NTNU – Trondheim
Norwegian University of
Science and Technology

Design and implementation of a Kalman Filter based estimator for temperature control

Thomas Hasfjord

Master of Science in Cybernetics and Robotics
Submission date: May 2014
Supervisor: Geir Mathisen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Norwegian University of Science and Technology (NTNU)

Master Thesis

Design and implementation of a Kalman Filter based estimator for temperature control

Author:
Thomas Hasfjord

Supervisor:
Geir Mathisen

A thesis submitted in fulfilment of the requirements for the degree of MSc. Engineering Cybernetics

in the

Department of Engineering Cybernetics

May 2014

NTNU

Abstract

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

MSc. Engineering Cybernetics

Design and implementation of a Kalman Filter based estimator for temperature control

by Thomas Hasfjord

The search for improved energy-efficient building solutions seem to be given much attention in the recent years. Both academically and commercially there is large interest to improve building energy economy by applying advanced control systems.

This thesis explores the possibility of implementing a simplified building model for optimal control based on calculation of physical constants and offline estimation. The data is collected from a case study of an apartment located in Trondheim, Norway. By applying an Extended Kalman Filter (EKF) with state and parameter estimation the estimator can compensate for unmeasured disturbances.

The simulation based results show that even with a simple model is implemented in the EKF accurate results are achieved even when large disturbances and missing data occur The computational complexity of the model is kept as low as possible, making it suitable for real time implementation. The proposed model with EKF has twice the number of states as there are rooms in the building, this means that the computational complexity scales fairly well when applied to larger buildings. The performance and complexity of the approach should be suitable for use in an model-based optimal building temperature control system.

Preface

This thesis is submitted in fulfillment of the requirements of MSc. Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU)it constitutes 30 points and is part of a smart grid project by SINTEF, in collaboration with eGotham[1] and I3RES[2].

The author would like to thank Geir Mathisen for proposing and supervising the thesis assignment, Giancarlo Marafioti for co-supervising and providing guidance for the thesis, also for allowing the use of his apartment and collecting data for the case study. Lastly fellow students for five exciting years at the university.

Table of Content

Abstract ii

Preface iii

Table of Content iv

List of Figures vii

List of Tables ix

Abbreviations xi

Physical Constants xiii

Symbols xv

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Motivation . . . 2
- 1.3 Design Constrains . . . 2
- 1.4 Contribution . . . 3
- 1.5 Thesis organization . . . 3

2 Theory 5
- 2.1 Fundamentals of heat transfer . . . 5
- 2.2 State space systems . . . 10
- 2.3 Model Predictive Control (MPC) . . . 12
- 2.4 Kalman filter . . . 13
  - 2.4.1 Parameter estimation . . . 15
  - 2.4.2 The Extended Kalman Filter (EKF) . . . 16

3 State of the art of house thermal flow modeling 19
- 3.1 Sub-space methods (4SID) . . . 20
- 3.2 Prediction Error Methods (PEM) . . . 20
- 3.3 Deterministic Semi-Physical Modeling (DSPM) . . . 21
- 3.4 Probalistic Semi-Physical Modeling (PSPM) . . . 22

4 Model design **23**
4.1 Discretization . . . 28
4.2 Augmented model for EKF . . . 28

5 Implementation **29**

6 Case study: Apartment in Trondheim **35**
6.1 Model set-up . . . 36
6.2 Building description . . . 38
6.2.1 Areas and volume . . . 38
6.2.2 Building components . . . 40
6.3 Experimental set-up . . . 44

7 Results and discussion **47**
7.1 Selection of dataset for estimation . . . 47
7.2 Model testing . . . 53
7.2.1 Experiment 1: Grey-box estimation with 4 parameters . . . 53
7.2.2 Experiment 2: Using calculated values for transmittance . . . 57
7.2.3 Experiment 3: Compensating for differences in external walls . . . 60
7.3 Kalman filter testing . . . 65
7.3.1 Tuning . . . 65
7.3.2 Experiment 4: Using EKF with state and parameter estimation . . . 66
7.4 Computational time . . . 71
7.5 General remarks . . . 71

8 Conclusion and further work **75**

A Code documentation **79**
A.1 Model . . . 79
A.2 RoomTemperatureSet . . . 81
A.3 TemperatureSet . . . 82
A.4 EKF . . . 84
A.5 Support functions . . . 85

B Plots **89**
B.1 Selection of dataset for estimation . . . 90
B.2 Experiment 1 . . . 95
B.3 Experiment 2 . . . 100
B.4 Experiment 3 . . . 105
B.5 Kalman Filter test . . . 110

Bibliography **115**

List of Figures

1.1 Diagram of complete control system 1

2.1 Drawing of proposed control volume for thermal flow 8

5.1 Diagram of complete control system 29
5.2 Class diagram 31
5.3 Example of input datafile 32

6.1 Drawing of apartment layout for case study 35
6.2 3D-drawing of apartment 38
6.3 Cross-section of inner walls 41
6.4 Cross-section of outer walls 42
6.5 Cross-section of floor 42

7.1 Selection of dataset for estimation: Room 5: Main bedroom 48
7.2 Selection of dataset for estimation: Room 4: Living room 49
7.3 Selection of dataset for estimation: Room 6: Kitchen 51
7.4 Selection of dataset for estimation: Room 2: Entrance 51
7.5 Selection of dataset for estimation: Wind speed 52
7.6 Experiment 1: Room 1: Bathroom 54
7.7 Experiment 1: Room 5: Main bedroom 55
7.8 Experiment 1: Room 6: Kitchen 55
7.9 Experiment 2: Room 6: Kitchen 58
7.10 Experiment 2: Room 3: Guest bedroom 58
7.11 Experiment 3: Room 4: Living room 61
7.12 Experiment 3: Outside 62
7.13 Experiment 3: Wind speed 62
7.14 Experiment 3: Room 2: Entrance 63
7.15 Experiment 4: Room 5: Main bedroom 67
7.16 Experiment 4: Room 4: Living room 68
7.17 Experiment 4: Estimated coefficients 68
7.18 Experiment 4: Room 3: Guest bedroom 69

B.1 Selection of dataset for estimation: Room 1: Bathroom 90
B.2 Selection of dataset for estimation: Room 2: Entrance 91
B.3 Selection of dataset for estimation: Room 3: Guest bedroom 91
B.4 Selection of dataset for estimation: Room 4: Living room 92
B.5 Selection of dataset for estimation: Room 5: Main bedroom 92
B.6 Selection of dataset for estimation: Room 6: Kitchen 93

B.7 Selection of dataset for estimation: Outside 93
B.8 Selection of dataset for estimation: Basement 94
B.9 Selection of dataset for estimation: Wind speed 94
B.10 Experiment 1: Room 1: Bathroom 95
B.11 Experiment 1: Room 2: Entrance 96
B.12 Experiment 1: Room 3: Guest bedroom 96
B.13 Experiment 1: Room 4: Living room 97
B.14 Experiment 1: Room 5: Main bedroom 97
B.15 Experiment 1: Room 6: Kitchen 98
B.16 Experiment 1: Outside 98
B.17 Experiment 1: Wind speed 99
B.18 Experiment 2: Room 1: Bathroom 100
B.19 Experiment 2: Room 2: Entrance 101
B.20 Experiment 2: Room 3: Guest bedroom 101
B.21 Experiment 2: Room 4: Living room 102
B.22 Experiment 2: Room 5: Main bedroom 102
B.23 Experiment 2: Room 6: Kitchen 103
B.24 Experiment 2: Outside 103
B.25 Experiment 2: Wind speed 104
B.26 Experiment 3: Room 1: Bathroom 105
B.27 Experiment 3: Room 2: Entrance 106
B.28 Experiment 3: Room 3: Guest bedroom 106
B.29 Experiment 3: Room 4: Living room 107
B.30 Experiment 3: Room 5: Main bedroom 107
B.31 Experiment 3: Room 6: Kitchen 108
B.32 Experiment 3: Outside 108
B.33 Experiment 3: Wind speed 109
B.34 Experiment 4: Room 1: Bathroom 110
B.35 Experiment 4: Room 2: Entrance 111
B.36 Experiment 4: Room 3: Guest bedroom 111
B.37 Experiment 4: Room 4: Living room 112
B.38 Experiment 4: Room 5: Main bedroom 112
B.39 Experiment 4: Room 6: Kitchen 113
B.40 Experiment 4: Outside 113
B.41 Experiment 4: Estimated parameters 114

List of Tables

2.1 Table of thermal resistances in the modes of transfer 7

5.1 Detailed specifications of test computer 30

6.1 Wall areas between rooms $[m^2]$ 38
6.2 Air volume and external areas 39
6.3 Heater power $[W]$ 40
6.4 Values used for calculating surface thermal transmittance 41
6.5 Values for thermal resistance in inner walls 41
6.6 Values for thermal resistance in outer walls 41
6.7 Values for thermal resistance in floors 43
6.8 Transmittances for building components 43
6.9 Experiment equipment list 44

7.1 Average wind speed and direction over the three days of testing 50
7.2 Experiment 1: Grey-box estimation values 53
7.3 Experiment 2: Grey-box estimation value 57
7.4 Experiment 3: Grey-box estimation values 60
7.5 Experiment 1-3: room 1-3: Model error 63
7.6 Experiment 1-3: room 4-6: Model error 64
7.7 Iteration time for estimator 71

8.1 All experiments: room 1-3: Model/estimator error 75
8.2 All experiments: room 4-6: Model/estimator error 75

A.1 Properties for object Model 79
A.2 Methods for object Model 80
A.3 Properties for object RoomTemperatureSet 81
A.4 Methods for object RoomTemperatureSet 81
A.5 Properties for object TemperatureSet 82
A.6 Methods for object TemperatureSet 83
A.7 Properties for object EKF 84
A.8 Methods for object EKF 84
A.9 Support functions for system I 86
A.10 Support functions for system II 87
A.11 Support functions for system III 88

Abbreviations

API	**A**application **P**rogramming **I**nterface
ARMAX	**A**uto**R**egressive–**M**oving-**A**verage model with e**X**ogenous inputs
BEPST	**B**uilding **E**nergy **P**erformance **S**imulation **T**ools
CDKF	**C**ontinuous-**D**iscrete **K**alman **F**ilter
CFD	**C**omputational **F**luid **D**ynamics
C.V.	**C**ontrol **V**olume
CSV	**C**omma **S**eparated **V**alue (File extension .csv)
CTF	**C**onduction **T**ransfer **F**unction
DSPM	**D**eterministic **S**emi-**P**hysical **M**odeling
EKF	**E**xtended **K**alman **F**ilter
EM	**E**xpectation-**M**aximization
HVAC	**H**eating **V**entilation and **A**ir **C**onditioning
MIMO	**M**ultiple **I**nput **M**ultiple **O**utput
ML	**M**aximum **L**ikelihood
MPC	**M**odel **P**redictive **C**ontrol
MRI	**M**PC **R**elevant **I**dentification
PEM	**P**rediction **E**rror **M**ethods
PID	**P**roportional-**I**ntegral-**D**erivative (-controller)
PLS	**P**artial **L**east **S**quares
PSPM	**P**robalistic **S**emi-**P**hysical **M**odeling
SISO	**S**ingle **I**nput **S**ingle **O**utput
SOC	**S**ystem **O**n a **C**hip
UKF	**U**nscented **K**alman **F**ilter
4SID	**S**ubspace **S**tate **S**pace **S**ystem **ID**entification

Physical Constants

Stefan-Boltzmanns Constant $\sigma = 5.670 \times 10^{-8}[Wm^{-2}K^{-4}]$

Symbols

A	Area	m^2
$\mathbf{A}$	State-matrix	
$\mathbf{B}$	Input-matrix	
$\mathbf{C}$	Output-matrix	
C	Heat capacitance	JK^{-1}
C_v	Heat capacity by volume	$Jm^{-3}K^{-1}$
$\mathbf{D}$	Input-output-matrix	
d	Length of conduction path trough building component	m
E_{st}	Stored energy in thermodynamic system	J
h_c	Conduction heat transfer coefficient	$Wm^{-2}K^{-1}$
h_r	Radiation heat transfer coefficient	$Wm^{-2}K^{-1}$
k	Thermal conductivity	$Wm^{-1}K^{-1}$
$\mathbf{K}$	Disturbance-matrix	
$\mathbf{K}_F$	Kalman filter gain	
L	Length of conduction path trough wall	m
P	Power	W (Js^{-1})
$\mathbf{P}$	Kalman filter covariance matrix	
q''_x	Heat flux	Jm^{-2}
Q	Energy transfered to/from thermodynamic system	J
$\mathbf{Q}$	Plant covariance matrix	
R	Thermal resistance in component	m^2KW^{-1}
$\mathbf{R}$	Measurement noise covariance matrix	
T	Temperature	K
$\bar{T}$	Mean temperature	$^\circ C$
U	Thermal transmittance	$Wm^{-2}K^{-1}$

$\mathbf{u}$	Input vector	
$\bar{v}$	Mean wind speed	m/s
V	Volume	m^3
W	Work	J
$\mathbf{x}$	State vector	
$\hat{\mathbf{x}}$	Estimated state vector	
$\mathbf{y}$	Vector of outputs from plant	
$\mathbf{\Gamma}$	Linearized plant noise matrix	
ε	Emissivity	
θ	Vector of parameters	
λ	Design thermal conductivity	$Wm^{-2}K^{-1}$
Λ	Vector of eigenvalues	
$\mathbf{\Lambda}$	Linearized measurement noise matrix	
μ	Efficiency	
ξ	Augmented state vector	
ρ	Volumetric mass density	kg/m^3
σ^2	Covariance	
$\mathbf{\Sigma}$	Linearized output matrix	
$\mathbf{\Phi}$	Linearized state matrix	
$\mathbf{\Omega}$	Augmented state matrix	
$\mathcal{O}$	Observability matrix	
$\mathcal{C}$	Controllability matrix	

Chapter 1

Introduction

1.1 Background

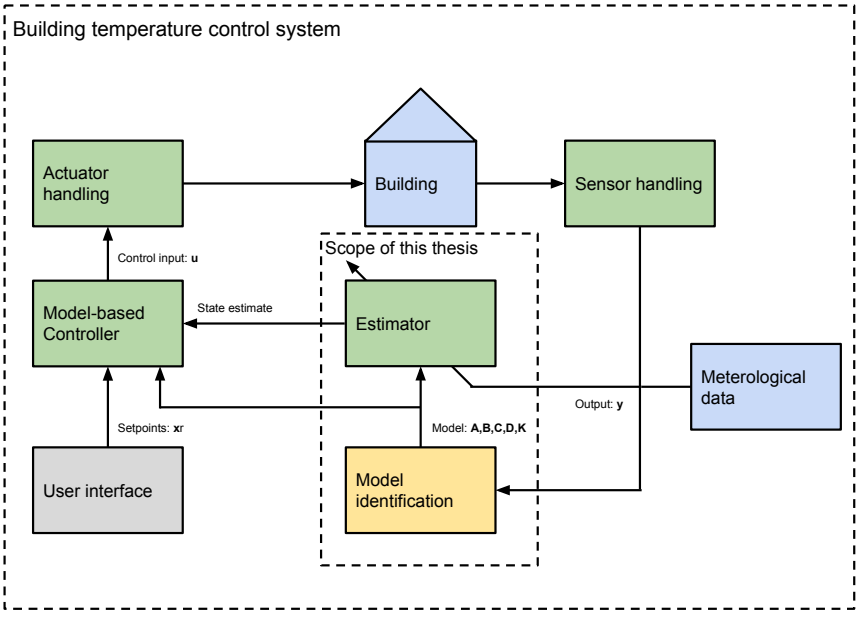

FIGURE 1.1: Diagram of complete control system

The thesis is part of a smart grid project at SINTEF who aims at implementing a complete model based system for building temperature control. The system should be easy to use and preferably not require elaborate redesigning for each individual building.

This thesis focuses on the modeling and estimation part of the control system, leaving the controller to be implemented in later works.

1.2 Motivation

In developed countries, Heating Ventilation and Air-Conditioning systems (HVAC) for buildings constitutes 10-20% of the total energy consumption. This number also seem to be on the rise according to Pérez-Lombard et al. [3]. In general the field of energy optimization seems to be on most contractors minds, and is also a popular subject in current building journals, therefore new methods for saving energy in buildings are welcome both in industry and academia.

One way of improving the energy consumption of buildings is to implement more efficient control strategies. Such strategies can be implemented without expensive rebuilding to previously constructed buildings and is suggested to be a key element for realizing the potential of new energy saving buildings [4]. A promising method of optimal control that have been used with success as building temperature control is the Model Predictive Controller or MPC.

1.3 Design Constrains

The model for implementing an optimal control strategy should be designed in such a way that it is simple enough to run in real time on a System on a Chip or similar. The complexity of the model will affect the runtime in this limited environment, so efforts should be made to simplify the model without sacrificing precision. In addition the model structure should be as general as possible, to prevent having to make adaptations when applying to a different building. Since the model is to be part of a larger system portability and modularity is important, to have a easy implementation and inclusion in a larger program.

