temperatures is used.

5.3 Model predictive control

Although a PID controller is able to hold the desired temperature it does not have the ability to calculate an energy optimal gain. This can be achieved by a controller that includes a cost function (e.g. MPC). This section introduces the concept of model predictive control and implements such a controller in order for it to control the temperature of the room.

Model predictive control (MPC) is one of the latest and most advanced controller structures used in process control today. It quickly became one of the most popular control strategies and now it is vastly used in process industry. Two

important reasons for its popularity may be its natural and logical way of including constraints, and its ability to “look into the future” and compute an optimal strategy to reach its goals without violating the given constraints. According to Maciejowski (2002) the main reason for MPC’s success is

1. It handles multi-variable control problems naturally.
2. It can take account of actuator limitations.
3. It allows operation closer to constraints, which frequently leads to more profitable operation.
4. There is plenty of time for the necessary on-line computations since its usage are on systems with relatively low control update rates.

MPC is often used in addition to the basic PID controller. When so, the MPC controller is usually wrapped around the PID controller(s). This is called cascade control. However in this report the PID controllers used to control the indoor temperature will be exchanged with MPC controllers. Then the energy usage in the rooms will be calculated for both of these cases and compared to see if the MPC control method is more energy efficient than the PID controller.

The traditional MPC is based on discrete signals. Thus the model found in Chapter 4 needs to be discretized. How this is done is elaborated in Section 5.3.3. The reason for mentioning it here is that it makes it easier to understand the concept of receding horizon.

5.3.1 Receding horizon

Predictive control is based on an idea called the “receding horizon”. This idea is thoroughly explained in Maciejowski (2002) and the reader is referred to this book for extensive details. However a short summary of the idea will be given here. As previously stated a model predictive controller includes an internal, discretized and possibly simplified model of the real system. This model is used to predict the future response of the real system, starting at the current time, over a future prediction horizon. Its goal is to reach a reference point within its horizon. There are many input trajectories that may lead to this reference point, but only one that promises the best behavior considering the constraints. Once this is found the first element of the input trajectory is applied to the real system. The system has now moved one time step and the entire procedure is repeated from the current position. The prediction horizon remains of the same length and is thus shifted one step further into the future, i.e. it remains the same length in front of the current time. This is also the reason why it is called a receding

5. AUTOMATIC CONTROL

horizon. In summary the following sequence of action can be assumed at time step k (Maciejowski 2002):

1. Obtain measurements, $y(k)$.
2. Compute the required plant input, $u(k)$.
3. Apply $u(k)$ to the plant.

5.3.2 State space representation

Usually an MPC controller is based on a discrete state-space model of the real system on the following form.

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) \quad (5.1)$$

$$\hat{y}(k|k) = C_y\hat{x}(k|k) \quad (5.2)$$

$$\hat{z}(k|k) = C_z\hat{x}(k|k) \quad (5.3)$$

where x is the state vector, u is the input vector, y is the vector of measured outputs and z is the vector of outputs to be controlled. In this report the model of the building found from the system identification procedure in Chapter 4 will be transformed into such a state-space representation. Thus this will be the model that the MPC controller makes its predictions from.

The model from Chapter 4 consists of a set of simple transfer functions where the transfer function from the ventilation to the room also consists of time delays. Unfortunately time delays cannot be represented directly in state space since they are not rational functions. Thus a rational function that approximates the time delay needs to be found. There are several approximation methods that exists but in this report the Padè approximation has been chosen. In the simplest case (first order Padè approximation) this can be written as follows

$$e^{-\tau s} = \frac{e^{-\frac{\tau}{2}s}}{e^{+\frac{\tau}{2}s}} \approx \frac{1 - \frac{\tau}{2}s}{1 + \frac{\tau}{2}s} \quad (5.4)$$

The function `pade(sys,N)` in `Matlab` finds the N 'th order Padè approximation of the system `sys`. This has been done for the transfer functions between the ventilation temperature and the temperature of the rooms. See Appendix D.7 for details of how this is done. The result of this approximation for room k133

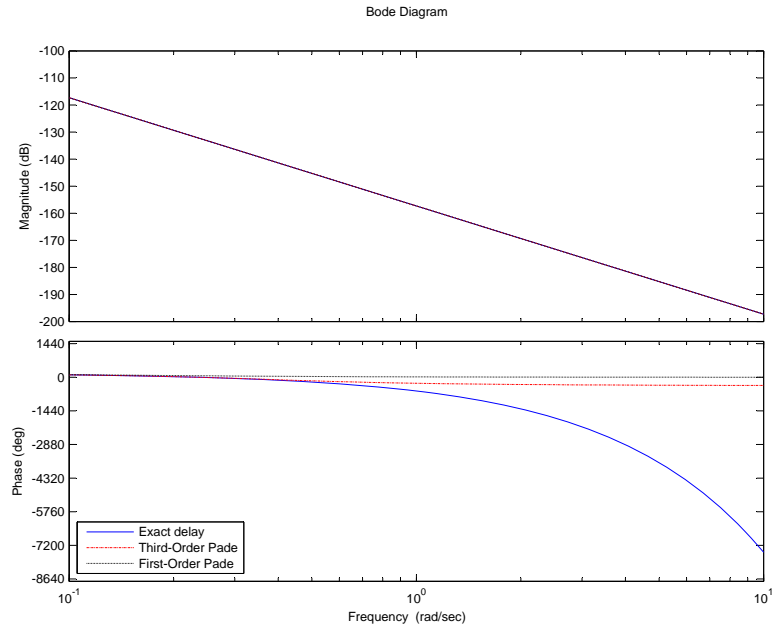


Figure 5.12: Comparison of the freq. response of the original model and the approximate model of room k133 found using bode plot

can be seen in Figure 5.12. This figure shows that the magnitude of the original model and the approximated model has an exact match. However, the phase deviates from the original model at approximately 0.3 rad/s for the first order approximation. The third order approximation does not deviate significantly until 0.7 rad/s. Hence the third order Padè approximation has an extended frequency band and is thus the preferred model approximation, although it means that the model becomes more complex, meaning that it consists of two additional poles compared to the first order Padè approximation. The transfer function from this approximation for room k133 is as follows

$$G(s) = \frac{-0.1617s^3 + 0.1449s^2 - 0.05407s + 0.008072}{11840000s^5 + 10620000s^4 + 3971000s^3 + 596000s^2 + 782.4s + 0.04992} \quad (5.5)$$

For the rest of the rooms in the building the third order Padè approximation produces a transfer function with the same structure, but with different values. The reader can run the m-file `mpc_preparations.m` to see the transfer functions found for the other rooms.

Having found a set of rational transfer functions for the entire building the next step is to express these transfer functions in state space form. A rational transfer

5. AUTOMATIC CONTROL

function can be transformed into state space using the `Matlab` command `ss(sys)`. Where `sys` is the system to be transformed into state space form. For room k133 the transfer function from the radiator to the room is defined as

$$G(s) = \frac{0.73171}{1013.5s + 1} \quad (5.6)$$

This result is taken from Chapter 4, Table 4.4, where also the other transfer functions of the system are found. To transform this into state space the `Matlab` function `ss(G(s))` can be used. The result of this transformation is shown below

$$\dot{x} = Ax + Bu \quad (5.7)$$

$$y = Cx + Du \quad (5.8)$$

where $A = -0.0009867$, $B = 0.03125$, $C = 0.0231$ and $D = 0$.

This state space representation only shows the connection between the radiator and the room temperature. However the ventilation and the outdoor temperature also affects the indoor temperature. The same procedure as above can be used to create state space representations from these transfer functions. The model of the room is now represented by three state space representations. However a single state space representation that includes every transfer function is preferred. This is possible using the `Matlab` command `ss_roomnumber = ss([TF_radiator, TF_outdoor__temp, TF_ventilation])`. The m-file `mpc_preparations.m` includes the code for this for every room and this can be seen in Appendix D.7. The file is also available on the CD attached to this report. In Appendix C.1 the continuous time state space representation found for each room can be seen.

5.3.3 Discretization of the state space

As mentioned earlier the model of the system which is going to be used in the MPC control scheme needs to be discretized. This is done using the `Matlab` function `c2d(sys, Ts)`. This function takes the system to be discretized and the chosen sample time as inputs. By default the method uses zero-order hold to convert the continuous signal into a discrete signal. The important thing to ensure when using this function is to chose the sample time according to the sampling theorem found in Proakis & Manolakis (2007). This means that `Ts` should be set as

$$Ts > 2 \cdot F_{max} \quad (5.9)$$

where F_{max} is the maximum frequency of the signal. In state space form this means that the sample time should be at least two times as fast as the smallest time constant of the system. From Table 4.4 in Chapter 4 it is clear that the lowest time constant is 985.05. Thus the sample time should be at least twice as fast as this. However this is a time constant found using system identification and hence it might be inaccurate. Considering this, the sample time is chosen to be 200 seconds. This should ensure that all the dynamics of the system is captured. But in `Matlab` the system is found to be not observable when the sampling time is larger than 117 seconds (see Section 5.3.4 for an explanation of observability). The cause of this is probably the Padè approximation of the time delay from the ventilation. A non observable system is not acceptable since the MPC control scheme needs to use an observer to estimate the states of the model. This observer is not able to estimate the states if the system is not observable. Hence the sampling time is chosen to be 50 seconds which is found to be well beneath the value where the system is no longer observable. With this sampling interval a discretized state space representation of the system is found. The reader is once again referred to the file `mpc_preparations.m` in Appendix D.7 to see how this was done for each room in this system. This file is also available in the CD included in this report and the reader is referred to this to see the results of the discretization for each room. These discrete state space representations ensures that the dynamics of the system has been captured and that the model of the room is observable even after the discretization.

5.3.4 Observer

The states of the state space representation found in the previous section are unknown. This is unfortunate since the MPC control scheme needs to know each state or an estimate of the states in order to function as intended. This is due to the presence of state feedback (see Chen (1999) for more details). Fortunately the unknown states can be estimated by a state estimator, also known as an observer. This is possible only if the system is observable (Chen 1999).

Observability

According to Chen (1999) a system represented in state space form is said to be observable if the *observability matrix* (Equation 5.10) found from the n dimen-

5. AUTOMATIC CONTROL

sional pair (\mathbf{A}, \mathbf{C}) has rank n (full column rank).

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (5.10)$$

In `Matlab` the observability matrix can be found for a state space system `sys` using the command `obsv(sys)`. When this has been done one needs to check if \mathbf{O} has full rank. In `Matlab` this is done by checking the difference `length(A)-rank(O)`. If this is different from 0 the system is not observable. Unfortunately it turns out that the `Matlab` function `obsv` is very vulnerable to numerical approximations and hence it is not a recommended method to check observability of a system. Another and more reliable way to check observability of a system is to use Theorem 6.06 in Chen (1999). This theorem states that it is possible to transform every state space equation into an equivalent transformation where the observable and the non-observable states are separated respectively into the matrices \mathbf{A}_o and $\mathbf{A}_{\bar{o}}$. See Chen (1999) for more details. The `Matlab` function `[Abar,Bbar,Cbar,T,k] = obsvf(A,B,C)` does this decomposition automatically and also produces a vector, `k`, where its sum is equal to the number of states in the matrix \mathbf{A}_o , the observable portion of the original \mathbf{A} matrix. If the number of states in the matrix \mathbf{A}_o is equal to the original it can be concluded that the original \mathbf{A} matrix is observable.

The previous procedure is carried out for each of the rooms since they now are individual systems, i.e. there is no connection between the room temperatures anymore (see Chapter 4). In the file `mpc_preparations.m` (see Appendix D.7) this is checked for every room. As mentioned earlier the results from this is that the state space systems for each room is observable as long as the discretization time step is chosen to be below 117 seconds. Since the system in this report is discretized with a time step of 50 seconds it is observable and hence it is possible to use a state observer to estimate the unknown states of the system (i.e. the system is observable).

Luenberger observer

In this report the Luenberger observer design was first chosen, however the tuning of this observer was found to be difficult. This section is a short summary of this

observer design and the results from the tuning of the observer. First the observer and the theory behind it will be presented. Second the result from the tuning of the observer will be shown and its shortcomings will be explained.

The Luenberger observer is described in Chapter 8.4 in Chen (1999) and the following description of this observer is mainly from this chapter. Given the following discrete linear system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (5.11a)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (5.11b)$$

The Luenberger observer is given as

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{l}(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k)) \quad (5.12)$$

assuming knowledge of \mathbf{A} , \mathbf{B} and \mathbf{C} . This can be written as

$$\hat{\mathbf{x}}(k+1) = (\mathbf{A} - \mathbf{lC})\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{l}\mathbf{y}(k) \quad (5.13)$$

Then the error between the actual state and the estimated state is defined to be

$$\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) \quad (5.14)$$

Differentiating $\mathbf{e}(k)$ and then substituting Equation 5.11a and Equation 5.13 into this equation gives

$$\begin{aligned} \mathbf{e}(k+1) &= \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) \\ &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) - (\mathbf{A} - \mathbf{lC})\hat{\mathbf{x}}(k) - \mathbf{B}\mathbf{u}(k) - \mathbf{l}(\mathbf{C}\mathbf{x}(k)) \\ &= (\mathbf{A} - \mathbf{lC})\mathbf{x}(k) - (\mathbf{A} - \mathbf{lC})\hat{\mathbf{x}}(k) = (\mathbf{A} - \mathbf{lC})(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) \end{aligned} \quad (5.15)$$

this can also be written as

$$\hat{\mathbf{e}}(k) = (\mathbf{A} - \mathbf{lC})\mathbf{e}(k) \quad (5.16)$$

5. AUTOMATIC CONTROL

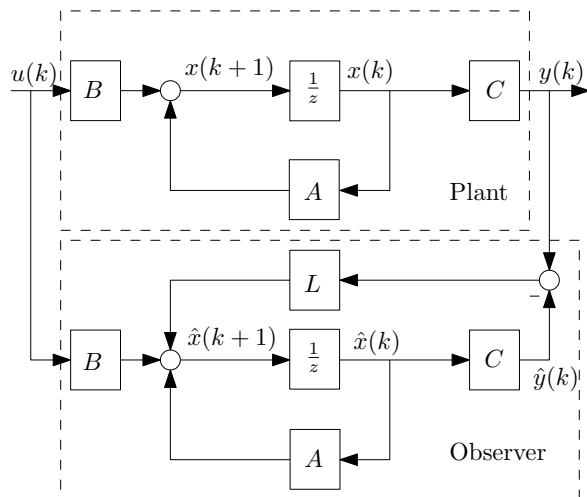


Figure 5.13: Luenberger observer (Magnussen 2010)

due to the definition of \mathbf{e} above.

The matrix \mathbf{l} is chosen freely by the designer of the observer and it should be chosen so that the eigenvalues of $(\mathbf{A} - \mathbf{I}\mathbf{C})$ all are negative. Then, if the largest eigenvalue is $-\sigma$, all entries of \mathbf{e} will approach zero at a rate of at least $e^{-\sigma t}$. This means that even if the initial error is large the estimated state will approach the actual state rapidly, i.e. the observer is stable. If the pair (\mathbf{A}, \mathbf{C}) is observable the eigenvalues of $(\mathbf{A} - \mathbf{I}\mathbf{C})$ can be chosen arbitrarily. As previously shown this is the case for the systems (each rooms) in this report. Figure 5.13 shows the structure of the discrete Luenberger observer together with the plant.

In order for the `Simulink` setup shown in Figure 5.13 to function one needs to chose a suitable matrix \mathbf{l} . Since the system is observable this can be chosen arbitrarily so that the matrix $(\mathbf{A} - \mathbf{I}\mathbf{C})$ only has negative eigenvalues. In addition to this the eigenvalues of the estimator should be faster than the desired eigenvalues of the state feedback. This can be done using the pole-placement `Matlab` algorithm `l = place(A', C', p)'`. Here \mathbf{p} is the chosen placement of the poles of $(\mathbf{A} - \mathbf{I}\mathbf{C})$. For every system the poles of the observer should be chosen so that they are faster than the fastest pole of the original system. Hence the poles are chosen to be

$$\mathbf{p} = -0.5 \cdot \min(|\text{eig}(\mathbf{A})|) \cdot [0 \ 0.15 \ 0.3 \ 0.45 \ 0.6 \ 0.75 \ 0.9] \quad (5.17)$$

This means that the slowest pole of the observer is more than twice as fast as the slowest pole of the system. However it turns out that this cannot be done, i.e.

the `Matlab` command `place` is not able to place the poles at these locations. The reason for this is that the selected poles are too close to each other. By changing the desired poles to

$$\mathbf{p} = -1 \cdot \min(|\text{eig}(\mathbf{A})|) \cdot [0 \ 0.15 \ 0.3 \ 0.45 \ 0.6 \ 0.75 \ 0.9] \quad (5.18)$$

no errors occur but a warning is issued where it is stated that the poles differ by more than ten percent from the specified location. This means that the poles could not be placed at these positions either. By trial and error the positions of the poles are changed until this warning is no longer present. This happens at the following pole placement

$$\mathbf{p} = -5 \cdot \min(|\text{eig}(\mathbf{A})|) \cdot [0 \ 0.15 \ 0.3 \ 0.45 \ 0.6 \ 0.75 \ 0.9] \quad (5.19)$$

This means that the observer poles are up to five times slower than the fastest pole of the original system. In addition to this the `l`-matrix found using the `Matlab` algorithm `place` consists of very large values. This would lead to a system that is not very robust which of course is not preferable and another observer should be considered to be used instead.

Kalman filter

The Kalman filter can be used both to filter out system disturbances (e.g. waves or wind) or measurement disturbances from a feedback loop and to estimate unknown states. In this section its primary goal is to estimate the states of the discrete system found in Section 5.3.3, but it is also used to filter out the measurement disturbances.

The following is a short introduction to the theory behind the Kalman filter and is mainly from Simon (2006). The Kalman filter estimates the state of a state space system given by the following equation

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (5.20)$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \quad (5.21)$$

This estimation is done by computing Equation 5.22 - 5.26 for each time step and these are the equations that constitute the Kalman filter (Simon 2006).

5. AUTOMATIC CONTROL

$$\mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \quad (5.22)$$

$$\mathbf{K}_k = \frac{\mathbf{P}_k^- \mathbf{C}_k^T}{\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k} \quad (5.23)$$

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}_{k-1} \mathbf{u}_{k-1} = \text{a priori state estimate} \quad (5.24)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-) = \text{a posteriori state estimate} \quad (5.25)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (5.26)$$

The following is a short description of the notation used in these equations. $\hat{\mathbf{x}}_k$ is the state estimate and \mathbf{P}_k is the covariance matrix of the estimate. The minus (-) and the plus (+) indicates respectively a priori and a posteriori estimates. \mathbf{K}_k is the optimal Kalman gain and \mathbf{I} is the identity matrix. In order for this algorithm to function properly one needs to enter some initial values for $\hat{\mathbf{x}}_0^+$ and \mathbf{P}_0^+ . Otherwise the algorithm would fail during its first step. These initial values are important tuning parameters in the Kalman filter. $\hat{\mathbf{x}}_0^+$ is the initial guess of the current states while \mathbf{P}_0^+ indicates how much one is to trust these initial states. The covariance matrices \mathbf{Q}_k and \mathbf{R}_k are defined as

$$\mathbf{Q}_k = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & Q_n \end{bmatrix} \quad (5.27)$$

$$\mathbf{R}_k = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & R_m \end{bmatrix} \quad (5.28)$$

$$(5.29)$$

where \mathbf{Q}_k is a square matrix with dimension equal to the state matrix \mathbf{A} , and the matrix \mathbf{R}_k is square with dimension equal to the number of measurements. Usually these matrices are diagonal and their parameters are set by the user to tune the Kalman filter in order to find good estimates of the output(s) and the state(s). \mathbf{Q}_k can be thought of as a matrix that states how much the model is to be trusted (i.e. how correct is the model compared to the real system). \mathbf{R}_k can be thought of as a matrix that states how much the measured variable is to be trusted (i.e. are the measurements correct?). When simulating, Equations

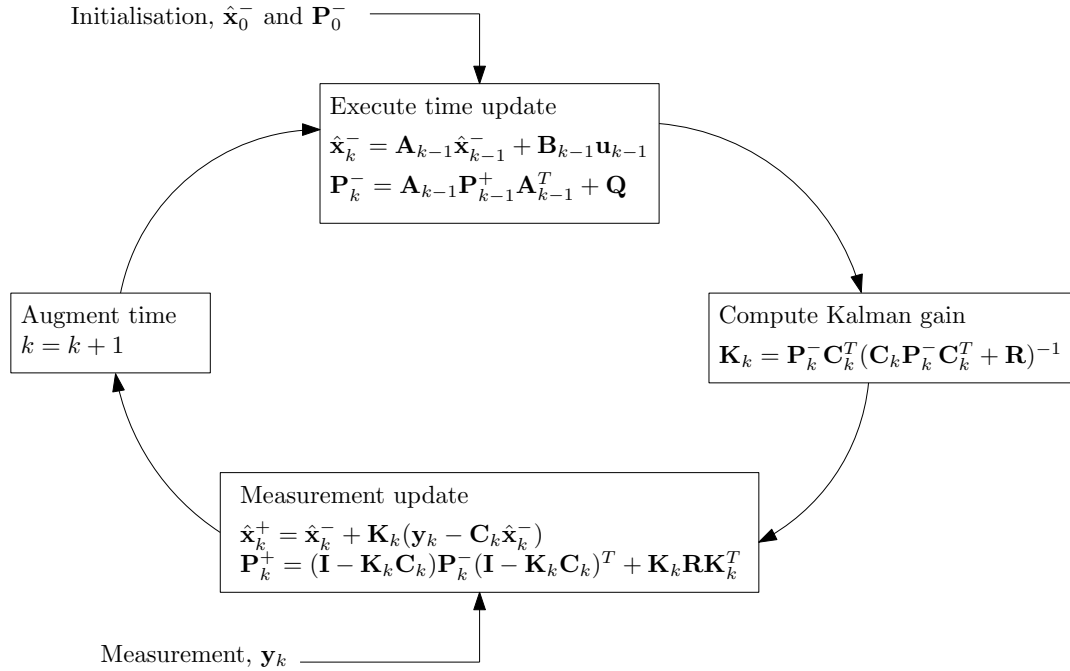


Figure 5.14: Kalman filter loop (Ersdal 2011)

5.22-5.26 are being calculated at every time step, i.e. it behaves like a for loop that is repeated at each time step. This loop is called the Kalman filter loop and is illustrated in Figure 5.14. If correctly tuned the Kalman filter will remove the measurement noise and within a short period of time find good estimates of the states and the output of the system. A more thorough introduction to the Kalman filter can be found in Simon (2006) or in Brown (1997).

In this report the Kalman filter has been implemented in `Simulink` using an `Embedded Matlab` function. Figure 5.15 shows how this function is included in the controller. The Kalman filter is labeled “Kalman filter” and the interior of this function can be seen in Appendix D.6. The observant reader might also notice that some noise has been added to the measurement (see Figure 5.15). The reason for this is that sensors used in real applications often is affected by noise. Hence the model now is a better approximation of a real system. The Kalman filter is tuned so that it filter out this noise and finds an approximation of the real measurement. As mentioned earlier it also finds good estimates of the system states $\hat{\mathbf{x}}_k$. Now that a discrete model of the system has been found and each state is estimated it is possible to implement the MPC control scheme. This will be elaborated in the next section.

5. AUTOMATIC CONTROL

5.3.5 MPC, theory and implementation

The theory presented in this section is mainly from Maciejowski (2002). This theory and the introduction made in Section 5.3 together with the receding horizon idea can be thought of as a short summary of the theory behind the MPC controller. This section will in addition to the theory behind the MPC control scheme explain the implementation of the MPC made in this report. In order to make the theory and the implementation of the controller as clear as possible the simplest MPC scheme is implemented first, then additional functions and capabilities are added one by one.

As briefly mentioned at the beginning of Section 5.3 the MPC control scheme includes a cost function. This cost function is used to penalize deviations of the predicted controlled outputs, $\hat{z}(k+i|k)$, from a reference trajectory (Maciejowski 2002). In an MPC controller this is the function that should be minimized and it can be defined as

$$V(k) = \sum_{i=H_w}^{H_p} \|\hat{z}(k+i|k) - r(k+i|k)\|_{Q(i)}^2 + \sum_{i=0}^{H_u-1} \|\Delta\hat{u}(k+i|k)\|_{R(i)}^2 \quad (5.30)$$

where $\hat{z}(k+i|k)$ denotes the predicted future value (at time $k+i$) of the variable which are to be controlled at time k . $r(k+i|k)$ is the future reference trajectory at time k and $\Delta\hat{u}(k+i|k)$ is the future estimated change of input at time k . H_p is the prediction horizon, H_w is the ‘window’ parameter and H_u is the control horizon. Note that $H_u \leq H_p$ and that the change of input signal $\Delta\hat{u}(k+i|k) = 0$ for $i \geq H_u$. $Q(i)$ and $R(i)$ is respectively the penalizing matrices for deviations from the reference trajectory (i.e. the error) and changes in the input vector. All of these variables affects the behavior of the controller, thus they can and should be used as tuning parameters. According to Maciejowski (2002) the cost function above can also be written as

$$V(k) = \|\mathcal{Z}(k) - \mathcal{T}(k)\|_{\mathcal{Q}}^2 + \|\Delta\mathcal{U}(k)\|_{\mathcal{R}}^2 \quad (5.31)$$

where

$$\begin{aligned} \mathcal{Z}(k) &= \begin{bmatrix} \hat{z}(k + H_w | k) \\ \vdots \\ \hat{z}(k + H_p | k) \end{bmatrix} \\ \mathcal{T}(k) &= \begin{bmatrix} \hat{r}(k + H_w | k) \\ \vdots \\ \hat{r}(k + H_p | k) \end{bmatrix} \\ \Delta \mathcal{U}(k) &= \begin{bmatrix} \Delta \hat{u}(k | k) \\ \vdots \\ \Delta \hat{u}(k + H_u - 1 | k) \end{bmatrix} \end{aligned}$$

and the matrices \mathcal{Q} and \mathcal{R} are given by

$$\mathcal{Q} = \begin{bmatrix} Q(H_w) & 0 & \dots & 0 \\ 0 & Q(H_w + 1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q(H_p) \end{bmatrix} \quad (5.32a)$$

$$\mathcal{R} = \begin{bmatrix} R(0) & 0 & \dots & 0 \\ 0 & R(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R(H_u - 1) \end{bmatrix} \quad (5.32b)$$

The matrices \mathcal{Q} and \mathcal{R} respectively penalizes deviation from the reference trajectory and the use of the controlled variable. Simply spoken one can say that higher values of the matrix \mathcal{Q} leads to a more aggressive regulator while higher values of the matrix \mathcal{R} leads to a more energy efficient regulator. The observant reader might notice that these two matrices are in some sense in conflict with each other and that it is difficult to end up with a controller that is very quick and at the same time uses small amounts of energy.

Solving the predictive control problem

In Maciejowski (2002) it is shown that $\mathcal{Z}(k)$ (defined above) has the following form

5. AUTOMATIC CONTROL

$$\mathcal{Z}(k) = \Psi x(k) + \Upsilon u(k-1) + \Theta \Delta \mathcal{U}(k) \quad (5.33)$$

where the matrices Ψ , Υ and Θ are defined as

$$\Psi = \begin{bmatrix} \mathbf{C}_z & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_z & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_z \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A}^{H_u} \\ \mathbf{A}^{H_u+1} \\ \vdots \\ \mathbf{A}^{H_p} \end{bmatrix} \quad (5.34a)$$

$$\Upsilon = \begin{bmatrix} \mathbf{C}_z & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_z & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_z \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \vdots \\ \sum_{i=0}^{H_u-1} \mathbf{A}^i \mathbf{B} \\ \sum_{i=0}^{H_u} \mathbf{A}^i \mathbf{B} \\ \vdots \\ \sum_{i=0}^{H_p-1} \mathbf{A}^i \mathbf{B} \end{bmatrix} \quad (5.34b)$$

$$\Theta = \begin{bmatrix} \mathbf{C}_z & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_z & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_z \end{bmatrix} \begin{bmatrix} \mathbf{B} & \dots & \mathbf{0} \\ \mathbf{AB} + \mathbf{B} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} \mathbf{A}^i \mathbf{B} & \dots & \mathbf{B} \\ \sum_{i=0}^{H_u} \mathbf{A}^i \mathbf{B} & \dots & \mathbf{AB} + \mathbf{B} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-1} \mathbf{A}^i \mathbf{B} & \dots & \sum_{i=0}^{H_p-H_u} \mathbf{A}^i \mathbf{B} \end{bmatrix} \quad (5.34c)$$

Code that generates these matrices automatically regardless of the prediction horizon, control horizon, number of inputs, states and outputs has been made to make the MPC controller as dynamical as possible. The code that generates these matrices can be seen in Appendix D.8. The tracking error is defined as

$$\mathcal{E}(k) = \mathcal{T}(k) - \Psi x(k) - \Upsilon u(k-1) \quad (5.35)$$

Combining Equation 5.35, 5.33 and 5.31 it is possible to rewrite the cost function $V(k)$ as

$$\begin{aligned} V(k) &= \|\Theta \Delta \mathcal{U}(k) - \mathcal{E}(k)\|_{\mathcal{Q}}^2 + \|\Delta \mathcal{U}(k)\|_{\mathcal{R}}^2 \\ &= \mathcal{E}(k)^T \mathcal{Q} \mathcal{E}(k) - 2\Delta \mathcal{U}(k)^T \Theta^T \mathcal{Q} \mathcal{E}(k) + \Delta \mathcal{U}(k)^T [\Theta^T \mathcal{Q} \Theta + \mathcal{R}] \Delta \mathcal{U}(k) \end{aligned} \quad (5.36)$$

It can be shown that the above equation has a special characteristics consisting of a constant term and two terms that respectively are multiplied by the term $\Delta \mathcal{U}^T(k)$ and its square. Thus $V(k)$ can be written as

$$V(k) = const - \Delta \mathcal{U}^T(k) \mathcal{G} + \Delta \mathcal{U}^T(k) \mathcal{H} \Delta \mathcal{U}(k) \quad (5.37)$$

where \mathcal{G} and \mathcal{H} are defined as

$$\mathcal{G} = 2\Theta^T \mathcal{Q} \mathcal{E}(k) \quad (5.38)$$

$$\mathcal{H} = \Theta^T \mathcal{Q} \Theta + \mathcal{R} \quad (5.39)$$

Since neither \mathcal{G} , nor \mathcal{H} depends on $\Delta \mathcal{U}(k)$ it is possible to find the optimal $\Delta \mathcal{U}(k)$ by calculating its gradient and set it to zero. This leads to the following optimal set of future inputs

$$\Delta \mathcal{U}(k)_{opt} = \frac{1}{2} \mathcal{H}^{-1} \mathcal{G} \quad (5.40)$$

However, finding the optimal solution should never be accomplished by calculating this expression. The reason for this is that \mathcal{H} often can be ill-conditioned (Maciejowski 2002). Ill-conditioned means that \mathcal{H} can become singular (non-invertible) if some of its entries are changed ever so slightly. To avoid the problem just stated another solution found by computing the solution as a least-squares problem is suggested. Since $\mathcal{Q} \geq 0$ and $\mathcal{R} \geq 0$ it is possible to compute their square-roots:

5. AUTOMATIC CONTROL

$$\begin{aligned} S_{\mathcal{Q}}^T S_{\mathcal{Q}} &= \mathcal{Q} \\ S_{\mathcal{R}}^T S_{\mathcal{R}} &= \mathcal{R} \end{aligned}$$

If the matrices \mathcal{Q} and \mathcal{R} is diagonal the square root can be found by taking the square root of each element. If however this is not the case the `Matlab` function `chol(A)` can be used. This function uses the Cholesky algorithm to compute the square-roots of the matrix \mathbf{A} (Maciejowski 2002). When the square roots of \mathcal{Q} and \mathcal{R} has been found one should use this to find an optimal input trajectory $\Delta\mathcal{U}(k)$. In Maciejowski (2002) it is shown that the least-squares solution of the equation

$$\begin{bmatrix} S_{\mathcal{Q}}\{\Theta\Delta\mathcal{U}(k) - \mathcal{E}(k)\} \\ S_{\mathcal{R}}\Delta\mathcal{U}(k) \end{bmatrix} = \mathbf{0} \quad (5.41)$$

is the optimal input trajectory $\Delta\mathcal{U}(k)$. This equation can be solved in `Matlab` by using the “backslash” operator:

$$\Delta\mathcal{U}(k)_{opt} = \begin{bmatrix} S_{\mathcal{Q}}\Theta \\ S_{\mathcal{R}} \end{bmatrix} \setminus \begin{bmatrix} S_{\mathcal{Q}}\mathcal{E}(k) \\ 0 \end{bmatrix} \quad (5.42)$$

Note that only the first part of this solution is used (i.e. only the first step is applied to the output). This was also mentioned in Section 5.3.1.

The `Simulink` implementation of the MPC scheme explained above can be seen in Figure 5.15. The optimal change of input is calculated using an `Embedded Matlab function` in `Simulink` (labeled `K_MPC` in Figure 5.15). Appendix D.9 shows the interior of this function. The output from this implementation, labeled $u(k)$ in Figure 5.15, is the temperature of the water sent to the radiator while the input labeled $y(k)$ is the discretized indoor temperature. That is the indoor temperature sampled at the same sample rate as the MPC (see Section 5.3.3). This means that the PID controller seen in Figure 5.7 is simply replaced by the MPC control scheme from 5.15.

Feed-forward from measured disturbances

It has been assumed that the current disturbances from the outdoor temperature and the ventilation is known due to sensors that measures these variables. This is

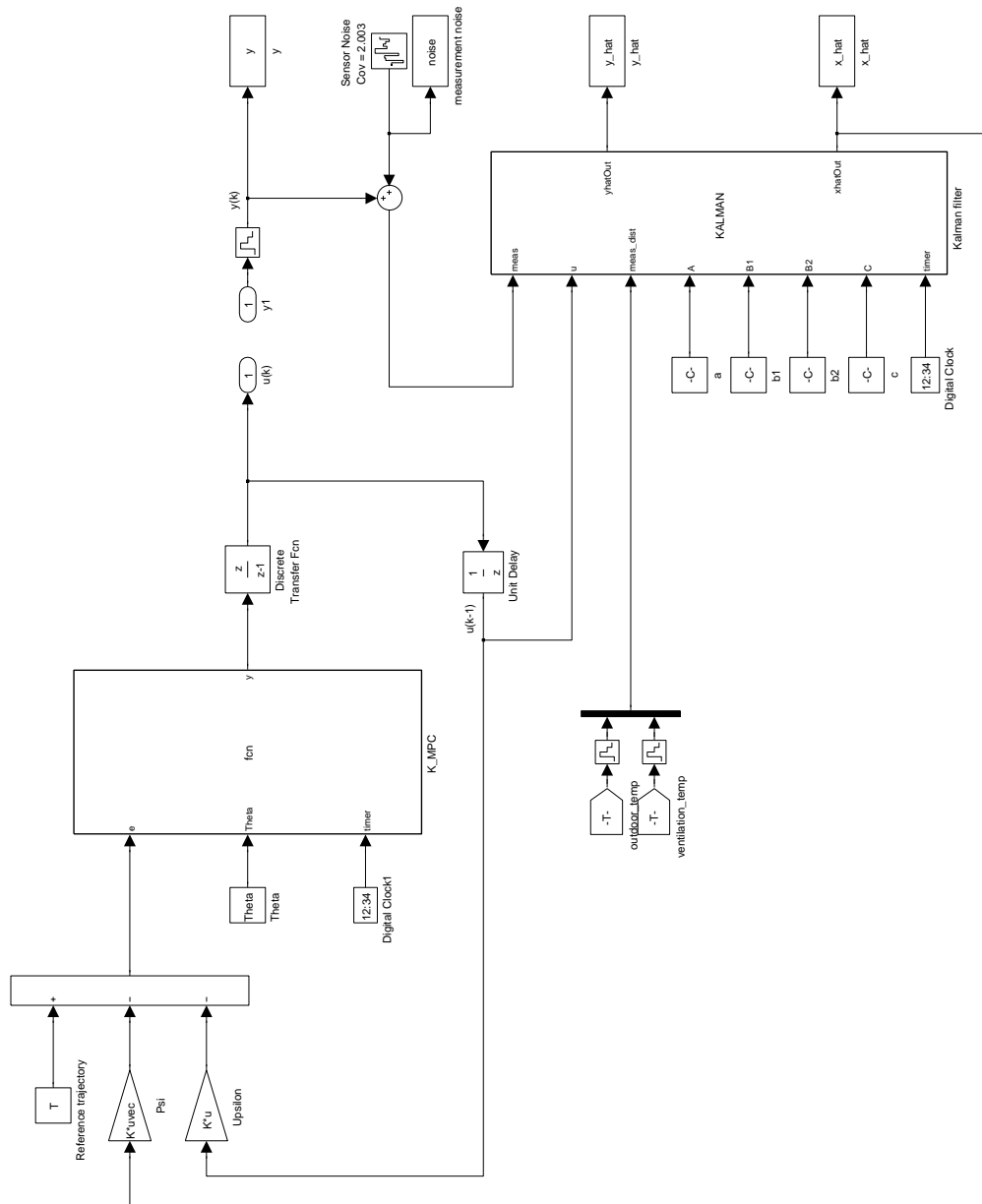


Figure 5.15: MPC control scheme without feed-forward and constraints

5. AUTOMATIC CONTROL

a reasonable assumption and the additional information provided by these sensors can and should be exploited. As stated earlier in this chapter; known disturbances can be canceled out by a feed-forward connection earlier than what will be the case with only a feedback connection. Thus the MPC should be extended with a feed-forward connection. As it turns out it is a fairly easy task to implement feed-forward into the MPC control scheme. The following is a short description of how this can be done.

First the discrete state-space model from Equation 5.1 has to be modified into the following equation

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) + B_d d_m(k) \quad (5.43)$$

$$\hat{y}(k|k) = C_y \hat{x}(k|k) \quad (5.44)$$

$$\hat{z}(k|k) = C_z \hat{x}(k|k) \quad (5.45)$$

where the B-matrix from the original state-space system has been split into two parts. One which is multiplied by the measured disturbances vector $d_m(k)$, and one which is multiplied with the input $u(k)$. Following from this change, $\mathcal{Z}(k)$ needs to be restated as

$$\mathcal{Z}(k) = \Psi x(k) + \Upsilon u(k-1) + \Theta \Delta \mathcal{U}(k) + \Xi \mathcal{D}_m(k) \quad (5.46)$$

where

$$\mathcal{D}_m = \begin{bmatrix} d_m(k) \\ \hat{d}_m(k+1|k) \\ \hat{d}_m(k+2|k) \\ \vdots \\ \hat{d}_m(k+H_p-1|k) \end{bmatrix} \quad (5.47)$$

and

$$\Xi = \begin{bmatrix} C_z B_d & \mathbf{0} & \dots & \mathbf{0} \\ C_z A B_d & C_z B_d & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ C_z A^{H_p-1} B_d & C_z A^{H_p-2} B_d & \dots & C_z B_d \end{bmatrix} \quad (5.48)$$

As seen from the equation above the predicted output is calculated with the assumption that the disturbance or an estimate of the disturbance is known some time into the future. Of course the behavior of future disturbances is usually not known, hence an approximation of this disturbance is usually applied. The most common thing to do is to assume that the disturbance will remain as it was for the last measurement (Maciejowski 2002). This is what has been done in this report and thus $d_m(k) = \hat{d}_m(k+1|k) = \dots = \hat{d}_m(k+H_p-1|k)$. Finally the tracking error needs to be redefined to include the feed-forward connection. This is done simply by including $\Xi \mathcal{D}_m(k)$ in Equation 5.35 which leads to

$$\mathcal{E}(k) = \mathcal{T}(k) - \Psi x(k) - \Upsilon u(k-1) - \Xi \mathcal{D}_m(k) \quad (5.49)$$

The `Simulink` model following from these changes can be seen in Figure 5.16. The `Embedded Matlab function` labeled `K_MPC` is the exact same as for the controller without feed-forward connection. The new `Matlab` function labeled “Feedforward” generates the matrix \mathcal{D}_m from the measurements of the disturbance. The interior of this function can be seen in Appendix D.11.

Constraints

When there are constraints present $\Delta \mathcal{U}(k)$ will be restricted by these constraints. Hence these constraints must be taken into account when calculating $\Delta \mathcal{U}(k)$. Typical constraints are actuator slew rates, actuator ranges and constraints on the controlled variables. According to Maciejowski (2002) these can be written as

$$E \begin{bmatrix} \Delta \mathcal{U}(k) \\ 1 \end{bmatrix} \leq 0 \quad (5.50)$$

$$F \begin{bmatrix} \mathcal{U}(k) \\ 1 \end{bmatrix} \leq 0 \quad (5.51)$$

$$G \begin{bmatrix} \mathcal{Z}(k) \\ 1 \end{bmatrix} \leq 0 \quad (5.52)$$

where E , F and G are matrices of suitable dimensions and $\Delta \mathcal{U}(k)$, $\mathcal{U}(k)$ and $\mathcal{Z}(k)$

5. AUTOMATIC CONTROL

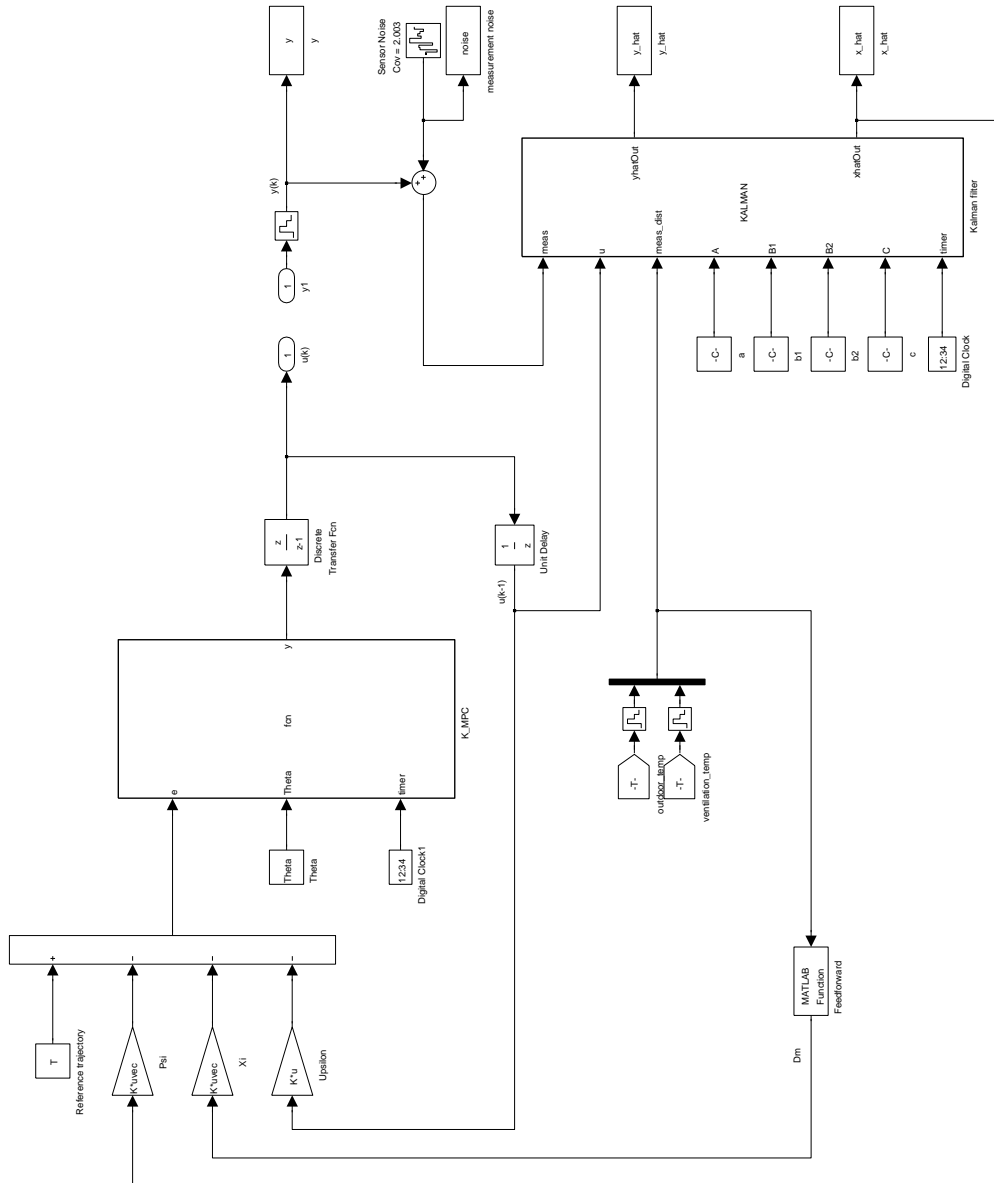


Figure 5.16: MPC control scheme with feed forward

is defined as

$$\Delta\mathcal{U}(k) = \begin{bmatrix} \Delta\hat{u}(k|k)^T \\ \vdots \\ \Delta\hat{u}(k + H_u - 1|k)^T \end{bmatrix} \quad (5.53a)$$

$$\mathcal{U}(k) = \begin{bmatrix} \hat{u}(k|k)^T \\ \vdots \\ \hat{u}(k + H_u - 1|k)^T \end{bmatrix} \quad (5.53b)$$

$$\mathcal{Z}(k) = \begin{bmatrix} \hat{z}(k + H_w|k)^T \\ \vdots \\ \hat{z}(k + H_p|k)^T \end{bmatrix} \quad (5.53c)$$

All of these constraints needs to be expressed using $\Delta\mathcal{U}$. According to Maciejowski (2002) this can be done by rewriting the inequalities from Equation 5.50 into the following form

$$\begin{bmatrix} W \\ \mathcal{F} \\ \Gamma\Theta \end{bmatrix} \Delta\mathcal{U}(k) \leq \begin{bmatrix} w \\ -\mathcal{F}_1 u(k-1) - f \\ -\Gamma[\Psi\hat{x}(k) + \Upsilon u(k-1) + \Xi D_m(k)] - g \end{bmatrix} \quad (5.54)$$

The reader is referred to Maciejowski (2002) for a detailed explanation of this transformation. The above equation can be written as

$$\Omega\theta \leq \omega \quad (5.55)$$

where

$$\Omega = \begin{bmatrix} W \\ \mathcal{F} \\ \Gamma\Theta \end{bmatrix} \quad (5.56)$$

$$\theta = \Delta\mathcal{U}(k) \quad (5.57)$$

5. AUTOMATIC CONTROL

and

$$\omega = \begin{bmatrix} w \\ -\mathcal{F}u(k-1) - f \\ -\Gamma[\Psi\hat{x}(k) + \Upsilon u(k-1) + \Xi D_m(k)] - g \end{bmatrix} \quad (5.58)$$

The cost function to be minimized by the controller is still the same as before;

$$\min_{\Delta\mathcal{U}} \Delta\mathcal{U}^T(k)\mathcal{H}\Delta\mathcal{U}(k) - \Delta\mathcal{U}^T(k)\mathcal{G} \quad (5.59)$$

but it is now subject to the inequality constraints defined in Equation 5.54 or equivalently 5.55. Fortunately this minimization is on the form

$$\min_{\theta} \frac{1}{2}\theta^T\Phi\theta + \phi^T\theta \quad (5.60)$$

subject to

$$\Omega\theta \leq \omega \quad (5.61)$$

This is known as the *Quadratic Programming* (QP) problem and can be solved using standard algorithms. In `Matlab` one can use the command `quadprog` to solve such a problem. For our minimization problem the quadratic problem is stated as

$$\min_{\Delta\mathcal{U}} \frac{1}{2}\Delta\mathcal{U}^T(2\mathcal{H})\Delta\mathcal{U} + (-\mathcal{G})^T\Delta\mathcal{U} \quad (5.62)$$

subject to

$$\Omega\theta \leq \omega \quad (5.63)$$

Note that in order to transform Equation 5.59 into the exact same form as Equation 5.60 \mathcal{H} needs to be multiplied by 2 and \mathcal{G} needs to be multiplied by -1 . For

the same reason as stated in the unconstrained problem (\mathcal{H} can be ill-conditioned) it is preferable to create an algorithm that solves the QP problem in “square root” form

$$\min_{\Delta \mathcal{U}} \left\| \begin{bmatrix} S_{\mathcal{Q}}\{\Theta \Delta \mathcal{U}(k) - \mathcal{E}(k)\} \\ S_{\mathcal{R}} \Delta \mathcal{U}(k) \end{bmatrix} \right\|^2 \quad (5.64)$$

subject to

$$\Omega \theta \leq \omega \quad (5.65)$$

However such an algorithm has not been created for this report. The reason for this is both due to time constraints and the fact that in this case the `Matlab` function `quadprog` solves the problem without generating any errors. Hence in order to solve the quadratic problem this `Matlab` command has been chosen. To implement the MPC scheme with constraints in `Simulink` the `Embedded Matlab function` from the previous design needs to be altered by including the constraints and finding the optimal solution $\Delta \mathcal{U}(k)_{opt}$ by solving the QP problem. The new interior design of this embedded function can be seen in Appendix D.10 while Figure 5.17 and Figure 5.18 shows the implementation in `Simulink`.

Challenges with embedded Matlab functions

There are some challenges that needs to be worked out in order for the above MPC schemes to function properly with the `Embedded Matlab function` blocks from `Simulink`. First, embedded `Matlab` functions does not have the ability to see the variables present in the workspace. It only recognizes variables that are either generated within the function or variables that are inputs to the function. Thus every variable that are used by the `Embedded Matlab function` needs to be declared within the function or defined as an input to the function. Second, not every function is supported for code generation. This includes the function `quadprog` which is used in this report. `Matlab` generates an error if this function is called within an `Embedded Matlab function`. A solution to avoid this problem is to declare the function to be extrinsic. Using such a deceleration ensures that the function is not compiled, but dispatched into `Matlab` for execution during simulation of the model. The disadvantage of declaring a function extrinsic is that `Simulink` no longer creates C code of the model, hence it can only be used

5. AUTOMATIC CONTROL

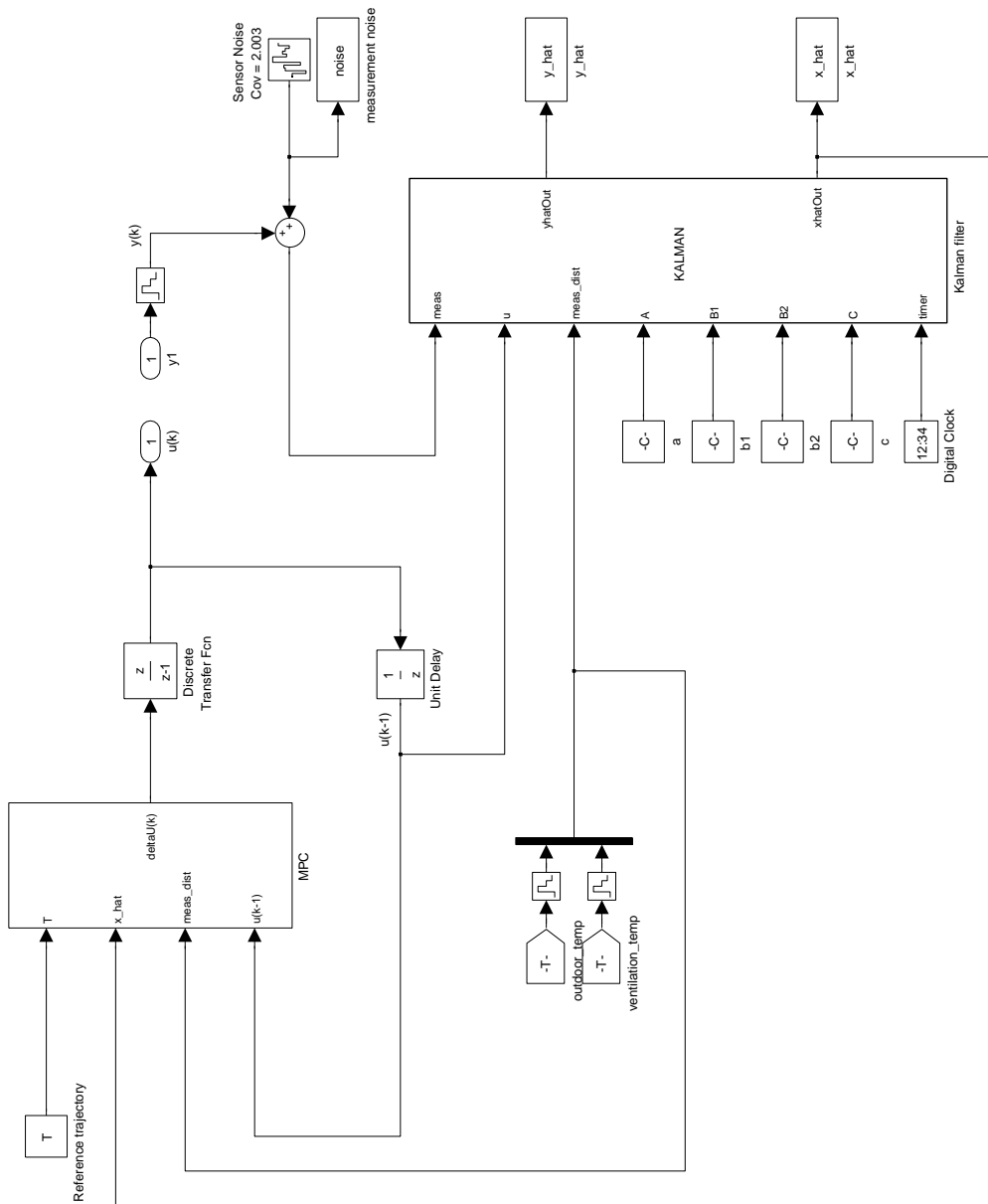


Figure 5.17: MPC control scheme with feed forward and constraints

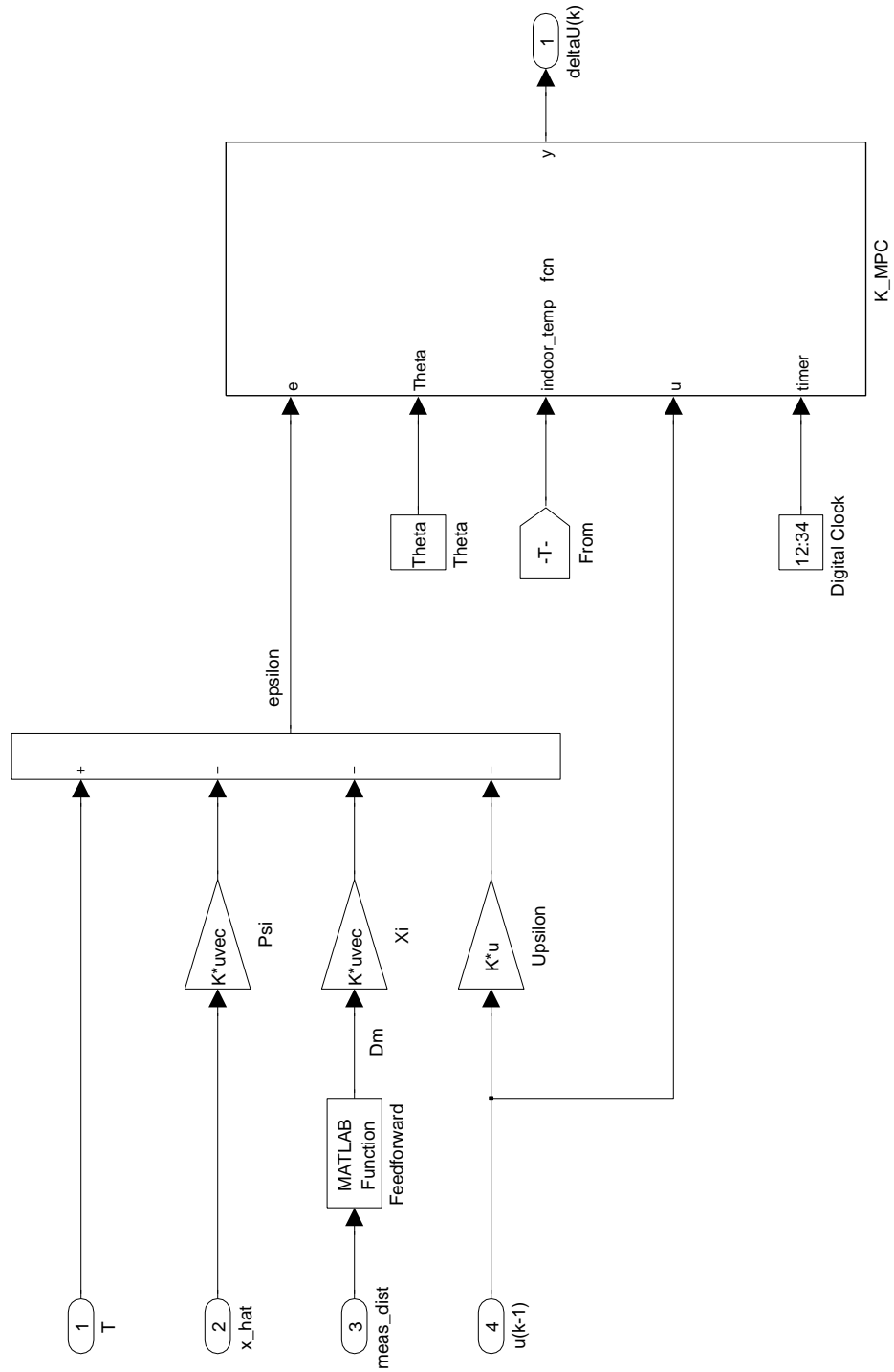


Figure 5.18: The internal structure of the block labeled MPC in Figure 5.17

5. AUTOMATIC CONTROL

for simulation. If the model is to be used on a real system (e.g. a real building) C code needs to be generated and the `Embedded Matlab function` can not be used. If this is the case then each `Embedded Matlab function` that includes functions that are called extrinsic needs to be replaced by the `Simulink block` called `Matlab function`. However this function has only one input which needs to include every external variable used within the function. In addition to this the `Matlab function` does not accept matrices but only vectors as input and hence each matrix to be included needs to be dismantled before the input and put together again inside the function. This is of course very cumbersome and since this report only runs simulations of the model and is not to be used within a real application the `Embedded Matlab function` has been chosen.

Tuning

Due to time constraints extensive tuning of the MPC has not been done, but parameters that works satisfying, but not optimal, for room k133 has been found for both the Kalman filter and the MPC. A description of these parameters and their impact on the result have been given earlier in Section 5.3 and will not be restated here. The parameter values chosen for the Kalman filter can be seen below

$$\mathbf{Q}_k = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (5.66)$$

$$\mathbf{R}_k = 5 \quad (5.67)$$

$$\mathbf{P}_0^+ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.68)$$

$$\hat{\mathbf{x}}_0^+ = [180 \ 150 \ 0 \ 0 \ 0 \ -5 \ 350]^T \quad (5.69)$$

For the MPC the following tuning parameters have been chosen

$$\mathcal{Q} = \begin{bmatrix} 100 & 0 & \dots & 0 \\ 0 & 100 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 100 \end{bmatrix} \quad (5.70)$$

$$\mathcal{R} = \begin{bmatrix} 10 & 0 & \dots & 0 \\ 0 & 10 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 10 \end{bmatrix} \quad (5.71)$$

$$H_p = 35 \quad (5.72)$$

$$H_u = 35 \quad (5.73)$$

$$H_w = 1 \quad (5.74)$$

where the dimension of \mathcal{Q} and \mathcal{R} respectively is determined by the length of the prediction horizon and the control horizon. For the prediction and control horizon given above the dimensions of \mathcal{Q} and \mathcal{R} is thus 35x35.

Verification of the MPC controller

The MPC controller from the previous section has to be tested for different scenarios. This section includes two examples that shows that the control scheme indeed does function as intended. First it will be shown that it is able to hold a desired indoor set temperature, then it will be shown that it also is able to control a room temperature that is varying, e.g. a room that should hold a certain temperature at daytime and another temperature at nighttime. This is the same procedure as was done to verify the PID controller

Example 5.5 Constant room temperature

This example shows how the MPC controller from Section 5.3 is able to hold the temperature of room k133 at the desired temperature. In this example the set temperature is 21°C. Figure 5.19 shows how the indoor temperature holds a fixed temperature even though there are a considerable disturbance from the outdoor temperature acting on the room. The outdoor disturbance is the same as was used for the verification of the PID controller and can be seen in Figure

5. AUTOMATIC CONTROL

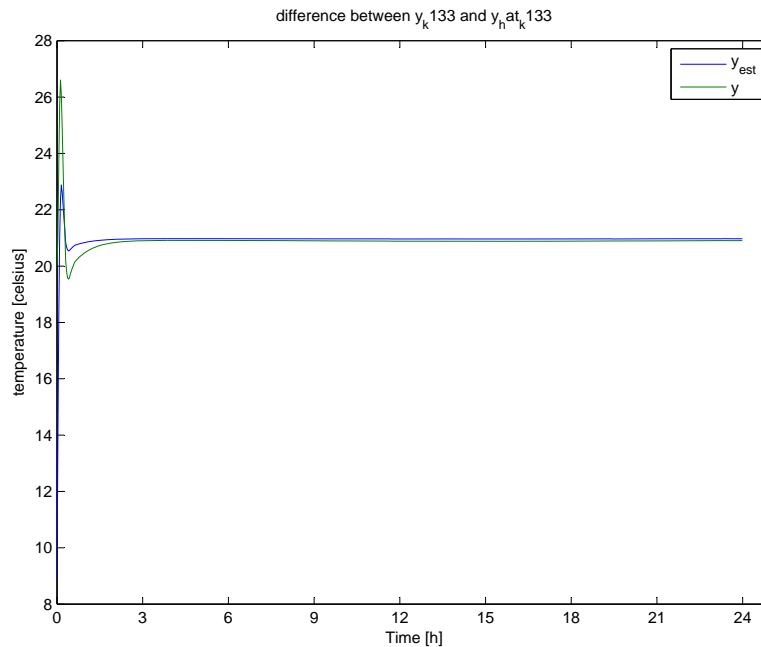


Figure 5.19: Fixed room temperature and estimated room temperature with MPC controller: Set temperature: 21°C. Disturbance from a varying outdoor temperature.

5.8. Figure 5.19 shows both the estimated indoor temperature and the actual indoor temperature. Studying the figure reveals that there is a slight deviation between the estimated and the real temperature. This deviation is due to inaccuracy or simplifications during the system identification in Chapter 4. In Chapter 7 this is discussed more thoroughly.

Example 5.6 Varying room temperature

This example shows how the MPC controller from Section 5.3 is able to vary the temperature of room k133 according to changes in the desired temperature. In this example the set temperature is chosen to be 21°C at daytime and lowered to 18°C at nighttime. Figure 5.20 shows how the indoor temperature holds the desired temperature even though the room is affected by the outdoor disturbances seen in Figure 5.8. This figure also reveals that the estimated temperature is slightly higher than the real indoor temperature. Again this is

5.3 Model predictive control

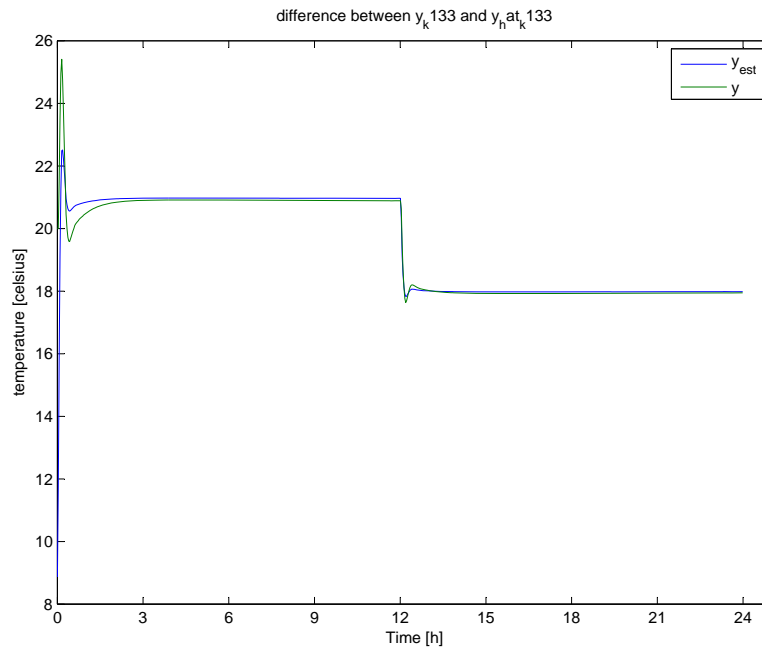


Figure 5.20: Varying room temperature and estimated room temperature with MPC controller. Daytime temperature: 21°C. Nighttime temperature: 18°C. Disturbance from a varying outdoor temperature. $H_p = H_u = 35$.

due to the fact that the model from Chapter 4 is not a perfect representation of the original system.

The reader might notice that even though the controller is able to hold the desired temperature it experiences some oscillation before settling down at the desired temperature. This oscillation can be avoided with a better tuned controller but due to time constraints extensive attempts of this has not been done in this report. However it was attempted to increase the horizons H_p and H_u . The results from this simulation can be seen in Figure 5.21. Comparing this figure with Figure 5.20 reveals that increasing the horizons from 35 to 100 has an almost negligible effect on the temperature response. Further attempts to make the room temperature act as desired should be done unfortunately this has not been done due to time constraints. One can however assume that further restrictions on the use of the controlled variable would decrease this oscillation although this would cause the system to react slower. Also editing the initial values selected in the Kalman filter can help the controller to avoid these oscillations.

5. AUTOMATIC CONTROL

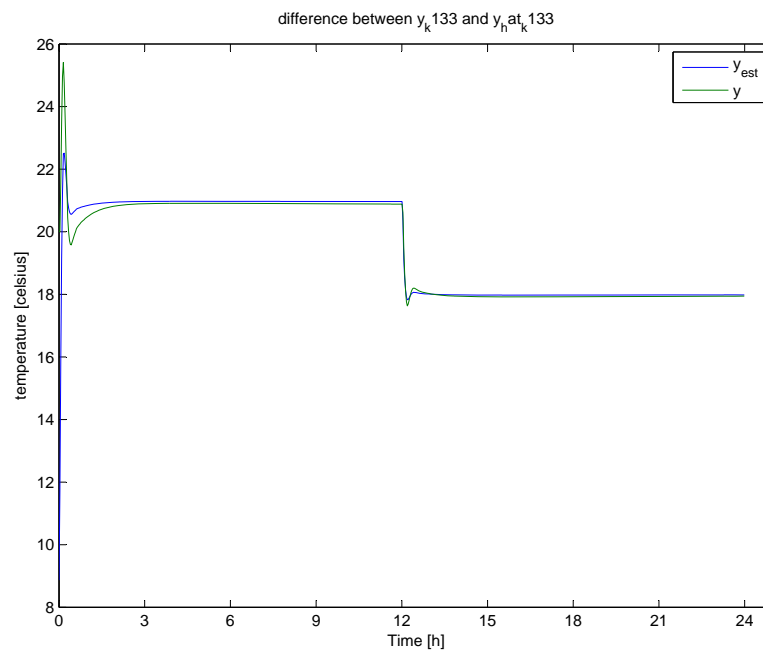


Figure 5.21: Varying room temperature and estimated room temperature with MPC controller. Daytime temperature: 21°C. Nighttime temperature: 18°C. Disturbance from a varying outdoor temperature. $H_p = H_u = 100$.