Because of the cold climate in Scandinavia the energy consumption focus is mostly on heating, while other, more temperate parts of the world might emphasize air-conditioning.

This difference must be taken into account when designing energy saving schemes. A design suitable for cold climate is not necessarily optimal for a warmer one.

1.4 Contribution

The main contribution of this thesis is a framework for model design and implementation for a building temperature control system. It also provides a case study of a real building to verify that the used methods can give an accurate description of the building in question. It points out the main challenges when applying the model to a larger optimal control based system.

The scripts and algorithms are constructed such that it should be easy to apply to a different building or to rewrite in any other programming language.

1.5 Thesis organization

The thesis looks at temperature control in buildings from a control engineering point of view, the chapters are organized as follows: First relevant theory for understanding the methods and physics being applied are presented in Chapter 2, then a short review of current modeling methods in Chapter 3, before the general building model is introduced mathematically in Chapter 4. The thesis continues by describing the proposed implementation in Chapter 5, before the case study is presented in Chapter 6. Chapter 7 displays and discusses the results from the case study. Finally Chapter 8 concludes the thesis and provides proposals for further work.

A comprehensive documentation of objects and functions implemented is included in Appendix A. All plots from the case study simulations are included in Appendix B.

Chapter 2

Theory

This chapter considers the fundamental physical and mathematical premises for model design and identification of heat flow in buildings. It is a short overview of necessary methods and standards needed to motivate choices made in later chapters of the thesis.

2.1 Fundamentals of heat transfer

Heat transfer is an important subject in many engineering disciplines, this section reviews the basics of heat transfer theory to motivate chosen model design. The theory is compiled from [5] and can be omitted for readers familiar with heat transfer and thermodynamics.

It is known from fundamental physics theory that there are three ways heat can be transfered: *conduction* through a substance from more energetic particles to less, *convection* from a surface to a moving fluid, thermal *radiation* from a heated surface. Heat transfer can be quantified in terms of rate equations given that the temperature distribution $T(\cdot)$ is known. For conduction the rate equation is known as Fourier's law, and can be stated for a plane as:

$$q''_{cond} = -k\Delta T \tag{2.1}$$

Where q''_x is the heat flux $[W/m^2]$, k is the transport property, known as the thermal conductivity $[W/m \cdot K]$, the sign implies that the heat always flow from the hotter part of the substance to a colder one.

For a linear heat distribution the equation can be simplified:

$$q''_{cond} = -\frac{\lambda}{L}(T_2 - T_1) \tag{2.2}$$

Where T_1 and T_2 is the temperatures on each surface of the plane, L is the length of the plane in the heat gradient direction, λ is the design thermal conductivity defined as: $\lambda = kL$.

For convection the governing rate equation is known as Newton's law of cooling, and can be stated as:

$$q''_{conv} = h_c(T_s - T_\infty) \tag{2.3}$$

Where T_s and T_∞ is the temperature of the surface and the fluid [K], respectively, h is the convection heat transfer coefficient.

Incoming radiation to a surface can be absorbed or reflected, when it is absorbed the thermal energy of the surface increases, while when reflected the radiation is returned to the environment. Surfaces also emit thermal radiation depending on the emissivity, a completely absorbent surface is called a blackbody and has an emissivity of 0. The governing rate equation is Stephan-Boltzmann law, here defined for a gray surface:

$$q''_{rad} = \varepsilon\sigma(T_s^4 - T_{sur}^4) \tag{2.4}$$

Where $0 \leq \varepsilon \leq 1$ is the surface emissivity, σ is the Stephan-Boltzmann constant, T_s and T_{sur} is the temperature at the radiating surface and the surrounding surface, respectively. For convenience the equation sometimes is expressed as:

$$q_{rad} = h_r A(T_s - T_{sur}) \tag{2.5}$$

Where h_r is the radiation heat transfer coefficient defined as:

$$h_r \equiv \varepsilon\sigma(T_s + T_{sur})(T_s^2 + T_{sur}^2) \tag{2.6}$$

This is a form of linearizion, notice that the dependence on temperature differences are lower in h_r than in the original equation, and the form corresponds with the formulas of conduction and convection. For the remainder the radiation heat transfer coefficient can be seen as a constant.

From the three equations the thermal resistance of each mode can be formulated

Transfer mode	**Rate of heat transfer**	**Resistance**
Conduction	$\dot{Q} = \frac{kA}{L}(T_1 - T_2)$	$\frac{L}{kA}$
Convection	$\dot{Q} = h_c A(T_s - T_\infty)$	$\frac{1}{h_c A}$
Radiation	$\dot{Q} = h_r A(T_s - T_{sur})$	$\frac{1}{h_r A}$

TABLE 2.1: Table of thermal resistances in the modes of transfer

h_r defined as in Equation 2.6

To achieve a balance equation the first law of thermodynamics is applied to a control volume:

$$\Delta E_{st}^{tot} = Q - W \tag{2.7}$$

Where E_{st}^{tot} is the total change in energy stored in the system, Q is the net heat transfered to the system, and W is the work the system performs.

For a system that does not perform any work, the only change in stored energy comes through heat transfer. Further more the only stored energy of interest is internal energy from temperature. Differentiating Equation 2.7 with respect to time gives the rate equation:

$$\dot{E}_{st} = -\dot{Q} = -\dot{Q}_{ht} + \dot{Q}_e \tag{2.8}$$

Where $\dot{Q}_{ht}$ is the energy loss rate due to heat transfer and $\dot{Q}_e$ is the energy gain rate from electrical heating.

From definition the heat capacitance is given by: $C = \frac{dE_{st}}{dT} = C_v V$ so the net change of

internal energy can be written as:

$$\frac{dE_{st}}{dt} = \frac{dE_{st}}{dT}\frac{dT}{dt} = C\frac{dT}{dt} \tag{2.9}$$

Considering a two dimensional control volume that encapsulates the inside of the room, with a small air film separating the control surfaces from the walls. It is assumed no internal heat generation within the walls and they all are considered homogeneous. Wall corners are ignored, so equal area is assumed on both sides of walls, and only the effect of a single wall is considered and the heat flow through said wall is assumed to be one dimensional.

$$C_v V\frac{dT}{dt} = A_s\left[q''_{cond}(T_{s_1}, T_{s_2}) - (q''_{conv}(T_{s_2}, T_\infty) + q''_{rad}(T_{s_2}, T_{sur}))\right] + \dot{Q}_e \tag{2.10}$$

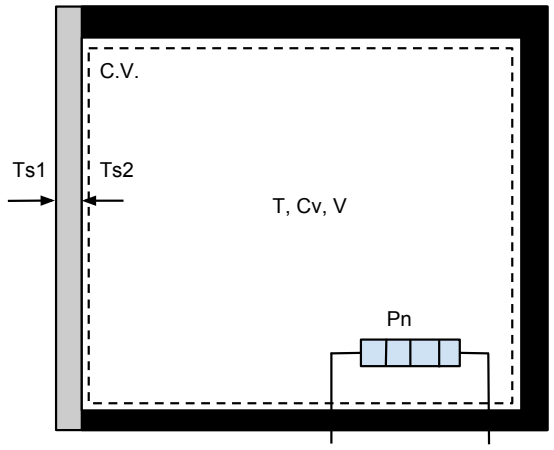

FIGURE 2.1: Drawing of proposed control volume for thermal flow

The general differential equation can be stated as:

Notice that the conduction is dependent on the temperature difference on each side of the wall T_{s_1} and T_{s_2} respectively, while the convection is dependent on the temperature

difference on the inside of the wall T_{s_2} and air-temperature T_∞. Radiation is dependent on the inside wall temperature and the temperature on the surroundings being irradiated by the wall T_{sur} i.e. the various surfaces of the room such as walls and ceiling. This causes the equation to be non-homogeneous, one solution is to separate it into several single order differential equations, where temperature in room and temperature on walls are separate states. Whit this type of differential equation the number of states needed will depend on both the number of rooms and the number of walls between the rooms, causing the complexity to increase greatly when adding more rooms. A simplification is needed to make the system scalable and more computationally efficient.

Rate of energy is also known as power, so the rate of energy supplied by heating can be given by the heaters nominal power rating P_n and an efficiency coefficient μ seeing how electrical heaters typically have a high efficiency the heating power can be approximated to the nominal power:

$$\dot{Q}_e = \mu P_n \approx P_n \tag{2.11}$$

Construction engineers often use a calculated total thermal transmittance U when designing [6]. This is a standardized way of calculating thermal flow through composite building components. The calculation principle is stated next:

- Obtain thermal resistance for each homogeneous part of the component
- Combine these individual resistances to obtain the total thermal resistance of the component, including (where appropriate) the effect of surface resistances.

Numerical values for common building materials are given in [7].

By using this principle a homogeneous single order differential equation can be obtained for each room:

$$\frac{dT}{dt} = \sum_{i=1}^{\mathbb{C}} \frac{UA_i}{C_v V}(T_i - T) + \frac{P_n}{C_v V} K_h \tag{2.12}$$

where T is the temperature in the room, assumed to be equal over the whole volume, i is the index of a space in or around the building, $\mathbb{C}$ is the subset of space indexes that are connected to current room.

With this representation the heater is assumed to heat the whole volume of air equally at all times, realistically this is not possible, so a coefficient K_h is introduced to compensate for the dispersion effect of heater power.

The thermal transmittance U will vary with different building components, so it needs to be calculated for all the different type of components in the building.

2.2 State space systems

Most physical systems can be described by continuous differential equations, as shown heat flow is no exception. A practical and well known representation of multiple differential equations is the state space representation.

The general linear continuous state space system can be stated as:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}\mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{v}(t) \end{aligned} \tag{2.13}$$

Where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is the input vector, $\mathbf{y}(t) \in \mathbb{R}^p$ is the output vector. $\mathbf{w}(t)$ is a vector of plant noise, $\mathbf{v}(t)$ is a vector of measurement noise, both usually considered to be zero mean Gaussian white noise.

Continuous state space systems are useful for describing physics and mathematical manipulation, but when using a digital computer to control a given system it must be sampled and discretized The continuous state space system in Equation 2.13 can be discretized by using Zero-Order-Hold as in Chen [8]:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_d\mathbf{x}_k + \mathbf{B}_d\mathbf{u}_k + \mathbf{K}_d\mathbf{w}_\mathbf{k} \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_\mathbf{k} \end{aligned} \tag{2.14}$$

Where the discretized system matrices are given by:

$$\begin{aligned} \mathbf{A}_d &= e^{\mathbf{A}T_s} = I + \mathbf{A}T_s + \frac{\mathbf{A}^2T_s^2}{2} + \cdots \approx I + \mathbf{A}T_s \\ \mathbf{B}_d &= \int_0^{T_s} e^{\mathbf{A}\tau}d\tau\mathbf{B} \approx \int_0^{T_s} Id\tau\mathbf{B} = T_s\mathbf{B} \end{aligned} \tag{2.15}$$

T_s is the sampling period, $\mathbf{C}$ and are unchanged from the continuous system. In the remaining part this thesis all state space systems are considered discrete and the subscript d is dropped from the system matrices.

There are a number of important properties associated with discrete state space systems that often restricts the use of the system and associated control methods. First of all there is stability. Stability is defined in Chen [8] as a dynamic system that stays within a certain region when it is initiated in said region. Linear discrete systems are considered stable if all eigenvalues of the A matrix is within the unit circle or

$$|\Lambda_i| < 1, \qquad i = \{1, \ldots, n\} \tag{2.16}$$

Again n is the number of states, or order of system, Λ is a vector containing eigenvalues of matrix A.

The *observability* is a property that states if the internal states of the system can be reached if the outputs are known. Mathematically observability is explored by examining the observability matrix $\mathcal{O}$.

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{2.17}$$

If the rank of the observability matrix is larger or equal to the order of the system it is observable. $rank(\mathcal{O}) \geq n$

The *controllability* is a property that states if all internal states can be affected by the inputs of the system. The controllability is explored by examining the controllability matrix $\mathcal{C}$.

$$\mathcal{C} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \ldots \quad \mathbf{A}^{n-1}\mathbf{B}] \tag{2.18}$$

If the rank of the controllability matrix is larger or equal to the order of the system it is controllable $rank(\mathcal{C}) \geq n$

2.3 Model Predictive Control (MPC)

There are a number of optimal control algorithms that may be applied to a thermal system, MPC is one of the most common and well known methods. It has been applied to building temperature control in Fux et al. [9], Bălan et al. [10], Ma et al. [11], Zacekova et al. [12], Široký et al. [13] the latter reporting energy savings from 15%-28% depending on house properties.

The MPC is a form of controller in which the current control move is obtained by solving a finite horizon open loop optimal control problem for each sampling instance. The optimization yields an optimal control sequence and the first control move in sequence is applied to the plant Mayne et al. [14]. MPC is popular because it handles constrains and is an optimal control approach. MPC can be extended in many ways to solve different type of control problems, but has a large computational load compared to a simpler PID control. In a practical system the states need to be estimated from the output to cancel disturbances and measurement noise, this set up is called output feedback MPC as opposed to state feedback [15].

MPC with output feedback can be summarized in the following algorithm:

for $t = 0, 1, 2, \ldots$ **do**

Compute the estimate of the current state $\hat{x}_t$ based on the measured data until time t.

Solve a dynamic optimization problem on the prediction horizon from t to $t + N$ with $\hat{x}_t$ as the initial condition.

Apply the first control move u_t from the solution above

end

Algorithm 1: Output feedback MPC algorithm Foss and Heirung [15]

A typical discrete time dynamic optimization problem can be stated as:

$$\min_{\mathbf{z}\in\mathbb{R}^n} f(\mathbf{z}) = \sum_{t=0}^{N-1} \frac{1}{2}\hat{\mathbf{x}}_{t+1}^T \mathbf{Q}_t \hat{\mathbf{x}}_{t+1} + \mathbf{d}_{\hat{\mathbf{x}}_{t+1}} \hat{\mathbf{x}}_t + \frac{1}{2}\mathbf{u}_t^T \mathbf{R}_t \mathbf{u}_t + \mathbf{d}_{ut}^T \mathbf{u}_t \tag{2.19}$$

Subject to: (2.20)

$$\hat{\mathbf{x}}_{t+1} = f(\hat{\mathbf{x}}_t, \mathbf{u}_t, \hat{\theta}_t) \tag{2.21}$$

$$\mathbf{x}^{low} \leq \hat{\mathbf{x}}_t \leq \mathbf{x}^{high} \tag{2.22}$$

$$\mathbf{u}^{low} \leq \mathbf{u}_t \leq \mathbf{u}^{high} \tag{2.23}$$

$$-\Delta\mathbf{u}^{high} \leq \Delta\mathbf{u}_t \leq \Delta\mathbf{u}^{high} \tag{2.24}$$

$$\Delta\mathbf{u}_t = \mathbf{u}_t - \mathbf{u}_{t-1} \tag{2.25}$$

$$\mathbf{z}^T = (x_1^T, \ldots, x_N^T, u_0, \ldots u_{N-1}^T) \tag{2.26}$$

$$n = N \cdot (n_x + n_u) \tag{2.27}$$

Where $\mathbf{x} \in \mathbb{R}^n$ are the states, $\mathbf{u} \in \mathbb{R}^m$ are the inputs, $\mathbf{Q}_t$ is the *positive semidefinite* weighing matrix for states at time t, $\mathbf{R}_t$ is the *positive semidefinite* weighing matrix for inputs at time t, $f(\hat{\mathbf{x}}_t, \mathbf{u}_t, \hat{\theta}_t)$ is the dynamic model to be designed, $\hat{\theta}_t$ is the model parameter estimate, the initial value $\mathbf{x}_0$ is assumed known.

By solving this dynamic optimization problem, the optimal control sequence can be obtained. One key property of optimization is whether the problem is convex or non-convex, non-convex problems are generally harder to solve. In order to generate a convex optimization problem the function $f(\cdot)$ and the inequality constrains must be linear, further more the cost function must be convex [15]. If this is not the case the estimator $f(\cdot)$ can be linearized along the prediction horizon.

2.4 Kalman filter

Kalman filtering was first introduced in Kalman [16] and is well known and popular solution to state estimation problems. The filter operates by propagating the mean and covariance of the state though time and is the optimal linear filter [17]. First an *a priori* estimate $\hat{\mathbf{x}}_{k+1|k}$ of the state is made, then when a new measurement is available, a *a posteriori* filtering is done revealing the smoothed estimate $\hat{\mathbf{x}}_{k|k}$. This is done for every

time-step in a discrete kalman filter, but the estimation part can also be performed more often than the filtering, this technique is called continuous discrete kalman filtering. The theory in this section is mostly based on [17] and [18].