6

Energy consumption

6.1 Calculation of energy consumption

When calculating energy consumption of any system it is convenient to express this as watt [W] or as kilowatt hours [kWh] since these are the most commonly used units describing energy usage. Watt is defined as one joule per second:

$$W = \frac{J}{s} = \frac{N \cdot m}{s} = \frac{kg \cdot m^2}{s^3} \quad (6.1)$$

To find the energy consumption of a component one needs to find a way to express the difference in energy input and energy output of that component and express this as watt or kilowatt hours. This will be done in the following two sections.

6.1.1 Energy usage in the ventilation system

In the ventilation system the following components consumes energy:

- Fan (one for supply air and one for return air)
- Circulation pump
- Heat recovery unit
- Heating coil

6. ENERGY CONSUMPTION

Table 6.1: Energy consumption of components in the ventilation system

Unit	Watt	Kilowatt hours per day
Fan	5.5 kW	132 kWh/day
Heat recovery unit	60 W	1.44 kWh/day
Circulation pump	60 W	1.44 kWh/day

It is easy to find the energy usage of the fans, the circulation pump and the heat recovery unit since these are fixed. Usually these components have the energy usage written in the specification sheet, and some of them even have it written on the component itself. The energy consumption of the fan, the circulation pump and the heat recovery unit of the system used in this report can thus be found by investigating each components specification sheet. Table 6.1 is a list over these components and their energy consumption.

If one needs to find how much energy one of these components consumes over a certain period of time, say for instance a period of one month then one needs only to multiply the amount of kilowatt the component consumes by the number of hours in that month. An example where the number of days in a particular month is 30 gives the following equation when calculating the energy usage of the fan which produces the supply air: $5.5kW \cdot 24hours \cdot 30days = 3960kWh/month$. To calculate the energy usage of the heating coil is not as straight forward as with the above mentioned components. How to calculate this will be explained in the following section.

Heating coil

The heating coil transfers energy from the water flowing through it via the metal of the coil and to the air flowing through it. This energy flow can be explained by the second law of thermodynamics (see Section 3.1.2). This means that the water flowing through the heating coil will loose some energy on the way through the component. Thus the water temperature flowing out of the heating coil will have a different temperature than the water flowing into it. Since both of these temperatures are known it is possible to express the energy consumption of the heating coil by an equation including the temperature difference. One needs to obtain an expression for energy in the water. The easiest way to do this is by looking at the units of the water properties (e.g. specific heat, density, temperature and flow) and find a combination of these that gives the same units as energy (i.e. W or J/s). Using this approach leads to the following equation for energy in water:

6.1 Calculation of energy consumption

$$W = qw_{hc} \cdot cp_w \cdot \rho_w \cdot T \quad (6.2)$$

where

qw_{hc} = flow of water through heating coil ($\frac{m^3}{s}$)

cp_w = specific heat capacity of water ($\frac{J}{kg \cdot K}$)

ρ_w = density of water ($\frac{kg}{m^3}$)

T = temperature (K)

When finding the energy consumption of the heating coil the above equation can be used, albeit with one difference. The temperature should be replaced by the temperature difference between the water flowing into the heating coil and the water flowing out of it. Thus Equation 6.2 changes to:

$$W = qw_{hc} \cdot cp_w \cdot \rho_w \cdot \Delta T(t) \quad (6.3)$$

where

$\Delta T(t)$ = temperature difference (K)

It should be noted that $\Delta T(t)$ is a function of time and hence it is not constant but varies with time. The reason for this is that the outdoor temperature changes with time. Therefore, to find out the amount of energy the heating coil consumes over a certain period of time, one needs to sum the energy usage of the coil at every time-step.

It is also possible to calculate energy consumption by inspecting how the energy of the air changes over the heating coil. Although now we need to take the heating coils efficiency into account since the actual energy consumption of the heating coil is the change of energy in the water flow, not the flow of air. This leads to the following equation

$$W = \left(\frac{1}{\eta}\right) \cdot qa_{hc} \cdot cp_a \cdot \rho_a \cdot \Delta T(t) \quad (6.4)$$

where

6. ENERGY CONSUMPTION

Table 6.2: Energy consumption of components in the radiator circuit

Unit	Watt	Kilowatt hours per day
Circulation pump	60 W	1.44 kWh/day

qa_{hc} = flow of air through heating coil ($\frac{m^3}{s}$)

cp_a = specific heat capacity of air ($\frac{J}{kg \cdot K}$)

ρ_a = density of air ($\frac{kg}{m^3}$)

ΔT = temperature difference (K)

η = efficiency of the heating coil (dimensionless, value between 0 and 1)

Both Equation 6.2 and 6.4 leads to the same result, namely the energy consumption of the heating coil. The heating coils efficiency will be regarded as 100 % in this report (i.e. $\eta = 1$).

6.1.2 Energy usage in the radiator system

In the radiator system the following components consumes energy:

- Circulation pump
- Each radiator

The energy consumption of the circulation pump is fixed. Hence it can be found in the specification sheet or it might be written on the component itself. The energy consumption of this component can be seen in Table 6.2.

Radiators

Calculating the amount of energy used by the radiators is, as for the heating coil, not as intuitive. In this report the electrical analogy used to model the room is utilized for this purpose. The radiators are modeled as seen in Section 3.3.7 and the output of these models are the radiator temperature. Information about this temperature, the temperature of the room and the resistors between these two temperatures can be used to calculate the energy consumption. This is possible since the temperature in the room model is represented by voltage. Thus Joule's first law, which states the amount of power dissipated over a resistor in terms of

the current running through it and its resistance, can be used to calculate the energy consumption (Narasimhan 1999):

$$P = I^2 \cdot R = \frac{U^2}{R} \quad (6.5)$$

where P is the energy consumption given in watt, I is the current, U is the voltage and R is the resistor value. The voltage across the resistors in the room model is the temperature difference between the radiator temperature and the indoor temperature. The resistor values are calculated using the equations found in Appendix A.1. Thus the power dissipated by the radiator can be expressed by this equation:

$$P = \frac{U^2}{R} = \frac{(rad_temp - room_temp)^2}{R_{conv_rad}} + \frac{n \cdot (rad_temp - room_temp)^2}{R_{rad_rad}} \quad (6.6)$$

where

rad_temp = radiator temperature (°C)

room_temp = room temperature (°C)

R_{conv_rad} = convective resistance between the radiator and the indoor air (Ω)

R_{rad_rad} = thermal radiation resistance between the radiators and the wall (Ω)

n = number of walls and ceiling in the room (dimensionless)

6.2 Energy savings

Why is it so important for corporations, companies and even single families to minimize energy consumptions in buildings? The answer to this is quite easy; energy is expensive. Corporations generally want the net income to be as large as possible. Minimizing the operational cost of buildings would be a good start when trying to accomplish this. Lowering the energy consumption in buildings also has an environmental aspect, but in this report only the economical aspect will be explored.

6. ENERGY CONSUMPTION

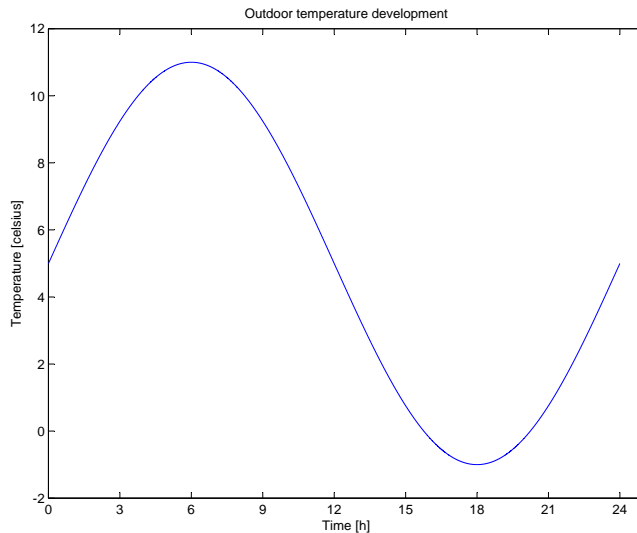


Figure 6.1: Outdoor temperature development

6.2.1 Energy savings in the ventilation system

In this section the ventilation system and its energy consumption will be compared for different scenarios. This will be done by starting off with the simplest case using only a heating coil to heat up the air and the necessary fans to transport the air. Then improvements will be added to this basic setup one by one. This is done with the intent to give the reader a perspective of the possible economical savings that can be made by changing the parameters of the controller or by other changes in the way the system operates (e.g. recycle the air by night or introducing a heat exchanger).

When comparing the energy savings for the different scenarios it is imperative that the outdoor conditions are equal in each simulation. Thus the outdoor air temperature in this section will be varying equally each day. In this section the temperature variation is computed as follows. A sine wave with amplitude 6°C and a frequency of one cycle per day is added to a constant temperature of 5°C . This results in the temperature development shown in Figure 6.1. It should be noted that this is not equal to the air temperature into the heating coil. This is because of the fact that the air travels through a fan before entering the heating coil. The fan contributes to a heating of the air and in this report this temperature change is equal to 1.5°C . Hence the temperature of the air sent into the heating coil is 1.5°C warmer than the outdoor temperature. This can be seen in Figure 6.2.

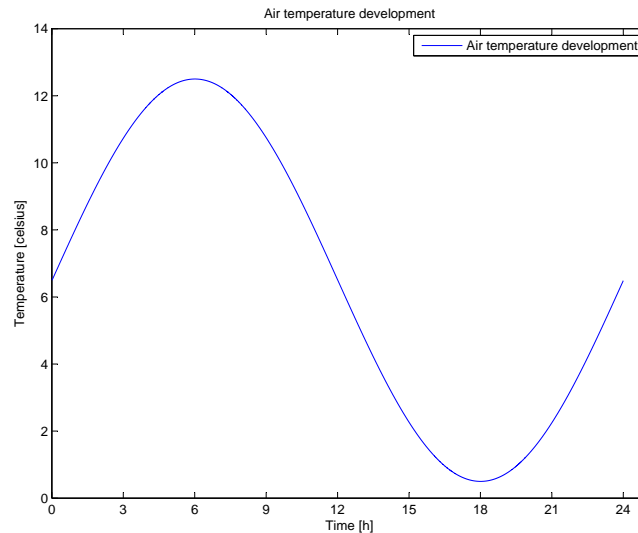


Figure 6.2: Temperature development of air sent into the heating coil

Constant temperature

Starting of with the simplest case where the outdoor air simply travels through a heating coil before entering the rooms of the building. As now will be shown there is much to save by doing something as simple as reducing the temperature of the ventilated air by a degree or two. The difference in energy consumption between a ventilation temperature of 18°C and a ventilation temperature of 16°C during one day is shown in Figure 6.3. The energy consumption given in kWh over this period of time is 295.2 kWh and 243.19 kWh for a ventilation temperature of 18°C and 16°C respectively. This is approximately a saving of 17.62 %. Assuming each kWh has a cost of NOK 1 gives a saving of NOK 52 per day. Over a period of one year this accumulates to NOK 18 980. Of course reducing the ventilation temperature sent to the rooms also has its disadvantages, namely that the degree of comfort for people in the building might decrease. One of the main purpose of HVAC systems is to maintain a comfortable indoor environment. Thus the ventilation temperature should not be set to low. This will lead to a sensation of cold draft in the room.

Varying temperature

To exploit the fact that the energy consumption is considerable lower when the set point temperature is low one needs to consider the idea of lowering the ventilation temperature by night or when the building or certain rooms in the building is not

6. ENERGY CONSUMPTION

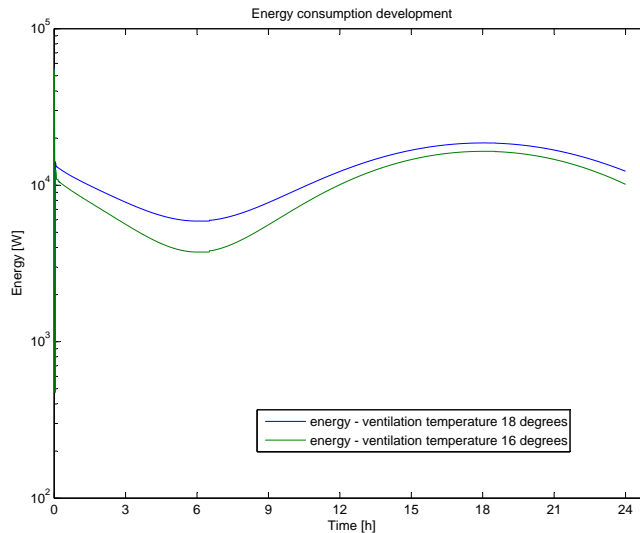


Figure 6.3: Energy consumption with ventilation temperature of 18°C and 16°C

in use. Figure 6.4 shows the energy consumption when the temperature is 18°C 12 hours of the day and 16°C the rest of the day. Comparing this result with the one found in Figure 6.3 shows that the energy consumption is considerable lower than with a fixed and high temperature. It can also be seen that the energy consumption is larger than with a fixed and low temperature, but now the comfort of the people in the building is maintained. Calculating the amount of kWh used in this case gives 269 kWh, which is close to half of the saving found in the previous section. Meaning that a saving of approximately 8.87 % is achieved. The peaks that can be seen in Figure 6.4 is where there is a set point change in the simulation. When the set point temperature is lowered, the energy consumption decreases. This is natural and comes from the shunt valve closing while waiting for the set point temperature to be reached. The opposite effect is seen at the start of the simulation.

When choosing a solution like this one should be aware of the following problem: If the ventilation air temperature is set below the buildings room temperature the ventilation will contribute to a heat loss in each room. This is clearly not economically and with this in mind one should not set the ventilation air temperature considerable lower than the set point of each rooms temperature. According to Novakovic (2010) ventilation temperature is usually preheated to 18°C at the inlet and heated to room temperature just before entering the room. In this report only the preheating will be discussed.

It should be noted that several of todays modern HVAC systems recirculates the air by night instead of only lowering the ventilation temperature. This means that

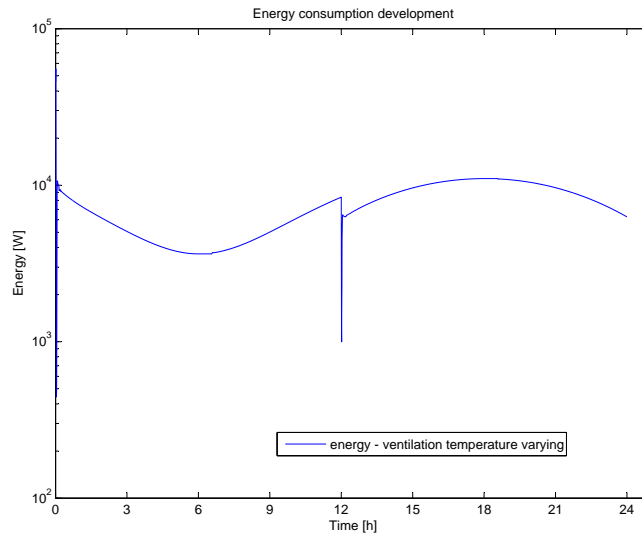


Figure 6.4: Energy consumption with ventilation temperature at 18°C for half of the day and 16°C the rest of the day.

no fresh air is brought into the building. Instead the warm waste air is brought back into each room. It is obvious that this is a more economical solution since cold air from the outside does not need to be heated in this case. This solution has not been simulated in this report but the reader should know of the possibility.

Varying air volume

Another easy method of saving energy in the ventilation system is by decreasing the rate of air volume sent to the building by some factor. Earlier in this report the rate of air flow through the ventilation system has been set to $0.857m^3/s$. This might seem like a small amount, but in 24 hours the volume of air that has been heated to a preset temperature is actually $74044.8m^3$. This corresponds to heating a room of dimensions $3m \cdot 100m \cdot 246.8m$ to the desired temperature in 24 hours. It is easy to understand that in order to heat up a volume of this size a large amount of energy is needed. If however the rate of air flow is decreased by a small amount the volume of air to be heated would be considerably smaller. Thus one can reduce the energy consumption considerably by simply reducing the air flow into the building by a minor fraction. Figure 6.5 shows the energy consumption of the ventilation system when the air quantity is reduced to $0.7713m^3/s$, namely 90% of the previous value. To make it easier to compare the energy consumptions for high and low air volumes the original energy consumption, namely when the air quantity is at 100%, is also included in the figure. It should be noted that the

6. ENERGY CONSUMPTION

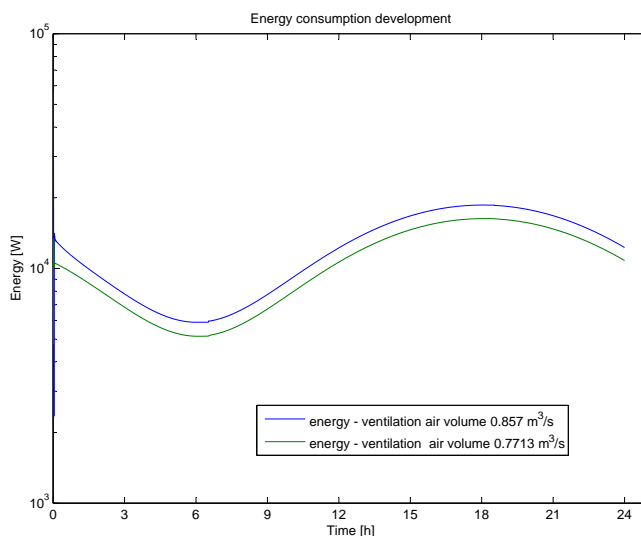


Figure 6.5: Energy consumption with air flow at $0.857m^3/s$ and $0.7713m^3/s$.

set point temperature is now fixed to 18°C . This is done to see the actual saving made by this change alone and not in combination with the changes made earlier.

As one can see from Figure 6.5 the economical aspect of reducing the air flow only a small fraction is considerable. Calculating the energy usage per day when the air flow is reduced to 90%, namely $0.7713m^3/s$, gives 257.1 kWh. As shown before the energy consumption per day for air flow of the normal rate was 295.2 kWh. This gives a difference of 38.1 kWh per day, meaning a possible saving of 12.9 %. Again, assuming a price of NOK 1 per kWh, gives an annual saving of NOK 13 906.5.

As in the previous section one should bear in mind that reducing the air flow to the building might have consequences for the people residing there. If the fresh air in a room reaches a certain lower value, the comfort of the people would decrease because of the rising CO_2 -values. In addition to the fact that people in a working environment becomes tired and less effective when there is not enough fresh air, there are laws that requires the indoor air quality to be above a certain minimum. The Norwegian Labour Inspection Authority operates with a norm value of $1000ppm$ (Arbeidstilsynet 2006). Thus the quality of the air must stay above this value.

It should be noted that it is also possible to vary the ventilated air volume based on the CO_2 values in each room. This would lead to substantial alteration of the ventilation system with the introduction of several controllable dampers and commutator transformers installed on the fans to be able control the fan speed.

In addition to this, sensors measuring CO^2 levels must be installed in every room. This will not be done in this report, but the reader should know that the possibility exists and that this would lead to an even lower energy consumption.

Introducing the heat recovery unit

Until now the ventilation system has consisted of only one component, namely the heating coil. Normally a modern ventilation system also has a heat recovery unit installed. This unit transfers heat from the waste air to the supply air. Hence it can be seen as a way of recycling the heat. See Section 3.6.2 for details of the mathematical modeling of this unit. The ventilation system is now the same as seen previously in Figure 3.31.

Plotting the outdoor air temperature together with the air temperature after the heat recovery unit shows that this unit contributes to a significant rise in air temperature (see Figure 6.6). This means that when the air reaches the heating coil it has a higher temperature than in the previous sections. From the figure it is also seen that the heat recovery unit is most efficient when the difference in the outdoor temperature and the outlet temperature is at its most. This is natural and follows from Fourier's law (Equation 3.1). Thus installing a heat recovery unit in front of the heating coil means that the heating coil would require less energy in order to heat the air temperature to the predefined set point.

The energy consumption of the heating coil in this case, compared with the one found without the heat recovery unit, can be seen in Figure 6.7. This energy consumption accumulates to 202.9 kWh over a period of 24 hours. Compared to the energy consumption of 295.2 kWh found for the heating coil without the heat recovery unit gives a saving of 92.3 kWh or 31.27 %. This is a considerable amount and once again assuming a price of NOK 1 per kWh, gives an annual saving of NOK 33 689.5.

One can now ask oneself the following question; does not the energy used by the heat recovery unit cancel out the energy saving in the heating coil? The answer to this is no. This is because the heat recovery unit is passive in the sense that it utilize the second law of thermodynamics (see Section 3.1.2). This means that it extracts energy from the waste air and pass it on to the colder supply air simply because of the temperature difference between those two temperatures ¹. This means that the heat recovery unit consumes no energy other than the one

¹If the supply air is warmer than the waste air the heat recovery unit would actually work as a cooling element. Hence it is usually turned off if the outdoor temperature reaches a certain value.

6. ENERGY CONSUMPTION

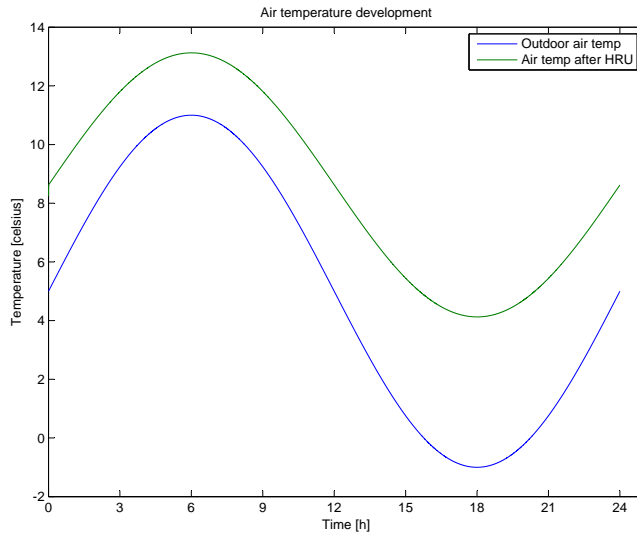


Figure 6.6: Air temperature after the HRU compared with the temperature before the HRU.

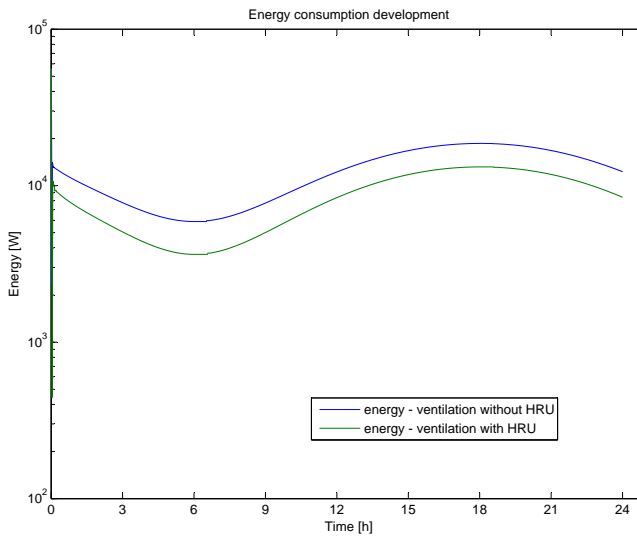


Figure 6.7: Energy consumption with and without HRU.

required to provide a more efficient heat exchange. In a heat recovery unit with a rotational disc as used in this report, the only energy consumed is the energy required to rotate the metal disc. In this report it is done using a 60W motor. This energy consumption is negligible compared to the amount of energy needed to raise the temperature by the same amount using a heating coil.

Varying the controller parameters

The final topic that will be discussed regarding economical savings in the ventilation system is the effect of changing the parameters of the PI-controller. As will be shown, tuning of these parameters can have an economical benefit. The ventilation controller in this report has two tuning parameters, namely Kp_{hc} and Ki_{hc} . Simply speaking one can say that raising these values leads to a more aggressive controller. While lowering these values leads to a less aggressive controller. The following example shows how a change in these parameters leads to a higher energy consumption.

Example 6.1 Varying controller parameters and their effect on energy consumption - ventilation

Given the following parameters for the controller:

- $Kp_{hc} = 0.8$
- $Ki_{hc} = 0$

Simulating with this controller, which is considerable more aggressive than the one found to work satisfying in Chapter 5, leads to a energy consumption of 206.75kWh. As seen before, simulating with the controller parameters set to $Kp_{hc} = 0.2$ and $Ki_{hc} = 0.002$, leads to an energy consumption of 202.9 kWh. When comparing these results it is clear that the aggressive controller uses 3.85 kWh more energy per day. Assuming a price of NOK 1 per kWh gives an annual saving of NOK 1405.25. This is not very much, but it proves that there is an economical aspect in adjusting the controllers parameters. It should be noted that if using a controller like this the set temperature cannot be followed. This is due to the relatively high gain a controller like this produces compared to the time delays present in the system even though the ventilation temperature is relatively close to the set point temperature. The reader should also know that using a controller like this, where the parameters are set to high, would lead to considerable wear and

6. ENERGY CONSUMPTION

tear of certain components (e.g. valves). This effect can easily be seen by changing the controller parameters of the ventilation system. The files needed to do so are included in the CD attached to this report. One of the main reasons why this leads to a controller that extensively varies the valve opening is the time delays introduced by including models of the pipes and ducts to the ventilation system. This introduction leads to extensive time delays between the air temperature at the temperature sensor and the actual air temperature out from the heating coil. This means that the controller is not informed of the fact that it has opened the valve enough before it is too late and then the controller starts to close the valve again. This leads to a valve opening that constantly oscillates between open and closed and hence its life time is greatly reduced. This is of course not a desired behavior and hence the parameters of the controller are chosen small enough to ensure that this does not happen.

6.2.2 Energy savings in the radiator circuit using a PID controller

In this section the radiator circuit and its energy consumption will be compared for different scenarios. This will be done by starting off with the simplest case, namely holding a constant temperature in each room. Then improvements will be added to this basic setup one by one. This is done with the intent to give the reader a perspective of the possible economical saving that can be made by changing the parameters of the controller or by other changes in the way the system operates (e.g. low temperature by night). As for the ventilation system, it is imperative that the outdoor conditions are equal in each simulation. Considering this the outdoor temperature is chosen to be equal to what it was when calculating the energy consumption of the ventilation system, namely 5°C together with a sine wave of amplitude 6°C.

Constant room temperature

In the simplest case the room temperature is controlled to a fixed value. In this case the fixed value is chosen to be 21°C. Then it is changed to 18°C and the energy consumption of these two fixed values are compared. Figure 6.8 shows the difference in energy consumption for these room temperatures. Calculating the energy consumption for all rooms in the north-south wing of the building

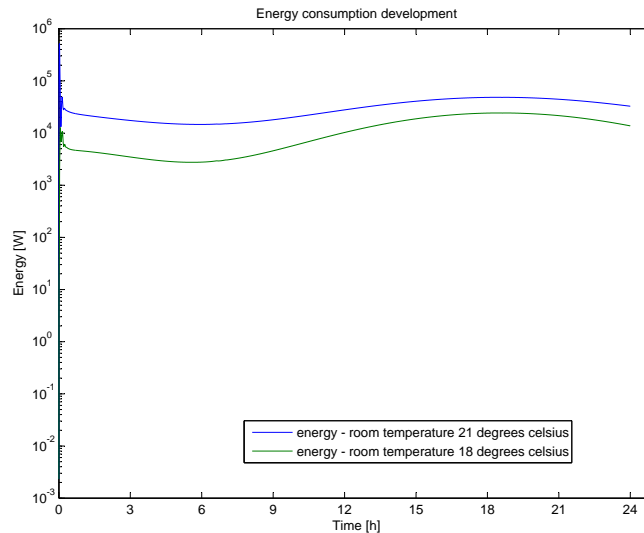


Figure 6.8: Difference in energy consumption for the buildings radiators with a room temperature of 21°C and a room temperature of 18°C.

over a period of one day with a room temperature of 21°C gives 736.68 kWh. This accumulates to an annual energy consumption of 268 888.2 kWh. Again assuming that the price of one kWh is NOK 1, gives an annual cost of NOK 268 888.2 in order to heat up the building. If the temperature is set to 18°C the energy consumption reduces to 286.76 kWh. This gives an annual cost of NOK 104 667.4. Comparing these two costs reveals that reducing the room temperature by 3°C gives an annual saving of 164 220.8 kWh. In other words; a saving of 61.07%. Of course reducing the room temperature has its disadvantages. The most obvious one is that the comfort for people in the building will decrease as the temperature decreases. One of the main purpose of HVAC systems is to maintain a comfortable indoor environment. Thus the room temperature should not be set too low.

Varying room temperature

From the last section it is clear that one can reduce the energy consumption substantially by lowering the temperature of each room. But as discussed, this will lead to a less comfortable environment for the people residing there. Thus having a comfortable room temperature only when there are people present in the building (e.g. by night) seems like a good idea. Using this idea the temperature is decreased by 3°C during the night, from a temperature of 21°C during the day. Figure 6.9 shows the affect this has on the energy consumption. The total

6. ENERGY CONSUMPTION

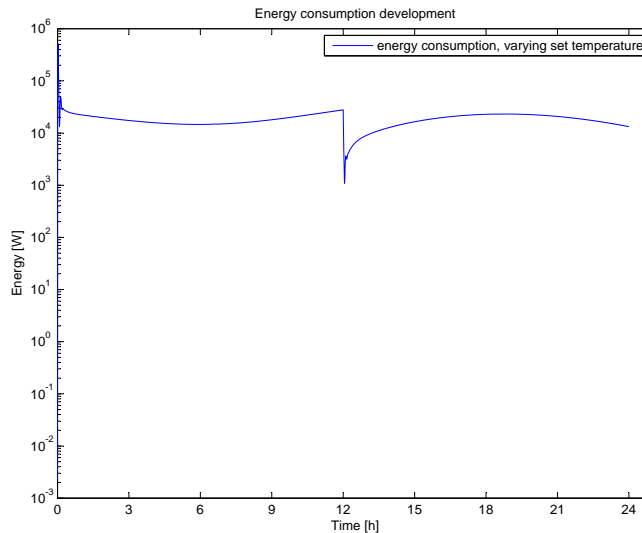


Figure 6.9: Energy consumption of the buildings radiators with varying room temperature.

energy consumption in this case is 451.84 kWh, which accumulates to an annual consumption of 164 921.6 kWh. A total cost of NOK 1 per kWh gives an annual cost of NOK 164 921.6. When comparing the energy consumption found in this section with the one found for a constant temperature of 21°C it is clear that the savings made by lowering the temperature by night is 103 966.6 kWh. This is a saving of 38.66 %.

Varying the controller parameters

The final topic that will be discussed regarding economical savings in the radiator circuit is the consequences by changing the parameters of the PI-controller. As will be shown, tuning of these parameters can have an economical benefit. The radiator controller in this report has two tuning parameters, namely K_p _{room} and K_i _{room}. As for the ventilation controller; raising these values leads to a more aggressive controller, while lowering these values leads to a less aggressive controller. The following example shows how a change in these parameters leads to a higher energy consumption.

Example 6.2 Varying controller parameters and their effect on energy consumption - radiator circuit

Given the following parameters for the controller:

- $Kp_{room} = 500$
- $Ki_{room} = 0$

Simulating with this controller, which is considerable more aggressive than the one found to work satisfying in Chapter 5 and without a integral action, leads to an energy consumption of 741.29 kWh. As seen before, simulating with the controller parameters set to $Kp_{hc} = 50$ and $Ki_{hc} = 0.05$, leads to an energy consumption of 736.68 kWh. Comparing these results gives an additional energy consumption of 4.6 kWh per day. Assuming a price of NOK 1 per kWh gives an annual extra cost of NOK 1682.65. Not very much, but in addition to this the temperature response in the room is not as desired. There is a large overshoot at the start of the simulation due to the aggressiveness of the controller. Also there is a offset, meaning that the controller is not able to reach the current set point. Since this controller leads to a poor temperature response the controller found in Chapter 5 should be used instead.

6.2.3 Energy savings in the radiator circuit using MPC

This section will present the energy consumption for the exact same scenarios as in Section 6.2.2 with one major difference: The MPC controller from Section 5.3 will be used instead of the PID controller to control the indoor temperature. Everything else, including the outdoor temperature development, is exactly as it was in Section 6.2.2. The results following from these simulations will be compared with the results from Section 6.2.2 so that the difference in energy consumption is made clear.