To derive the general kalman filter the discrete linear system from Equation 2.13 is considered:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}\mathbf{w}(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{v}(t)\end{aligned}$$

Where $\mathbf{x}_k \in \mathbb{R}^n, \mathbf{u}_k \in \mathbb{R}^m$, and $\mathbf{y}_k \in \mathbb{R}^p$, $\mathbf{w}_k$ and $\mathbf{v}_k$ are uncorrelated zero mean white noise processes and have known covariance matrices Q_k and R_k:

$$\begin{aligned}w_k &= \mathcal{N}(0, \mathbf{Q}_k)\\ v_k &= \mathcal{N}(0, \mathbf{R}_k)\end{aligned} \tag{2.28}$$

Further more they are assumed uncorrelated:

$$E\begin{bmatrix} w_k w_k^T & w_k v_k^T \\ v_k w_k^T & v_k v_k^T \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_k & 0 \\ 0 & \mathbf{R}_k \end{bmatrix} \tag{2.29}$$

the initial values are given by:

$$\begin{aligned}\mathbf{x}_0 &= E[\mathbf{x}_0]\\ \mathbf{P}_0 &= E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)]\end{aligned} \tag{2.30}$$

Where P_0 is the initial value of the estimation error covariance P_k.

The a priori estimate can be stated as:

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= \mathbf{A}_k\hat{\mathbf{x}}_{k|k} + \mathbf{B}_k\mathbf{u}_k\\ \mathbf{P}_{k+1|k} &= \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T + \mathbf{K}_k\mathbf{Q}_k\mathbf{K}_k^T\end{aligned} \tag{2.31}$$

When the new measurement is available the estimate is filtered:

$$\begin{aligned}\mathbf{K}_F &= \mathbf{P}_{k|k-1}\mathbf{C}_k^T(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k)^{-1}\\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_F(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1})\\ \mathbf{P}_{k|k} &= [\mathbf{I} - \mathbf{K}_F\mathbf{C}_k]\mathbf{P}_{k|k-1}\end{aligned} \tag{2.32}$$

$\mathbf{K}_F$ is the kalman gain. When correctly tuned the kalman filter should give an accurate estimation of the state $\mathbf{x}_k$ given the output $\mathbf{y}_k$, this type of configuration is normally used to filter out noise.

The estimation step can be performed more often that the update, which leads to the well known Continuous-Discrete Kalman Filter (CDKF) [19]. This type of filter can provide a state estimation between samples, and in that way it may be more accurate than only relying on sampled values. It can also be used to increase the possible sampling interval without sacrificing accuracy, lastly the CDKF can compensate for missing or incorrect data, if the acquired sample is missing the filter can rely on its estimate and discard measured value.

2.4.1 Parameter estimation

Kalman filters can be used for both state and parameter estimation, when this is the case, the resulting problem is called a dual estimation problem.

The state space model of the system with unknown parameters can be stated as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}(\theta)\mathbf{x}_k + \mathbf{B}(\theta)\mathbf{u}_k + \mathbf{K}(\theta)\mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{H}(\theta)\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{2.33}$$

Where θ is a vector of unknown parameters to be estimated.

When using the Extended Kalman Filter for solving the dual problem, the state vector is augmented to include an estimate of the unknown parameters:

$$\begin{aligned} \xi_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \theta_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{A}(\theta_k)\mathbf{x}_k + \mathbf{B}(\theta_k)\mathbf{u}_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \mathbf{\Gamma}(\theta_k) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{w}_k \\ \mathbf{n}_k \end{bmatrix} \\ \mathbf{y}_k &= \mathbf{H}(\theta_k)\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{2.34}$$

Where $\mathbf{n}_k$ is white Gaussian noise of appropriate strength as to allow exploration of the parameter space [20].

Using KF to solve a dual problem will result in a non-linear augmented model, this can be seen from Equation 2.34: The next instance of states are dependent on a product of the states in the linear state space system and the estimated parameter, which is also a

state in the augmented system. Several extensions exist for handling non-linearities in Kalman filtering.

2.4.2 The Extended Kalman Filter (EKF)

The simplest form of non-linearities handling is simply linearizing the system around a certain point. This technique is used in the Extended Kalman Filter (EKF) which is known to be sub-optimal, but well proven method in practice.

A general non-linear system can be formulated as:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{2.35}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k) \tag{2.36}$$

Where $\mathbf{x}_k \in \mathbb{R}^n, \mathbf{u}_k \in \mathbb{R}^m$, and $\mathbf{x}_k \in \mathbb{R}^p$, $\mathbf{w}_k$ and $\mathbf{v}_k$ are defined as in Equation 2.28
The linearized system matrices are found by calculating the Jacobians:

$$\begin{aligned} \mathbf{\Phi}_k &= \left.\frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \\ \mathbf{\Gamma}_k &= \left.\frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\bar{\mathbf{w}}} \end{aligned} \tag{2.37}$$

Where $\mathbf{\Phi}_k$ is the linearized state matrix, $\mathbf{\Gamma}_k$ is the linearized plant noise matrix, $\bar{\mathbf{w}}$ is the mean value of the plant noise.
The prediction step is performed like in Equation 2.31 only now with the nonlinear model function $f(\cdot)$.

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, \bar{\mathbf{w}}) \tag{2.38}$$

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k \mathbf{P}_{k|k} \mathbf{\Phi}_k^T + \mathbf{\Lambda}_k \mathbf{Q}_k \mathbf{\Lambda}_k^T \tag{2.39}$$

Notice that the state estimate is made on the non-linear system, while the covariance matrix estimate is made on the linearized one. The rest of the Jacobians are found based

on the predicted state estimate $\hat{x}_{k+1|k}$:

$$\begin{aligned} \mathbf{\Sigma}_k &= \left.\frac{\partial h(\mathbf{x}, \mathbf{v})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} \\ \mathbf{\Lambda}_k &= \left.\frac{\partial h(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v}}\right|_{\mathbf{v}=\bar{\mathbf{v}}} \end{aligned} \tag{2.40}$$

Where $\mathbf{\Sigma}_k$ is the linearized output matrix, $\mathbf{\Lambda}_k$ is the linearized measurement noise matrix, $\bar{\mathbf{v}}$ is the mean value of the measurement noise.

If no non-linearities is present in a system matrix the partial derivative equals the system matrix, for instance: if no non-linearities is present in $h(\mathbf{x}, \mathbf{v})$ then

$$\begin{aligned} \mathbf{\Sigma}_k &= \left.\frac{\partial h(\mathbf{x}, \mathbf{v})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} = \mathbf{C}_k \\ \mathbf{\Lambda}_k &= \left.\frac{\partial h(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v}}\right|_{\mathbf{v}=\bar{\mathbf{v}}} = \mathbf{I} \end{aligned} \tag{2.41}$$

Finally the Kalman filtering step is applied as in Equation 2.32:

$$\begin{aligned} \mathbf{K}_F &= \mathbf{P}_{k|k-1}\mathbf{C}_k^T(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_F(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{P}_{k|k} &= [\mathbf{I} - \mathbf{K}_F\mathbf{C}_k]\mathbf{P}_{k|k-1} \end{aligned} \tag{2.42}$$

It is important to note that while Kalman filtering is optimal in the case of a linear system, the EKF is sub-optimal in the case of a non-linear system. This can still give adequate performance in the case when the nonlinearities are small, but when the system is highly nonlinear the performance is degraded.

The Jacobians needs to be calculated for every time step, this increases the computational load compared to normal Kalman filtering. Other methods for nonlinear system exist such as the Unscented Kalman Filter (UKF) and particle filtering. These subjects are eventually not considered in this thesis. While the UKF has a computational burden similar to the EKF [18], the particle filtering is a more complex algorithm [20, 21]. These methods may be investigated if the EKF does not perform in a satisfactory manner.

Chapter 3

State of the art of house thermal flow modeling

Modeling of thermal flows in houses has been subject to substantial research over the last two decades. Computer aided simulation tools for predicting the thermal behavior are many and well used by civil engineers and architects, these tools are called building Energy Performance Simulation Tools or BEPST Privara et al. [22], a comparison of available tools is presented in Crawley et al. [23].

In control engineering however the BEPST created models are too complex and therefore computationally expensive to implement as a part of a practical control system. Another problem is that these models are usually not explicit and therefore unsuitable for model based control. Anyway, BEPST has been used as a support for the explicit model used in control, in [22] it is suggested to use a simulation tool to provide well conditioned data for model identification.

While the emergence of model based control has provided the need for simplified models, the increase of computing power over the last twenty years have made more advanced models feasible. Still the complexity of the model is key to provide a precise enough model, while keeping runtime within the constrains of a real time implementation. The following is a short review of the different modeling methods mentioned in literature, for a comparison of these methods the reader is referred to Privara et al. [24].

3.1 Sub-space methods (4SID)

Sub-space methods are black-box estimation techniques that condition the input to output data by projecting or performing a decomposition of the system matrices and then estimate using least squares [25]. One method that is particularly useful in the modeling of buildings and has been implemented in Privara et al. [26] is the Subspace State Space System Identification or 4SID.

The 4SID algorithm can provide an estimation of the system order as well as matrices of the state space description Verhaegen and Dewilde [27]. Another apparent advantage is the state space representation, which makes it suitable for estimator design. The greatest disadvantage with 4SID and other black-box methods is that while the state space model may represent the input-output characteristics of the system, the explicit model structure may be different from the true system, so the connection to the physical system may be lost.

3.2 Prediction Error Methods (PEM)

Prediction error methods are statistically based identification techniques. Their objective is to minimize the error between the one step ahead prediction and the current value, this is done by updating the parameters of a predefined model structure [28]. Prediction error methods usually applies Auto Regressive Moving Average with eXogenous inputs (ARMAX) model structure. This structure and identification method has been successfully used as a room temperature model in Wu and Sun [29]. PEM models best accommodates SISO systems, this is impractical since buildings normally are MIMO in nature. Using a SISO-ARMAX model to represent a HVAC-system has been done by Yiu and Wang [30], here the authors concluded that multiple SISO-ARMAX models connected to represent each component had a lower performance than a MIMO-ARMAX model. The general PEM optimization problem can be represented as:

$$\hat{\theta} = \arg\min_{\theta} \sum_{k=1}^{N} \ell(\varepsilon_k(\theta)) \tag{3.1}$$

Where $\ell(\bullet)$ is an appropriate scalar function, θ is the vector of unknown parameters, ε_k is the prediction error at time k i.e. $\varepsilon_k = y_k - \hat{y}_k$, $\hat{y}_k$ is the estimate of y at time k.

A special case of PEM is the MPC relevant identification or MRI. Similar to PEM, MRI minimizes the error between current and predicted future values. The difference is that MRI uses a multi step prediction instead of one-step ahead. It is sensible when applying MPC that the number of time steps estimated is equal to the MPC prediction horizon, in this way the runtime of both algorithms may be reduced. The optimization problem is reformulated as:

$$\theta^* = \arg\min_{\hat{\theta}} \sum_{i=1}^{P} \sum_{k=0}^{N-i} (y_{k+i} - Z_{k+1,\hat{\theta}}\hat{\theta})^2 \tag{3.2}$$

Where $Z(q) = [u_{q-n_d}, \ldots, u_{q-n_b}, y_{q_1}, \ldots, y_{q-n_a}]$ with $q = k + i$ is the regressor, $\hat{\theta} = [\hat{b}_{n_d}, \ldots, \hat{b}_{n_b,\hat{a_1},\ldots,\hat{a_{n_a}}}]$ is the vector of estimated parameters, P is the number of steps used, N is the size of the dataset used for estimation, n_a and n_b respectively denotes the number of delayed inputs and outputs, n_d is the delay between the input and output,

The advantage of using a multi step prediction error is discussed in Zacekova et al. [12]. Using a case study the authors conclude that the MRI method gives better estimation result compared to a one-step ahead PEM faced with highly correlated data, an additional advantage is achieved when estimating using Partial Least Squares (PLS).

3.3 Deterministic Semi-Physical Modeling (DSPM)

By modeling the heat flow and capacity of an object as passive electrical components like resistors and capacitors a thermoelectric analogy can be derived. Furthermore several effects can be combined into the same passive components using the lumped capacitance method [5, Chapter 5]. Many papers suggest using this type of gray-box modeling technique and several leading research groups on optimal control such as UC Berkley [11], KU Leven [31], and ETH Zürich [32] are using some form of thermoelectric analogy. A discrete state space representation based on RC-networks can be formulated in many ways as described in Jiménez and Madsen [33].

The levels of complexity in the analogue models varies, in Fux et al. [9] a simple 1R1C model is used but the parameters and states are estimated on-line. This dual estimation problem is solved using the EKF. Fraisse et al. [34] uses a 3R4C model for each wall,

and calculate thermal resistances and capacitances from physical considerations. Furthermore several wall models are aggregated for simplification. The authors conclude that the resulting model can still be valid depending on the similarity of the thermal properties of the walls aggregated.

3.4 Probalistic Semi-Physical Modeling (PSPM)

This approach is based on a physical description of the system and represented by a set of stochastic differential equations. The parameters of the model are estimated through e.g. a Maximum Likelihood (ML) algorithm using the prior knowledge of the system parameters. The estimation can be represented mathematically as:

$$\theta^*_{ML} = \arg\max_{\theta}\{\ln(L(\theta, Y_1^N|y_0))\} \tag{3.3}$$

$$L(\theta, Y_1^N|y_0) = \prod_{k=1}^{N} \frac{exp(-\varepsilon_k^T R_{k|k-1}^{-1}\varepsilon_k/2)}{(\sqrt{2\pi})^l\sqrt{det(R_{k|k-1})}} p(y_0|\theta) \tag{3.4}$$

Where L is the likelihood function, Y_1^N is a vector of N measurements y_0 is the initial conditions, l is the dimension of the problem defined by the number of outputs, θ is the vector of unknown parameters, $p(y_0|\theta)$ is the probability of initial conditions on parameters, ε_k are residuals and $R_{k|k-1}$ is the residual covariance matrix,

This optimization problem can be solved with the EM-algorithm [35, 36]. Kristensen et al. [37] uses EKF to solve the ML problem recursively, the algorithm is implemented in a software tool named CTSM. PSPM is applied to a case study of a building in Bacher and Madsen [38] with promising results. The method is computationally heavy for complex models, so it is suggested application is to small scale models [24].

Chapter 4

Model design

This chapter uses heat transfer theory to derive a simple state space system for the thermal flow for a an apartment located in Trondheim, Norway. However the design presented is valid for a general building. The single order homogeneous differential equation 2.12 is expanded to include terms for all the walls connected to a given room. This can be seen as a network of RC-circuits with one resistor per wall and one capacitance per room. The electrical heating is seen as a power supply, providing constant power equal to the nominal power of electrical heating elements. The heat flow though the ceiling is ignored under the assumption that the temperature in the space above the room is equal to the room in an apartment building, if this is not likely for the building to be modeled it should either be included in the model or handled as a disturbance.

$$C_a\dot{Q} = C_a\frac{dT_a}{dt} = \frac{1}{R_{ab}}(T_b - T_a) + P_aK_hh \tag{4.1}$$

h is a variable defining if the heater is on or off:

$$h(t) = \begin{cases} 1 & \text{heater is on at time } t \\ 0 & \text{heater is off at time } t \end{cases} \tag{4.2}$$

Some parameters can be calculated from looking at the geometry of the building:

$$C_a\dot{T_a} = A_{ab}(T_b - T_a)U + P_aK_hh \tag{4.3}$$

Where U is a general conductance coefficient defined as $U = \frac{1}{R_T}$ that only depend on the wall materials, the capacitance can be calculated by assuming that it is dominated by the properties of air inside the room, therefore the heat capacity of other materials inside the room is ignored:

$$C_a \approx V_a C_v \tag{4.4}$$

Where C_v is the volumetric heat capacity of the air inside the room. The value of this can be found in a physics table for general properties for instance [5, Appendi A]. For air at $300K(27°C)$, atmospheric pressure and typical room humidity the volumetric heat capacity is found to be:

$$C_v = 1169 \left[\frac{J}{m^3 K}\right] \tag{4.5}$$

This property is dependent on density, so it will vary with temperature, this will cause the state space system to be time variant. At this stage the heat capacity is assumed constant at stated value.

$$\dot{T}_a = \frac{A_{ab}}{V_a C_v}(T_b - T_a)U + \frac{P_a}{V_a C_v} K_h h \tag{4.6}$$

The general differential equation can be stated as:

$$\dot{T}_i = \frac{1}{V_i C_v} \sum^{j \in \mathbb{C}_i} A_{ij}(T_j - T_i)U + P_i K_h h_i \tag{4.7}$$

Where $i \in \mathbb{N}(1:n)$, n is the number of states, $\mathbb{C}_i$ is the subset of indexes for temperatures connected to current temperature. If the walls are made from different materials or have different thickness U will vary, typically the value for U will be lower for outer walls, since they are usually better insulated.

It is important to observe that the outside temperature is not a state in the model, since the buildings inside temperature has little to no effect on the outside temperature. The outside temperature is therefore assumed to be an input. If there are other adjacent, measured temperatures that can not be considered a state, the should be modeled as an input.

The outside temperature can not be known a-priori, so the model will in practice have to rely on accurate forecast data for outside temperature input. The Norwegian Meteorological Institute offers a API for collecting forecast data for most places in Norway[39].

The basement temperature can not be acquired through forecast data, but as can be seen from data later in the thesis, the basement temperature varies a great deal less than other measured temperatures. It is therefore assumed to be constant over the simulation time window.

The state space model can be formulated as:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{K}\mathbf{v} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{w} \end{aligned} \tag{4.8}$$

System matrices are defined as:

$$\mathbf{A} = \begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,n} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n,1} & K_{n,2} & \cdots & K_{n,n} \end{bmatrix} \tag{4.9}$$

Where

$$K_{ij} = \begin{cases} U_I \frac{A_{ij}}{V_i C_v} & j \in \mathbb{C}_i \cup i \neq j \\ -U_I \frac{1}{V_i C_v} \sum^{j \in \mathbb{C}_i} A_{ij} - U_O \frac{1}{V_i C_v} \sum^{j \in \mathbb{E}} A_{iE} - U_B \frac{1}{V_i C_v} \sum^{j \in \mathbb{B}} A_{iB} & \text{if } i = j \\ 0 & j \notin \mathbb{C}_i \end{cases} \tag{4.10}$$

The function for the diagonal terms Equation 4.11 are explained next:

$$\mathbf{A}_{j,j} - U_I \frac{1}{V_i C_v} \sum^{j \in \mathbb{C}_i} A_{ij} - U_E \frac{1}{V_i C_v} \sum^{j \in \mathbb{E}} A_{iE} - U_B \frac{1}{V_i C_v} \sum^{j \in \mathbb{B}} A_{iB} \tag{4.11}$$

From Equation 4.7 it can be seen that all terms will include the state itself, so the connection with external temperature and others modeled as inputs will be included as a term in the diagonal of the **A**-matrix. Here the space $\mathbb{E}$ is the indexes of the external temperatures connected to state j, $\mathbb{O}$ is the indexes of the other temperatures connected to state j, U_E is the conductance though external walls and U_O is the conductance

through other building mass i.e. floor to basement.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & U_I \frac{A_{12}}{V_1 C_v} & \cdots & U_I \frac{A_{1n}}{V_1 C_v} \\ U_I \frac{A_{21}}{V_2 C_v} & \mathbf{A}_{2,2} & \cdots & U_I \frac{A_{2n}}{V_2 C_v} \\ \vdots & \vdots & \ddots & \vdots \\ U_I \frac{A_{n1}}{V_n C_v} & U_I \frac{A_{n2}}{V_n C_v} & \cdots & \mathbf{A}_{n,n} \end{bmatrix} \tag{4.12}$$

It is assumed that there is an electrical oven in each room and the measured temperatures in adjacent spaces, such as outside and basement, are modeled as inputs. The input matrix becomes:

$$\mathbf{B} = \begin{bmatrix} K_{h1} & 0 & \cdots & 0 & K_{o1} & K_{b1} \\ 0 & K_{h2} & \cdots & 0 & K_{o2} & K_{b2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & K_{hn} & K_{on} & K_{bn} \end{bmatrix} \tag{4.13}$$

As with the A-matrix certain easy measurable parameters can be taken from the adjacent temperature coefficients:

$$\mathbf{B} = \begin{bmatrix} \frac{P_1}{V_1 C_v} K_h & 0 & \cdots & 0 & U_E \frac{A_{1o}}{V_1 C_v} & U_B \frac{A_{1b}}{V_1 C_v} \\ 0 & \frac{P_2}{V_2 C_v} K_h & \cdots & 0 & U_E \frac{A_{2o}}{V_2 C_v} & U_B \frac{A_{2b}}{V_2 C_v} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \frac{P_n}{V_n C_v} K_h & U_E \frac{A_{no}}{V_n C_v} & U_B \frac{A_{nb}}{V_n C_v} \end{bmatrix} \tag{4.14}$$

Here U_E is transmittance through external walls and U_B is transmittance through the floor to the basement.

The plant noise vector $\mathbf{v}$ in Equation 4.8 is assumed to contain zero mean Gaussian white noise with known covariance:

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}^T \tag{4.15}$$

Where:

$$\mathbf{v} = v(t) = \mathcal{N}(0, \sigma_{vv}^2) \tag{4.16}$$

The plant noise vector is defined as:

$$\mathbf{K} = \begin{bmatrix} e_1 & 0 & \cdots & 0 \\ 0 & e_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_n \end{bmatrix} \tag{4.17}$$

The system matrix for the output function is defined as:

$$\mathbf{C} = \begin{bmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_n \end{bmatrix} \tag{4.18}$$

Where:

$$c_i = \begin{cases} c_i = 1 & \text{temperature } i \text{ is measured} \\ c_i = 0 & \text{temperature } i \text{ is not measured} \end{cases} \tag{4.19}$$

The measurement noise vector $\mathbf{w}$ from Equation 4.8 is also assumed to contain zero mean Gaussian white noise with known covariance:

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}^T \tag{4.20}$$

Where

$$w_i(t) = \mathcal{N}(0, \sigma_{ww}^2) \tag{4.21}$$

In a normal inhabited building there may be a number of disturbances that are not stochastic in nature and, there may be unmodeled heat sources and sinks like the inhabitants opening a window, or lighting fireplaces. These disturbances are also time variant in nature, so they need to be handled by some form of adaptive scheme if they are to be compensated. The EKF with state- and parameter estimation could provide such functionality and will be introduced shortly.

4.1 Discretization

By using Zero-Order-Hold as defined in 2.15 the discretized system is found:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}\mathbf{e}_\mathbf{k} \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{e}_\mathbf{k} \end{aligned} \tag{4.22}$$

4.2 Augmented model for EKF

When introducing an EKF with parameter estimation the state space model is adjusted in the following way:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{\Omega}(\theta_k)\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}\mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{4.23}$$

Where the augmented state matrix is defined as:

$$\mathbf{\Omega}(\theta_k) = \mathbf{A} + \mathrm{diag}(\theta_k) \tag{4.24}$$

The disturbance parameter vector is defined as:

$$\theta_k = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_n] \tag{4.25}$$

The system is extended to include the parameters to be estimated:

$$\begin{aligned} \xi_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \theta_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{\Omega}(\theta_k)\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \mathbf{K} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{w}_k \\ \mathbf{n}_k \end{bmatrix} \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{4.26}$$

Chapter 5

Implementation

This chapter covers the computer program used to calculate and test the estimator and model. In this early stage of implementation process, the program is just designed for testing the case study. However the classes and structure are designed to be general and portable. In a real time system the data will be a stream of sensor and meteorological data, for the case study previously collected sensor data is stored in files, and simulated offline.

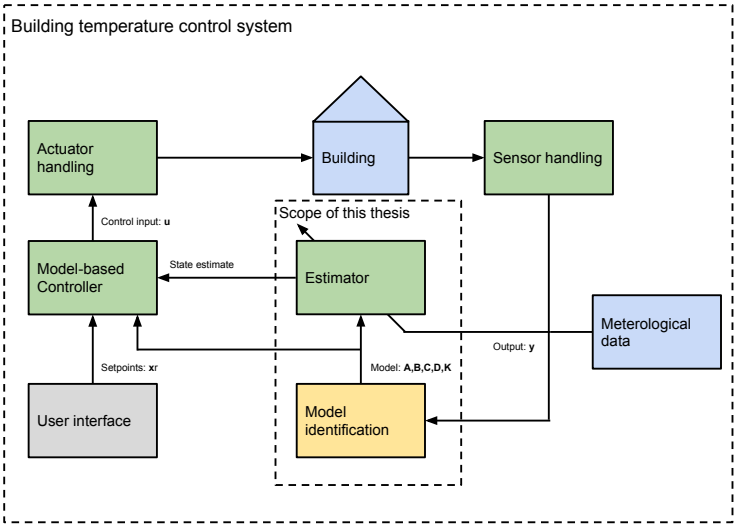

FIGURE 5.1: Diagram of complete control system

Operating system	**Processor**	**Clock frequency**	**RAM**
Windows 7 Enterprise 64-bit	Intel Core i5-2500	3.30GHz	8.0 Gb

TABLE 5.1: Detailed specifications of test computer

Figure 5.1 shows the over all structure of the control system proposed, the colors of blocks represents what the code could be running on. Green represents a real time system, most likely a SoC or micro-controller. Yellow is a Matlab-script that can run offline. Blue is external inputs from sensors and data, and grey is the user interface. In this thesis Matlab is used for the estimator, model and offline identification scheme. The computer specifications are summarized in Table 5.1.

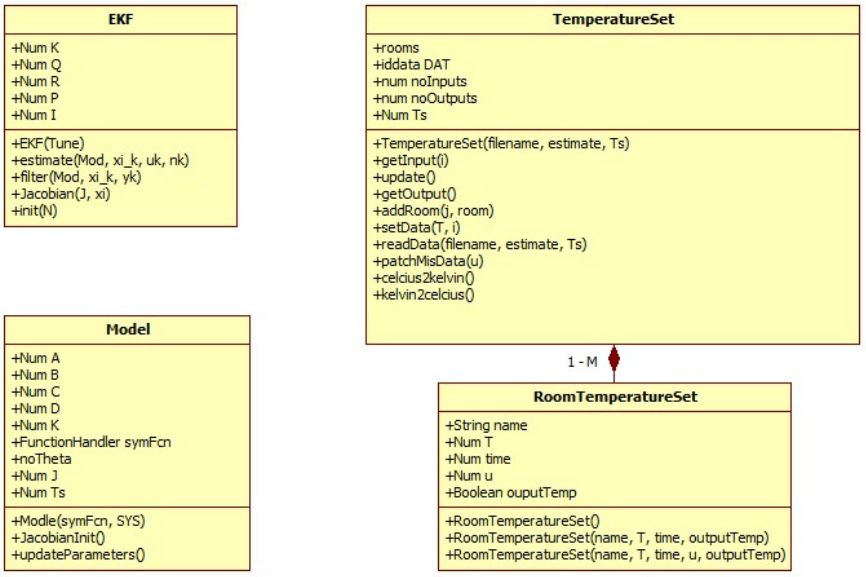

FIGURE 5.2: Class diagram

There are three main objects in the implementation, the rest of the code is support functions that is specific to the case study:

- **Model**
- **Kalman Filter**
- **Temperature dataset**

The Model object contains all information about the identified model, and can be constructed from a general state space system and a symbolic function defining the relationship between the augmented system matrices and the parameters to be estimated. This function is the basis for both updating the model with new parameter estimates, and calculating a matrix of Matlab-functions that define the Jacobian of the system.

The EKF-object contains functions to estimate and update the temperature estimate which are both mathematical functions based on the KF-theory. It also contain the covariance matrices (R, P, Q) and the Kalman Filter gain matrix K_f. The EKF-object has a function for calculating the numerical value of the Jacobian based on estimated state $\hat{\xi}$ and the Jacobian matrix contained in the Model-object.

The TemperatureSet-object contains data from all rooms in the house saved in a vector of RoomTemperatureSet-objects. The constructor extracts data based on the filename of

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Bathroom		Entrance		Guest bedroom		Living room		Main bedroom		Kitchen		Outside		Basement		Time
2	Input	Temp	Input	Temp	Input	Temp	Input	Temp	Input	Temp	Input	Temp	Input	Temp	Input	Temp	
3	0	20,5	0	18	0	17,3	0	21,7	1	20,6	0	19,7	6,4		11,8		07.04.2014 00:00
4	0	20,4	0	18	0	17,3	0	21,6	1	20,5	0	19,7	6,4		11,8		07.04.2014 00:10
5	0	20,4	0	17,9	0	17,2	0	21,5	1	20,4	0	19,6	6,2		11,8		07.04.2014 00:20
6	0	20,4	0	17,9	0	17,2	0	21,4	1	20,3	0	19,6	6,2		11,8		07.04.2014 00:30
7	0	20,4	0	17,9	0	17,2	0	21,3	1	20,2	0	19,6	6,1		11,8		07.04.2014 00:40
8	0	20,4	0	17,9	0	17,2	0	21,2	1	20,2	0	19,5	6,1		11,8		07.04.2014 00:50
9	0	20,4	0	17,8	0	17,2	0	21,1	1	20,1	0	19,5	6,1		11,8		07.04.2014 01:00
10	0	20,4	0	17,8	0	17,2	0	21	1	20	0	19,4	5,9		11,8		07.04.2014 01:10
11	0	20,4	0	17,8	0	17,2	0	20,9	1	19,9	0	19,4	5,9		11,8		07.04.2014 01:20
12	0	20,4	0	17,8	0	17,1	0	20,8	1	19,9	0	19,3	5,9		11,8		07.04.2014 01:30
13	0	20,4	0	17,8	0	17,2	0	20,7	1	19,9	0	19,3	5,9		11,8		07.04.2014 01:40
14	0	20,4	0	17,8	0	17,2	0	20,6	1	19,8	0	19,3	5,9		11,8		07.04.2014 01:50
15	0	20,4	0	17,8	0	17,1	0	20,5	1	19,7	0	19,2	5,8		11,8		07.04.2014 02:00
16	0	20,4	0	17,7	0	17,1	0	20,4	1	19,7	0	19,2	5,9		11,8		07.04.2014 02:10
17	0	20,4	0	17,7	0	17,1	0	20,3	1	19,6	0	19,2	5,9		11,8		07.04.2014 02:20
18	0	20,4	0	17,7	0	17,1	0	20,2	1	19,6	0	19,1	5,8		11,8		07.04.2014 02:30

FIGURE 5.3: Example of input datafile

a .mat-file with a matrix named "DATA" containing data for all sensors. The structure of the DATA-matrix is assumed to be:

$$DATA = \begin{bmatrix} u_1(t=1) & y_1(t=1) & u_2(t=1) & y_2(t=1) & \dots & u_M(t=1) & y_M(t=1) & t \\ u_1(t=2) & y_1(t=2) & u_2(t=2) & y_2(t=2) & \dots & u_M(t=2) & y_M(t=2) & t \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ u_1(t=N) & y_1(t=N) & u_2(t=N) & y_2(t=N) & \dots & u_M(t=N) & y_M(t=N) & t \end{bmatrix} \tag{5.1}$$

Where $u_i(t)$ is the *input* of room i at time t and $y_i(t)$ is the *output* of room i at time t t is a time and date object converted to Matlab-format. Figure 5.3 shows an example of the structure of a spreadsheet-file that will generate a correct DATA-matrix when imported.

The offline-estimation uses the Matlab `greyest()`-function which again uses a combination of system identification algorithms to find the parameters that best fit the data. The algorithms used in `greyest()` are[40]:

- Subspace Gauss-Newton direction
- Adaptive Gauss-Newton
- Levenberg-Marquardt -method
- Gradient method
- Trust region reflective method[1]

[1]Requires Optimization toolbox for Matlab, not installed on version used in case study

In this thesis the `greyest` is used in auto-mode, which means it tests all the stated methods, and uses the one which estimate best fit the data. For further reading on said methods the reader is referred to [28], and [41].

The Structure of the resulting model is defined by a Matlab-function `odefunction`, for instructions on how to generate a correct `odefunction` the reader is referred to [40].

Each experiment of the case study has its own script, Algorithm 2 shows the pseudocode of the scripts. All necessary files for running the experiments are included on the CD-ROM attached to the thesis.

Import data for *estimation* to `TemperatureSet`-instance
Import data for *simulation* to `TemperatureSet`-instance
Run offline estimation
if *EKF is used* **then**
  Load symbolic function handle `symFcn`
  Resample identified model
  Construct instance of Model-object
  Run Model-instance with EKF
else
  Run identified model
end
Plot results
Calculate eigenvalues, observability-matrix and controllability-matrix

Algorithm 2: Experiment pseudocode

For further details of implementation the reader is referred to Appendix A or see help-files using the Matlab `help`-command.

Chapter 6

Case study: Apartment in Trondheim

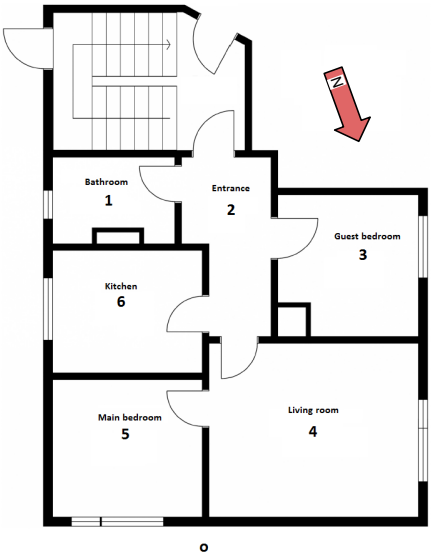

FIGURE 6.1: Drawing of apartment layout for case study

This chapter considers an apartment set-up with temperature sensors in each room, real data is collected over a period and the model results are compared to the measurements. The apartment is a typical Scandinavian wooden house with glass wool insulation in outer walls, and inner walls made of plaster and wood. In addition to the sensors in each room, sensors are placed outside and in the basement directly under the apartment. The apartment is heated mainly with electrical heaters, but there is also a fireplace placed on the living room wall adjacent to the guest bedroom. A portable heater is sometimes used in the kitchen. Due to practical concerns the heater status of the bathroom heater

is not measured, the model will have to handle it as a unknown disturbance. The sensors are placed in a manner that give the most accurate representation of the room temperature, with no direct solar radiation and not to close to unmeasured disturbances such as windows, doors and fireplace.