Constant room temperature

As in Section 6.2.2 the simplest case where the indoor temperature is fixed is studied first. Figure 6.10 shows the energy consumption for room k133 using a MPC controller compared with the energy consumption for the same room while using the traditional PID controller. As can be seen from this figure the MPC control scheme consumes approximately the same amount of energy as the PID controller. Figure 6.11 shows a close up of the same energy consumption. In this figure it is clear that the MPC actually uses a little less energy to hold

6. ENERGY CONSUMPTION

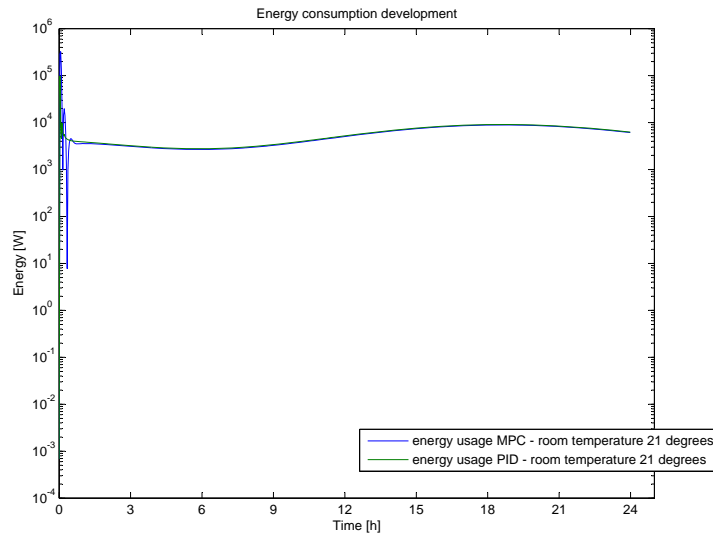


Figure 6.10: Difference in energy consumption for room k133 using different controllers. Fixed set temperature.

the temperature at the desired temperature. This can be explained by the fact that the MPC actually is not able to hold the desired temperature exactly. The main reason for this is that the model used by the MPC to control the system is not 100% accurate or that the Kalman filter is not tuned perfectly and hence the states are inaccurate. Thus the MPC believes it actually has reached its set point.

When summarizing the energy consumption for room k133 it becomes clear that the MPC actually uses more energy than the PID. This is in contradiction to what is seen in Figure 6.10 and Figure 6.11. There are several reasons for this. First, the MPC scheme is initialized with a very different temperature than what was the case for the PID controller. This gives the PID controller an advantage at the start up of the comparison. However once the MPC has settled both control schemes follows each other closely. Second, the MPC controller has not been tuned in order to be as efficient as possible. This is another disadvantage which, given more time, could have been removed.

Varying room temperature

In this section the energy consumption of the radiator using the MPC controller will be compared with the energy consumption while using the PID controller. This is the same as was done in the last section, but now the desired room temperature is varying such that the temperature is low during the night and

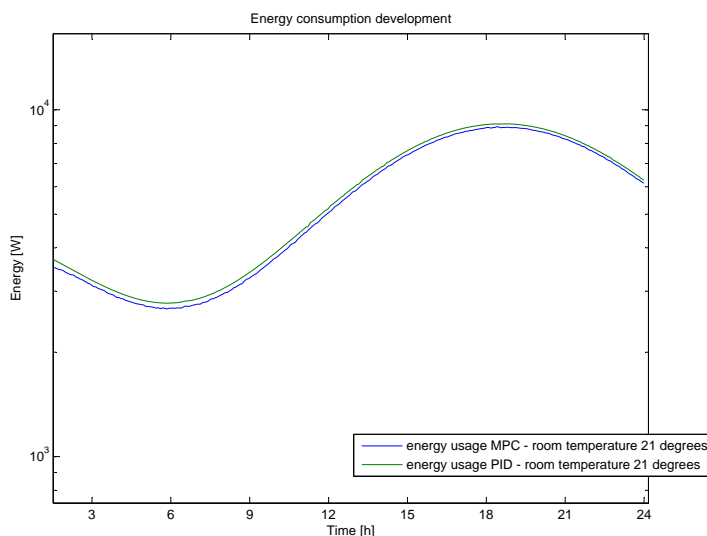


Figure 6.11: A close-up of the difference in energy consumption for room k133 using different controllers. Fixed set temperature.

high during the day. The temperature during the day is set to be 21°C while it is lowered by 3 degrees during the night. The resulting energy consumption for this scenario can be seen in Figure 6.12.

From this figure it is clear that the MPC controller uses approximately the same amount of energy as the PID controller. This is the same result as was given when the temperature was fixed. However it is also seen that during the transition between daytime temperature and nighttime temperature the MPC controller oscillates more than the PID controller. This causes the MPC controller to use more energy than the PID controller, but a reasonable assumption is that this effect can be removed with additional tuning of the MPC controller. However time has not allowed the writer to verify this.

With the two examples given above and the discussion that have been made for these it can not be stated that the MPC is more energy efficient, nor less energy efficient, than the PID. In order to do this some additional tuning of the MPC is needed and several more scenarios needs to be simulated. However to time constraint this has not been done. The results from this section and assumptions of simulation results which has not been done will be elaborated more thoroughly in Chapter 7.

6. ENERGY CONSUMPTION

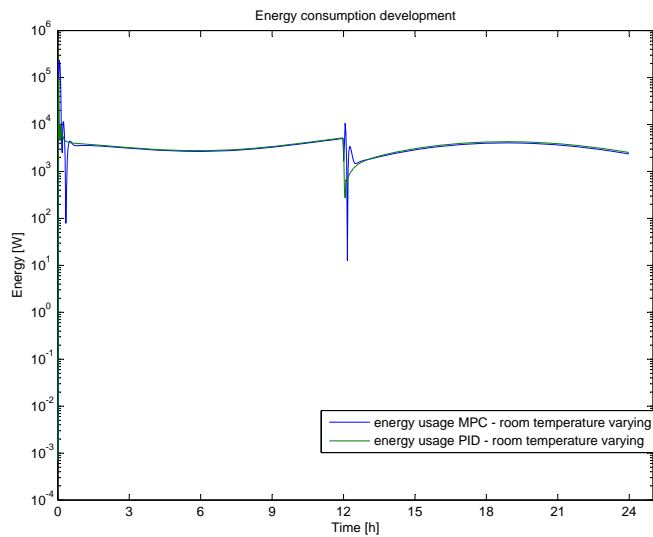


Figure 6.12: Difference in energy consumption for room k133 using different controllers. Varying set temperature.

7

Discussion

This chapter discusses the choices made throughout the report and the results found from the choices that have been made. In order to make it easier for the reader this will be done in chronological order. First, the model of the building from Chapter 3 and its validity will be discussed. Second, the system identification from Chapter 4 and its results. Third, the MPC control scheme from Section 5.3 and operational strategies made in this section will be discussed. Finally, the results from Chapter 6 will be discussed. Since time has not allowed the writer to tune the MPC controller optimally, where optimal means to use as little energy as possible, anticipated results from a fine tuned MPC controller will also be discussed in this Chapter.

The models of each room and the models of each component in the ventilation system are based on mathematical equations. These equations are based on physical balance laws with certain simplifications. The simplifications are made in order to keep the mathematical expressions as simple as possible, while still retaining a model that resembles the physical reality closely. When simplifications are introduced the behavior of the models diverges from the real behavior of the system. It is important that this deviation is not too large. Otherwise the model cannot be said to be valid.

For the room model it has been assumed a constant temperature in the entire room. This is a major simplification since the temperature of the air will be varying throughout the room. Near the windows the temperature will be low due to the vicinity of the outdoor temperature, while the temperature near the radiators will be much higher than in the rest of the room. The temperature differences in the room causes the air to circulate. The reason for this is that the density of hot air is less than the density for cold air. Thus hot air ascends, while

7. DISCUSSION

cold air descends and there will be a natural circulation of air inside the room. The model in this report neglects the air circulation within the rooms due to the extensive computational resources this requires. Hence the room in this report has been modeled as having a single temperature with a certain thermal capacity. Even though this is a major simplification it may be used since the air flow inside each room is not of interest in this report. Hence one mean temperature for each room is an accurate enough description of the rooms state.

Another simplification made in the model of the ventilation system is that it is limited to a constant flow of air. This is a drawback of the model design since simulation results of a varying airflow will not be valid. In modern ventilation systems it is usually possible to vary the airflow according to the current necessity of fresh air. This is usually done by connecting a frequency transformer to the fans motor. The frequency transformer can then be used to control the speed of the fan and hence the flow of air through the ventilation system. This component needs to be feed with a control signal that specify the speed that the motor should be driven at. This can be done by a controller which receives signals from sensors in each room of the building and based on this information (e.g. CO² levels) makes a decision to speed up or slow down the fan. As previously mentioned such a system cannot be simulated using the models from Chapter 3. To be able to do so these mathematical models needs to be altered so that they are valid also for varying airflow. Attempts to implement models with varying airflow has been done (see Section 3.6.7). However these attempts where unsuccessful with an unrealistic behavior. Another possibility is to use an ad hoc solution where several models of the ventilation system from Chapter 3 (modeled for different fixed air flows) are used together with a switch which selects the model to use based on the current airflow. The proposed ad hoc solution has not been tested, but without to much effort it should be possible to implement it. However the best solution would be to create a single model that also is valid for varying airflow. To summarize, the ventilation model in this report has a major drawback: It is not valid for varying airflow. Nonetheless, as seen in Chapter 5 and Chapter 6, many examples that shows the energy consumption while still holding the temperature as desired and without violating any of the ventilation systems shortcomings can still be simulated. Also the fixed airflow has been reduced in one of the simulations to visualize the saving varying airflow might lead to.

The fact that the fans in a ventilation system contributes to a temperature rise in the ventilated air makes the placement of these important when designing such a system. One should utilize the energy contribution of this component in order to save energy. Hence the most economical placement of the supply air fan will be after the heat recovery unit and before the heating coil. For the waste air fan the most economical placement will be just before the air enters the heat recovery

unit. This should ensure that the temperature difference of the two air flows in the heat recovery unit are as large as possible. The greater the temperature difference of the HRU is the greater is the effect of it. Hence the air flowing into the heating coil will have a higher temperature than what otherwise would be the case.

The radiator model from Chapter 3 is a very simplified model and in contrast to the other models it is not based on any physical laws. However it is based on the expected behavior of a radiator, based on real world experiences. This means that it takes a while for the radiator to heat up and it takes a while for it to cool down. Hence, even without the use of any physical laws, a model that behaves nearly as a real radiator has been found. One can discuss whether or not the time constant (see Section 3.3.7) is chosen correctly, but even if chosen to large or to small the system behavior will still be similar, although with a slower or faster response for the radiator. This means that the controller parameters found previously in the report might need to be altered so that the system behaves according to its intention. Nevertheless with this choice of radiator model the system behaves naturally and as desired.

With the preceding discussion in mind the models found in Chapter 3 are said to be valid, meaning accurate enough, as long as the air flow and water flow through each component is constant.

In Chapter 4 a simplified model of the building from Section 3.5, which also can be discretized, is found using system identification. This model is then used in the implementation of the MPC controller from Section 5.3. In order for the MPC scheme to function as intended it is imperative that the result from this identification and simplification procedure is a model that reacts the same way as the original system. This means that it is very important that the signals used to excite and validate the system during the identification process are varying enough both in frequency and amplitude. This should ensure that the identification method chosen is able to identify every characteristic property of the real system. If failing to do so the identified model will not react or behave as the original and the MPC will not function properly. In this report the excitation signals are chosen so that they are guaranteed to reflect most of the original systems behavior, however the connection between the adjoining rooms that are present in the original model have been neglected since it was found to have little or close to no effect on the temperature of the room to be identified. This was shown in Chapter 4. Thus, the reader should be aware of the fact that the models from Chapter 4 is yet another approximation of an exact and real model and hence it may be the source of control problems or inaccuracies that may occur.

In Section 5.3 the MPC has been chosen to simply replace the PID controller

7. DISCUSSION

in each room and hence only control the temperature of the room. The choice of this control structure can be discussed. It might be argued that a single MPC controller that controls each of the rooms temperatures and the ventilation temperature should be used instead. One reason why this might be better is that the controller then has multiple variables under its control. In this case it might be that the controller would find it more economical to raise the ventilation temperature to maintain the temperature of the room instead of the radiator temperature. However in this report it has been decided that the temperature is to hold a constant temperature of 18°C even though such a ventilation temperature reduces the temperature of the indoor air. This would not be the case if the control structure just mentioned are chosen. Then the MPC should be able to vary the ventilation temperature if this is found to be economically beneficial. This decision is handled automatically by the controllers cost function. Another possibility that should not be overlooked is to use the ventilation as the only source of heating. This means that the radiators are not present anymore and that the MPC or PID controller is used to adjust the ventilated temperature so that the room temperature maintains its desired value. In order to implement such a design the PID or MPC controller needs to be feed with the current temperature as has been done throughout this report, but the control signals should be sent to the shunt valve instead of the radiators. Also the control parameters needs to be changed. For the MPC controller to function in this case the identified model needs to be altered so that it includes the characteristic behavior of the ventilation system. It should be noted that the building in this report has a wall thickness of 15 cm with standard insulation and otherwise is constructed of wood. This results in a wall that does not insulate very good, hence to use the ventilation as the only source of heating in this building is not recommended. The use of this controller design is only common in modern and energy efficient buildings. Such buildings are sometimes also referred to as energy neutral buildings.

The Kalman filter introduced in Section 5.3 uses the identified model together with the input and the disturbed output to estimate both the real value of the output and the states of the identified model. It was shown in Section 5.3 that the estimated room temperature does not match the real room temperature exactly. This is due to the fact that the identified model is an approximation of the real system and hence the MPC controller believes it has reached its desired temperature when in fact there is a small off-set between the desired temperature and the real temperature. However this off-set is only about 0.03°C and thus it can be neglected since no one who resides in the building would feel the difference. If however it is imperative that the room holds exactly a certain temperature this small off-set should be taken into consideration. An easy way to fix this would be to adjust the desired temperature so that it compensates for this off-

set. Removing the noise from the measured output and choosing the tuning parameter R in the Kalman filter to 0 would also remove this off-set. However this would lead to a less realistic model since measurements usually includes some form of disturbance.

The MPC controller assumes that the states of the system is known. In this report the Kalman filter estimates the states, but since these are based on the identified model from Chapter 4, and one of the transfer function from this chapter is further estimated by the third order Padè approximation, the states have no physical interpretation. Hence it is impossible to check whether or not the states from the Kalman filter are correct. The estimated output from the Kalman filter can however be compared to the real output of the system. This is found to be accurate enough (see the previous section). Without the knowledge of whether or not the estimated states are correct it is not as easy to tune the Kalman filter optimally, but when simulating the entire system, including the MPC controller and the Kalman filter, it has been shown that the desired temperature is reached within a reasonable short time.

In Chapter 6 the calculation of energy consumption for each component used in this report was explained. As previously shown some of the components modeled consumes a varying amount of energy, namely the heating coil and each radiator, while the other components has a fixed energy consumption. Thus the only components where the energy consumption is altered over time is the heating coil and the radiators. Chapter 6 has shown the energy consumption of these components for different scenarios. Some of those results will be further discussed now.

The energy consumption of the radiators for varying room temperature (see Section 6.2.2) needs some detailed explanation. The observant reader might notice that the energy consumption for this case is suspiciously close to the energy consumption for the case where the room temperature is set to 18°C. Hence it is also relatively far from the case where the room temperature is fixed at 21°C. It is easy to interpret this as a fault. From a superficial point of view one could think that this energy consumption should stay close to the middle of these two values. However this is not the case. The explanation for this is quite simple but not that obvious.

In the case of varying room temperature from Section 6.2.2 the set point temperature of the room drops together with the outdoor temperature. Meaning a low set temperature for the rooms when the outdoor temperature is low and a high set temperature for the rooms when the outdoor temperature is high (see Figure 5.9). Thus the energy consumption required to hold the set temperature is relatively low during the entire simulation. Hence it is close to the one found

7. DISCUSSION

for a fixed low temperature and this explains why the radiators uses only a little bit more energy than when the temperature is set to a constant low value.

However, if the outdoor temperature is shifted 180 degrees so that the outdoor temperature behavior is opposite to the behavior seen in Figure 5.9. Or if the reference temperature is opposite of what Figure 5.9 shows. Then the opposite happens, meaning that the energy consumption for a varying temperature is now close to the energy consumption for a fixed temperature of 21°C.

Thus it has been shown that the energy consumption of the radiators is as expected even though it did not seem that way at first glance. Simulating the system with a constant outdoor temperature verifies this. This discussion also reveals that the amount of energy saved by lowering the temperature by night is varying depending on the outdoor temperature each night. In the case described in Section 6.2.2 the energy saved was 38.66 %. With an outdoor temperature varying differently the percent saved can change substantially. For instance if the outdoor temperature is shifted by 180 degrees the savings made are considerable lower. The CD included in this report contains the necessary files needed to run simulations of different scenarios which verifies this.

Due to time constraints the writer has not been able to tune the MPC controller to function as efficiently as desired. This was shown in Section 6.2.3 where the results shows little to none energy savings made by changing the PID controller with the MPC controller. Since the MPC scheme in this report only controls one variable substantial saving can not be expected either, but the controller should be able to cope with set point changes more efficiently due to its ability to “look into the future”. However if implementing the control structure proposed above where the MPC controls multiple variables it can be assumed that, when correctly tuned, it would be able to control the temperature of the room while using less energy than the basic PID controller.

In the results presented in Section 5.3 and Section 6.2.3 an MPC controller with tuning parameters as shown in Section 5.3 has been used. Changing these parameters can greatly reduce or enhance the performance of the controller. Selecting a prediction horizon that is considerably longer gives the MPC the advantage that it predicts the future behavior of the room temperature further into the future. With the chosen prediction horizon from Section 5.3 and the knowledge of the time step used in the discretization of the model used in the MPC it is possible to calculate how far into the future the MPC “looks”. For the results given earlier in the report the prediction horizon was chosen to be 35. With a time step of 50 seconds as has been used in this report one can conclude that the controller “looks” approximately 29 minutes into the future ($35 \cdot 50 = 1750seconds \approx 29minutes$).

The MPC control scheme in this report has been implemented as dynamic as possible. This means that the matrices Ψ , Υ , Θ and Ξ which are varying depending on the choice of prediction horizon, control horizon and the number of inputs, states and outputs are generated automatically. The only thing that is required by the user is to simply choose appropriate values for the mentioned constants. Thus the code that generates these matrices can be used in any MPC implementation regardless of what it is to control. For this report it means that it is easier and a lot faster to test the system behavior for different prediction and control horizons. The code that creates these matrices can be seen in Appendix D.8.

Although it has not been done in this report, it should be noted that having one specific controller for each room would lead to a better temperature response for each room. Using the same controller in each room gives a poor response in the rooms that are considerable different in size compared to the medium sized room of which the controller has been designed. The easiest but cumbersome solution to this is to alter the controller parameters in each room so that the temperature response in each room regardless of size is as desired. The introduction of an adaptive controller in each room might solve this. Meaning that the same controller is used in each room. But this controller automatically changes its parameters to fit the size of the room within a reasonable short period of time.

The reader should also be aware of the fact that changing the controller parameters so that the controller becomes more aggressive, might lead to considerable wear and tear of the components involved in the feedback loop (e.g. valves or actuators). This was briefly mentioned in Section 6.2.1 and it is of course not favorable. Hence it should be taken into account when designing controllers. In this report though, this problem has not been assessed as the focus has been on energy savings rather than on the controller.

From Chapter 6 it is clear that there is a considerable amount of energy to save based on simple changes made to the HVAC system. Most of these modifications has a very low cost and it can easily be shown that within a few years or even less it is economically beneficial to introduce one or several of these modifications. The MPC controller has been shown to be a promising solution. But this controller structure is probably more efficient when implemented in a multi variable control problem where other controllers might struggle to find an optimal solution.

7. DISCUSSION

8

Conclusion and further works

In this report a building with its ventilation system has been modeled, and its temperature response has been studied. Simulation of this model has been done using Matlab with Simulink and its toolbox Simscape. The model has been simulated for different scenarios, with two different controllers, namely the PID controller and the MPC controller. From these simulations the energy consumption has been logged. Each of these scenarios has then been compared to each other and the possible economical savings has been made clear. A summary of these results using the PID controller are shown in Table 8.1 and 8.2.

These tables together with Chapter 6 and Chapter 7 clarifies the possible energy savings that can be made by simple and inexpensive changes to the buildings HVAC system. Table 8.2 makes it clear that the savings made with a varying set temperature depends largely on the outdoor temperature. Even though this is the case there will be a certain percentage of energy saved in any case.

The results found using the MPC controller indicates that it is possible to save even more energy by the use of such a controller instead of the PID controller. However it is not possible to make any conclusion of this based on the results presented in this report. This is due to the fact that it was not enough time to tune the controller to present better and several more results. However on a general basis it is known that the MPC controller, due to its ability to predict and to keep the cost function to a minimum, can be more energy efficient than the traditional PID controller.

In this report six rooms from Saltstraumen elementary school has been used as an example, but one can anticipate approximately the same percentage of saving by introducing the same changes to other buildings. Hence this report can be used to emphasize the possible economical savings in any buildings HVAC system.

8. CONCLUSION AND FURTHER WORKS

Table 8.1: Energy consumption, ventilation

Set temperature	Air volume	HRU	Consumption	Savings (%)
Fixed, 18°C	$0.857m^3/s$	No	295.2 kWh/day	-
Fixed, 16°C	$0.857m^3/s$	No	243.19 kWh/day	17.62 %
Varying, 18°C - 16°C	$0.857m^3/s$	No	269 kWh/day	8.87 %
Fixed, 18°C	$0.7713m^3/s$	No	257.1 kWh/day	12.9 %
Fixed, 18°C	$0.857m^3/s$	Yes	202.9 kWh/day	31.27 %

Table 8.2: Energy consumption, radiator circuit

Room set temperature	Consumption	Savings (%)
Fixed, 21°C	736.68 kWh/day	-
Fixed, 18°C	286.76 kWh/day	61.07 %
Varying, 21°C - 18°C	451.84 kWh/day	38.66 %
Varying, 18°C - 21°C	599 kWh/day	18.69 %

Further works

This section gives the reader a short introduction to further developments possible in the field of HVAC systems and possible extensions of this report.

By introducing more advanced models it is possible to achieve more accurate simulations than the one used in this report. Introducing models that handles varying air volumes a HVAC system using VAV-control can be modeled. By doing so several additional scenarios can be simulated, proving that there is a possibility for even larger economical savings. Actually the economical aspect of this has already been briefly shown in Section 6.2, in the part where the air volume is decreased to 90%. Also attempts of creating a model that includes varying air flow has been discussed but these attempts where unsuccessful and further investigation of why this fails should be done.

This report only indicates that the MPC is able to control the temperature of the room while using less energy. This should be investigated further so that one can come to a conclusion of whether or not this indication is correct. The MPC also has the ability to hold certain values within constraints. It can for instance be used to hold the levels of CO_2 below a certain value without continuously feeding the room with fresh air, but only bring in fresh air when necessary. This would lead to a more economical usage of fresh air (i.e. fresh air is only used when necessary). If a model that is valid also for varying airflow is found such

an ability can easily be implemented in the MPC controller from Section 5.3 by including the upper limit of the CO^2 as a constraint using the same procedure as explained in Section 5.3.

Another development possibility can be to make a program that is feed with room, ventilation and controller data, and on the basis of this automatically generates m-files and Simulink diagrams. The program should also be able to run simulations of these generated files and print out the results to the user. Such a program can be used to visualize possible energy savings that can be made in buildings by introducing the changes seen in this report and possibly several more advanced changes (e.g. a more advanced controller).

It would also be interesting to model a more energy efficient building and see if the ventilation system alone could control the temperature of the room. The MPC controller could also be feed with weather forecasts and predict optimal inputs based on this additional information. This should be implemented into the feed forward connection, a simple solution where the weather is assumed to be fixed has been implemented already.

8. CONCLUSION AND FURTHER WORKS

Bibliography

- Alvsvåg, . (2010). Hvac systems. modeling, simulation and control for energy saving in hvac-systems, *Technical report*, Norwegian University of Science and Technology.
- Arbeidstilsynet (2006). *Veiledning om klima og luftkvalitet på arbeidsplassen*, Direktoratet for arbeidstilsynet.
- ASHRAE (2009). *2009 ASHRAE Handbook: Fundamentals*, American Society of Heating, Refrigerating and Air-Conditioning Engineers, inc.
- Balchen, J. G., Andresen, T. & Foss, B. A. (2003). *Reguleringsteknikk*, Institutt for teknisk kybernetikk, NTNU.
- Brown, R. G. (1997). *Introduction to random signals and applied Kalman filtering*, John Wiley and Sons, Inc.
- Byggforsk, S. (n.d.a). Beregning av u-verdi og varmestrøm for konstruksjoner mot grunnen etter ns-en iso 13370, [Online]. <http://bks.byggforsk.no/DocumentView.aspx?sectionId=2&portalMenuId=0&nodeId=225&level=2&documentId=2623>. (Accessed December 11, 2010).
- Byggforsk, S. (n.d.b). Beregning av u-verdier etter ns-en iso 6946, [Online]. <http://bks.byggforsk.no/DocumentView.aspx?sectionId=2&portalMenuId=0&nodeId=225&level=2&documentId=208>. (Accessed December 11, 2010).
- Byggforsk, S. (n.d.c). U-verdier. tak, [Online]. <http://bks.byggforsk.no/DocumentView.aspx?documentId=212§ionId=2>. (Accessed December 11, 2010).
- Byggforsk, S. (n.d.d). U-verdier. vegger over terreng, [Online]. <http://bks.byggforsk.no/DocumentView.aspx?sectionId=2&portalMenuId=0&nodeId=225&level=2&documentId=211>. (Accessed December 11, 2010).

BIBLIOGRAPHY

- Chen, C.-T. (1999). *Linear system theory and design*, Oxford University Press, Inc.
- Djuric, N. (2010). Personal conversations with phd natasa djuric.
- Ersdal, A.-M. (2011). *Methods for ice model updating using a mobile sensor network*, Master's thesis, Norwegian University of Science and Technology.
- Haines, R. W. & Myers, M. E. (2010). *HVAC systems design handbook*, McGraw-Hill Companies, Inc.
- Hauge, E. H. & Støvneng, J. A. (2009). *Grunnleggende fysik - klassisk mekanikk og varmelære*, Tapir Akademisk Forlag.
- Instruments, N. (n.d.). Selecting a model structure in the system identification process, [Online]. <http://zone.ni.com/devzone/cda/tut/p/id/4028>. (Accessed April 22, 2011).
- Lienhard IV, J. H. & Lienhard V, J. H. (2008). *A heat transfer textbook*, Phlogiston Press.
- Ljung, L. (2006). *System identification. Theory for the user. Second edition*, Prentice Hall PTR.
- Ljung, L. (n.d.). System identification toolbox 7, [Online]. http://www.mathworks.com/help/pdf_doc/ident/ident.pdf. (Accessed December 11, 2010).
- Maciejowski, J. M. (2002). *Predictive Control with Constraints*, Pearson Education Limited.
- Magnussen, J. (2010). Increased energy efficiency in buildings using model predictive control, *Technical report*, Norwegian University of Science and Technology.
- Narasimhan, T. N. (1999). *Ohm's Law, Fick's Law, Joule's Law and Ground Water Flows*, Department of Materials Science and Mineral Engineering. Department of Environmental Science, Policy and Management Earth Sciences Division, Lawrence Berkeley National Laboratory. 457 Evans Hall, University of California at Berkeley.
- Nilsson, J. W. & Riedel, S. A. (2005). *Electric circuits*, 7th edn, Pearson Prentice Hall.
- NorDan (n.d.). Vindusboken, [Online]. <http://www.nordan.no/newsread/ReadImage.asp?WCI=GetByID&IMAGEID=1&DOCID=10838>. (Accessed December 11, 2010).

BIBLIOGRAPHY

- Novakovic, V. (1995). *Dynamisk modellering og simulering av klimasystemer*, Institutt for energi- og prosessteknikk, NTNU.
- Novakovic, V. (2010). Personal conversations with professor vojislav novakovic.
- Proakis, J. G. & Manolakis, D. G. (2007). *Digital signal processing. Principles, algorithms and applications*, fourth edn, Pearson Prentice Hall.
- Rockwool (n.d.) (n.d.). Varmeisolering, [Online]. <http://www.rockwool.no/r%C3%A5d+og+veiledning/takisolasjonsguiden/produktegenskaper+og+praktiske+forhold/varmeisolering>. (Accessed October 11, 2010).
- Simon, D. (2006). *Optimal state estimation*, John Wiley and Sons, Inc.
- Skogestad, S. & Postlethwaite, I. (2007). *Multivariable feedback control*, John Wiley & Sons, Ltd.
- Thue, J. V. (2006). *Husbyggingsteknikk; bygningsfysisk grunnlag, kapittel 4-8*, Institutt for bygg, anlegg og transport, NTNU.
- Underwood, C. P. (1999). *HVAC Control Systems: Modelling, analysis and design*, E & FN Spon.

BIBLIOGRAPHY

Appendices

Appendix A

Calculation of electrical components in the building model

This appendix is based on Chapter 19.6 in Novakovic (1995). First all the resistors used in this report will be explained then the capacitors will be defined. Every electrical component used are based on a physical definition of how heat transfers through materials either by convection, conduction and radiation, and how materials store heat or energy.

A.1 Calculation of thermal resistance

A.1.1 Outer thermal transmittance of external walls and ceiling

$xxxx_Roy$ is the outer thermal transmittance of external wall y in room $xxxx$. According to Novakovic (1995) this value may be defined as

$$xxxx_Roy = \frac{1}{U_{ext_wall} \cdot A_{wall}} \quad (A.1)$$

where

$$U_{ext_wall} = \text{U value of the external wall } \left(\frac{W}{m^2 \cdot K}\right)$$

A. CALCULATION OF ELECTRICAL COMPONENTS IN THE BUILDING MODEL

$$A_{wall} = \text{area of the wall (m}^2\text{)}$$

According to Byggforsk (n.d.b) the U value is a standardized measure of how easy heat escapes through an element (e.g. a wall or a window). It states how much heat per time unit (i.e. Watt (W)) that flows through an area of $1m^2$ when exposed to a constant temperature difference of 1K. Calculating the outer thermal transmittance of the ceiling is done in the exact same way as was done for the wall above. The only difference is that the U-value and the area of the wall must be changed to the one relevant for the ceiling. In this report the U-values of the walls and ceilings are found on the basis of knowledge about the construction (e.g. thickness and construction materials). This knowledge is then used together with Byggforsk (n.d.d) and Byggforsk (n.d.c) and the U-value for the buildings wall and ceiling has been found based on this.

A.1.2 Inner thermal transmittance of external walls and ceiling

$xxxx_Riy$ is the inner thermal transmittance of external wall y in room $xxxx$. According to Novakovic (1995) this value is defined as

$$xxxx_Riy = \left(\frac{\delta_{wall}}{2 \cdot \lambda_{wall}} + \frac{\delta_{sheathing}}{2 \cdot \lambda_{sheathing}} \right) \cdot \frac{1}{A_{wall}} \quad (\text{A.2})$$

where

$$\delta_{wall} = \text{thickness of the external wall (m)}$$

$$\lambda_{wall} = \text{heat conductivity of the wall (}\frac{W}{m \cdot K}\text{)}$$

$$\delta_{sheathing} = \text{thickness of the indoor sheathing of the wall (m)}$$

$$\lambda_{wall} = \text{heat conductivity of the wall (}\frac{W}{m \cdot K}\text{)}$$

$$A_{wall} = \text{area of the wall (m}^2\text{)}$$

Using A_{wall} is not entirely correct since the indoor area of the wall normally is a small fraction smaller than the outdoor area. In this report though, the outdoor area is equal to the indoor area. This leads to a small error when simulating but the error is very small and thus negligible. When calculating inner thermal transmittance of the ceiling the above equation is used, but with δ and λ values for the ceiling and its sheathing instead of the wall.

A.1.3 Thermal transmittance of internal walls

$xxxx_R_{internal_y}$ is defined as the thermal transmittance of internal wall y in room xxxx.

$$xxxx_R_{internal_y} = \frac{1}{2 \cdot U_{internal_wall} \cdot A_{wall}} \quad (A.3)$$

where

$$U_{internal_wall} = \text{U value of the internal wall } \left(\frac{W}{m^2 \cdot K}\right)$$

$$A_{wall} = \text{area of the wall } (m^2)$$

The reason for scaling the resistance by a factor of 2 in the equation above is that this is a wall shared with the adjoining room, and thus there is an equal resistor on the other side of the mass point. $U_{internal_wall}$ is the U-value of the entire wall. Thus, without this scaling the entire wall would have a greater resistance to heat conduction than it should. If the mass point do not lay in the center of the wall, the scaling factor must be altered, but it must always sum up to 4. E.g. 1 and 3, meaning that the mass point is positioned one fourth into the wall from one side and three fourth from the other side). Otherwise this would alter the heat conduction properties of the wall.

A.1.4 Thermal transmittance of the floor

$xxxx_R_{floor}$ is defined as the thermal transmittance of the floor in room xxxx.

$$xxxx_R_{floor} = \frac{1}{U_{floor} \cdot A_{floor}} \quad (A.4)$$

where

$$U_{floor} = \text{U value of the floor } \left(\frac{W}{m^2 \cdot K}\right)$$

$$A_{floor} = \text{area of the floor } (m^2)$$

It should be noted that the U-value of the floor actually depends on the floor area (Byggforsk n.d.a). However the floor area does not alter the U-value significantly and the area does not vary considerable in the different rooms either. Hence the U-value of the floors are considered constant in this report.

A. CALCULATION OF ELECTRICAL COMPONENTS IN THE BUILDING MODEL

A.1.5 Convective thermal resistance wall to air and ceiling to air

$xxxx_Rconv_wally_air$ is the convective thermal resistance from the inner surface of wall y to the indoor air. According to Novakovic (1995) this value is defined as

$$xxxx_Rconv_wally_air = \frac{1}{htc_{wallair} \cdot A_{wall}} \quad (A.5)$$

where

A_{wall} = area of the wall (m^2)

$htc_{wallair}$ = heat transfer coefficient ($\frac{W}{m^2 \cdot K}$)

$htc_{wallair}$ is normally around $3W/m^2K$ (Novakovic 1995). Again, the convective thermal resistance of the ceiling can be calculated in the same way as for the walls only by changing the variables to the one representing the ceiling. That is, changing A_{wall} to $A_{ceiling}$ and $htc_{wallair}$ to $htc_{ceilingair}$.

A.1.6 Thermal transmittance of the windows

$xxxx_Rwindow$ is the thermal transmittance of the windows in room $xxxx$. According to Novakovic (1995) this value is defined as

$$xxxx_Rwindow = \frac{1}{U_{window} \cdot A_{window}} - \frac{1}{\frac{1}{xxxx_Radiation_window_wally} + \frac{1}{xxxx_Rconv_window_air}} \quad (A.6)$$

where

A_{window} = area of the window (m^2)

U_{window} = U value of the window ($\frac{W}{m^2 \cdot K}$)

The U-value of the windows in this report are from NorDan (n.d.). $xxxx_Rconv_window_air$ and $xxxx_Radiation_window_wally$ is defined in the two next sections.

A.1.7 Convective thermal resistance window to air

$xxxx_Rconv_window_air$ is the convective thermal resistance between the indoor window surface and the indoor air. According to Novakovic (1995) this value is defined as

$$xxxx_Rconv_window_air = \frac{1}{htc_{windowair} \cdot A_{window}} \quad (A.7)$$

where

A_{window} = area of the window (m^2)

$htc_{windowair}$ = heat transfer coefficient ($\frac{W}{m^2 \cdot K}$)

$htc_{windowair}$ is normally around $3W/m^2K$ (Novakovic 1995)

A.1.8 Thermal radiative resistance window to wall

$xxxx_Rradiation_window_wall$ is the thermal radiation resistance between the windows and each of the walls including the ceiling. According to Novakovic (1995) this value is defined as

$$xxxx_Rradiation_window_wall = \frac{1}{htc_{windowwall} \cdot A_{window}} \quad (A.8)$$

where

A_{window} = area of the window (m^2)

$htc_{windowwall}$ = heat transfer coefficient ($\frac{W}{m^2 \cdot K}$)

$htc_{windowwall}$ is normally around $5W/m^2K$ (Novakovic 1995)

A.1.9 Convective thermal resistance in the air due to ventilation

$xxxx_Rconv_ventilation$ is the convective resistance due to ventilation air mixed with air already in room $xxxx$. According to Novakovic (1995) this value is defined as

A. CALCULATION OF ELECTRICAL COMPONENTS IN THE BUILDING MODEL

$$xxxx_Rconv_ventilation = \frac{1}{\dot{m}_{air_vent} \cdot cp_{air}} \quad (A.9)$$

where

\dot{m}_{air_vent} = mass of the ventilated air per second ($\frac{kg}{s}$)

cp_{air} = specific heat capacity of air ($\frac{J}{kg \cdot K}$)

In typical room conditions specific heat capacity of the air is usually given as 1012J/kgK.

A.1.10 Convective thermal resistance in the air due to infiltration

xxxx_Rconv_infiltration is the convective resistance due to infiltration air mixed with air already in room xxxx. According to Novakovic (1995) this value is defined as

$$xxxx_Rconv_infiltration = \frac{1}{\dot{m}_{air_infiltration} \cdot cp_{air}} \quad (A.10)$$

where

$\dot{m}_{air_infiltration}$ = mass of the infiltrated air per second ($\frac{kg}{s}$)

cp_{air} = specific heat capacity of air ($\frac{J}{kg \cdot K}$)

$\dot{m}_{air_infiltration}$ is found from the following equation

$$\dot{m}_{air_infiltration} = \frac{xxxx_room_volume \cdot \rho_{air} \cdot ACH_{infiltration}}{hours} \quad (A.11)$$

where

$ACH_{infiltration}$ = number of air changes per hour

$ACH_{infiltration}$ is given as 0.2 in this report. According to Novakovic (1995) typical values for this is in the range of 0.1-0.3.

A.1.11 Thermal radiative resistance radiator to wall

$xxxx_Rradiation_radiator_wall$ is the thermal radiation resistance between the radiators and the wall. According to Novakovic (1995) this value is defined as

$$xxxx_Rradiation_radiator_wall = \frac{1}{htc_{radiatorwall} \cdot A_{radiator} \cdot N_{radiator}} \quad (A.12)$$

where

$A_{radiator}$ = area of the radiator (m^2)

$htc_{radiatorwall}$ = heat transfer coefficient ($\frac{W}{m^2 \cdot K}$)

$N_{radiator}$ = number of radiators in the room

In Novakovic (1995) the calculation of this value does not include the number of radiators. Instead of using the area of one radiator and multiplying it with the number of resistors in the room, a total radiator area is used. When calculating the radiator effect on the ceiling the exact same method is used, but the resistor is connected to the ceiling instead of the wall. $htc_{radiatorwall}$ is normally around $6 - 8W/m^2K$ (Novakovic 1995).

A.1.12 Convective thermal resistance radiator to wall

$xxxx_Rconv_radiator$ is the convective resistance between the radiator and the indoor air. This value is defined in Novakovic (1995) as

$$xxxx_Rconv_radiator = R_{prod} \quad (A.13)$$

R_{prod} is calculated by dividing the temperature difference between the radiator and the air with the total heating effect of the radiators. Usually this heating effect is in the region of 1500-4000 Watt for each radiator. In this report 1kW is used. To find the total heating effect, assuming each radiator is of the same type and size, the heating effect of each radiator must be multiplied by the numbers of radiator. It should be noted that $xxxx_Rconv_radiator$ is defined as

$$xxxx_Rconv_radiator = R_{prod} - xxxx_Rradiation_radiator_wall \quad (A.14)$$

A. CALCULATION OF ELECTRICAL COMPONENTS IN THE BUILDING MODEL

in Novakovic (1995). However the usage of equation A.14 may cause `xxxx_Rconv_radiator` to become negative. A negative resistor is of course not possible and a consultation with PhD Natasa Djuric who is working together with Vojislav Novakovics reveals that this equation should be replaced by Equation A.13 (Djuric 2010). Hence A.13 is used when calculating `xxxx_Rconv_radiator`.

A.2 Calculation of heat capacities

A.2.1 Heat capacity of the air inside a room

`xxxx_Ca` is the heat capacity of the air inside room `xxxx`. According to Novakovic (1995) this value is defined as

$$xxxx_Ca = V_{room} \cdot cp_{air} \cdot \rho_{air} \quad (A.15)$$

where

V_{room} = volume of the room (m^3)

cp_{air} = specific heat capacity of air ($\frac{J}{kg \cdot K}$)

ρ_{air} = density of air ($\frac{kg}{m^3}$)

The specific heat capacity and density of air can be found in the file `constants.m` found in Appendix D and on the attached CD.

A.2.2 Heat capacity of the wall

`xxxx_Cwy` is the heat capacity of wall `y` in room `xxxx`. According to Novakovic (1995) this value is defined as

$$xxxx_Cwy = V_{wall} \cdot cp_{wall} \cdot \rho_{wall} \quad (A.16)$$

where

V_{wall} = volume of the wall (m^3)

cp_{air} = specific heat capacity of the wall ($\frac{J}{kg \cdot K}$)

A.2 Calculation of heat capacities

ρ_{air} = density of the wall ($\frac{kg}{m^3}$)

The density of the wall and the specific heat capacity of the wall are calculated with the assumption that the wall exists of 90 % mineral wool and 10 % wood. See Appendix D for the details of how this is calculated. The exact same procedure as shown in this section and the preceding section is used when calculating every other heat capacities (e.g. the floor and ceiling).

A. CALCULATION OF ELECTRICAL COMPONENTS IN THE BUILDING MODEL

Appendix B

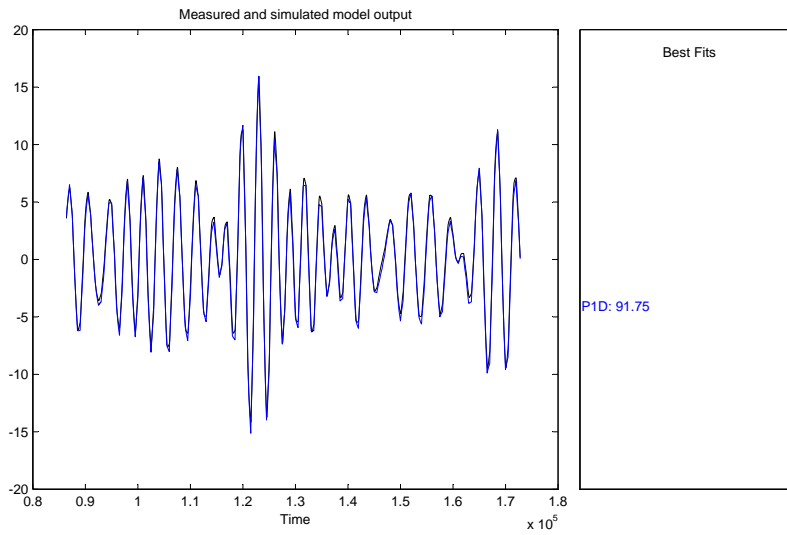
System identification

This appendix is a supplement to Chapter 4, System identification.

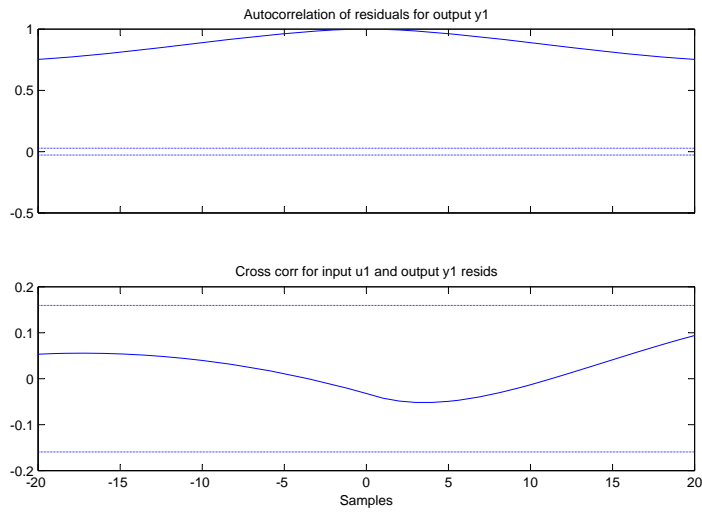
B.1 Model identification results

See Section 4.3.3 for the results for room k133.

B. SYSTEM IDENTIFICATION



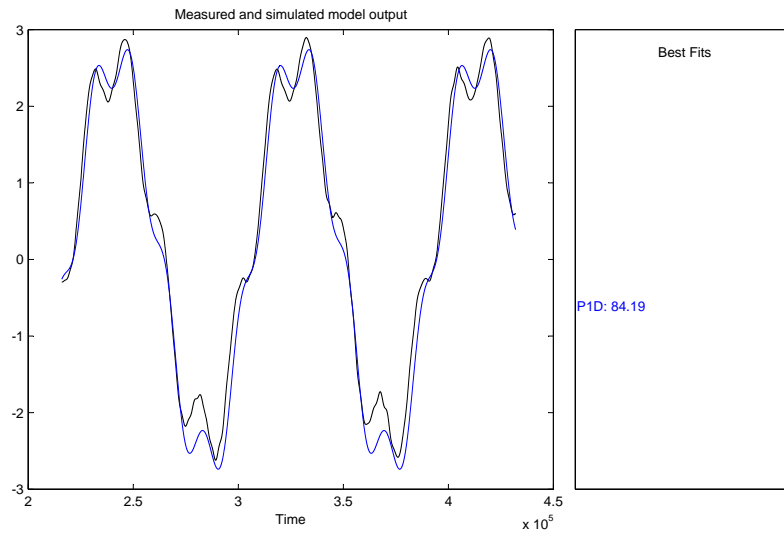
(a) Match



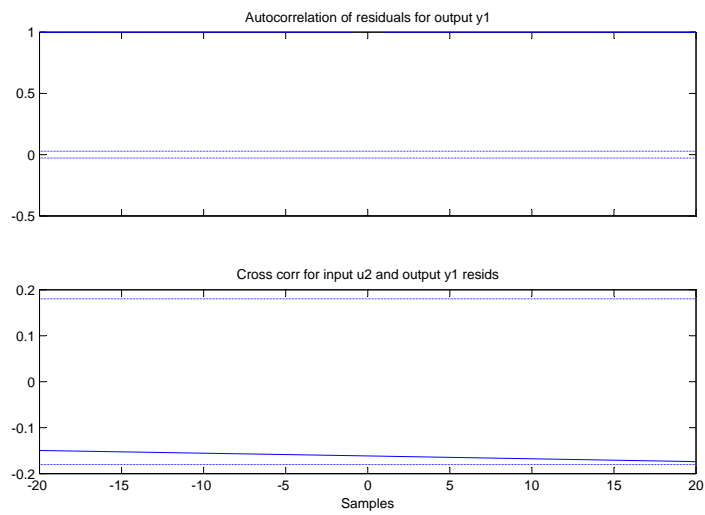
(b) Residual

Figure B.1: System identification results, room k134 - radiator.

B.1 Model identification results



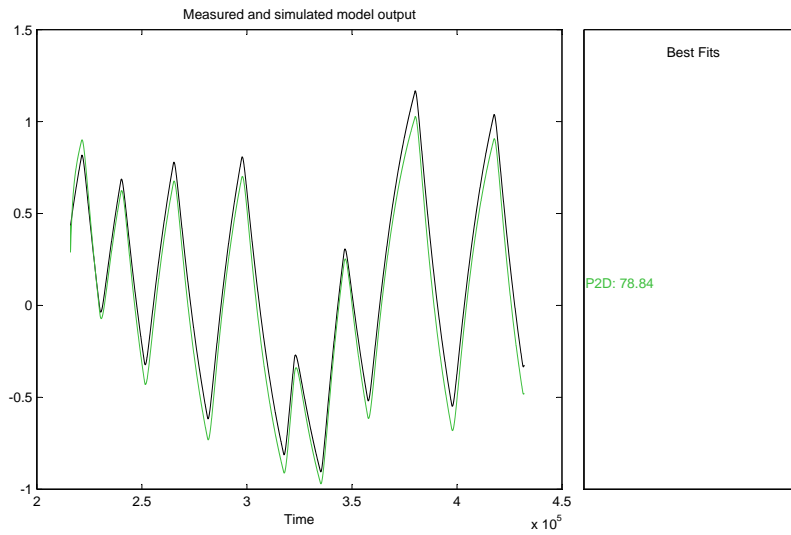
(a) Match



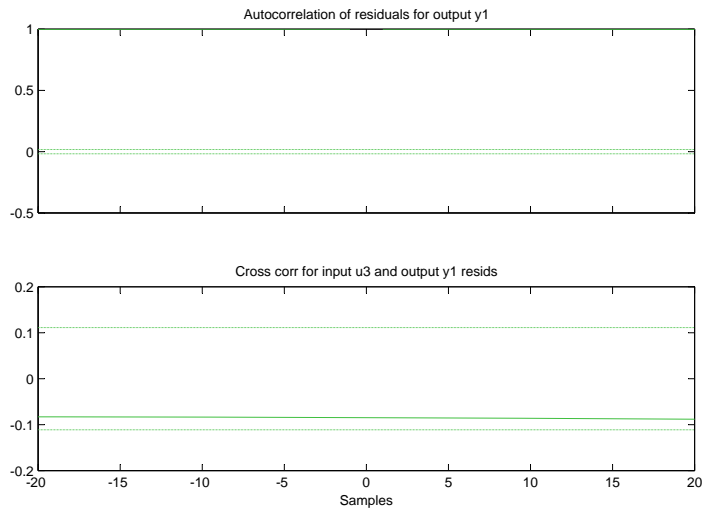
(b) Residual

Figure B.2: System identification results, room k134 - outdoor temperature.

B. SYSTEM IDENTIFICATION



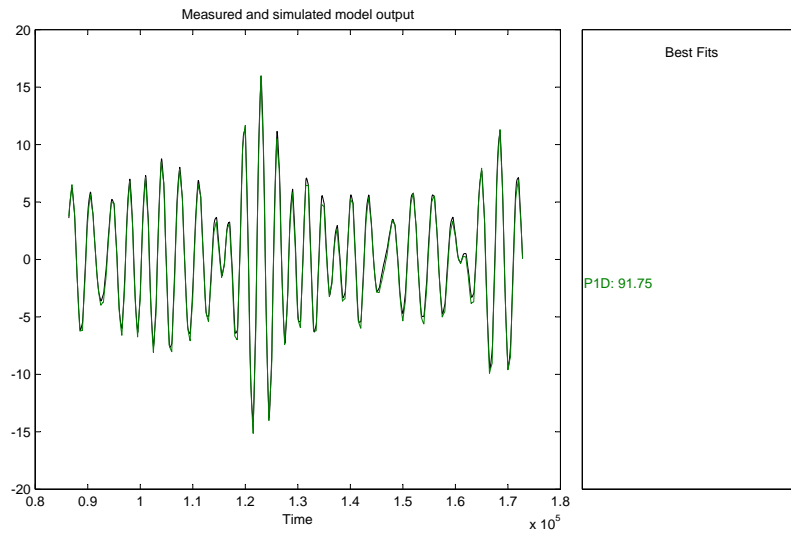
(a) Match



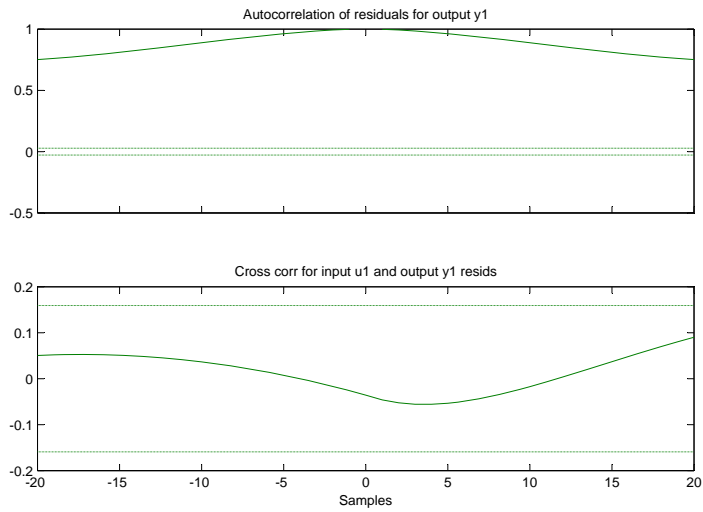
(b) Residual

Figure B.3: System identification results, room k134 - ventilation.

B.1 Model identification results



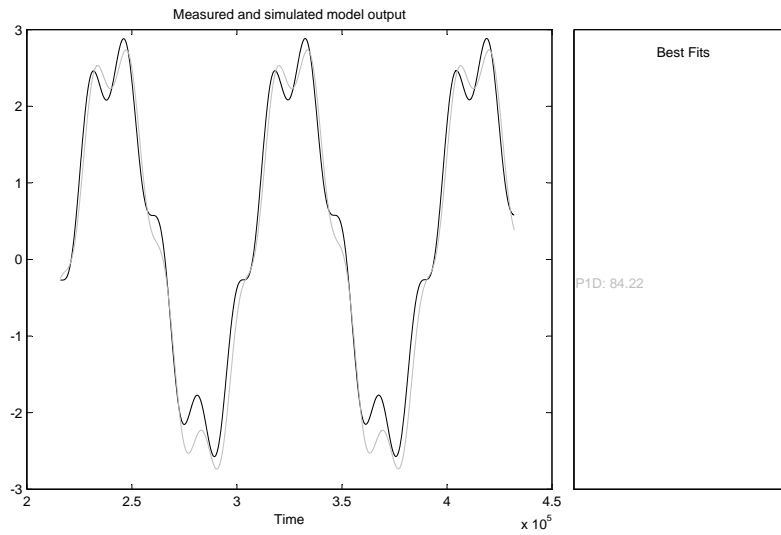
(a) Match



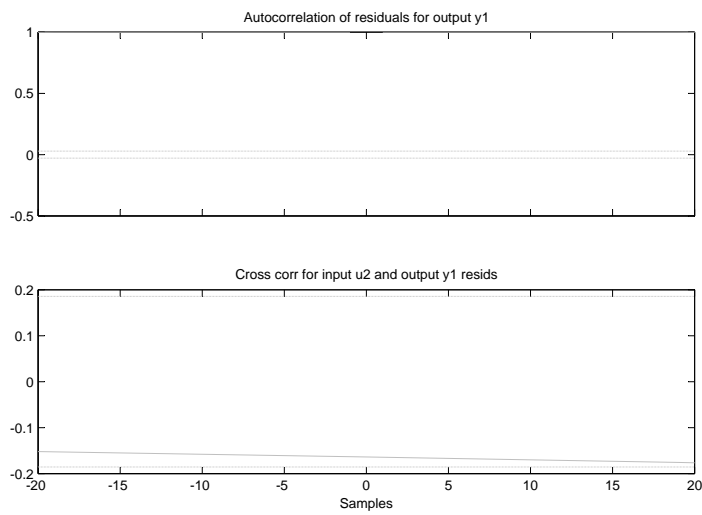
(b) Residual

Figure B.4: System identification results, room k135 - radiator.

B. SYSTEM IDENTIFICATION



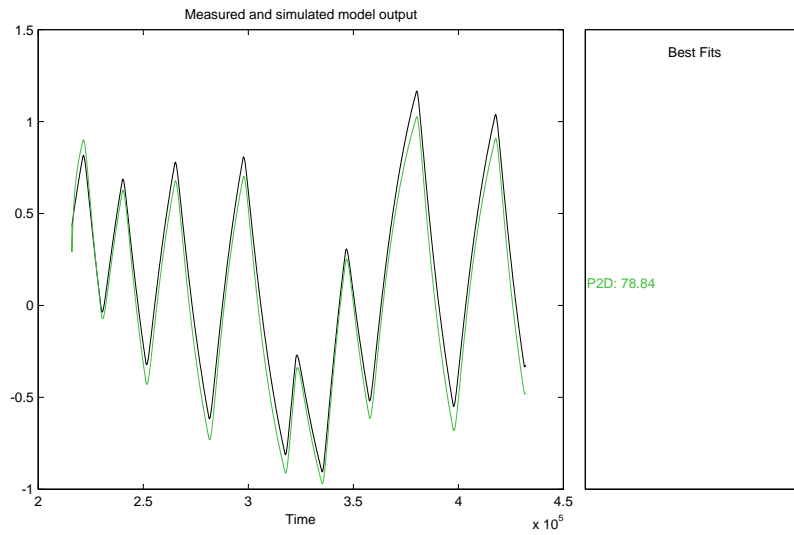
(a) Match



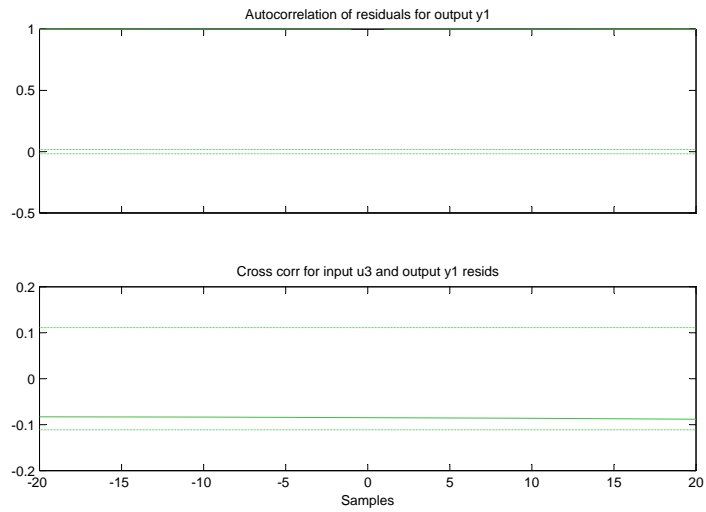
(b) Residual

Figure B.5: System identification results, room k135 - outdoor temperature.

B.1 Model identification results



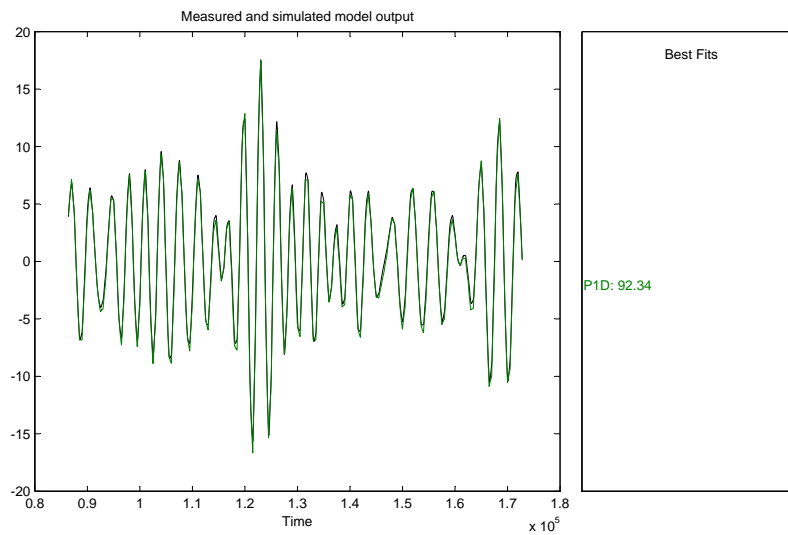
(a) Match



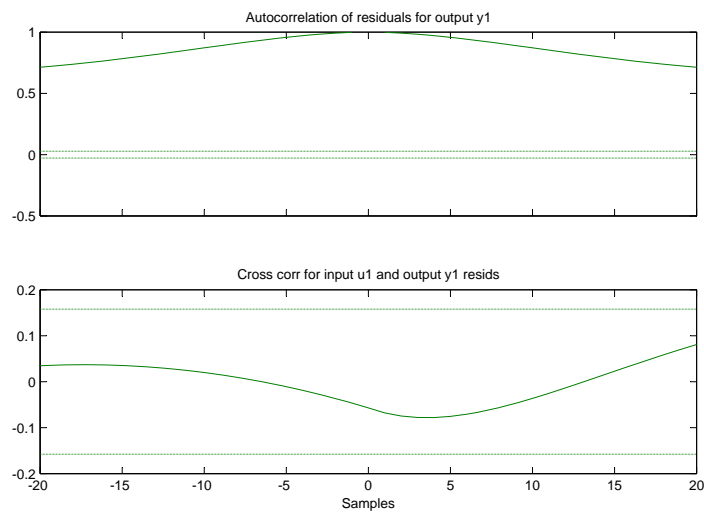
(b) Residual

Figure B.6: System identification results, room k135 - ventilation.

B. SYSTEM IDENTIFICATION



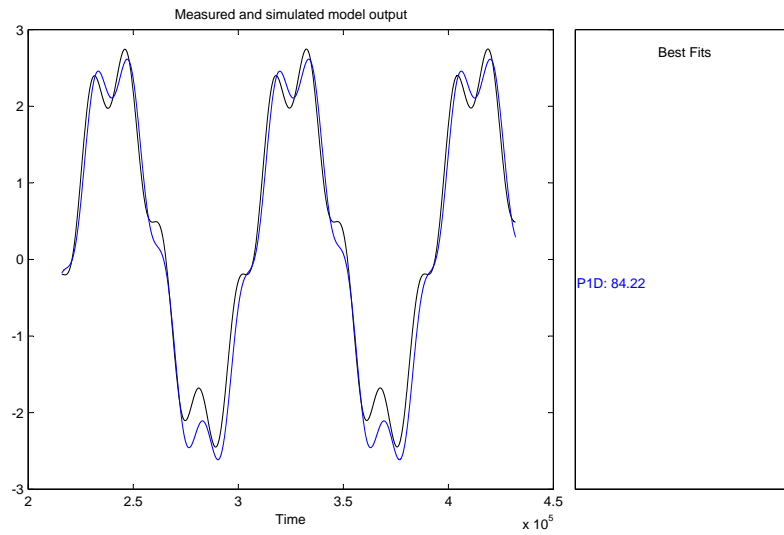
(a) Match



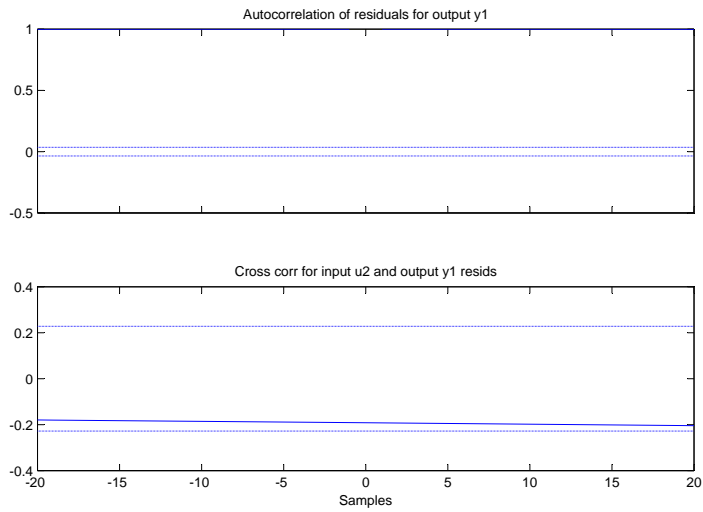
(b) Residual

Figure B.7: System identification results, room k136 - radiator.

B.1 Model identification results



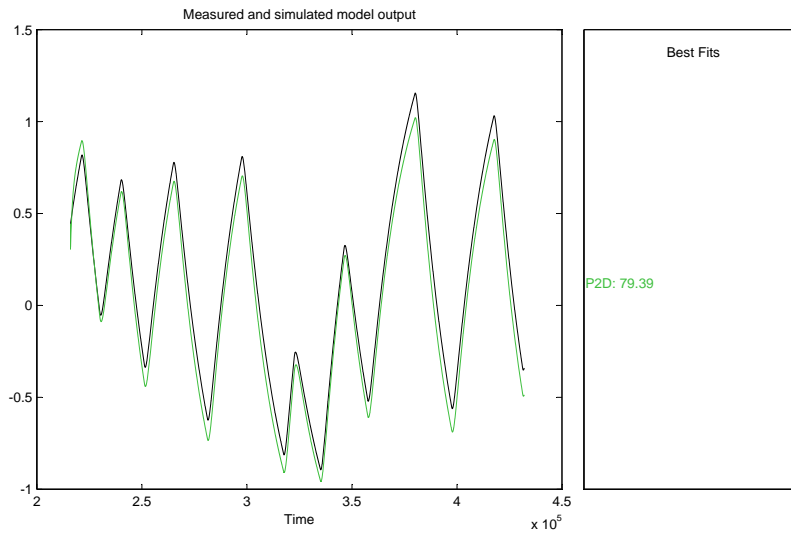
(a) Match



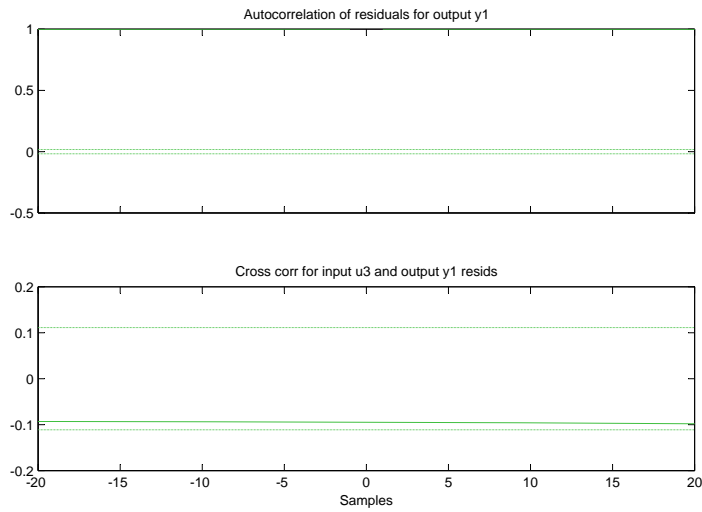
(b) Residual

Figure B.8: System identification results, room k136 - outdoor temperature.