6.1 Model set-up

Recalling Equation 4.7, the apartment differential equations can be stated as:

$$\begin{aligned}
C_v V_1 \frac{dx_1}{dt} &= \left[A_{12}(x_2 - x_1) + A_{16}(x_6 - x_1)\right] U_i + A_{1o}(u_o - x_1)U_o + A_{1b}(u_b - x_1)U_b + P_1 U_h u_1 + E_1 v \\
C_v V_2 \frac{dx_2}{dt} &= \left[A_{12}(x_1 - x_2) + A_{23}(x_3 - x_2) + A_{24}(x_4 - x_2) + A_{26}(x_6 - x_2)\right] U_i \\
&\quad + A_{2b}(u_b - x_2)U_b + P_2 U_h u_2 + E_2 v \\
C_v V_3 \frac{dx_3}{dt} &= \left[A_{23}(x_2 - x_3) + A_{34}(x_4 - x_3)\right] U_i + A_{3o}(u_o - x_3)U_o + A_{3b}(u_b - x_3)U_b + P_3 U_h u_3 + E_3 v \\
C_v V_4 \frac{dx_4}{dt} &= \left[A_{24}(x_2 - x_4) + A_{34}(x_3 - x_4) + A_{45}(x_5 - x_4)\right] U_i + A_{4o}(u_o - x_4)U_o \\
&\quad + A_{4b}(u_b - x_4)U_b + P_4 U_h u_4 + E_4 v \\
C_v V_5 \frac{dx_5}{dt} &= \left[A_{45}(x_4 - x_5) + A_{56}(x_6 - x_5)\right] U_i + A_{5o}(u_o - x_5)U_o + A_{5b}(u_b - x_5)U_b + P_5 U_h u_5 + E_5 v \\
C_v V_6 \frac{dx_6}{dt} &= \left[A_{16}(x_1 - x_6) + A_{26}(x_2 - x_6) + A_{56}(x_5 - x_6)\right] U_i + A_{6o}(u_o - x_6)U_o \\
&\quad + A_{6b}(u_b - x_6)U_b + P_6 u_6 + E_6 v
\end{aligned} \tag{6.1}$$

On state space form:

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{K}\mathbf{w} \\
\mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{v}
\end{aligned} \tag{6.2}$$

$$\mathbf{A} = \begin{bmatrix}
-\frac{K_{11}}{C_v V_1} & \frac{A_{12}}{C_v V_1} & 0 & 0 & 0 & \frac{A_{16}}{C_v V_1} \\
\frac{A_{12}}{C_v V_2} & -\frac{K_{22}}{C_v V_2} & \frac{A_{23}}{C_v V_2} & \frac{A_{24}}{C_v V_2} & 0 & \frac{A_{26}}{V_2} \\
0 & \frac{A_{23}}{C_v V_3} & -\frac{K_{33}}{C_v V_3} & \frac{A_{34}}{C_v V_3} & 0 & 0 \\
0 & \frac{A_{24}}{C_v V_4} & \frac{A_{34}}{C_v V_4} & -\frac{K_{44}}{C_v V_4} & \frac{A_{45}}{C_v V_4} & 0 \\
0 & 0 & 0 & \frac{A_{45}}{C_v V_5} & -\frac{K_{55}}{C_v V_5} & \frac{A_{56}}{C_v V_5} \\
\frac{A_{16}}{C_v V_6} & \frac{A_{26}}{C_v V_6} & 0 & 0 & \frac{A_{56}}{C_v V_6} & -\frac{K_{66}}{C_v V_6}
\end{bmatrix} \tag{6.3}$$

$$
\begin{aligned}
K_{11} &= [A_{12} + A_{16}]\, U_i + A_{1o}U_o + A_{1b}U_b \\
K_{22} &= [A_{11} + A_{23} + A_{24} + A_{26}]\, U_i + A_{2b}U_b \\
K_{33} &= [A_{23} + A_{34}]\, U_i + A_{3o} + A_{3b}) \\
K_{44} &= [A_{24} + A_{34} + A_{45}]\, U_i + A_{4o}U_o + A_{4b}U_b \\
K_{55} &= [A_{45} + A_{56}]\, U_i + A_{5o}U_o + A_{5b}U_b \\
K_{66} &= [A_{16} + A_{26} + A_{56}]\, U_i + A_{6o}U_o + A_{6b}U_b
\end{aligned} \tag{6.4}
$$

$$
\mathbf{B} = \begin{bmatrix}
\frac{P_1}{C_v V_1} K_h & 0 & 0 & 0 & 0 & 0 & \frac{A_{1o}}{C_v V_1} U_o & \frac{A_{1b}}{C_v V_1} U_b \\
0 & \frac{P_2}{C_v V_2} K_h & 0 & 0 & 0 & 0 & 0 & \frac{A_{2b}}{C_v V_1} U_b \\
0 & 0 & \frac{P_3}{C_v V_3} K_h & 0 & 0 & 0 & \frac{A_{3o}}{C_v V_3} U_o & \frac{A_{3b}}{C_v V_1} U_b \\
0 & 0 & 0 & \frac{P_4}{C_v V_4} K_h & 0 & 0 & \frac{A_{4o}}{C_v V_4} U_o & \frac{A_{4b}}{C_v V_1} U_b \\
0 & 0 & 0 & 0 & \frac{P_5}{C_v V_5} K_h & 0 & \frac{A_{5o}}{C_v V_5} U_o & \frac{A_{5b}}{C_v V_1} U_b \\
0 & 0 & 0 & 0 & 0 & \frac{P_6}{C_v V_6} K_h & \frac{A_{6o}}{C_v V_6} U_o & \frac{A_{6b}}{C_v V_1} U_b
\end{bmatrix} \tag{6.5}
$$

$$
\mathbf{C} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{6.6}
$$

$$
\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \end{bmatrix}^T \tag{6.7}
$$

$$
\mathbf{K} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{6.8}
$$

6.2 Building description

This section gives a description of the building in the case study. Area and building components of the buildings infrastructure are covered in detail and are necessary to derive calculated values for coefficients in the model.

6.2.1 Areas and volume

The areas and volume of the respective spaces are calculated from a 3D drawing of the apartment, all doors and windows are considered closed and part of the total wall area:

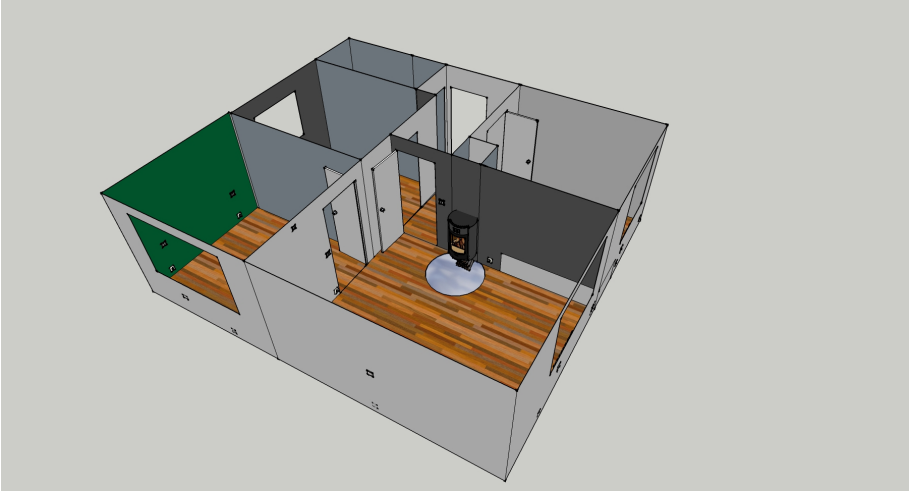

FIGURE 6.2: 3D-drawing of apartment

Name	**1**	**2**	**3**	**4**	**5**	**6**
1 Bathroom	-	2.904	0	0	0	6.24
2 Entrance	2.904	-	5.256	3.12	0	3.912
3 Guest bedroom	0	5.256	-	5.904	0	0
4 Living room	0	3.12	5.904	-	6.816	2.184
5 Main bedroom	0	0	0	6.816	-	7.392
6 Kitchen	6.24	3.912	0	2.184	7.392	-

TABLE 6.1: Wall areas between rooms $[m^2]$

The air volume for each room is known, as well as areas of external walls and area of floor that connects to the basement.

Name	**Floor area** $[m^2]$	**Volume** $[m^3]$	**External wall area** $[m^2]$
1 Bathroom	3.1691	7.6058	2.904
2 Entrance	4.2060	10.0944	-
3 Guest bedroom	8.6336	20.7206	6.81
4 Living room	16.275	39.0600	19.416
5 Main bedroom	8.8892	21.3341	14.328
6 Kitchen	7.8232	18.7757	6.0972

TABLE 6.2: Air volume and external areas

The nominal power for each room is given in Table 6.3, the bathroom has integrated floor heating with unknown power, a portable heater of 600W is used in the kitchen, which is equal to the kitchen heater power, the model doubles power in the kitchen when the portable heater is used.

Name	**Heater power** $[W]$
1 Bathroom	-
2 Entrance	600
3 Guest bedroom	600
4 Living room	1000
5 Main bedroom	800
6 Kitchen	600/1200

TABLE 6.3: Heater power $[W]$

6.2.2 Building components

How the different building components are comprised in this apartment is only partly known, some of it is based on known fact due to recent construction work and some of it is assumed with basis in common Scandinavian construction-practices. All values for thermal resistance are taken from [7] except for the Glava-insulation which is taken from [42]. The thermal transmittance of the air film close to the wall is calculated in accordance to [6, Annex A]

Average outside wind velocities from the last year in Trondheim is calculated from [43] and found to be $\bar{v} = 2,74m/s \approx 3m/s$ to match tables in cited standard. The average normal temperature is found to be $\bar{T} = 4,95°C \approx 5°C$. Values used for calculating surface resistance is found in Table 6.4. Note that the values for wind and temperature may vary greatly from this in the case study, but this simplification is needed to generate a time invariant model.

The resulting total transmittance is calculated:

	$\bar{T}$ $[°C]$	$\bar{\nu}$ $[m/s]$	h_r $[W/m^2K^1]$	h_c $[W/m^2K]$	R_s $[m^2K/W]$
Inside	20	-	5.13	2.5	0.131
Outside	5	3	4.365	4.2	0.117

TABLE 6.4: Values used for calculating surface thermal transmittance

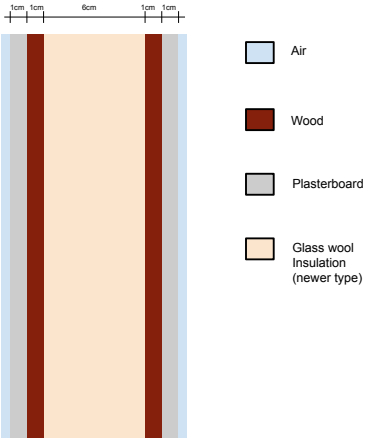

FIGURE 6.3: Cross-section of inner walls

Component	$\lambda[W/mK]$	$d[m]$	$R[m^2K/W]$	n
Surface effect	-	-	0.131	2
Plasterboard	0.21	0.01	0.048	2
Wood	0.14	0.01	0.071	2
Insulation (Glava 35)	0.035	0.06	1.579	1
Sum			2.214	

TABLE 6.5: Values for thermal resistance in inner walls

Component	$\lambda[W/mK]$	$d[m]$	$R[m^2K/W]$	n
Surface effect (inside)	-	-	0.131	1
Surface effect (outside)	-	-	0.117	1
Plasterboard	0.21	0.01	0.048	1
Wood 1	0.014	0.01	0.071	2
Air	0.12	0.013	0.1	1
Insulation (Older type)	0.04	0.1	2.5	1
Wood 2	0.14	0.03	0.214	1
sum			3.252	

TABLE 6.6: Values for thermal resistance in outer walls

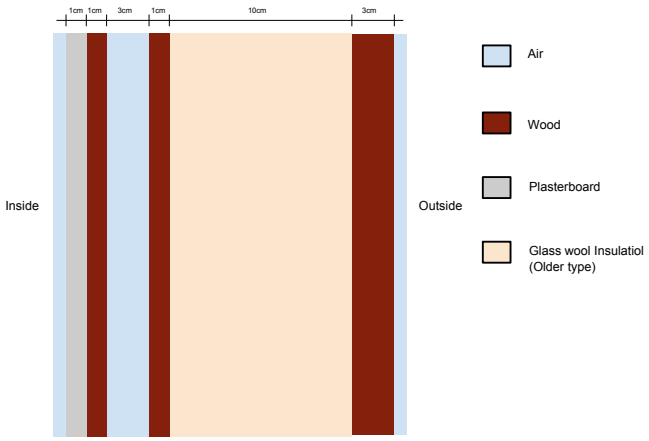

FIGURE 6.4: Cross-section of outer walls

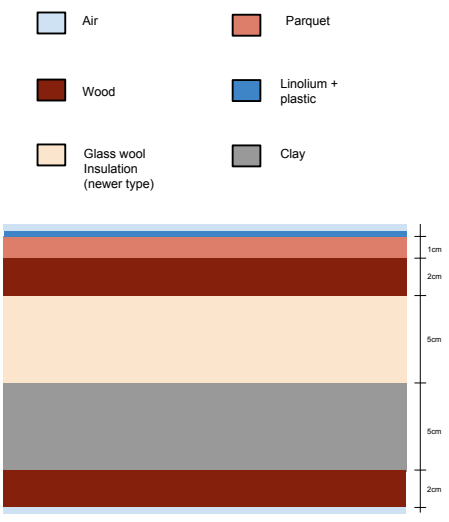

FIGURE 6.5: Cross-section of floor

Component	$\lambda[W/mK]$	$d[m]$	$R[m^2K/W]$	n
Surface effect (over)	-	-	0.172	1
Surface effect (under)	-	-	0.099	1
Linoleum	0.17	0.002	0.012	1
Parquet	0.23	0.002	0.009	1
Wood	0.14	0.02	0.143	2
Insulation (Glava 35)	0.035	0.05	1.429	1
Clay	1.5	0.05	0.033	1
sum			2.083	

TABLE 6.7: Values for thermal resistance in floors

Component	Inner walls	Outer walls	Floor
Thermal transmittance $[W/m^2K]$	0.0452	0.491	0.307

TABLE 6.8: Transmittances for building components

6.3 Experimental set-up

The equipment used for the case study is listed in Table 6.9.

Function	**Name**	**Producer**	**Precision**
Temperature sensor	TSS320	Proove AB	$\pm 1°C$
Wireless interface	Tellstic Duo	Telldus Technologies	-
SOC	Raspberry PI	Farnell	-

TABLE 6.9: Experiment equipment list

The temperature-sensors collect temperature in all rooms, in the basement and directly outside the building, while the sensors for heaters only collect a boolean that evaluate true if the heater is energized. The set-points for all heaters are unknown and heater dynamics is effectively ignored in the offline identified model.

The Raspberry Pi collects samples from the sensors and uploads the data to a Dropbox-folder via an SQLite interface. The data is saved as a Comma Separated Value file(.csv) which in turn is exported into an excel-spreadsheet, the spreadsheet also handles the portable heater by summarizing the contribution from kitchen and portable heater.

The sensors take measurements every ten minutes, it is assumed that they are synchronized or that any time between samples of the different sensor is of little importance in the broader picture. This can be reasoned from the fact that temperature changes happens slowly relative to the sampling time of ten minutes.

It is observed that some measurements are missing in the data, the sensors seem to skip collecting samples from time to time. In the spreadsheet-file missing data is handled by leaving the cell empty, this again will be interpreted by Matlab as NaN-values. The offline estimation does not handle NaN-values, so when offline estimation is done the values are patched by a sample and hold function, see Appendix A for details. When EKF is applied the missing data is estimated by the filter. Please note that when one of the inputs evaluates as NaN the whole vector of inputs collected at this time must be discarded.

Two different offline schemes are considered:

- Estimating multiple parameters with grey-box estimation scheme

- Calculating each wall resistance coefficient based on physical tables

The model is tested with and without applying the EKF with state and parameter estimation. All experiments are simulated in Kelvin and plotted in Celsius for easier reference.

Chapter 7

Results and discussion

In this chapter a discussion of the results from the case study and comparison of simulated model with data is given. Possible explanations for the results observed are stated, and comments on performance are made.