B. SYSTEM IDENTIFICATION



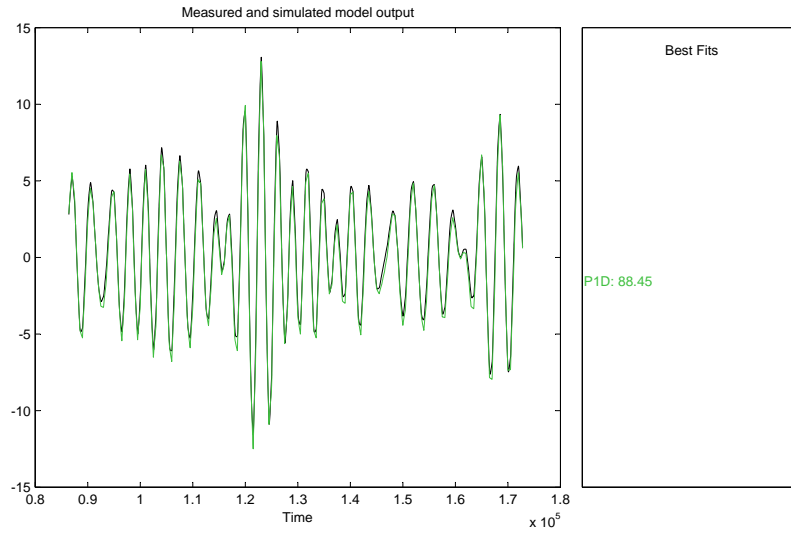
(a) Match



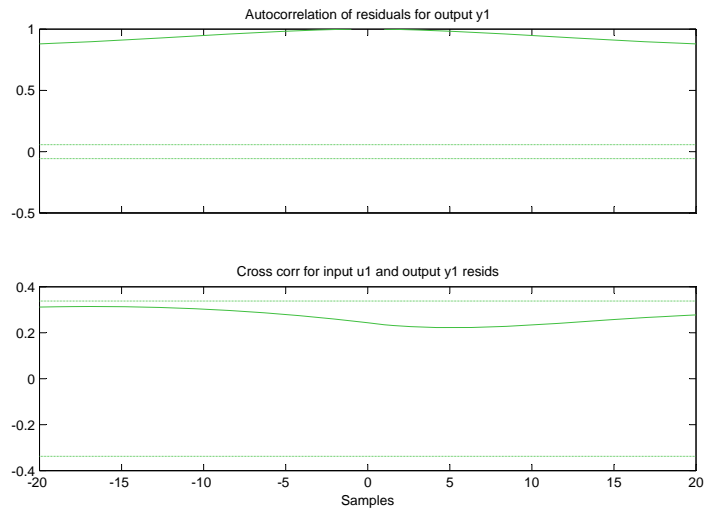
(b) Residual

Figure B.9: System identification results, room k136 - ventilation.

B.1 Model identification results



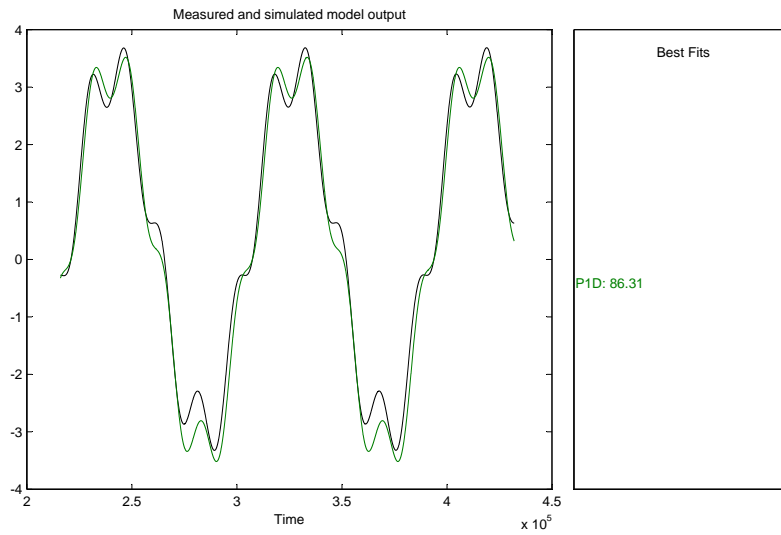
(a) Match



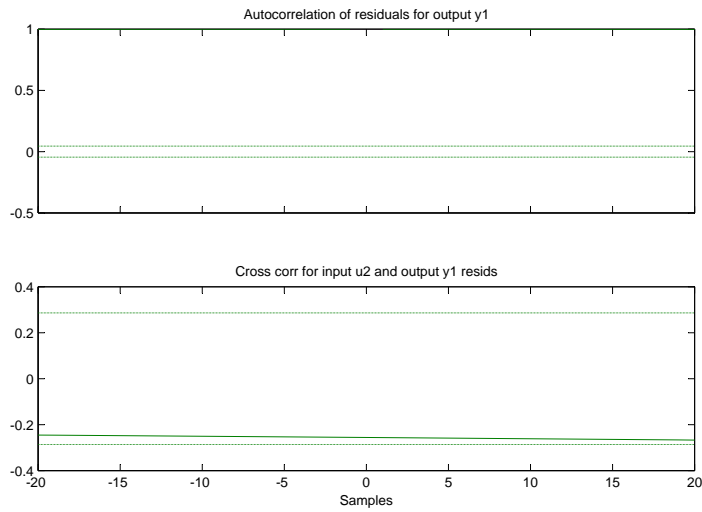
(b) Residual

Figure B.10: System identification results, room k143 - radiator.

B. SYSTEM IDENTIFICATION



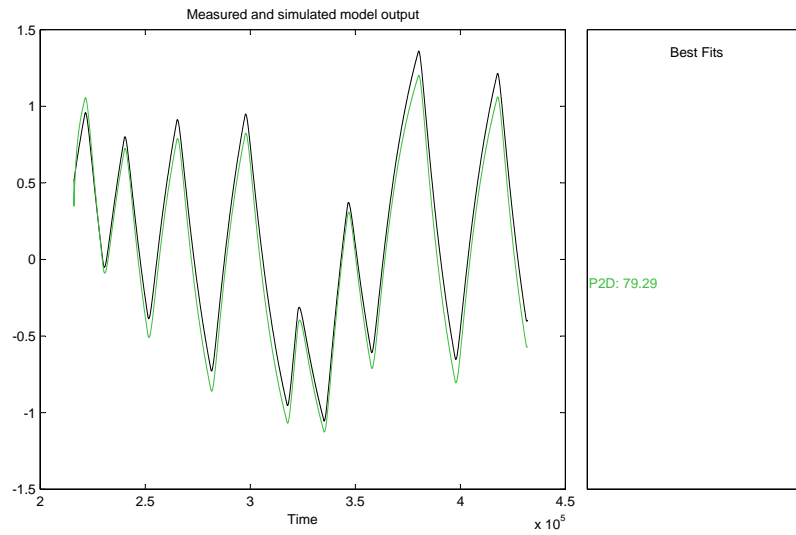
(a) Match



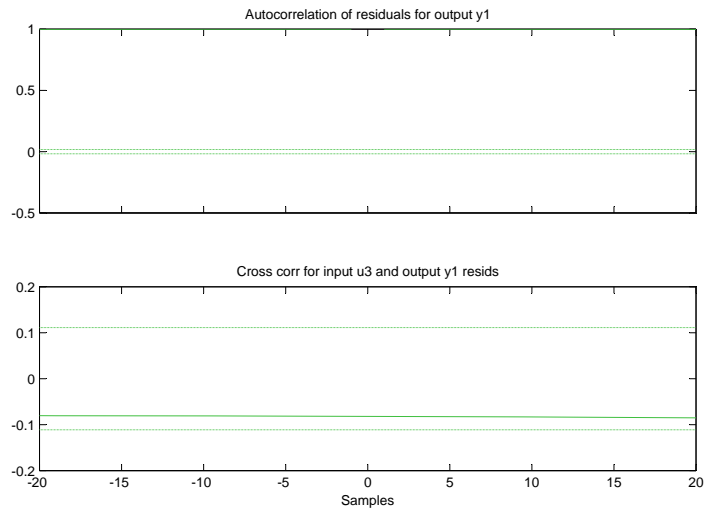
(b) Residual

Figure B.11: System identification results, room k135 - outdoor temperature.

B.1 Model identification results



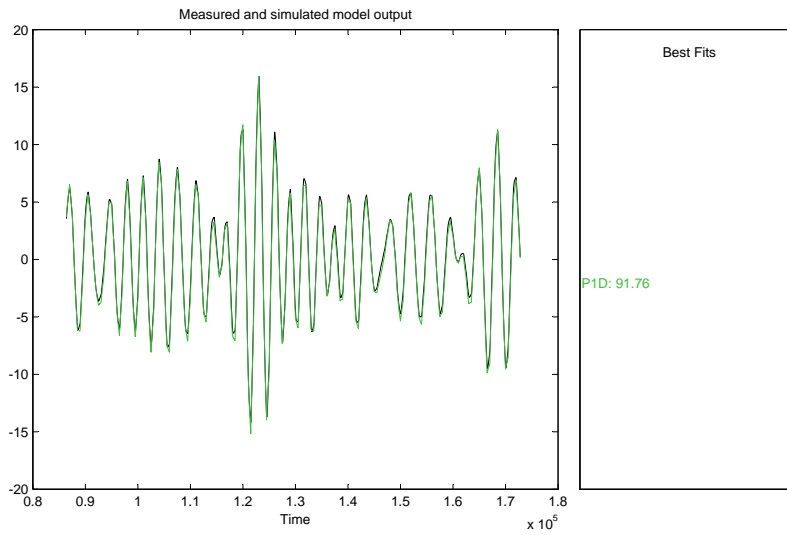
(a) Match



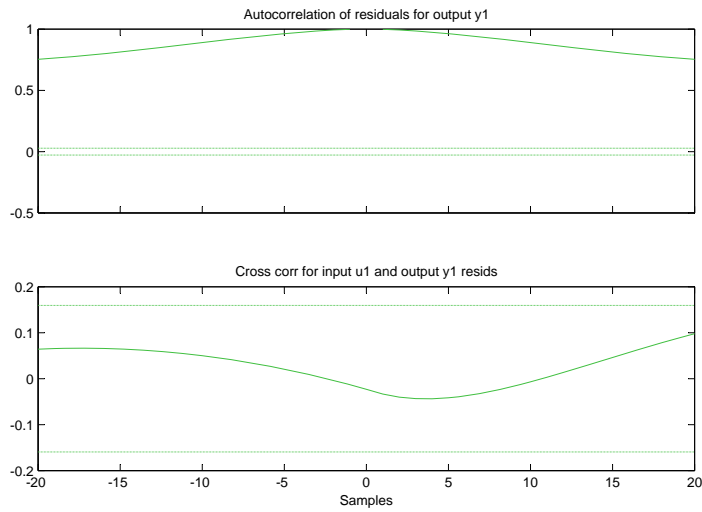
(b) Residual

Figure B.12: System identification results, room k143 - ventilation.

B. SYSTEM IDENTIFICATION



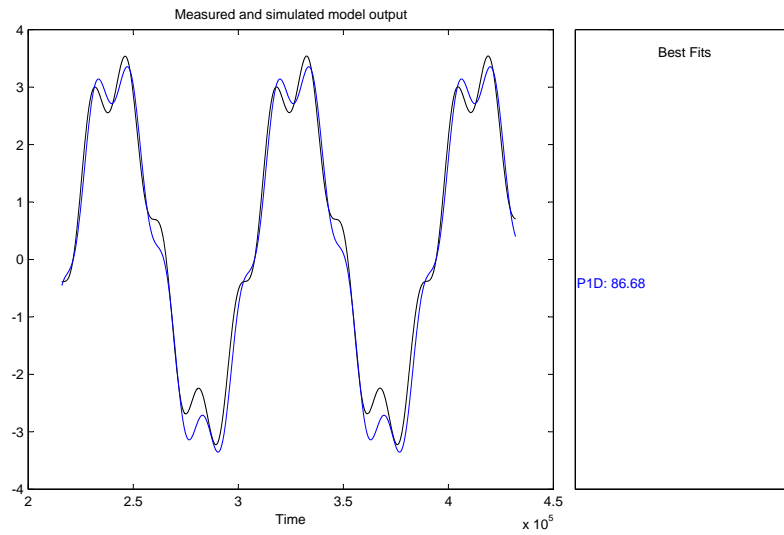
(a) Match



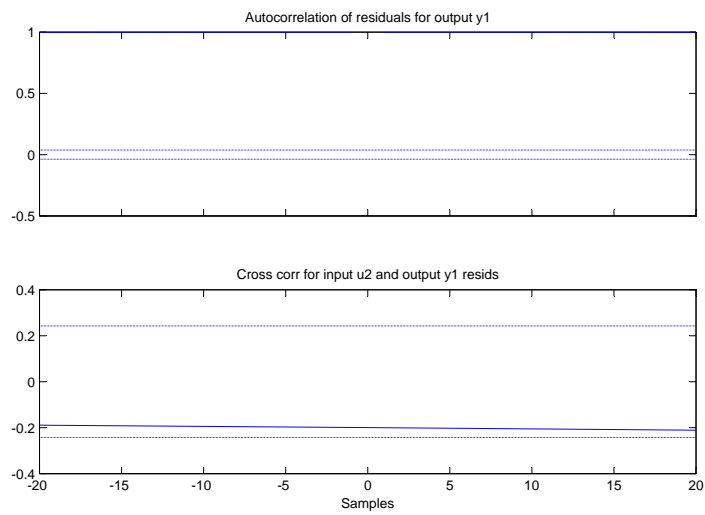
(b) Residual

Figure B.13: System identification results, room k164 - radiator.

B.1 Model identification results



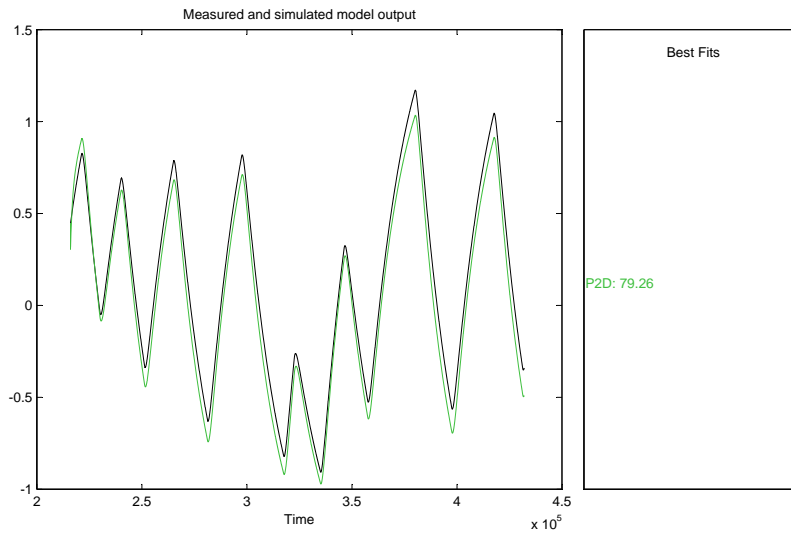
(a) Match



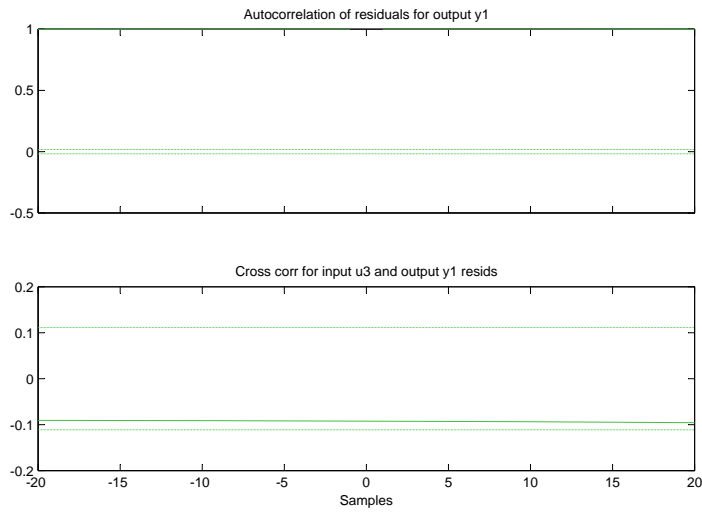
(b) Residual

Figure B.14: System identification results, room k164 - outdoor temperature.

B. SYSTEM IDENTIFICATION



(a) Match



(b) Residual

Figure B.15: System identification results, room k164 - ventilation.

Appendix C

Model predictive control

This appendix is a supplement to Section 5.3, Model predictive control.

C.1 State space representation of the rooms

C.1.1 Room k133

```

10.05.11 14:10 MATLAB Command Window 1 of 1

a =
      x1      x2      x3      x4      x5      x6      x7
x1 -0.0009867      0      0      0      0      0      0
x2      0 -0.0002734      0      0      0      0      0
x3      0      0 -0.8971      -0.3355 -0.2014 -0.01692 -0.001106
x4      0      0      0      0      0      0      0
x5      0      0      0      0.25      0      0      0
x6      0      0      0      0      0.01563      0      0
x7      0      0      0      0      0      0.0009766      0

b =
      u1      u2      u3
x1  0.03125      0      0
x2      0  0.007813      0
x3      0      0  0.01563
x4      0      0      0
x5      0      0      0
x6      0      0      0
x7      0      0      0

c =
      x1      x2      x3      x4      x5      x6      x7
y1  0.0231  0.004689      0 -8.744e-007  3.133e-006 -7.485e-005  0.01144

d =
      u1      u2      u3
y1  0      0      0

Continuous-time model.
>>

```

C.1.2 Room k134

,

10.05.11 14:11 MATLAB Command Window 1 of 1

```

a =
    x1    x2    x3    x4    x5    x6    x7
    -0.001005    0    0    0    0    0    0
    0    -0.0002596    0    0    0    0    0
    0    0    -0.8969    -0.3353    -0.2011    -0.01359    -0.0008146
    0    0    0    0    0    0    0
    0    0    0    0.25    0    0    0
    0    0    0    0    0.01563    0    0
    0    0    0    0    0    0.0009766    0

b =
    u1    u2    u3
    x1    0.03125    0    0
    x2    0    0.007813    0
    x3    0    0    0.01563
    x4    0    0    0
    x5    0    0    0
    x6    0    0    0
    x7    0    0    0

c =
    x1    x2    x3    x4    x5    x6    x7
    y1    0.02115    0.004564    0    -7.34e-007    2.63e-006    -6.283e-005    0.009605

d =
    u1    u2    u3
    y1    0    0    0

Continuous-time model.
>>

```

C. MODEL PREDICTIVE CONTROL

C.1.3 Room k135

```
10.05.11 14:11 MATLAB Command Window 1 of 1

a =
      x1      x2      x3      x4      x5      x6      x7
      -0.001015      0      0      0      0      0      0
      0      -0.0002608      0      0      0      0      0
      0      0      -0.8969      -0.3353      -0.2011      -0.01359      -0.0008143
      0      0      0      0      0      0      0
      0      0      0      0.25      0      0      0
      0      0      0      0      0.01563      0      0
      0      0      0      0      0      0.0009766      0

b =
      u1      u2      u3
      0.03125      0      0
      0      0.007813      0
      0      0      0.01563
      0      0      0
      0      0      0
      0      0      0
      0      0      0

c =
      x1      x2      x3      x4      x5      x6      x7
      0.02123      0.004578      0      -7.34e-007      2.63e-006      -6.283e-005      0.009605

d =
      u1      u2      u3
      0      0      0

Continuous-time model.
>>
```

C.1.4 Room k136

,

10.05.11 14:11 MATLAB Command Window 1 of 1

```

a =
      x1      x2      x3      x4      x5      x6      x7
x1 -0.000962      0      0      0      0      0      0
x2      0 -0.0002928      0      0      0      0      0
x3      0      0 -0.8969      -0.3354 -0.2012 -0.01443 -0.0008983
x4      0      0      0      0      0      0      0
x5      0      0      0      0.25      0      0      0
x6      0      0      0      0      0.01563      0      0
x7      0      0      0      0      0      0.0009766      0

b =
      u1      u2      u3
x1 0.03125      0      0
x2      0 0.007813      0
x3      0      0 0.01563
x4      0      0      0
x5      0      0      0
x6      0      0      0
x7      0      0      0

c =
      x1      x2      x3      x4      x5      x6      x7
y1 0.02324 0.00487      0 -7.82e-007 2.802e-006 -6.694e-005 0.01023

d =
      u1      u2      u3
y1 0 0 0

Continuous-time model.
>>

```

C. MODEL PREDICTIVE CONTROL

C.1.5 Room k143

```
10.05.11 14:12 MATLAB Command Window 1 of 1

a =
      x1      x2      x3      x4      x5      x6      x7
x1 -0.0007342      0      0      0      0      0      0
x2      0 -0.0003205      0      0      0      0      0
x3      0      0 -0.8968      -0.3352 -0.201 -0.01255 -0.0007722
x4      0      0      0      0      0      0      0
x5      0      0      0      0.25      0      0      0
x6      0      0      0      0      0.01563      0      0
x7      0      0      0      0      0      0.0009766      0

b =
      u1      u2      u3
x1      0.03125      0      0
x2      0      0.007813      0
x3      0      0      0.01563
x4      0      0      0
x5      0      0      0
x6      0      0      0
x7      0      0      0

c =
      x1      x2      x3      x4      x5      x6      x7
y1      0.01629      0.007126      0      -8.097e-007      2.901e-006      -6.93e-005      0.01059

d =
      u1      u2      u3
y1      0      0      0

Continuous-time model.
>>
```

C.1.6 Room k164

```

10.05.11 14:12 MATLAB Command Window 1 of 1

a =
      x1      x2      x3      x4      x5      x6      x7
x1 -0.0009801  0      0      0      0      0      0
x2  0 -0.0009801  0      0      0      0      0
x3  0  0 -0.8969  -0.3353 -0.2012 -0.01417 -0.0008735
x4  0  0  0      0      0      0      0
x5  0  0  0      0.25  0      0      0
x6  0  0  0      0      0.01563  0      0
x7  0  0  0      0      0      0.0009766  0

b =
      u1      u2      u3
x1  0.03125  0      0
x2  0  0.01563  0
x3  0  0  0.01563
x4  0  0  0
x5  0  0  0
x6  0  0  0
x7  0  0  0

c =
      x1      x2      x3      x4      x5      x6      x7
y1  0.02104  0.01048  0 -7.762e-007  2.781e-006  -6.644e-005  0.01016

d =
      u1 u2 u3
y1  0  0  0

Continuous-time model.
>>

```

C. MODEL PREDICTIVE CONTROL

Appendix D

M-files used in the model

This appendix includes the most important of the m-files used in the report. All these files are also included on the attached CD.

D. M-FILES USED IN THE MODEL

D.1 Initialization

17.06.11 13:46 C:\Users\alvsvag\Documents\My Dropbox\...\init_building_mpc.m 1 of 2

```
clear all
close all
clc %clears the command window

%% -----initial values-----
average_ext_temp = 5;
indoor_temp_adj_rooms = 18;
return_air_temp = 18;
internal_temp_room_init = 20;
internal_temp_wall_init = 20;
internal_temp_floor_init = 20;
internal_temp_ceiling_init = 20;
water_temp_from_supply = 110;
room_set_temp=21;
ventilation_temp_set = 18;

%PID values
Kp_room = 50;
Ki_room = 0.05;
Kd_room = 0;
Kp_hc = 0.2;
Ki_hc = 0.002;
Kd_hc = 0;

%definition of time
second = 1;
minute = 60*second;
hour = 60*minute;
day = 24*hour;

%% -----load parameters-----
%loads all parameters used in the model
run constants;
run room_values;
run input_parameters;
run preprocessing;
run mpc_preparations;
run mpc;

%% ----- simulation -----
numdays = 1; %number of days to simulate
tstart = 0; %Sim start time
tstop = numdays*day; % Sim stop time, day is equal to twenty-four hours
(60seconds*60minutes*24hours)
tstep = 2*minute-10; % Sim step time
t=tstart:tstep:tstop; %time array

%start simulation
sim active_building_new_model_mpc_one_room

%% ----- calculation of energy consumption -----

%energy consumption - ventilation
run energy_consumption_ventilation;
```

D.1 Initialization

17.06.11 13:46 C:\Users\alvsvag\Documents\My Dropbox\...\init_building_mpc.m 2 of 2

```
%energy consumption - radiators  
run energy_consumption_radiators;  
  
sim active_building_new_model_pid  
  
run energy_consumption_radiators2;
```

D. M-FILES USED IN THE MODEL

D.2 Constants

04.12.10 15:09 C:\Documents and Settings\Øyvind\Mine dokument...\constants.m 1 of 3

```
%% -----Constants-----
% -----

%-----global room values-----
room_height = 3;
depth_external_wall = 0.15;
depth_internal_wall = 0.10;
depth_floor = 0.25;
depth_external_ceiling = depth_floor;
depth_wall_sheathing = 0.002; %thickness of the wallpaper
depth_floor_sheathing = 0.003; %thickness of the floor covering (gulvbelegg)
depth_ceiling_sheathing = 0.01; %thickness of the ceiling covering (trekledning)
ACH_infiltration = 0.2; %air changes per hour (ACHs). typical value is 0.1-0.3.
ACH_ventilation = 1; %air changes per hour (ACHs)
%Radiators
Radiator_heat_effect = 2000; %W
htc_rad = 7; %normally between 6-8 W/(m^2*K)
A_rad = 1; %the radiators radiation area

%-----global U-factors and heat transfer coefficients-----
%Wall
U_external_wall = 0.31; %found in table 211 in u_value_wall.pdf from sintef byggforsk
U_internal_wall = 0.41; %found in table 214 in u_value_wall.pdf from sintef byggforsk
htc_wallair = 3; %typical values found in novakovic
lambda_external_wall = U_external_wall*depth_external_wall; %equation from sintef
byggforsk
lambda_wall_sheathing = 0.03; % lambda value for 6mm gypsum board (gipsplate) found
in varmekonduktivitet_varmemotstand_bygningsmaterialer.pdf from sintef byggforsk
%Window
U_window = 2.8; % Isolerglass 2-lags glass. Found in u_value_window.pdf from nor-dan
htc_windowwall = 5; %typical values found in novakovic
htc_windowair = 3; %typical values found in novakovic
%Ceiling
U_external_ceiling = 0.22; %found in table 22 in u_value_ceiling.pdf from sintef
byggforsk
htc_ceilingair = 3; %typical values found in novakovic
lambda_external_ceiling = U_external_wall*depth_external_ceiling; %equation from
sintef byggforsk
lambda_ceiling_sheathing = U_external_ceiling*depth_ceiling_sheathing;
%Floor
U_floor = 0.15; % found in table 23 in u_value_floor.pdf from sintef byggforsk (this
value depends on the area of the floor, but does not vary much. Therefor a fixed u-
value is used in every room although the areas are varying)

%-----physical values - external walls-----
%specific heat capacity - J/(kg*K)
cp_external_wall = 2700*0.1+1030*0.9; %assuming the wall consists of 20% wood with
specific heat capacity 2700J/(kg*K) and 80% insulation with specific heat capacity
1030J/(kg*K)
%density - kg/m^3
rho_external_wall = 500*0.1+31*0.9; %assuming the wall consists of 20% wood with
density 500kg/m^3 and 80% insulation with specific heat capacity 31kg/m^3

%-----physical values - internal walls-----
%specific heat capacity - J/(kg*K)
```

D.2 Constants

04.12.10 15:09 C:\Documents and Settings\Øyvind\Mine dokument...\constants.m 2 of 3

```
cp_internal_wall = 2700*0.1+1030*0.9; %assuming the wall consists of 20% wood with
specific heat capacity 2700J/(kg*K) and 80% insulation with specific heat capacity
1030J/(kg*K)
%density - kg/m^3
rho_internal_wall = 500*0.1+31*0.9; %assuming the wall consists of 20% wood with
density 500kg/m^3 and 80% insulation with specific heat capacity 31kg/m^3

%-----physical values - ceiling-----
%specific heat capacity - J/(kg*K)
cp_ceiling = 2700*0.1+1030*0.9; %assuming the wall consists of 20% wood with specific
heat capacity 2700J/(kg*K) and 80% insulation with specific heat capacity 1030J/
(kg*K)
%density - kg/m^3
rho_ceiling = 500*0.1+31*0.9; %assuming the ceiling consists of 20% wood with density
500kg/m^3 and 80% insulation with specific heat capacity 31kg/m^3

%-----physical values - floor-----
%specific heat capacity - J/(kg*K)
cp_floor = 2700*0.1+1030*0.9; %assuming the floor consists of 20% wood with specific
heat capacity 2700J/(kg*K) and 80% insulation with specific heat capacity 1030J/
(kg*K)
%density - kg/m^3
rho_floor = 500*0.1+31*0.9; %assuming the floor consists of 20% wood with density
500kg/m^3 and 80% insulation with specific heat capacity 31kg/m^3

%-----physical values - air-----
%conductivity - W/(m*K)
lambda_air = 0.025;
%density - kg/m^3
rho_air = 1.2;
%specific heat capacity - J/(kg*K)
cp_air = 1012;
%kinematic viscosity - m^2/s
v_air = 15*10^(-6);
%Prandtl's number - dimensionless
Pr_air = 0.72;

%-----physical values - water-----
%diffusivity = (thermal conductivity)/(volumetric heat capacity) - m^2/s
alpha_water = 0.00000016;
%conductivity - W/(m*K)
lambda_water = 0.66;
%density - kg/m^3
rho_water = 980;
%specific heat capacity - J/(kg*K)
cp_water = 4180;
%kinematic viscosity - m^2/s
v_water = 0.5*10^(-6);
%Prandtl's number - dimensionless
Pr_water = 3.6;

%-----other physical values-----
%conductivity steel - W/(m*K)
lambda_steel = 40;
%conductivity insulation - W/(m*K)
```

D. M-FILES USED IN THE MODEL

04.12.10 15:09 C:\Documents and Settings\Øyvind\Mine dokument...\constants.m 3 of 3

```
lambda_insulation = 0.05;  
%emissivity for tube surface  
epsilon_tube = 0.5;  
%emissivity for insulation surface  
epsilon_insulation = 0.8;  
%specific heat capacity steel - J/(kg*K)  
cp_steel = 460;  
%specific heat capacity copper - J/(kg*K)  
cp_copper = 390;  
%specific heat capacity aluminum - J/(kg*K)  
cp_aluminum = 900;  
%density steel - kg/m^3  
rho_steel = 7800;
```

```
%-----
```

D.3 Room values

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 1 of 11

```

%% Cardinal points and the definition of length, width and the different walls
%
%
%           North
%          /\
%          ||
%          ||
% West<---->---->East
%          ||
%          ||
%          \
%           South
%
%wallN = "north wall"
%wallE = "east wall"
%wallS = "south wall"
%wallW = "west wall"
%wallNE = "northeast wall"
%wallNW = "northwest wall"
%wallSE = "southeast wall"
%wallSW = "southwest wall"
%
%length = west<->east
%width = north<->south
%
%% -----classroom 133-----
%--Room dimensions--
k133_length = 8.325;
k133_width = 7.125;
k133_height = room_height;
%
k133_window_area = (1.32*1.4)+(0.8*1.4)+(0.87*0.36)+(0.87*0.71)+(0.87*0.36)+(0.87*0.71)+(0.87*0.36)+(0.87*0.71)+(0.87*0.36)+(0.87*0.71)+(0.87*0.36)+(0.87*0.71);
k133_wallN_area = (room_height*k133_length)-k133_window_area;
k133_wallE_area = room_height*k133_width;
k133_wallS_area = room_height*k133_length;
k133_wallW_area = room_height*k133_width;
k133_floor_area = k133_length*k133_width;
k133_ceiling_area = k133_floor_area;
%
k133_room_volume = room_height*k133_floor_area;
k133_wallN_volume = k133_wallN_area*depth_external_wall;
k133_wallE_volume = k133_wallE_area*depth_internal_wall;
k133_wallS_volume = k133_wallS_area*depth_internal_wall;
k133_wallW_volume = k133_wallW_area*depth_external_wall;
k133_floor_volume = k133_floor_area*depth_floor;
k133_ceiling_volume = k133_ceiling_area*depth_external_ceiling;
%
%--Calculation resistances (resistance the heat must go through to pass through a material)--
% xxxx_Roy = outer thermal transmittance of external wall y in room xxxx
k133_RoN = 1/(U_external_wall*k133_wallN_area);
k133_RoW = 1/(U_external_wall*k133_wallW_area);
k133_RoS = 1/(U_external_wall*k133_wallS_area);
% xxxx_Riy = inner thermal transmittance of external wall y in room xxxx

```

D. M-FILES USED IN THE MODEL

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 2 of 11

```
k133_RiN = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
(2*lambda_wall_sheathing)))*(1/k133_wallN_area); %1/(inner_wall_U*k133_wallN_area);
k133_RiW = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
(2*lambda_wall_sheathing)))*(1/k133_wallW_area);
k133_RiS = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
(2*lambda_wall_sheathing)))*(1/k133_wallS_area);
% xxxx_Rinternal_wallx = thermal transmittance of internal wall in room xxxx
k133_RinternalE = 1/(2*U_internal_wall*k133_wallE_area); %internal_wall_U is
multiplied by 2 since only half of the wall thickness is to be considered
% xxxx_Rconv_wallx_air = convective resistance between wall n and the air in room
xxxx
k133_Rconv_wallN_air = 1/(htc_wallair*k133_wallN_area); %htc_wallair=convective heat
transfer factor between wallsurface and air
k133_Rconv_wallE_air = 1/(htc_wallair*k133_wallE_area);
k133_Rconv_wallS_air = 1/(htc_wallair*k133_wallS_area);
k133_Rconv_wallW_air = 1/(htc_wallair*k133_wallW_area);
% xxxx_Rfloor = floor thermal transmittance in room xxxx
k133_Rfloor = 1/(U_floor*k133_floor_area);
% xxxx_Rceiling = ceiling thermal transmittance in room xxxx
k133_Rceiling = 1/(U_external_ceiling*k133_ceiling_area);
k133_Riceiling = ((depth_external_ceiling/(2*lambda_external_ceiling))+
(depth_ceiling_sheathing/(2*lambda_ceiling_sheathing)))*(1/k133_ceiling_area);
% xxxx_Rconv_ceiling_air = convective resistance between wall n and the air in room
xxxx
k133_Rconv_ceiling_air = 1/(htc_ceilingair*k133_ceiling_area); %
htc_ceilingair=convective heat transfer factor between ceiling surface and air
% xxxx_Rconv_window_air = window-air heat convection resistance
k133_Rconv_window_air = 1/(htc_windowair*k133_window_area);
% Radiation resistance wall-window
k133_Rradiation_window_wall = 1/(htc_windowwall*k133_window_area);
% xxxx_Rwindow = window thermal transmittance in room xxxx
k133_Rwindow = 1/(U_window*k133_window_area)-1/
(1/k133_Rconv_window_air+1/k133_Rradiation_window_wall);
% Calculation of infiltration (air leakage through the walls, ceiling and floor)
resistance
mass_dot_infiltration = (k133_room_volume*rho_air*ACH_infiltration)/hour; %air (kg/s)
into the room by infiltration
k133_Rconv_infiltration = 1/(mass_dot_infiltration*cp_air);
% Calculation of ventilation (air exchange in the room due to ventilation)
resistance. ACH_infiltration determines the rate
mass_dot_ventilation = (k133_room_volume*rho_air*ACH_ventilation)/hour; %air (kg/s)
into the room by infiltration. ACH_ventilation determines the rate
k133_Rconv_ventilation = 1/(mass_dot_ventilation*cp_air);
% xxxx_Rradiator = radiator thermal transmittance in room xxxx
k133_number_of_radiators = 4;
k133_Rradiation_radiator_wall = 1/(htc_rad*A_rad*k133_number_of_radiators);
% xxxx_Rconv_radiator = convective heat transfer between radiator surface temperature
and air
k133_Rprod = (60-20)/(Radiator_heat_effect*k133_number_of_radiators);
k133_Rconv_radiator = k133_Rprod;

%--Calculation of capacitances (heat stored in mass)--
% capacitance of the air in the room
k133_Ca = k133_room_volume*rho_air*cp_air;
% capacitance of the walls
```


D.3 Room values

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 3 of 11

```
k133_CwN = k133_wallN_volume*rho_external_wall*cp_external_wall;
k133_CwE = k133_wallE_volume*rho_internal_wall*cp_internal_wall;
k133_CwS = k133_wallS_volume*rho_external_wall*cp_external_wall;
k133_CwW = k133_wallW_volume*rho_external_wall*cp_external_wall;
% capasitance of the floor
k133_Cf = k133_floor_volume*rho_floor*cp_floor;
% capasitance of the ceiling
k133_Cc = k133_ceiling_volume*rho_ceiling*cp_ceiling;
%-----

%% -----classroom 134-----
%--Room dimensions--
k134_length = 4.650;
k134_width = 7.125;
k134_heigth = room_height;

k134_window_area = (1.32*1.4)+(0.8*1.4)+(2.29*0.71);
k134_wallN_area = (room_height*k134_length)-k134_window_area;
k134_wallE_area = room_height*k134_width;
k134_wallS_area = room_height*k134_length;
k134_wallW_area = room_height*k134_width;
k134_floor_area = k134_length*k134_width;
k134_ceiling_area = k134_floor_area;

k134_room_volume = room_height*k134_floor_area;
k134_wallN_volume = k134_wallN_area*depth_external_wall;
k134_wallE_volume = k134_wallE_area*depth_internal_wall;
k134_wallS_volume = k134_wallS_area*depth_internal_wall;
k134_wallW_volume = k134_wallW_area*depth_internal_wall;
k134_floor_volume = k134_floor_area*depth_floor;
k134_ceiling_volume = k134_ceiling_area*depth_external_ceiling;

%--Calculation resistances (resistance the heat must go through to pass through a
material)--
% xxxx_Roy = outer thermal transmittance of external wall y in room xxxx
k134_RoN = 1/(U_external_wall*k134_wallN_area);
k134_RoS = 1/(U_external_wall*k134_wallS_area);
% xxxx_Riy = inner thermal transmittance of external wall y in room xxxx
k134_RiN = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
(2*lambda_wall_sheathing)))*(1/k134_wallN_area); %1/(inner_wall_U*k133_wallN_area);
k134_RiS = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
(2*lambda_wall_sheathing)))*(1/k134_walls_area);
% xxxx_Rinternal_wallx = thermal transmittance of internal wall in room xxxx
k134_RinternalE = 1/(2*U_internal_wall*k134_wallE_area); %internal_wall_U is
multiplied by 2 since only half of the wall thickness is to be considered
k134_RinternalW = 1/(2*U_internal_wall*k134_wallW_area);
% xxxx_Rconv_wallx_air = convective resistance between wall n and the air in room
xxxx
k134_Rconv_wallN_air = 1/(htc_wallair*k134_wallN_area); %htc_wallair=convective heat
transfer factor between wallsurface and air
k134_Rconv_wallE_air = 1/(htc_wallair*k134_wallE_area);
k134_Rconv_wallS_air = 1/(htc_wallair*k134_wallS_area);
k134_Rconv_wallW_air = 1/(htc_wallair*k134_wallW_area);
% xxxx_Rfloor = floor thermal transmittance in room xxxx
k134_Rfloor = 1/(U_floor*k134_floor_area);
```

D. M-FILES USED IN THE MODEL

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 4 of 11

```
% xxxx_Rceiling = ceiling thermal transmittance in room xxxx
k134_Ro ceiling = 1/(U_external_ceiling*k134_ceiling_area);
k134_Riceiling = ((depth_external_ceiling/(2*lambda_external_ceiling))+
(depth_ceiling_sheathing/(2*lambda_ceiling_sheathing)))*(1/k134_ceiling_area);
% xxxx_Rconv_ceiling_air = convective resistance between wall n and the air in room
xxxx
k134_Rconv_ceiling_air = 1/(htc_ceilingair*k134_ceiling_area); %
htc_ceilingair=convective heat tranfer factor between ceiling surface and air
% xxxx_Rconv_window_air = window-air heat convection resistance
k134_Rconv_window_air = 1/(htc_windowair*k134_window_area);
% Radiation resistance wall-window
k134_Rradiation_window_wall = 1/(htc_windowwall*k134_window_area);
% xxxx_Rwindow = window thermal transmittance in room xxxx
k134_Rwindow = 1/(U_window*k134_window_area)-1/
(1/k134_Rconv_window_air+1/k134_Rradiation_window_wall);
% Calculation of infiltration (air leakage through the walls, ceiling and floor)
resistance
mass_dot_infiltration = (k134_room_volume*rho_air*ACH_infiltration)/hour; %air (kg/s)
into the room by infiltration
k134_Rconv_infiltration = 1/(mass_dot_infiltration*cp_air);
% Calculation of ventilation (air exchange in the room due to ventilation)
resistance. ACH_infiltration determines the rate
mass_dot_ventilation = (k134_room_volume*rho_air*ACH_ventilation)/hour; %air (kg/s)
into the room by infiltration. ACH_ventilation determines the rate
k134_Rconv_ventilation = 1/(mass_dot_ventilation*cp_air);
% xxxx_Rradiator = radiator thermal transmittance in room xxxx
k134_number_of_radiators = 2;
k134_Rradiation_radiator_wall = 1/(htc_rad*A_rad*k134_number_of_radiators);
% xxxx_Rconv_radiator = convective heat tranfer between radiator surface temperature
and air
k134_Rprod = (60-20)/(Radiator_heat_effect*k134_number_of_radiators);
k134_Rconv_radiator = k134_Rprod;

%--Calculation of capacitances (heat stored in mass)--
% capasitance of the air in the room
k134_Ca = k134_room_volume*rho_air*cp_air;
% capasitance of the walls
k134_CwN = k134_wallN_volume*rho_external_wall*cp_external_wall;
k134_CwE = k134_wallE_volume*rho_internal_wall*cp_internal_wall;
k134_CwS = k134_wallS_volume*rho_external_wall*cp_external_wall;
k134_CwW = k134_wallW_volume*rho_internal_wall*cp_internal_wall;
% capasitance of the floor
k134_Cf = k134_floor_volume*rho_floor*cp_floor;
% capasitance of the ceiling
k134_Cc = k134_ceiling_volume*rho_ceiling*cp_ceiling;
%-----

%% -----classroom 135-----
%--Room dimensions--
k135_length = 4.650;
k135_width = 7.125;
k135_heigth = room_height;

k135_window_area = (2.29*0.71)+(0.8*1.4)+(1.32*1.4);
k135_wallN_area = (room_height*k135_length)-k135_window_area;
```

D.3 Room values

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 5 of 11

```
k135_wallE_area = room_height*k135_width;
k135_walls_area = room_height*k135_length;
k135_wallW_area = room_height*k135_width;
k135_floor_area = k135_length*k135_width;
k135_ceiling_area = k135_floor_area;

k135_room_volume = room_height*k135_floor_area;
k135_wallN_volume = k135_wallN_area*depth_external_wall;
k135_wallE_volume = k135_wallE_area*depth_internal_wall;
k135_walls_volume = k135_walls_area*depth_internal_wall;
k135_wallW_volume = k135_wallW_area*depth_internal_wall;
k135_floor_volume = k135_floor_area*depth_floor;
k135_ceiling_volume = k135_ceiling_area*depth_external_ceiling;

%--Calculation resistances (resistance the heat must go through to pass through a
material)--
% xxxx_Roy = outer thermal transmittance of external wall y in room xxxx
k135_RoN = 1/(U_external_wall*k135_wallN_area);
k135_RoS = 1/(U_external_wall*k135_walls_area);
% xxxx_Riy = inner thermal transmittance of external wall y in room xxxx
k135_RiN = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/
(2*lambda_wall_sheathing)))*(1/k135_wallN_area); %1/(inner_wall_U*k133_wallN_area);
k135_RiS = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/
(2*lambda_wall_sheathing)))*(1/k135_walls_area);
% xxxx_Rinternal_wallx = thermal transmittance of internal wall in room xxxx
k135_RinternalW = 1/(2*U_internal_wall*k135_wallW_area); %internal_wall_U is
multiplied by 2 since only half of the wall thickness is to be considered
k135_RinternalE = 1/(2*U_internal_wall*k135_wallW_area);
% xxxx_Rconv_wallx_air = convective resistance between wall n and the air in room
xxxx
k135_Rconv_wallN_air = 1/(htc_wallair*k135_wallN_area); %htc_wallair=convective heat
transfer factor between wall surface and air
k135_Rconv_wallE_air = 1/(htc_wallair*k135_wallE_area);
k135_Rconv_walls_air = 1/(htc_wallair*k135_walls_area);
k135_Rconv_wallW_air = 1/(htc_wallair*k135_wallW_area);
% xxxx_Rfloor = floor thermal transmittance in room xxxx
k135_Rfloor = 1/(U_floor*k135_floor_area);
% xxxx_Rceiling = ceiling thermal transmittance in room xxxx
k135_Roceiling = 1/(U_external_ceiling*k135_ceiling_area);
k135_Riceiling = ((depth_external_ceiling/(2*lambda_external_ceiling))+
(depth_ceiling_sheathing/(2*lambda_ceiling_sheathing)))*(1/k135_ceiling_area);
% xxxx_Rconv_ceiling_air = convective resistance between wall n and the air in room
xxxx
k135_Rconv_ceiling_air = 1/(htc_ceilingair*k135_ceiling_area); %
htc_ceilingair=convective heat transfer factor between ceiling surface and air
% xxxx_Rconv_window_air = window-air heat convection resistance
k135_Rconv_window_air = 1/(htc_windowair*k135_window_area);
% Radiation resistance wall-window
k135_Rradiation_window_wall = 1/(htc_windowwall*k135_window_area);
% xxxx_Rwindow = window thermal transmittance in room xxxx
k135_Rwindow = 1/(U_window*k135_window_area)-1/
(1/k135_Rconv_window_air+1/k135_Rradiation_window_wall);
% Calculation of infiltration (air leakage through the walls, ceiling and floor)
resistance
mass_dot_infiltration = (k135_room_volume*rho_air*ACH_infiltration)/hour; %air (kg/s)
```

D. M-FILES USED IN THE MODEL

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 6 of 11

```
into the room by infiltration
k135_Rconv_infiltration = 1/(mass_dot_infiltration*cp_air);
% Calculation of ventilation (air exchange in the room due to ventilation)
resistance. ACH_infiltration determines the rate
mass_dot_ventilation = (k135_room_volume*rho_air*ACH_ventilation)/hour; %air (kg/s)
into the room by infiltration. ACH_ventilation determines the rate
k135_Rconv_ventilation = 1/(mass_dot_ventilation*cp_air);
% xxxx_Rradiator = radiator thermal transmittance in room xxxx
k135_number_of_radiators = 2;
k135_Rradiation_radiator_wall = 1/(htc_rad*A_rad*k135_number_of_radiators);
% xxxx_Rconv_radiator = convective heat tranfer between radiator surface temperature
and air
k135_Rprod = (60-20)/(Radiator_heat_effect*k135_number_of_radiators);
k135_Rconv_radiator = k135_Rprod;

%--Calculation of capacitances (heat stored in mass)--
% capasitance of the air in the room
k135_Ca = k135_room_volume*rho_air*cp_air;
% capasitance of the walls
k135_CwN = k135_wallN_volume*rho_external_wall*cp_external_wall;
k135_CwE = k135_wallE_volume*rho_internal_wall*cp_internal_wall;
k135_CwS = k135_wallS_volume*rho_external_wall*cp_external_wall;
k135_CwW = k135_wallW_volume*rho_internal_wall*cp_internal_wall;
% capasitance of the floor
k135_Cf = k135_floor_volume*rho_floor*cp_floor;
% capasitance of the ceiling
k135_Cc = k135_ceiling_volume*rho_ceiling*cp_ceiling;

%% -----classroom 136-----
%--Room dimensions--
k136_length = 8.250;
k136_width = 7.125;
k136_heigth = room_height;

k136_window_area = (1.32*1.4)+(0.8*1.4)+(0.87*0.36)+(0.87*0.71)+(0.87*0.36)+(0.87*0.
71)+(0.87*0.36)+(0.87*0.36)+(0.8*1.4)+(1.32*1.4);
k136_wallN_area = (room_height*k136_length)-k136_window_area;
k136_wallE_area = room_height*k136_width;
k136_wallS_area = room_height*k136_length;
k136_wallW_area = room_height*k136_width;
k136_floor_area = k136_length*k136_width;
k136_ceiling_area = k136_floor_area;

k136_room_volume = room_height*k136_floor_area;
k136_wallN_volume = k136_wallN_area*depth_external_wall;
k136_wallE_volume = k136_wallE_area*depth_internal_wall;
k136_wallS_volume = k136_wallS_area*depth_internal_wall;
k136_wallW_volume = k136_wallW_area*depth_internal_wall;
k136_floor_volume = k136_floor_area*depth_floor;
k136_ceiling_volume = k136_ceiling_area*depth_external_ceiling;

%--Calculation resistances (resistance the heat must go through to pass through a
material)--
% xxxx_Roy = outer thermal transmittance of external wall y in room xxxx
k136_RoN = 1/(U_external_wall*k136_wallN_area);
```

D.3 Room values

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 7 of 11

```
k136_RoS = 1/(U_external_wall*k136_wallS_area);
% xxxx_Riy = inner thermal transmittance of external wall y in room xxxx
k136_RiN = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
(2*lambda_wall_sheathing)))*(1/k136_wallN_area); %1/(inner_wall_U*k133_wallN_area);
k136_RiS = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
(2*lambda_wall_sheathing)))*(1/k136_wallS_area);
% xxxx_Rinternal_wallx = thermal transmittance of internal wall in room xxxx
k136_RinternalW = 1/(2*U_internal_wall*k136_wallW_area); %internal_wall_U is
multiplied by 2 since only half of the wall thickness is to be considered
k136_RinternalE = 1/(2*U_internal_wall*k136_wallE_area);
% xxxx_Rconv_wallx_air = convective resistance between wall n and the air in room
xxxx
k136_Rconv_wallN_air = 1/(htc_wallair*k136_wallN_area); %htc_wallair=convective heat
transfer factor between wallsurface and air
k136_Rconv_wallE_air = 1/(htc_wallair*k136_wallE_area);
k136_Rconv_wallS_air = 1/(htc_wallair*k136_wallS_area);
k136_Rconv_wallW_air = 1/(htc_wallair*k136_wallW_area);
% xxxx_Rfloor = floor thermal transmittance in room xxxx
k136_Rfloor = 1/(U_floor*k136_floor_area);
% xxxx_Rceiling = ceiling thermal transmittance in room xxxx
k136_Ro ceiling = 1/(U_external_ceiling*k136_ceiling_area);
k136_Riceiling = ((depth_external_ceiling/(2*lambda_external_ceiling))+
(depth_ceiling_sheathing/(2*lambda_ceiling_sheathing)))*(1/k136_ceiling_area);
% xxxx_Rconv_ceiling_air = convective resistance between wall n and the air in room
xxxx
k136_Rconv_ceiling_air = 1/(htc_ceilingair*k136_ceiling_area); %
htc_ceilingair=convective heat transfer factor between ceiling surface and air
% xxxx_Rconv_window_air = window-air heat convection resistance
k136_Rconv_window_air = 1/(htc_windowair*k136_window_area);
% Radiation resistance wall-window
k136_Rradiation_window_wall = 1/(htc_windowwall*k136_window_area);
% xxxx_Rwindow = window thermal transmittance in room xxxx
k136_Rwindow = 1/(U_window*k136_window_area)-1/
(1/k136_Rconv_window_air+1/k136_Rradiation_window_wall);
% Calculation of infiltration (air leakage through the walls, ceiling and floor)
resistance
mass_dot_infiltration = (k136_room_volume*rho_air*ACH_infiltration)/hour; %air (kg/s)
into the room by infiltration
k136_Rconv_infiltration = 1/(mass_dot_infiltration*cp_air);
% Calculation of ventilation (air exchange in the room due to ventilation)
resistance. ACH_infiltration determines the rate
mass_dot_ventilation = (k136_room_volume*rho_air*ACH_ventilation)/hour; %air (kg/s)
into the room by infiltration. ACH_ventilation determines the rate
k136_Rconv_ventilation = 1/(mass_dot_ventilation*cp_air);
% xxxx_Rradiator = radiator thermal transmittance in room xxxx
k136_number_of_radiators = 4;
k136_Rradiation_radiator_wall = 1/(htc_rad*A_rad*k136_number_of_radiators);
% xxxx_Rconv_radiator = convective heat transfer between radiator surface temperature
and air
k136_Rprod = (60-20)/(Radiator_heat_effect*k136_number_of_radiators);
k136_Rconv_radiator = k136_Rprod;

%--Calculation of capacitances (heat stored in mass)--
% capasitance of the air in the room
k136_Ca = k136_room_volume*rho_air*cp_air;
```

D. M-FILES USED IN THE MODEL

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 8 of 11

```
% capacitance of the walls
k136_CwN = k136_wallN_volume*rho_external_wall*cp_external_wall;
k136_CwE = k136_wallE_volume*rho_internal_wall*cp_internal_wall;
k136_CwS = k136_wallS_volume*rho_external_wall*cp_external_wall;
k136_CwW = k136_wallW_volume*rho_internal_wall*cp_internal_wall;
% capacitance of the floor
k136_Cf = k136_floor_volume*rho_floor*cp_floor;
% capacitance of the ceiling
k136_Cc = k136_ceiling_volume*rho_ceiling*cp_ceiling;
%-----

%% -----classroom 143-----
%--Room dimensions--
k143_lengthN = 9.748;
k143_lengthS = 8.572;
k143_width = 7.125;
k143_heigth = room_height;

k143_window_area = (1.32*1.4)+(0.8*1.4)+(0.8*1.4)+(2.09*1.4)+(2.09*1.4)+(0.8*1.4);
k143_wallN_area = (room_height*k143_lengthN)-k143_window_area;
k143_wallE_area = room_height*k143_width;
k143_wallS_area = room_height*k143_lengthS;
k143_wallW_area = room_height*k143_width;
k143_floor_area = 64.8;
k143_ceiling_area = k143_floor_area;

k143_room_volume = room_height*k143_floor_area;
k143_wallN_volume = k143_wallN_area*depth_external_wall;
k143_wallE_volume = k143_wallE_area*depth_internal_wall;
k143_wallS_volume = k143_wallS_area*depth_internal_wall;
k143_wallW_volume = k143_wallW_area*depth_internal_wall;
k143_floor_volume = k143_floor_area*depth_floor;
k143_ceiling_volume = k143_ceiling_area*depth_external_ceiling;

%--Calculation resistances (resistance the heat must go through to pass through a
material)--
% xxxx_Roy = outer thermal transmittance of external wall y in room xxxx
k143_RoN = 1/(U_external_wall*k143_wallN_area);
k143_RoS = 1/(U_external_wall*k143_wallS_area);
% xxxx_Riy = inner thermal transmittance of external wall y in room xxxx
k143_RiN = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/
(2*lambda_wall_sheathing)))*(1/k143_wallN_area); %1/(inner_wall_U*k133_wallN_area);
k143_RiS = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/
(2*lambda_wall_sheathing)))*(1/k143_wallS_area);
% xxxx_Rinternal_wallx = thermal transmittance of internal wall in room xxxx
k143_RinternalW = 1/(2*U_internal_wall*k143_wallW_area); %internal_wall_U is
multiplied by 2 since only half of the wall thickness is to be considered
k143_RinternalE = 1/(2*U_internal_wall*k143_wallE_area);
% xxxx_Rconv_wallx_air = convective resistance between wall n and the air in room
xxxx
k143_Rconv_wallN_air = 1/(htc_wallair*k143_wallN_area); %htc_wallair=convective heat
transfer factor between wallsurface and air
k143_Rconv_wallE_air = 1/(htc_wallair*k143_wallE_area);
k143_Rconv_wallS_air = 1/(htc_wallair*k143_wallS_area);
```

D.3 Room values

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 9 of 11

```
k143_Rconv_wallW_air = 1/(htc_wallair*k143_wallW_area);
% xxxx_Rfloor = floor thermal transmittance in room xxxx
k143_Rfloor = 1/(U_floor*k143_floor_area);
% xxxx_Rceiling = ceiling thermal transmittance in room xxxx
k143_Rceiling = 1/(U_external_ceiling*k143_ceiling_area);
k143_Riceiling = ((depth_external_ceiling/(2*lambda_external_ceiling))+
(depth_ceiling_sheathing/(2*lambda_ceiling_sheathing)))*(1/k143_ceiling_area);
% xxxx_Rconv_ceiling_air = convective resistance between wall n and the air in room
xxxx
k143_Rconv_ceiling_air = 1/(htc_ceilingair*k143_ceiling_area); %
htc_ceilingair=convective heat tranfer factor between ceiling surface and air
% xxxx_Rconv_window_air = window-air heat convection resistance
k143_Rconv_window_air = 1/(htc_windowair*k143_window_area);
% Radiation resistance wall-window
k143_Rradiation_window_wall = 1/(htc_windowwall*k143_window_area);
% xxxx_Rwindow = window thermal transmittance in room xxxx
k143_Rwindow = 1/(U_window*k143_window_area)-1/
(1/k143_Rconv_window_air+1/k143_Rradiation_window_wall);
% Calculation of infiltration (air leakage through the walls, ceiling and floor)
resistance
mass_dot_infiltration = (k143_room_volume*rho_air*ACH_infiltration)/hour; %air (kg/s)
into the room by infiltration
k143_Rconv_infiltration = 1/(mass_dot_infiltration*cp_air);
% Calculation of ventilation (air exchange in the room due to ventilation)
resistance. ACH_infiltration determines the rate
mass_dot_ventilation = (k143_room_volume*rho_air*ACH_ventilation)/hour; %air (kg/s)
into the room by infiltration. ACH_ventilation determines the rate
k143_Rconv_ventilation = 1/(mass_dot_ventilation*cp_air);
% xxxx_Rradiator = radiator thermal transmittance in room xxxx
k143_number_of_radiators = 3;
k143_Rradiation_radiator_wall = 1/(htc_rad*A_rad*k143_number_of_radiators);
% xxxx_Rconv_radiator = convective heat tranfer between radiator surface temperature
and air
k143_Rprod = (60-20)/(Radiator_heat_effect*k143_number_of_radiators);
k143_Rconv_radiator = k143_Rprod;

%--Calculation of capacitances (heat stored in mass)--
% capasitance of the air in the room
k143_Ca = k143_room_volume*rho_air*cp_air;
% capasitance of the walls
k143_CwN = k143_wallN_volume*rho_external_wall*cp_external_wall;
k143_CwE = k143_wallE_volume*rho_internal_wall*cp_internal_wall;
k143_CwS = k143_wallS_volume*rho_external_wall*cp_external_wall;
k143_CwW = k143_wallW_volume*rho_internal_wall*cp_internal_wall;
% capasitance of the floor
k143_Cf = k143_floor_volume*rho_floor*cp_floor;
% capasitance of the ceiling
k143_Cc = k143_ceiling_volume*rho_ceiling*cp_ceiling;
%-----

%% -----classroom 164-----
%--Room dimensions--
k164_length = 4.750;
k164_width = 7.125;
```

D. M-FILES USED IN THE MODEL

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 10 of 11

```
k164_heigth = room_height;

k164_window_area = (0.8*1.4)+(1.32*1.4)+(1.32*1.4)+(0.8*1.4);
k164_wallN_area = (room_height*k164_length)-k164_window_area;
k164_wallE_area = room_height*k164_width;
k164_wallS_area = room_height*k164_length;
k164_wallW_area = room_height*k164_width;
k164_floor_area = k164_length*k164_width;
k164_ceiling_area = k164_floor_area;

k164_room_volume = room_height*k164_floor_area;
k164_wallN_volume = k164_wallN_area*depth_external_wall;
k164_wallE_volume = k164_wallE_area*depth_internal_wall;
k164_wallS_volume = k164_wallS_area*depth_internal_wall;
k164_wallW_volume = k164_wallW_area*depth_internal_wall;
k164_floor_volume = k164_floor_area*depth_floor;
k164_ceiling_volume = k164_ceiling_area*depth_external_ceiling;

%--Calculation resistances (resistance the heat must go through to pass through a
material)--
% xxxx_Roy = outer thermal transmittance of external wall y in room xxxx
k164_RoN = 1/(U_external_wall*k164_wallN_area);
k164_RoE = 1/(U_external_wall*k164_wallE_area);
k164_RoS = 1/(U_external_wall*k164_wallS_area);
% xxxx_Riy = inner thermal transmittance of external wall y in room xxxx
k164_RiN = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
2*lambda_wall_sheathing)))*(1/k164_wallN_area); %1/(inner_wall_U*k133_wallN_area);
k164_RiE = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
2*lambda_wall_sheathing)))*(1/k164_wallE_area);
k164_RiS = ((depth_external_wall/(2*lambda_external_wall))+(depth_wall_sheathing/(
2*lambda_wall_sheathing)))*(1/k164_wallS_area);
% xxxx_Rinternal_wallx = thermal transmittance of internal wall in room xxxx
k164_RinternalW = 1/(2*U_internal_wall*k164_wallW_area); %internal_wall_U is
multiplied by 2 since only half of the wall thickness is to be considered
% xxxx_Rconv_wallx_air = convective resistance between wall n and the air in room
xxxx
k164_Rconv_wallN_air = 1/(htc_wallair*k164_wallN_area); %htc_wallair=convective heat
tranfer factor between wallsurface and air
k164_Rconv_wallE_air = 1/(htc_wallair*k164_wallE_area);
k164_Rconv_wallS_air = 1/(htc_wallair*k164_wallS_area);
k164_Rconv_wallW_air = 1/(htc_wallair*k164_wallW_area);
% xxxx_Rfloor = floor thermal transmittance in room xxxx
k164_Rfloor = 1/(U_floor*k164_floor_area);
% xxxx_Rceiling = ceiling thermal transmittance in room xxxx
k164_Roceiling = 1/(U_external_ceiling*k164_ceiling_area);
k164_Riceiling = ((depth_external_ceiling/(2*lambda_external_ceiling))+
(depth_ceiling_sheathing/(2*lambda_ceiling_sheathing)))*(1/k164_ceiling_area);
% xxxx_Rconv_ceiling_air = convective resistance between wall n and the air in room
xxxx
k164_Rconv_ceiling_air = 1/(htc_ceilingair*k164_ceiling_area); %
htc_ceilingair=convective heat tranfer factor between ceiling surface and air
% xxxx_Rconv_window_air = window-air heat convection resistance
k164_Rconv_window_air = 1/(htc_windowair*k164_window_area);
% Radiation resistance wall-window
k164_Rradiation_window_wall = 1/(htc_windowwall*k164_window_area);
```


D.3 Room values

07.06.11 15:32 C:\Users\alvsvag\Documents\My Dropbox\mast...\room_values.m 11 of 11

```
% xxxx_Rwindow = window thermal transmittance in room xxxx
k164_Rwindow = 1/(U_window*k164_window_area)-1/
(1/k164_Rconv_window_air+1/k164_Rradiation_window_wall);
% Calculation of infiltration (air leakage through the walls, ceiling and floor)
resistance
mass_dot_infiltration = (k164_room_volume*rho_air*ACH_infiltration)/hour; %air (kg/s)
into the room by infiltration
k164_Rconv_infiltration = 1/(mass_dot_infiltration*cp_air);
% Calculation of ventilation (air exchange in the room due to ventilation)
resistance. ACH_infiltration determines the rate
mass_dot_ventilation = (k164_room_volume*rho_air*ACH_ventilation)/hour; %air (kg/s)
into the room by infiltration. ACH_ventilation determines the rate
k164_Rconv_ventilation = 1/(mass_dot_ventilation*cp_air);
% xxxx_Rradiator = radiator thermal transmittance in room xxxx
k164_number_of_radiators = 2;
k164_Rradiation_radiator_wall = 1/(htc_rad*A_rad*k164_number_of_radiators);
% xxxx_Rconv_radiator = convective heat tranfer between radiator surface temperature
and air
k164_Rprod = (60-20)/(Radiator_heat_effect*k164_number_of_radiators);
k164_Rconv_radiator = k164_Rprod;

%--Calculation of capacitances (heat stored in mass)--
% capasitance of the air in the room
k164_Ca = k164_room_volume*rho_air*cp_air;
% capasitance of the walls
k164_CwN = k164_wallN_volume*rho_external_wall*cp_external_wall;
k164_CwE = k164_wallE_volume*rho_external_wall*cp_external_wall;
k164_CwS = k164_wallS_volume*rho_external_wall*cp_external_wall;
k164_CwW = k164_wallW_volume*rho_internal_wall*cp_internal_wall;
% capasitance of the floor
k164_Cf = k164_floor_volume*rho_floor*cp_floor;
% capasitance of the ceiling
k164_Cc = k164_ceiling_volume*rho_ceiling*cp_ceiling;
%-----
```

D. M-FILES USED IN THE MODEL

D.4 Input parameters

04.12.10 19:57 C:\Documents and Settings\Øyvind\Mine d...\input_parameters.m 1 of 1

```
%% -----Input parameters-----  
% -----  
  
%-----Shunt-----  
% theoretical regulation ratio - regulation port (Pa)  
r_rp = 50;  
% theoretical regulation ratio - shunt port (Pa)  
r_sp = 50;  
% differential pressure - regulation port (Pa)  
dp_rp = 1736;  
% differential pressure - shunt port (Pa)  
dp_sp = 1736;  
% differential pressure - pipe to shunt (Pa)  
dp_r = 300;  
% differential pressure - shunt pipe (Pa)  
dp_s = 150;  
  
%-----Heating coil-----  
% weight exclusive water content (kg)  
m_hc = 28;  
% water content (m^3)  
w_hc = 0.002;  
% water flow through the heating coil (m^3/s)  
qw_hc = 0.000205;  
% air flow through the heating coil (m^3/s)  
qa_hc = 0.857;  
% water temperature into the heating coil (celcius)  
twi_hc = 80;  
% air temperature into the heating coil (celcius)  
tai_hc = 5;  
% air temperature from the heating coil (celcius)  
tao_hc = 18;  
% heating coil material (1=Alu/Cu, 2=Cu/Cu)  
material_hc = 1;  
  
%-----Heat recovery unit-----  
% weight heat recovery unit  
m_hru = 28;  
%air speed through the heat recovery unit  
va_hru = 0.857;  
%specific heat capacity heat recovery unit  
cp_hru = 900; %aluminium  
% heat recovery unit efficiency  
eta = 0.75;
```

D.5 Preprocessing

13.12.10 19:22 C:\Users\alvsvag\Documents\My Dropbox\pros...\preprocessing.m 1 of 1

```

%% -----Preprocessing-----
% -----

%-----Heating coil-----
% water temperature out from the heating coil (celcius)
two_hc = twi_hc-((rho_air*qa_hc*cp_air)/(rho_water*cp_water*qw_hc))*(tao_hc-tai_hc);
%relative cooling of water
epsilon_water = (twi_hc-two_hc)/(twi_hc-tai_hc);
%relative heating of air
epsilon_air = (tao_hc-tai_hc)/(twi_hc-tai_hc);
%specific heat capacity heating coil
if(material_hc==1)
    cp_hc = 0.67*cp_aluminum+0.33*cp_copper;
elseif(material_hc==2)
    cp_hc = cp_copper;
end
%time constant in heating coil (s)
T_hc = (m_hc*cp_hc+rho_water*cp_water*v_water)/(rho_water*cp_water*qw_hc)*(1-epsilon_water);
%gain water to water
K_ww = 1-epsilon_water;
%gain water to air
K_wa = epsilon_air;
%gain air to water
K_aw = epsilon_water;
%gain air to air
K_aa = 1-epsilon_air;
%time constant in heating coil air side
T_aa = (1-epsilon_air-epsilon_water)/(1-epsilon_water)*T_hc;

%-----Heat recovery unit-----
%time constant in heat recovery unit (s)
Thru = (m_hru*cp_hru)/(2*va_hru*cp_air);
%gain fresh air to heated fresh air
K_12 = 1-eta;
%gain used air to heated fresh air
K_32 = eta;