Data used for offline estimation is based on a regular day, with inhabitants in the building. This is expected to create unknown disturbances, it follows that careful considerations needs to be made when selecting dataset for estimation, to prevent large disturbances being present.

7.1 Selection of dataset for estimation

This section displays the proposed dataset chosen for offline estimation and motivates why this time period was chosen.

Firstly the effect of heater dynamics can be seen clearly in Figure 7.1. It can be observed that a heater is turned in at the end of the day, the temperature can be seen as rising until a certain point, and then it ripples around a constant temperature value. The heater input however is seen as "on" for the whole period of the night. This type of dynamic can be explained by a heater controller, turning on and off the heater using hysteresis control. It can be seen that over different days the temperature in the room ripples around different values, this might be explained by the heater thermostat value being

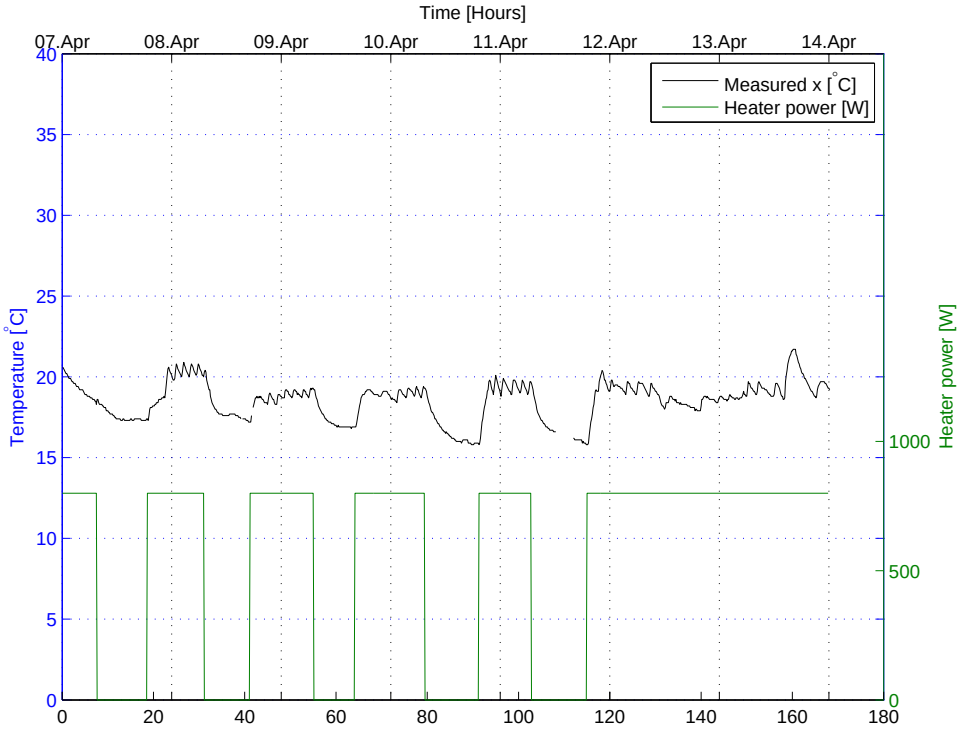

FIGURE 7.1: Selection of dataset for estimation: Room 5: Main bedroom

changed between different days. This effect seems to be consistent over a number of days, except for the weekend, when temperatures are generally higher. This is probably because the inhabitants are absent during the day on weekdays.

One disturbance that presumably will have a large effect is the fireplace in the living room. The plot from the living room can be seen in Figure 7.2
It can be observed that on the evening of Friday the 11th of April the temperature rises higher than the other days, and starts to fall during the night until seemingly stabilizing at a temperature equal to previous nights. During the weekend the heater seems to be on at all times, but still the temperature does not rise as much as it did on the evening in question. This may be explained by the fireplace. Choosing a dataset for estimation that includes said Friday will in other words not be recommended.

In the kitchen there are a number of possible disturbances, mainly from cooking and cleaning appliances, this can be seen from Figure 7.3. Each day there is a peak temperature in the morning corresponding with heater input, notice that two heaters are applied in the morning, and therefore the power is doubled. But there is also a second peak that does not seem to correspond to heater input, some time around four to six.

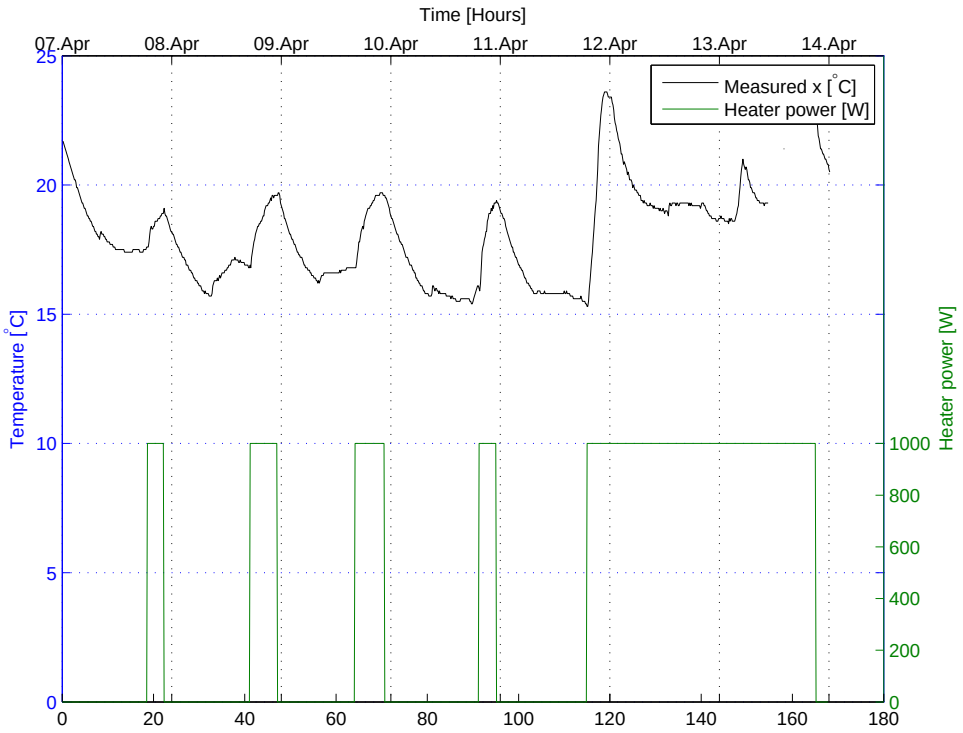

FIGURE 7.2: Selection of dataset for estimation: Room 4: Living room

This may be caused by cooking or other kitchen appliances being used. This effect seem to be present every day

The entrance is connected to most rooms in the model, and also has doors with unknown status. Open doors between rooms will cause a disturbance, since it enhances the connection between them greatly. There is also has a direct connection and door to the stairway outside the apartment, where the temperature is not measured. If this door is left open the disturbance may be great. This can be seen in Figure 7.4. By Examining the difference between 9th and 10th of April we see a discrepancy between heater input and room temperature, this may be caused by doors being left open the first day. It can be seen from the previous plots and the rest of the plots in section B.1 that the temperature outside and in the other rooms are relatively alike over these two days, thus it is reasonable to believe that this is caused by the outer door being left open.

The Guest bedroom heater is kept off during the whole week, so heater dynamics and dispersion effect can not be estimated for this room. As previously stated the bathroom heater status is unknown, and therefore it is expected to see error in the simulation

Time	**Day 1: 09.04**	**Day 2: 10.04**	**Day 3: 11.04**
Direction	South-South-East	South	West-South-West
Wind speed $[m/s]$	4.1	5.3	6.5

TABLE 7.1: Average wind speed and direction over the three days of testing

of the bathroom. In the rest of the plots seen in section B.1 large disturbances is not explicitly observed.

In this dataset the day where the smallest effect of disturbances are observed is the 9th if April, so this is the day chosen to be the dataset for offline estimation. Finding a day where the disturbance conditions are alike is almost impossible in an inhabited building, but the one that seems most alike is 9th of April where every room seem to have the same response except for the entrance. A better conditioned dataset could probably improve the results seen here, but it does not seem to be available at this time.

The weather in Trondheim these three days was mostly overcast, light wind with slight rain. A plot of wind speeds is included in Figure 7.5. As previously stated the outer wall transmittance is calculated under using the average wind speed for Trondheim over a whole year. It can be seen that days selected for simulation and estimation differs from these values. The average wind speed and direction is stated in Table 7.1.

All meteorological data is taken from [43] where the measurements are being made at Voll measuring station approximately 3 km in direct line from the apartment used in the case study. Data is not used in the model directly, but is included to help explain possible sources for model error.

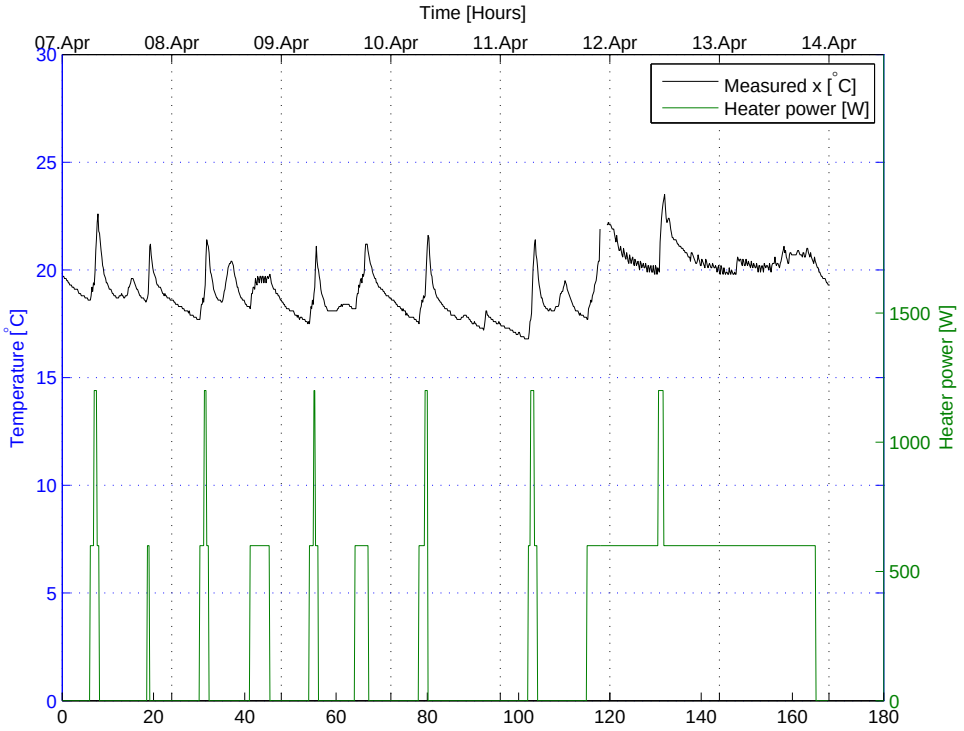

FIGURE 7.3: Selection of dataset for estimation: Room 6: Kitchen

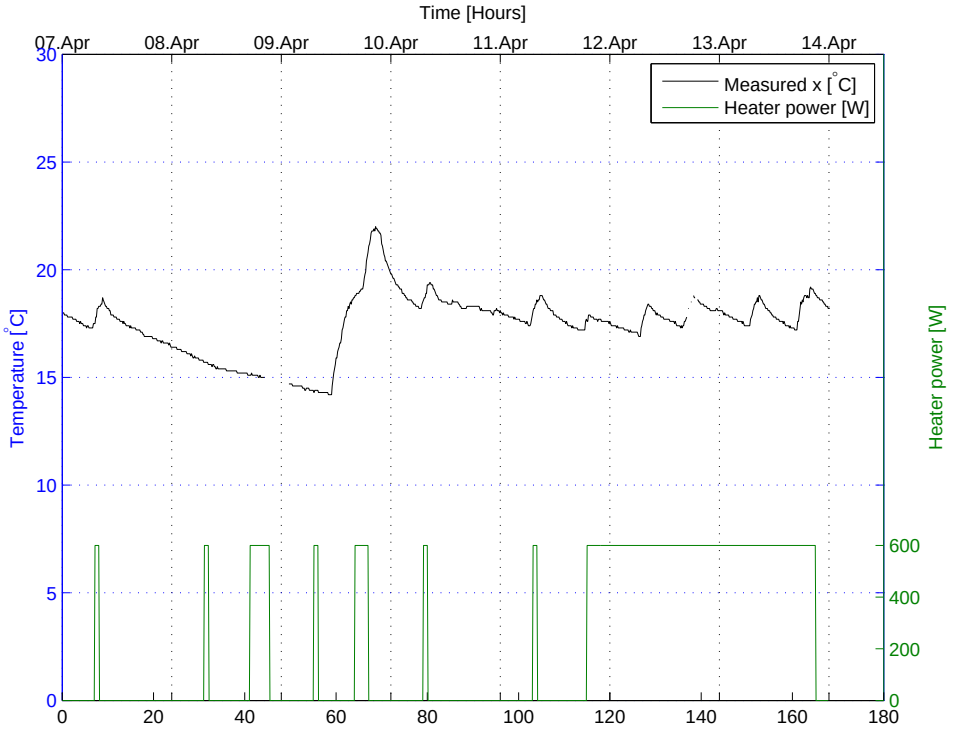

FIGURE 7.4: Selection of dataset for estimation: Room 2: Entrance

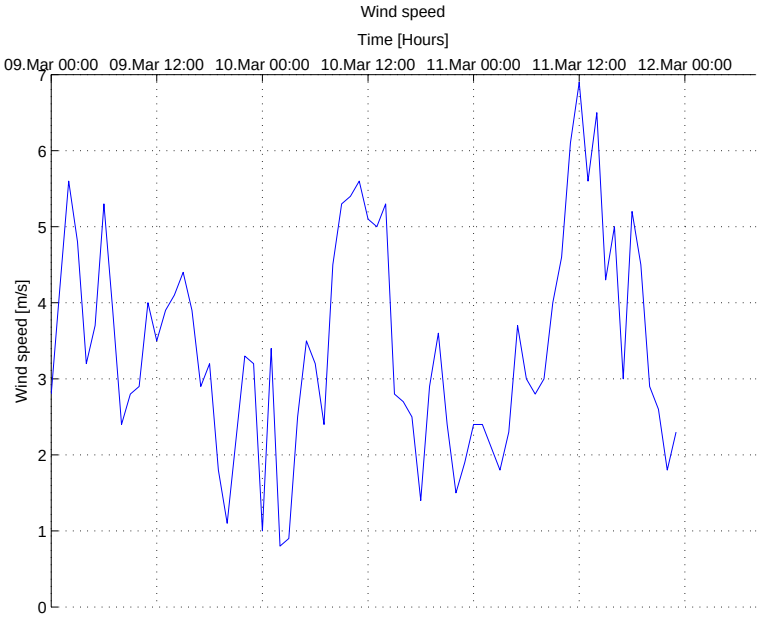

FIGURE 7.5: Selection of dataset for estimation: Wind speed

7.2 Model testing

7.2.1 Experiment 1: Grey-box estimation with 4 parameters

The first experiment is based on knowledge of the model structure, but nothing else. All coefficients are assumed unknown and need to be estimated. The offline estimation is run on data from 9th of April and verified against data from the 10th and 11th of April. The first day the model is verified against the same data used in the estimation. This is to provide a benchmark for each experiment to see how well this best case scenario fits the data.
For the estimation and model comparison missing data is patched using a sample and hold function, details in Appendix A.

The parameters to be estimated in this experiment are the temperature transmittance of inner, outer walls, floor and a general heating coefficient: $U_i, U_o U_b$ and K_h, respectively.

The values generated by estimation is stated in Table 7.2:

Parameter	U_i	U_o	U_b	K_h
Value	0.0007964	0.09956	0	0.004528

TABLE 7.2: Experiment 1: Grey-box estimation values

It is observed that the values are small and far from the previously calculated values seen in Table 6.8. U_b is zero, meaning the estimation calculated that improved results would be had if it where negative. This suggests that the model can not be accurately estimated with this simplified structure.

The eigenvalues of the model is calculated:

$$|\Lambda| = \begin{bmatrix} 0.9994 & 0.9963 & 0.9980 & 0.9981 & 0.9975 & 0.9973 \end{bmatrix} \tag{7.1}$$

All eigenvalues are slightly smaller than 1, thus this model is barely stable.