```

D.6 Kalman filter - embedded Matlab function

17.06.11 13:51 Block: active_building_new_model_mpc_one_roo.../Kalman filter 1 of 1

```
function [yhatOut, xhatOut] = KALMAN(meas, u, meas_dist, A, B1, B2, C, timer)
% This Embedded MATLAB Function implements a simple Kalman filter
% for estimating both the state and output of a linear, discrete-time,
% time-invariant, system given by the following state-space equations:
%
%  $x(k) = A x(k-1) + B1 u_1(k) + B2 d(k) + w(k)$ 
%  $y(k) = C x(k-1) + v(k)$ 
%
% Define storage for the variables that need to persist
% between time periods.
persistent xhat P Q_kalman R_kalman %A B1 B2 C
if isempty(P)
% First time through the code so do some initialization
xhat = [180 150 0 0 0 -5 350]';
P = 1*eye(7);
Q = [100 100 100 100 100 100 100]; %how much is the model trusted? (low values
indicate that the model is believed to be approximately correct)
Q_kalman = diag(Q);
R_kalman = 5; %how much is the measured variable trusted? (low values indicate
that the measurement is believed to be approximately correct)
end

if timer == 0
xhatOut = xhat;
yhatOut = C*xhatOut;
else
% Propagate the state estimate and covariance matrix:
xhat = A*xhat + B1*u + B2*meas_dist;
P = A*P*A' + Q_kalman;
% Calculate the Kalman gain
K = P*C'/(C*P*C' + R_kalman);
% Calculate the measurement residual
resid = meas - C*xhat;
% Update the state and error covariance estimate
xhat = xhat + K*resid;
P = (eye(size(K,1))-K*C)*P*(eye(size(K,1))-K*C)' + K*R_kalman*K';
% Post the results
xhatOut = xhat;
yhatOut = C*xhatOut;
end
```

D.7 MPC preparations

17.06.11 13:58 C:\Users\alvsvag\Documents\My Dropbox\m...\mpc_preparations.m 1 of 6

```
%% ----- Transfer functions from the identification procedure -----  
%TF values for identified model  
Kp1k133 = 0.73171;  
Kp2k133 = 0.134;  
Kp3k133 = 0.16171;  
Tp1k133 = 1013.5;  
Tp2k133 = 3658;  
Tp31k133 = 14872;  
Tp32k133 = 795.83;  
Td1k133 = 0;  
Td2k133 = 0;  
Td3k133 = 13.396;  
Kp1k134 = 0.65748;  
Kp2k134 = 0.13735;  
Kp3k134 = 0.18424;  
Tp1k134 = 994.89;  
Tp2k134 = 3851.95;  
Tp31k134 = 16077;  
Tp32k134 = 999.19;  
Td1k134 = 0;  
Td2k134 = 0;  
Td3k134 = 13.396;  
Kp1k135 = 0.6535;  
Kp2k135 = 0.13713;  
Kp3k135 = 0.1843;  
Tp1k135 = 985.05;  
Tp2k135 = 3834.2;  
Tp31k135 = 16084;  
Tp32k135 = 999.06;  
Td1k135 = 0;  
Td2k135 = 0;  
Td3k135 = 13.396;  
Kp1k136 = 0.72889;  
Kp2k136 = 0.12995;  
Kp3k136 = 0.178;  
Tp1k136 = 1003.8;  
Tp2k136 = 3415.4;  
Tp31k136 = 15509;  
Tp32k136 = 939.29;  
Td1k136 = 0;  
Td2k136 = 0;  
Td3k136 = 13.396;  
Kp1k143 = 0.69344;  
Kp2k143 = 0.1737;  
Kp3k143 = 0.21437;  
Tp1k143 = 1362;  
Tp2k143 = 3119.9;  
Tp31k143 = 15543;  
Tp32k143 = 1090.2;  
Td1k143 = 0;  
Td2k143 = 0;  
Td3k143 = 13.396;  
Kp1k164 = 0.67087;  
Kp2k164 = 0.16706;  
Kp3k164 = 0.1817;
```

D. M-FILES USED IN THE MODEL

17.06.11 13:58 C:\Users\alvsvag\Documents\My Dropbox\m...\mpc_preparations.m 2 of 6

```
Tp1k164 = 1020.3;
Tp2k164 = 3517.6;
Tp3k164 = 15645;
Tp32k164 = 957.58;
Td1k164 = 0;
Td2k164 = 0;
Td3k164 = 13.396;

%TF
s = tf('s');
%k133
k133_radiator = Kp1k133/(Tp1k133*s+1);
k133_outdoor_temp = Kp2k133/(Tp2k133*s+1);
k133_ventilation = Kp3k133*exp(-Td3k133*s)/((Tp31k133*Tp32k133)*s^2+
(Tp31k133+Tp32k133)*s+1);
k133_ventilation_no_td = Kp3k133/((Tp31k133*Tp32k133)*s^2+(Tp31k133+Tp32k133)*s+1);
%k134
k134_radiator = Kp1k134/(Tp1k134*s+1);
k134_outdoor_temp = Kp2k134/(Tp2k134*s+1);
k134_ventilation = Kp3k134*exp(-Td3k134*s)/((Tp31k134*Tp32k134)*s^2+
(Tp31k134+Tp32k134)*s+1);
%k135
k135_radiator = Kp1k135/(Tp1k135*s+1);
k135_outdoor_temp = Kp2k135/(Tp2k135*s+1);
k135_ventilation = Kp3k135*exp(-Td3k135*s)/((Tp31k135*Tp32k135)*s^2+
(Tp31k135+Tp32k135)*s+1);
%k136
k136_radiator = Kp1k136/(Tp1k136*s+1);
k136_outdoor_temp = Kp2k136/(Tp2k136*s+1);
k136_ventilation = Kp3k136*exp(-Td3k136*s)/((Tp31k136*Tp32k136)*s^2+
(Tp31k136+Tp32k136)*s+1);
%k143
k143_radiator = Kp1k143/(Tp1k143*s+1);
k143_outdoor_temp = Kp2k143/(Tp2k143*s+1);
k143_ventilation = Kp3k143*exp(-Td3k143*s)/((Tp31k143*Tp32k143)*s^2+
(Tp31k143+Tp32k143)*s+1);
%k164
k164_radiator = Kp1k164/(Tp1k164*s+1);
k164_outdoor_temp = Kp2k164/(Tp2k164*s+1);
k164_ventilation = Kp3k164*exp(-Td3k164*s)/((Tp31k164*Tp32k164)*s^2+
(Tp31k164+Tp32k164)*s+1);

%% ----- padè approximation of the TF with time dealys above -----
% %first order padè approx
% P1_k133_ventilation = pade(k133_ventilation,1);
% P1_k134_ventilation = pade(k134_ventilation,1);
% P1_k135_ventilation = pade(k135_ventilation,1);
% P1_k136_ventilation = pade(k136_ventilation,1);
% P1_k143_ventilation = pade(k143_ventilation,1);
% P1_k164_ventilation = pade(k164_ventilation,1);

%third order padè approx
P3_k133_ventilation = pade(k133_ventilation,3);
P3_k134_ventilation = pade(k134_ventilation,3);
P3_k135_ventilation = pade(k135_ventilation,3);
```

D.7 MPC preparations

17.06.11 13:58 C:\Users\alvsvag\Documents\My Dropbox\m...\mpc_preparations.m 3 of 6

```
P3_k136_ventilation = pade(k136_ventilation,3);
P3_k143_ventilation = pade(k143_ventilation,3);
P3_k164_ventilation = pade(k164_ventilation,3);

% % compare the freq. resp. of the original and the approx. models using bode plot
% %k133
% figure('Name','freq. resp. P3_k133_ventilation','NumberTitle','off')
% h = bodeoptions;
% h.PhaseMatching = 'on';
% bodeplot(k133_ventilation, '-b', P3_k133_ventilation, '-.r', P1_k133_ventilation, ':k',
{0.1 10},h)
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'SouthWest')
% %k134
% figure('Name','freq. resp. P3_k134_ventilation','NumberTitle','off')
% h = bodeoptions;
% h.PhaseMatching = 'on';
% bodeplot(k134_ventilation, '-b', P3_k134_ventilation, '-.r', P1_k134_ventilation, ':k',
{0.1 10},h)
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'SouthWest')
% %k135
% figure('Name','freq. resp. P3_k135_ventilation','NumberTitle','off')
% h = bodeoptions;
% h.PhaseMatching = 'on';
% bodeplot(k135_ventilation, '-b', P3_k135_ventilation, '-.r', P1_k135_ventilation, ':k',
{0.1 10},h)
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'SouthWest')
% %k136
% figure('Name','freq. resp. P3_k136_ventilation','NumberTitle','off')
% h = bodeoptions;
% h.PhaseMatching = 'on';
% bodeplot(k136_ventilation, '-b', P3_k136_ventilation, '-.r', P1_k136_ventilation, ':k',
{0.1 10},h)
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'SouthWest')
% %k143
% figure('Name','freq. resp. P3_k143_ventilation','NumberTitle','off')
% h = bodeoptions;
% h.PhaseMatching = 'on';
% bodeplot(k143_ventilation, '-b', P3_k143_ventilation, '-.r', P1_k143_ventilation, ':k',
{0.1 10},h)
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'SouthWest')
% %k164
% figure('Name','freq. resp. P3_k164_ventilation','NumberTitle','off')
% h = bodeoptions;
% h.PhaseMatching = 'on';
% bodeplot(k164_ventilation, '-b', P3_k164_ventilation, '-.r', P1_k164_ventilation, ':k',
{0.1 10},h)
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'SouthWest')
%
% % compare the time domain responses of the original and approx. models using step
% %k133
% figure('Name','time domain resp. k133_ventilation','NumberTitle','off')
% step(k133_ventilation, '-b', P3_k133_ventilation, '-.r', P1_k133_ventilation, ':k')
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'Southeast')
% %k134
% figure('Name','time domain resp. k134_ventilation','NumberTitle','off')
```

D. M-FILES USED IN THE MODEL

17.06.11 13:58 C:\Users\alvsvag\Documents\My Dropbox\m...\mpc_preparations.m 4 of 6

```
% step(k134_ventilation, '-b', P3_k134_ventilation, '-.r', P1_k134_ventilation, ':k')
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'Southeast')
% %k135
% figure('Name', 'time domain resp. k135_ventilation', 'NumberTitle', 'off')
% step(k135_ventilation, '-b', P3_k135_ventilation, '-.r', P1_k135_ventilation, ':k')
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'Southeast')
% %k136
% figure('Name', 'time domain resp. k136_ventilation', 'NumberTitle', 'off')
% step(k136_ventilation, '-b', P3_k136_ventilation, '-.r', P1_k136_ventilation, ':k')
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'Southeast')
% %k143
% figure('Name', 'time domain resp. k143_ventilation', 'NumberTitle', 'off')
% step(k143_ventilation, '-b', P3_k143_ventilation, '-.r', P1_k143_ventilation, ':k')
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'Southeast')
% %k164
% figure('Name', 'time domain resp. k164_ventilation', 'NumberTitle', 'off')
% step(k164_ventilation, '-b', P3_k164_ventilation, '-.r', P1_k164_ventilation, ':k')
% legend('Exact delay', 'Third-Order Pade', 'First-Order Pade', 'Location', 'Southeast')

%% ----- Transform the model into state space form -----
% % transform each TF into SS
% ss_k133_radiator = ss(k133_radiator);
% ss_k133_outdoor_temp = ss(k133_outdoor_temp);
% ss_k133_ventilation = ss(P3_k133_ventilation);
% ss_k134_radiator = ss(k134_radiator);
% ss_k134_outdoor_temp = ss(k134_outdoor_temp);
% ss_k134_ventilation = ss(P3_k134_ventilation);
% ss_k135_radiator = ss(k135_radiator);
% ss_k135_outdoor_temp = ss(k135_outdoor_temp);
% ss_k135_ventilation = ss(P3_k135_ventilation);
% ss_k136_radiator = ss(k136_radiator);
% ss_k136_outdoor_temp = ss(k136_outdoor_temp);
% ss_k136_ventilation = ss(P3_k136_ventilation);
% ss_k143_radiator = ss(k143_radiator);
% ss_k143_outdoor_temp = ss(k143_outdoor_temp);
% ss_k143_ventilation = ss(P3_k143_ventilation);
% ss_k164_radiator = ss(k164_radiator);
% ss_k164_outdoor_temp = ss(k164_outdoor_temp);
% ss_k164_ventilation = ss(P3_k164_ventilation);

% transform all TF of each room into 1 state space representation
ss_k133 = ss([k133_radiator, k133_outdoor_temp, P3_k133_ventilation]);
ss_k134 = ss([k134_radiator, k134_outdoor_temp, P3_k134_ventilation]);
ss_k135 = ss([k135_radiator, k135_outdoor_temp, P3_k135_ventilation]);
ss_k136 = ss([k136_radiator, k136_outdoor_temp, P3_k136_ventilation]);
ss_k143 = ss([k143_radiator, k143_outdoor_temp, P3_k143_ventilation]);
ss_k164 = ss([k164_radiator, k164_outdoor_temp, P3_k164_ventilation]);

%% ----- Discretized system -----
%time step
Ts = 50; %chosen so that it captures the dynamics of the system

%discrete system to be used with the mpc and the kalman filter
discrete_ss_k133 = c2d(ss_k133, Ts);
discrete_ss_k134 = c2d(ss_k134, Ts);
```


D.7 MPC preparations

17.06.11 13:58 C:\Users\alvsvag\Documents\My Dropbox\m...\mpc_preparations.m 5 of 6

```
discrete_ss_k135 = c2d(ss_k135,Ts);
discrete_ss_k136 = c2d(ss_k136,Ts);
discrete_ss_k143 = c2d(ss_k143,Ts);
discrete_ss_k164 = c2d(ss_k164,Ts);

%% ----- Check observability -----
[Abar1,Bbar1,Cbar1,T1,k1] = obsvf(discrete_ss_k133.a,discrete_ss_k133.b,
discrete_ss_k133.c);
obs_k133 = length(discrete_ss_k133.a)-sum(k1) %each state in the system ss_k133 is
observable if this expression is equal to 0
[Abar2,Bbar2,Cbar2,T2,k2] = obsvf(discrete_ss_k134.a,discrete_ss_k134.b,
discrete_ss_k134.c);
obs_k134 = length(discrete_ss_k134.a)-sum(k2) %each state in the system ss_k134 is
observable if this expression is equal to 0
[Abar3,Bbar3,Cbar3,T3,k3] = obsvf(discrete_ss_k135.a,discrete_ss_k135.b,
discrete_ss_k135.c);
obs_k135 = length(discrete_ss_k135.a)-sum(k3) %each state in the system ss_k135 is
observable if this expression is equal to 0
[Abar4,Bbar4,Cbar4,T4,k4] = obsvf(discrete_ss_k136.a,discrete_ss_k136.b,
discrete_ss_k136.c);
obs_k136 = length(discrete_ss_k136.a)-sum(k4) %each state in the system ss_k136 is
observable if this expression is equal to 0
[Abar5,Bbar5,Cbar5,T5,k5] = obsvf(discrete_ss_k143.a,discrete_ss_k143.b,
discrete_ss_k143.c);
obs_k143 = length(discrete_ss_k143.a)-sum(k5) %each state in the system ss_k143 is
observable if this expression is equal to 0
[Abar6,Bbar6,Cbar6,T6,k6] = obsvf(discrete_ss_k164.a,discrete_ss_k164.b,
discrete_ss_k164.c);
obs_k164 = length(discrete_ss_k164.a)-sum(k6) %each state in the system ss_k164 is
observable if this expression is equal to 0

%% ----- Check controllability-----
[Abar7,Bbar7,Cbar7,T7,k7] = ctrbf(discrete_ss_k133.a,discrete_ss_k133.b,
discrete_ss_k133.c);
ctrb_k133 = length(discrete_ss_k133.a)-sum(k7); %the system ss_k133 is controllable
if this expression is equal to 0
[Abar8,Bbar8,Cbar8,T8,k8] = ctrbf(discrete_ss_k134.a,discrete_ss_k134.b,
discrete_ss_k134.c);
ctrb_k134 = length(discrete_ss_k134.a)-sum(k8); %the system ss_k134 is controllable
if this expression is equal to 0
[Abar9,Bbar9,Cbar9,T9,k9] = ctrbf(discrete_ss_k164.a,discrete_ss_k164.b,
discrete_ss_k164.c);
ctrb_k164 = length(discrete_ss_k164.a)-sum(k9); %the system ss_k164 is controllable
if this expression is equal to 0

%% ----- Luenberger observer -----
% A-1C, design of the l-matrix
% p = -5*min(abs(eig(ss_k133b.a)))*[0 .15 .3 .45 .6 .75 .9]
% l_k133 = place(discrete_ss_k133.a',discrete_ss_k133.c',p)'
% l_k134 = place(discrete_ss_k134.a',discrete_ss_k134.c',p)'
% l_k135 = place(discrete_ss_k135.a',discrete_ss_k135.c',p)'
% l_k136 = place(discrete_ss_k136.a',discrete_ss_k136.c',p)'
% l_k143 = place(discrete_ss_k143.a',discrete_ss_k143.c',p)'
% l_k164 = place(discrete_ss_k164.a',discrete_ss_k164.c',p)'
```

D. M-FILES USED IN THE MODEL

17.06.11 13:58 C:\Users\alvsvag\Documents\My Dropbox\m...\mpc_preparations.m 6 of 6

```
%% -----Kalman filter -----  
% see embedded matlab function named kalman filter
```

D.8 Creation of MPC matrices

17.06.11 15:54 C:\Users\alvsvag\Documents\My Dropbox\master_2011\cd\mpc.m 1 of 2

```

%% ----- MPC -----

%% prediction horizon and intitial values
H_p = 35; %remember to also change H_p in feedforward.m and K_MPC (embedded matlab
function)
H_u = H_p; %remember to also change H_u in K_MPC (embedded matlab function)
H_w = 1;

%% system
n = 7; %number of states
l = 1; %number of inputs
m = 1; %number of controlled (measured) outputs. If this parameter changes remember
to also change it in K_MPC (embedded matlab function)
d = 2; %number of measured disturbance inputs

% Epsilon(k) = T(k) - Psi*x(k) - Upsilon*u(k-1) - Xi*Dm(k) (see simulink block
labeled mpc)
T1 = ones(m*(H_p-H_w+1), 1)*21; %reference value
T2 = ones(m*(H_p-H_w+1), 1)*18; %reference value

%% -----room:k133-----
A_k133 = discrete_ss_k133.a; %dimension n*n
B1_k133 = discrete_ss_k133.b(:,1); %dimension n*1
B2_k133 = discrete_ss_k133.b(:,[2 3]); %dimension n*d
C_k133 = discrete_ss_k133.c; %dimension m*n
C_blockdiag_k133 = blkdiag2(C_k133,H_p); %see equation 2.70 in Maciejowski

%% Define matrices required for the prediction vector equation
% Z(k) = Psi*x(k) + Upsilon*u(k-1) + Theta*DeltaU(k) + Xi*Dm(k)
Psi_k133 = zeros(m*(H_p-H_w+1), n);
Upsilon_k133 = zeros(m*(H_p-H_w+1), 1);
Theta_k133 = zeros(m*(H_p-H_w+1), 1*H_u);
Xi_k133 = zeros(m*(H_p-H_w+1), m*(H_p-H_w+1));

%% Creation of Psi, Upsilon, Theta and Xi matrices for z(k+1),z(k+2),...,z(k+Hp):
%Psi
for q = 1:m*(H_p-H_w+1)
    i = q*n-(n-1);
    Psi_k133 = Psi_k133 + C_blockdiag_k133(:, i:i+n-1)*A_k133^q; %tested OK
end

%Upsilon
Upsilon_temp_k133 = zeros(H_p*n,m);
for p = 0:m*(H_p-H_w+1)-1
    i=p*n+1;
    if p==0
        Upsilon_temp_k133(i:i+n-1,:) = (A_k133^p)*B1_k133;
    else
        Upsilon_temp_k133(i:i+n-1,:) = Upsilon_temp_k133(i-n:i-1)+(A_k133^p)*B1_k133;
    end
end
Upsilon_k133 = C_blockdiag_k133*Upsilon_temp_k133;

%Theta

```

D. M-FILES USED IN THE MODEL

17.06.11 15:54 C:\Users\alvsvag\Documents\My Dropbox\master_2011\cd\mpc.m 2 of 2

```
Theta_k133(:,(1:1)) = Upsilon_k133;
for r = 1:H_u-1
    Theta_k133(:,(1*r+1:1*r+1)) = shiftd(Upsilon_k133,0,r); %tested OK
end

%Xi
Xi1_k133 = zeros(m*(H_p-H_w+1), d);
for s = 0:m*(H_p-H_w+1)-1
    if s==0
        i = 1;
        Xi1_k133 = C_blockdiag_k133(:, i:i+n-1)*B2_k133;
    else
        i = s*n+1;
        Xi1_k133 = Xi1_k133 + C_blockdiag_k133(:, i:i+n-1)*(A_k133^s)*B2_k133;
    end
end

Xi_k133(:,(1:d)) = Xi1_k133;
for t = 1:H_p-1
    Xi_k133(:,(d*t+1:d*t+d)) = shiftd(Xi1_k133,0,t); %tested OK
end
```

D.9 MPC function without constraints

08.06.11 13:59 Block: active_building_new_model_mpc_simple/Building.../K_MPC 1 of 1

```

function y = fcn(e,Theta,timer)
%#eml

H_p = 35; %prediction horizon
H_u = H_p;

%% -----Objective weight matrices-----
Q_mpc = 100.0; %penalizes deviation from set point
R_mpc = 100; %penalizes the use of input (energy)
Q_mat_mpc = eye(H_p)*Q_mpc;
R_mat_mpc = eye(H_u)*R_mpc;

%variables that needs to persist between time periods
persistent prev_timer K_MPC

if isempty(prev_timer)
    % First time through the code initialize prev_timer to ensure that timer ~=
prev_timer
    prev_timer = -1;
    K_MPC = 0; %pre-declares the output type and number to avoid the following error:
Function output 'y' cannot be of MATLAB type.
end

%if else loop that enables the optimization to run only at a spesific interval
(each time the timer value changes)
if timer~=prev_timer
    %% Calculation of deltaU

    % Alternative 1
    % H = Theta'*Q_mat*Theta + R_mat;
    % K_MPC = [1, zeros(1,H_p - 1)]*inv(H)*(Theta)'*Q_mat;

    % Alternative 2 (best alternative, no problem even if Theta is ill-conditioned)
    Sq = chol(Q_mat_mpc); %squareroot of Q_mat
    Sr = chol(R_mat_mpc); %squareroot of R_mat

    Kfull = [Sq*Theta; Sr]\[Sq; zeros(H_p,H_p)];
    l = 1; %number of inputs
    K_MPC = Kfull(1:l,:);

    %pre-declares the output type and number to avoid the following error: Function
output 'y' cannot be of MATLAB type.
    y = K_MPC*e;
    prev_timer = timer;
else
    y = K_MPC*e;
end

```

D. M-FILES USED IN THE MODEL

D.10 MPC function with constraints

17.06.11 13:53 Block: active_building_new_model_mpc_one_room/Buildi.../K_MPC 1 of 2

```
function y = fcn(e,Theta,indoor_temp,u,timer)
%#eml
eml.extrinsic('quadprog');
eml.extrinsic('blkdiag2');

H_p = 100; %prediction horizon
H_u = H_p;
m = 1; %number of controlled (measured) outputs

%% -----Objective weight matrices-----
Q_mpc = 100; %penalizes deviation from set point
R_mpc = 10; %penalizes the use of input (energy)
Q_mat_mpc = eye(H_p)*Q_mpc;
R_mat_mpc = eye(H_u)*R_mpc;

%% -----Creation of the constraint matrices-----

%stating the constraints:
% dU: no constraints
% U: current indoor temperature <= U <= 70 degrees celsius
% z: no constraints

%constraints on U
F = zeros(2*H_u, m*H_u+1);
Fstart = [-1/indoor_temp, 1; 1/70, -1]; %constraints on U (see example 2.3 in
Maciejowski)
%build F
F_blockdiag = blkdiag2(Fstart(:,1),H_u);
F(:,1:H_u) = F_blockdiag;
for u = 1:H_u
    F(2*u-1:2*u,m*H_u+1) = Fstart(:,2); %tested OK
end

%convert the constraints to constraints on dU
%constraints on U
F_dU = zeros(2*H_u, m*H_u);
for v = 1:H_u
    F_dU(:,v) = sum(F(:,v:H_u),2); %tested OK
end

%variables that needs to persist between time periods
persistent prev_timer dU_opt

if isempty(prev_timer)
    % First time through the code initialize prev_timer to ensure that timer ~=
prev_timer
    prev_timer = -1;
    dU_opt = 0; %pre-declares the output type and number to avoid the following
error: Function output 'y' cannot be of MATLAB type.
end

%if else loop that enables the optimization to run only at a spesific interval
(each time the timer value changes)
if timer~=prev_timer
    %set up as a QP problem
```

D.10 MPC function with constraints

17.06.11 13:53 Block: active_building_new_model_mpc_one_room/Buildi.../K_MPC 2 of 2

```
% min x'*H*x - G'*x subject to: OMEGA*x <= omega
% x
G = 2*Theta'*Q_mat_mpc*e;
H = Theta'*Q_mat_mpc*Theta + R_mat_mpc;
largeOmega = F_dU;
omega = -F_dU(:,1)*u - F(:,end);

%solve the QP problem (calculation of deltaU)
%minimizes 1/2*x'*H*x + G'*x subject to the restrictions OMEGA*x <= omega
dU = quadprog(2*H,-G,largeOmega,omega); %NB! 2*H and -G due to the definition of
the minimization problem

% %minimizes 1/2*x'*H*x + G'*x
% dU = quadprog(2*H,-G); %NB! 2*H and -G due to the definition of the
minimization problem

dU_opt = [1, zeros(1,H_u - 1)]*dU;
y = dU_opt;
prev_timer = timer;
else
y = dU_opt;
end
```

D. M-FILES USED IN THE MODEL

D.11 Feed forward

07.06.11 14:03 C:\Users\alvsvag\Documents\My Dropbox\master...\feedforward.m 1 of 1

```
function y = feedforward(u)
%#eml

meas_dist = 2; %number of measured disturbances
Hp = 35; %prediction horizon

y = zeros(meas_dist*Hp,1);
[n,m] = size(y);
i=1;
while(i<n)
    y(i:i+meas_dist-1) = u; %tested OK
    i = i+meas_dist;
end
```


D.12 Energy consumption radiators

07.06.11 14:02 C:\Users\alvsvag\Documents\...\energy_consumption_radiators.m 1 of 3

```

%% -----energy consumption - radiator circuit -----

%room k133
temp_diff_radiator_k133 = k133_radiator_temp-k133_indoor_air_temperature;

[energy_consumption_radiator_k133] = (temp_diff_radiator_k133.^2)*
(1/k133_Rconv_radiator+5/k133_Rradiation_radiator_wall); %5 due to the fact that the
radiator radiates an equal amount of energy to 4 walls and one ceiling

energy_radiator_sum = 0;
for i=1:length(t)
    energy_radiator_sum=energy_radiator_sum+energy_consumption_radiator_k133(i);
end

average_energy_radiator = energy_radiator_sum/(length(t-1)*1000); % divided by 1000
to get average energy in kilowatt.

kilowatt_hours_used_k133 = average_energy_radiator*numdays*24

%room k134
temp_diff_radiator_k134 = k134_radiator_temp-k134_indoor_air_temperature;

[energy_consumption_radiator_k134] = (temp_diff_radiator_k134.^2)*
(1/k134_Rconv_radiator+5/k134_Rradiation_radiator_wall); %5 due to the fact that the
radiator radiates an equal amount of energy to 4 walls and one ceiling

energy_radiator_sum = 0;
for i=1:length(t)
    energy_radiator_sum=energy_radiator_sum+energy_consumption_radiator_k134(i);
end

average_energy_radiator = energy_radiator_sum/(length(t-1)*1000); % divided by 1000
to get average energy in kilowatt.

kilowatt_hours_used_k134 = average_energy_radiator*numdays*24

%room k135
temp_diff_radiator_k135 = k135_radiator_temp-k135_indoor_air_temperature;

[energy_consumption_radiator_k135] = (temp_diff_radiator_k135.^2)*
(1/k135_Rconv_radiator+5/k135_Rradiation_radiator_wall); %5 due to the fact that the
radiator radiates an equal amount of energy to 4 walls and one ceiling

energy_radiator_sum = 0;
for i=1:length(t)
    energy_radiator_sum=energy_radiator_sum+energy_consumption_radiator_k135(i);
end

average_energy_radiator = energy_radiator_sum/(length(t-1)*1000); % divided by 1000
to get average energy in kilowatt.

kilowatt_hours_used_k135 = average_energy_radiator*numdays*24

%room k136
temp_diff_radiator_k136 = k136_radiator_temp-k136_indoor_air_temperature;

```

D. M-FILES USED IN THE MODEL

07.06.11 14:02 C:\Users\alvsvag\Documents\...\energy_consumption_radiators.m 2 of 3

```
[energy_comsumption_radiator_k136] = (temp_diff_radiator_k136.^2)*  
(1/k136_Rconv_radiator+5/k136_Rradiation_radiator_wall); %5 due to the fact that the  
radiator radiates an equal amount of energy to 4 walls and one ceiling  
  
energy_radiator_sum = 0;  
for i=1:length(t)  
    energy_radiator_sum=energy_radiator_sum+energy_comsumption_radiator_k136(i);  
end  
  
average_energy_radiator = energy_radiator_sum/(length(t-1)*1000); % divided by 1000  
to get average energy in kilowatt.  
  
kilowatt_hours_used_k136 = average_energy_radiator*numdays*24  
  
%room k143  
temp_diff_radiator_k143 = k143_radiator_temp-k143_indoor_air_temperature;  
  
[energy_comsumption_radiator_k143] = (temp_diff_radiator_k143.^2)*  
(1/k143_Rconv_radiator+5/k143_Rradiation_radiator_wall); %5 due to the fact that the  
radiator radiates an equal amount of energy to 4 walls and one ceiling  
  
energy_radiator_sum = 0;  
for i=1:length(t)  
    energy_radiator_sum=energy_radiator_sum+energy_comsumption_radiator_k143(i);  
end  
  
average_energy_radiator = energy_radiator_sum/(length(t-1)*1000); % divided by 1000  
to get average energy in kilowatt.  
  
kilowatt_hours_used_k143 = average_energy_radiator*numdays*24  
  
%room k164  
temp_diff_radiator_k164 = k164_radiator_temp-k164_indoor_air_temperature;  
  
[energy_comsumption_radiator_k164] = (temp_diff_radiator_k164.^2)*  
(1/k164_Rconv_radiator+5/k164_Rradiation_radiator_wall); %5 due to the fact that the  
radiator radiates an equal amount of energy to 4 walls and one ceiling  
  
energy_radiator_sum = 0;  
for i=1:length(t)  
    energy_radiator_sum=energy_radiator_sum+energy_comsumption_radiator_k164(i);  
end  
  
average_energy_radiator = energy_radiator_sum/(length(t-1)*1000); % divided by 1000  
to get average energy in kilowatt.  
  
kilowatt_hours_used_k164 = average_energy_radiator*numdays*24  
  
%% -----total energy consumption radiators-----  
[sum_energy_comsumption_radiator] = [energy_comsumption_radiator_k133]+  
[energy_comsumption_radiator_k134]+[energy_comsumption_radiator_k135]+  
[energy_comsumption_radiator_k136]+[energy_comsumption_radiator_k143]+  
[energy_comsumption_radiator_k164];
```

D.12 Energy consumption radiators

07.06.11 14:02 C:\Users\alvsvag\Documents\...\energy_consumption_radiators.m 3 of 3

```
sum_kilowatt_hours_used =  
kilowatt_hours_used_k133+kilowatt_hours_used_k134+kilowatt_hours_used_k135+kilowatt_h  
ours_used_k136+kilowatt_hours_used_k143+kilowatt_hours_used_k164
```

D. M-FILES USED IN THE MODEL

,

D.13 Energy consumption ventilation

07.06.11 14:03 C:\Users\alvsvag\Document...\energy_consumption_ventilation.m 1 of 1

```
%% -----energy consumption - ventilation-----

temp_diff = water_to_HC-water_from_HC;
[energy_consumption_ventilation] = qw_hc*cp_water*rho_water*temp_diff;

energy_ventilation = 0;
for i=1:length(t)
    energy_ventilation=energy_ventilation+(20*energy_consumption_ventilation(i)); %←
times 20 to get pr.sec. (since tstep equals 20 sec.)
end

energy_vent_kilo = energy_ventilation/1000
```