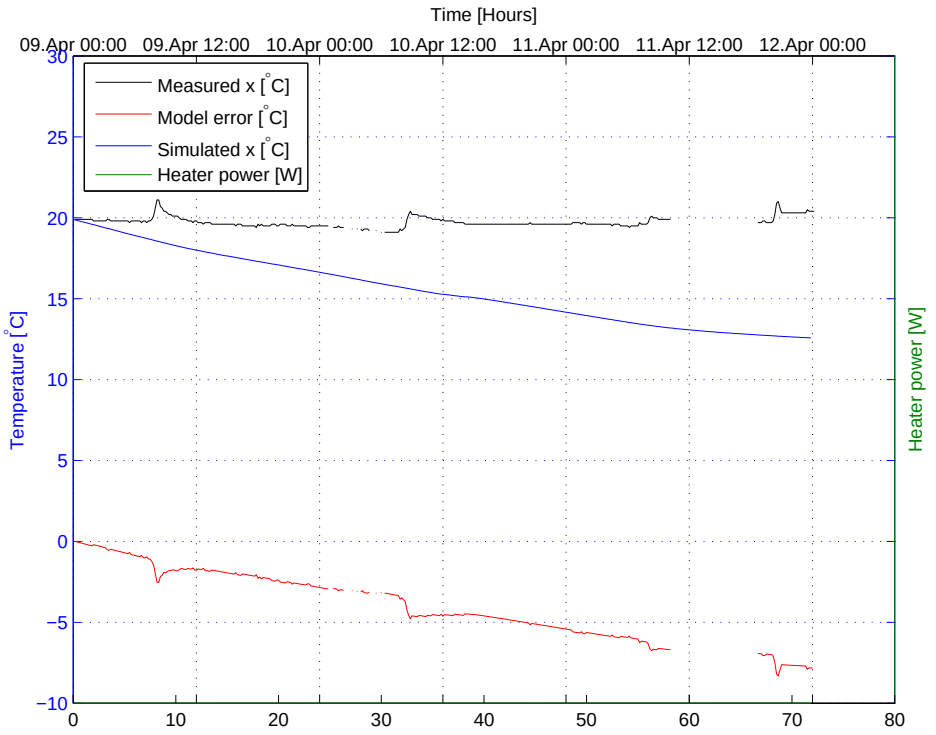

FIGURE 7.6: Experiment 1: Room 1: Bathroom

From examining the plot of the bathroom in Figure 7.6 it can be seen that model performance for this room is rather poor. The temperature in the model seems to fall under the measured values, and is steadily falling throughout the simulation. This could be due to lack of measurement of heater input in this room.

The worst performance of this model can be seen in the plot of the main bedroom Figure 7.7. The model overshots even in the benchmark period, and it can be seen that it has a steady temperature gradient, going up or down depending on whether the heater is active or not. This suggests that the model is too simple to follow the dynamics in this room, and could also have something to do with the observed heater dynamics, since it only seem to occur in this room where heater dynamics are apparent. The estimation can not react to both the general large dynamics of the daily temperature changes in addition to the smaller changes that the heater controller contributes. This is in fact poorly conditioned data for this type of estimation.

It can be seen from the plot in the kitchen Figure 7.8 that the model does not seem to follow rapid dynamics, but rather follows a line where mean error is smallest. This is because of the identification scheme itself. All methods used in offline estimation choose

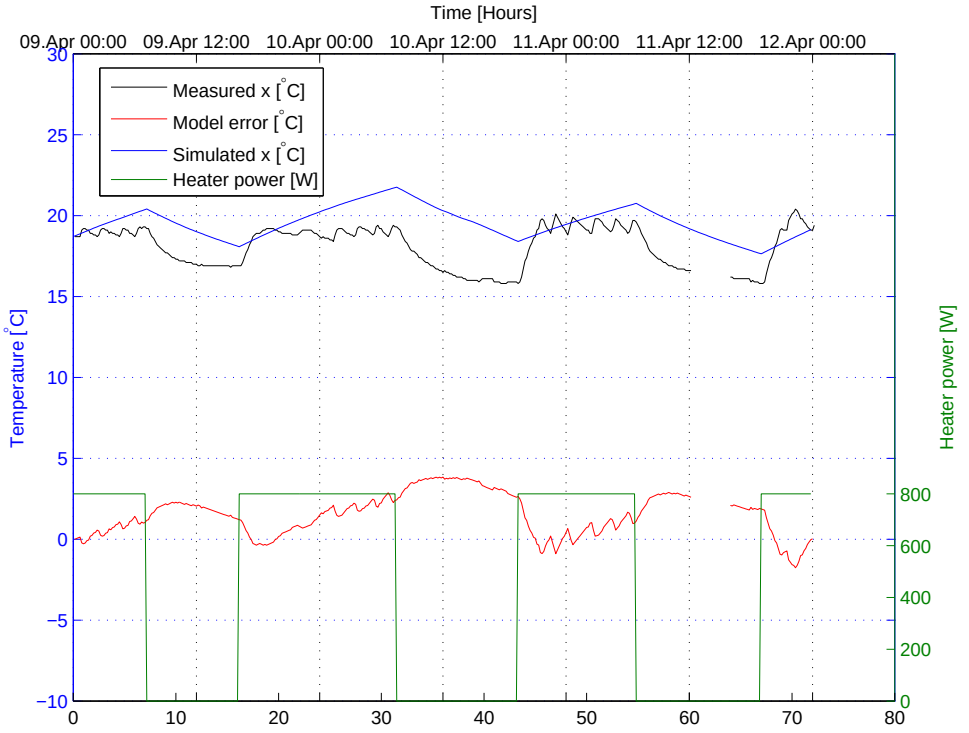

FIGURE 7.7: Experiment 1: Room 5: Main bedroom

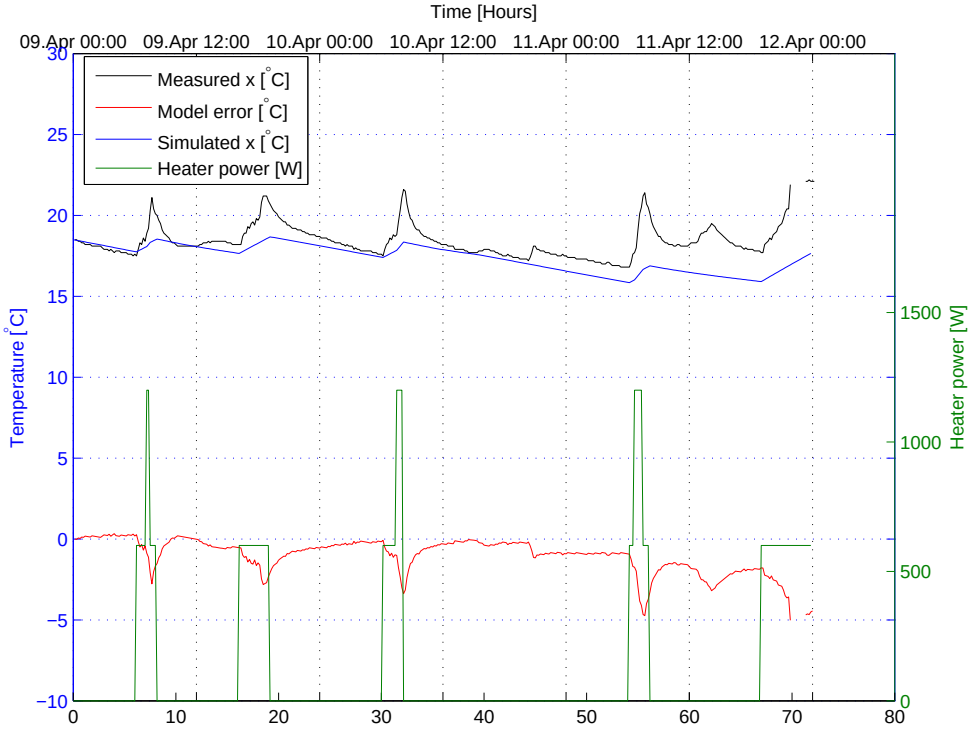

FIGURE 7.8: Experiment 1: Room 6: Kitchen

values whose error is the smallest, it could be that with this complex dataset and simple model it is simply not feasible to estimate it correctly without having more knowledge of the physical system. Same tendency can be observed for the remaining plots that can be seen in section B.2.

The results shown in this test suggest that the model structure is to simplified to produce a proper model with estimation alone. The parameters shown in Table 7.2 differs a lot from the values calculated in section 6.2 for thermal transmittance. It could be that the model needs some form of physical anchoring to provide an accurate simulation, next the calculated physical constants is included to see if the results can be improved.

7.2.2 Experiment 2: Using calculated values for transmittance

The second experiment uses the values for the transmittances from Table 6.8, but with no calculated value for the heater-coefficient K_h it will have to be estimated:

The variable estimated is shown in Table 7.3.

Parameter	K_h
Value	0.02013

TABLE 7.3: Experiment 2: Grey-box estimation value

The eigenvalues of the model is calculated:

$$|\Lambda| = \begin{bmatrix} 0.8400 & 0.7556 & 0.7092 & 0.6074 & 0.4451 & 0.4734 \end{bmatrix} \tag{7.2}$$

Unlike the previous experiment these values are further away from 1, which means the stability margin is larger for this model.

This model seems to better react to rapid changes in the temperature, this can be seen from Figure 7.9.

The peaks generated by heater activity is better represented in the model. It it observed however that a large bias is present and this causes large errors in the model compared to data.

The bias can especially be seen in the plots from the Guest bedroom Figure 7.10. Here it is observed that the bias is approximately $25°C$. This could be due to differences in wall materials and effect from windows. In the other rooms similar bias effects are seen, see section B.3 for plots of all rooms.

By calculating a general temperature transmittance for all outer walls the effect of windows are ignored. It is expected that windows will have a large effect on the temperature flow from inside rooms to outside, so effort needs to be made to compensate for differences in walls. The proposed solution is to estimate a coefficient relating the wall transmittance to the window effect in the respective room. A different approach could

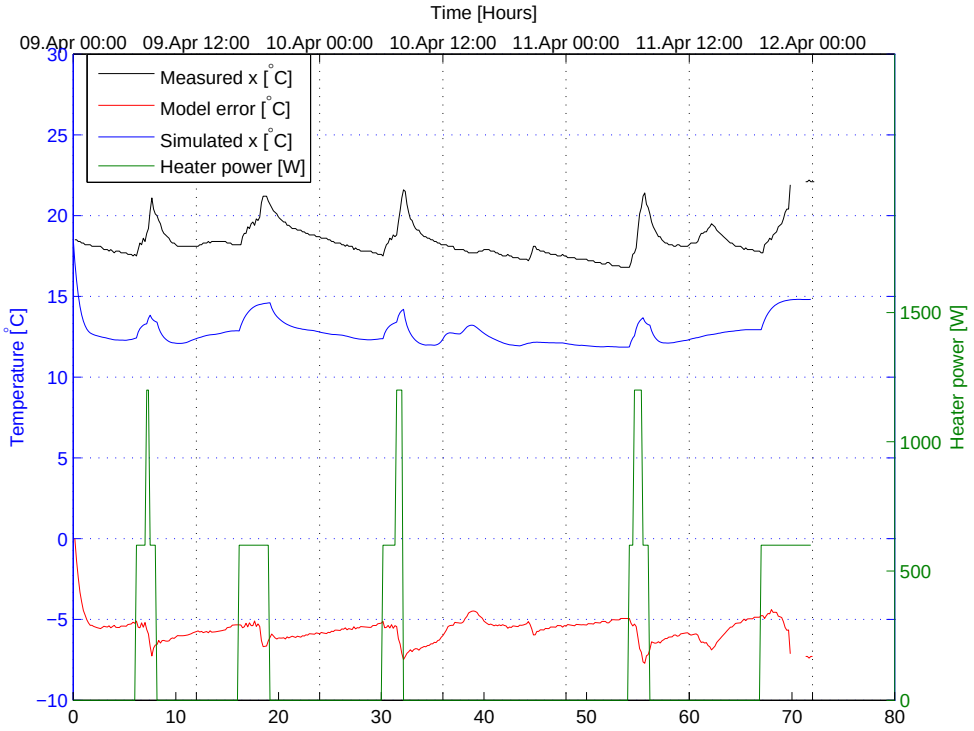

FIGURE 7.9: Experiment 2: Room 6: Kitchen

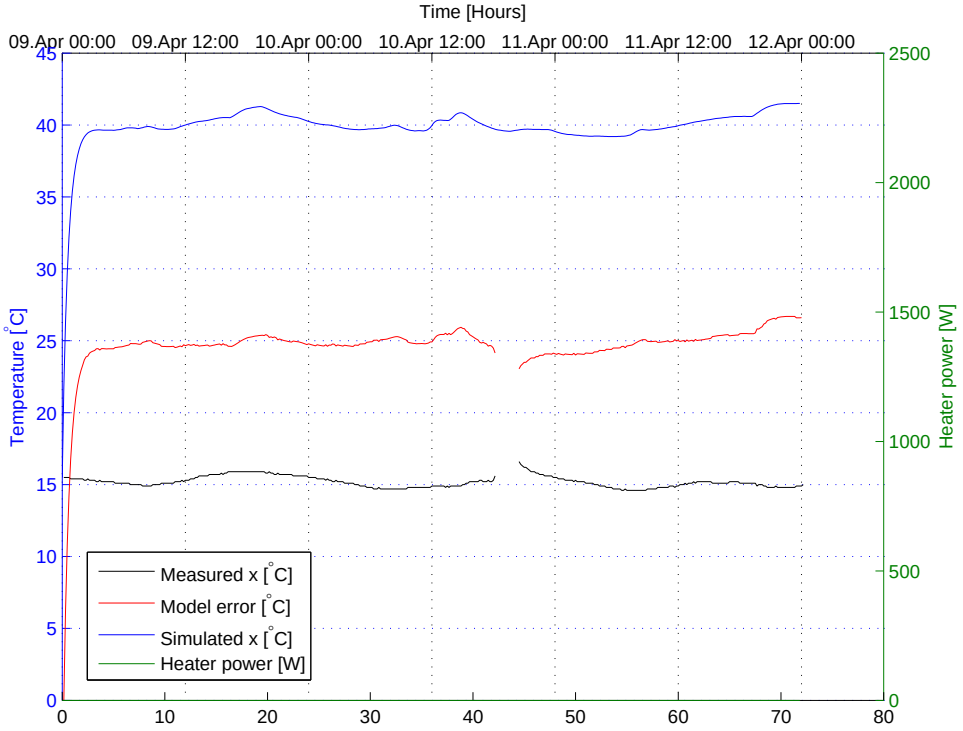

FIGURE 7.10: Experiment 2: Room 3: Guest bedroom

be to calculate transmittance for windows, and include them in the model. But seeing how wind and solar effect on windows may differ greatly depending which way they are facing the window-effect may have to be estimated regardless.

7.2.3 Experiment 3: Compensating for differences in external walls

In an effort to compensate for the large bias observed in the previous experiment, more parameters are included in the offline estimation. The proposed experiment estimates coefficients to allow different transmittance in outer walls. The new transmittance is stated as:

$$U_{oi} = U_o K_i \tag{7.3}$$

Where U_{oi} is the transmittance in outer wall for room i, U_o is the calculated transmittance for outer walls, and K_i is the parameters to be estimated.

In this experiment the following parameters are estimated: $K_h, K_1, K_3, K_4, K_5, K_6$.

Parameter	K_h	K_1	K_3	K_4	K_5	K_6
Value	0.01925	1.126	0.5638	1.065	1.057	1.089

TABLE 7.4: Experiment 3: Grey-box estimation values

The eigenvalues of the model is calculated:

$$|\Lambda| = \begin{bmatrix} 0.8400 & 0.7556 & 0.7092 & 0.6074 & 0.4451 & 0.4734 \end{bmatrix} \tag{7.4}$$

From the plot of the living room temperature Figure 7.11 a number of effects can be seen. No bias-effect is observed, but the model seems to react too quickly in relation to data, even in the benchmark-period of the first day this effect is seen. This could be because of the simplification of the model itself. The model assumes that the temperature is equal in the whole volume of the room, when in reality the energy introduced in the system needs more time to heat the whole volume to equal temperature. There is an internal heat flow dynamics that is not modeled, similar to the previously mentioned dispersion effect in heaters. The fact that this effect can best be seen in the living room where the air volume is the largest further asserts this argument. For very large rooms this effect could be substantial, if this is the case, then it should be considered to split the large room into multiple states.

At the end of Friday the 11th of April the measured temperature rises steeply, but the model does not follow, as previously stated it seems like the fireplace was in use on that Friday. The use of the fireplace would explain the large error in the model at this time.

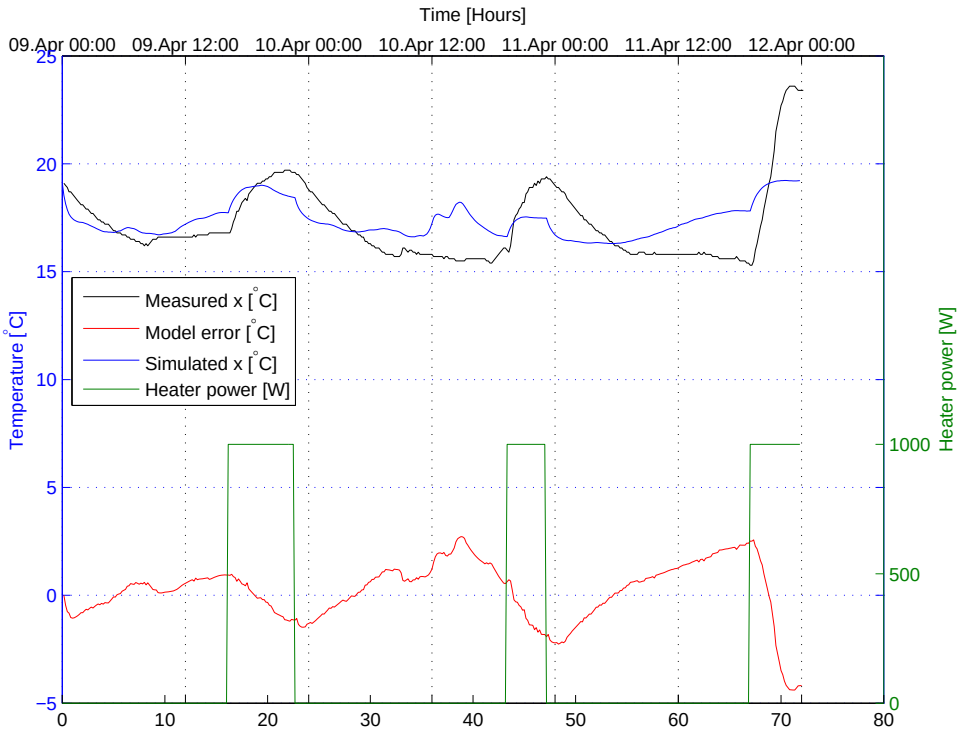

FIGURE 7.11: Experiment 3: Room 4: Living room

Another thing that can be observed from this plot is that there is two consecutive peaks in the model dynamics around mid-day of the 10th of April. Examining the outdoor temperature at the time in question Figure 7.12, there can be seen that the outside temperature peaks at the same time.

It would seem that the model overemphasizes the effect of outside temperature at this time, similar effects can be seen in the plots for Main bedroom and Kitchen, section B.4. Solar radiation though the windows during the time when estimation data is collected could explain this, although meteorological data shows that the weather was overcast for the day in question. Another explanation could be different wind-chilling effects. It can be seen from Table 7.1, that the average wind speed was lower during the period used for gathering data for estimation than the following days, and from plot Figure 7.13 the peak wind speed for each day is around mid-day.

Finally the entrance is considered in Figure 7.14 where the model performs poorly, an maximum error of $\approx 5°C$ is observed. This can be explained by a door being left open as previously mentioned. Again such disturbance cannot be known a priori, so it will have to be handled later. Choosing a different dataset for estimation may have improved

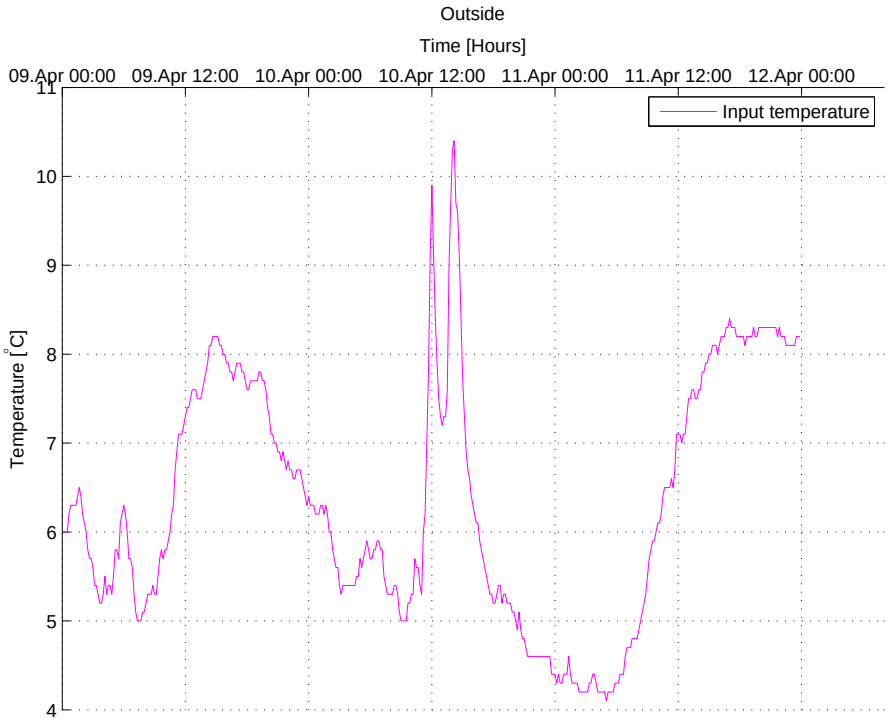

FIGURE 7.12: Experiment 3: Outside

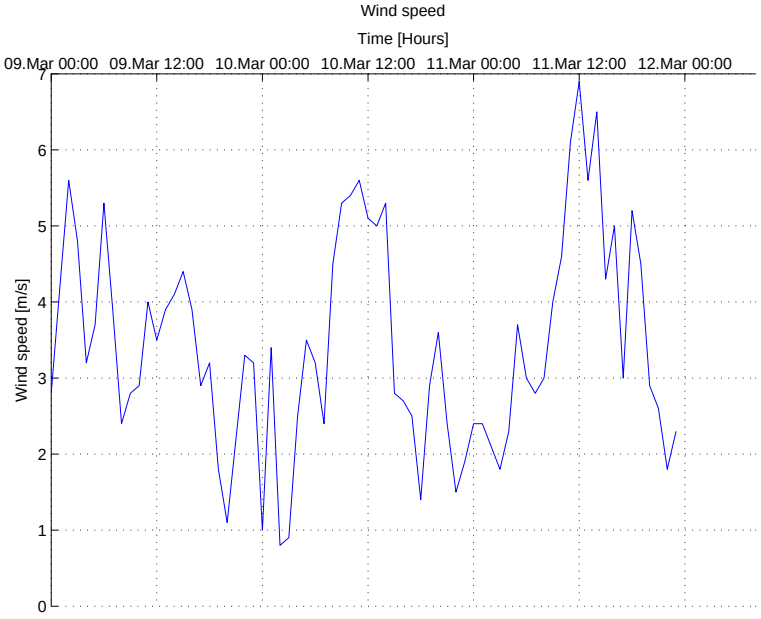

FIGURE 7.13: Experiment 3: Wind speed

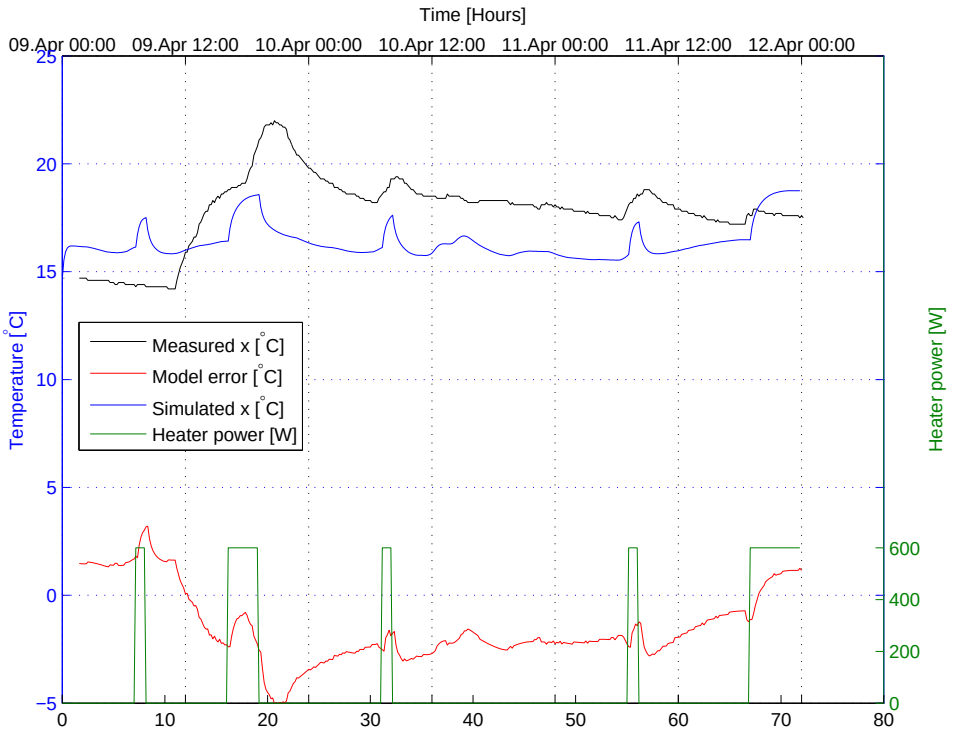

FIGURE 7.14: Experiment 3: Room 2: Entrance

the situation, but a dataset with few or no disturbances present over several days is not found in this case study. If implemented in a real system it is unlikely that the user can generate perfect conditions for estimation, so it is a useful test to see if the estimator can compensate for this large error in the identified model.

This section shows that by comparing the simulated temperature to the data different performances of the model is obtained, some rooms follow the measured temperature with small errors, while others have larger errors due to disturbances and poorly conditioned data. A summary of model error is presented in Table 7.5 and Table 7.6.

	Bathroom		**Entrance**		**Guest bedroom**	
	mean	max	mean	max	mean	max
Experiment 1 [$°C$]	4.2	8.3	1.3	5.5	0.3	1.7
Experiment 2 [$°C$]	6.4	7.5	2.7	7.3	24.6	26.6
Experiment 3 [$°C$]	0.5	1.6	2.0	5.0	0.4	1.7

TABLE 7.5: Experiment 1-3: room 1-3: Model error

It can be seen that the mean error is relatively low for most rooms, but larger in the entrance. The maximum error value is also found in the entrance close to $5°C$ and in the living room. The maximum error in the living room seems to be caused by the fireplace.

	Living room		**Main bedroom**		**Kitchen**	
	mean	max	mean	max	mean	max
Experiment 1 $[°C]$	1.8	9.5	1.7	3.8	1.3	5.0
Experiment 2 $[°C]$	2.4	7.0	6.2	8.6	5.7	7.7
Experiment 3 $[°C]$	1.1	4.4	1.0	2.8	0.5	2.0

TABLE 7.6: Experiment 1-3: room 4-6: Model error

The rest of the rooms have relatively low values for error, but there is still potential for improvement. In the next section a Kalman Filter will be applied and results are expected to improve.

7.3 Kalman filter testing

By introducing an Extended Kalman Filter with state- and parameter estimation a temperature estimator is implemented. Parameters are introduced in each state of the model to compensate for unknown time varying disturbances. The initial model used in the Kalman Filter is the one from experiment 3, since it is seen to have the over all best performance.

7.3.1 Tuning

The EKF is tuned using the covariances of the three noises present in the model: measurement noise $\mathbf{v}_k$, plant noise $\mathbf{w}_k$, and parameter exploration noise $\mathbf{n}_k$.

The Covariance of measurement noise is assumed to be small, since it can be seen that there is not much noise present in the measurements.

$$\sigma_{vv}^2 = 0.1 \tag{7.5}$$

The Covariance noise of the plant is calculated from the square of the mean model error found in previous experiments, so:

$$\sigma_{ww}^2 = \begin{bmatrix} 0.25 & 4.0 & 0.16 & 1.21 & 1.0 & 0.25 \end{bmatrix} \tag{7.6}$$

Parameter exploration noise is found by tuning, it can be seen that the higher the value is chosen to be the quicker the parameter changes. It can also be seen that these values are small compared to the other covariances. The values are chosen large enough for the EKF to react to rapid changes, but without introducing additional noise in the state estimation:

$$\sigma_{nn}^2 = \begin{bmatrix} 5 \times 10^{-7} & 6 \times 10^{-7} & 4 \times 10^{-8} & 2 \times 10^{-7} & 2 \times 10^{-7} & 10^{-8} \end{bmatrix} \tag{7.7}$$

The values in Equation 7.5, 7.5 and 7.7 are related to the EKF design matrices in the following way:

$$
\begin{aligned}
R &= \sigma_{vv}^2 I_{1\times 12} \\
Q &= \text{diag}\left(\begin{bmatrix}\sigma_{ww}^2 & \sigma_{nn}^2\end{bmatrix}\right)
\end{aligned}
\tag{7.8}
$$

7.3.2 Experiment 4: Using EKF with state and parameter estimation

The EKF sampling interval is selected to be 10 times as quick as the original sampling time, so a new state is estimated every minute. If missing data occurs the EKF will estimate the temperature until valid data is available. The resampling of the model is done by Matlab-function `d2d()` which uses zero-order-hold [40]. The EKF-algorithm is written in pseudocode in Algorithm 3

for k = *1:N* **do**
  Estimate state based on previous estimates, and input-values
  Calculate Jacobian based on current estimate
  Calculate new Kalman filter covariance matrix P based on new Jacobian
  if (*Valid measurement available* $\mathbf{y}(k)$) **then**
    Calculate Kalman Filter gain K_f
    Filter estimate using output values from plant
    Recalculate Kalman Filter covariance matrix P
  end
end

Algorithm 3: EKF pseudocode

Notice that when an invalid input is detected, the filter must discard the whole input vector at that time, therefore and potentially valuable information is lost.

It can be seen in the plot from the main bedroom Figure 7.15 that the estimate now follows fast dynamics from heaters. Since the kalman filter is updated with measurements every ten minutes the EKF can adapt to rapid changes and time varying disturbances. When however large periods of missing data occurs the performance deteriorates. It can be clearly seen on the 11th of April that when data input goes missing for eight hours the EKF has an error of about $2°C$. When the EKF estimates states it uses both the previously identified model and the online estimated parameter. The error may originate from both these sources. It could be that the online estimated time varying disturbance coefficient is no longer valid, because the nature of the disturbances has changed. Another explanation is that the model overemphasizes the effect of outside temperature, as seen in the previous test. In any case, eight hours means that the

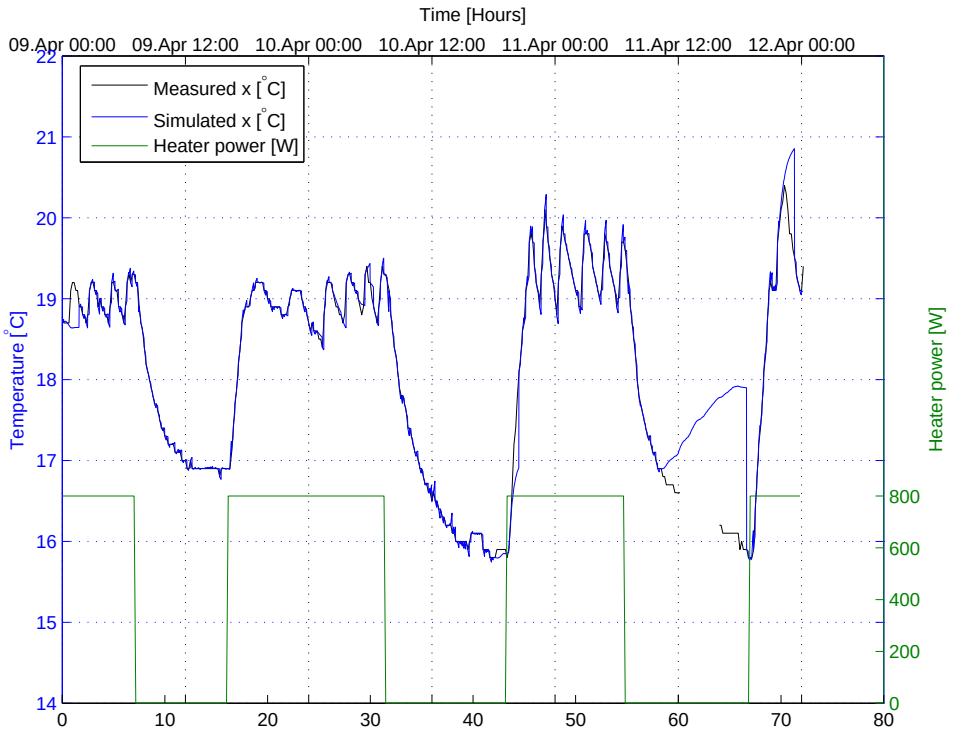

FIGURE 7.15: Experiment 4: Room 5: Main bedroom

kalman filter has estimated states for 480 samples, and an initial small error in state estimation might grow large in this time. When applying the EKF as an estimator in an MPC the prediction horizon might be smaller than 480 steps.

Examining the other plots shows in section B.5 the same tenancy can be observed, but the error is generally smaller when missing data occurs. The plot from the living room Figure 7.16 shows that the EKF compensates for the disturbance from the fireplace observed in previous tests.

It can also be seen that the EKF has a small error at the beginning of the test, this may be explained by missing data. Examining the plot of the parameters estimated by the EKF Figure 7.17 it can be seen that they do not change at this time. This means that no update is performed and previous values for parameters are being applied, the same can be seen for the period on the 11th of April where data has been missing for a long time. The parameters themselves does not seem to converge against a constant value, but follows the system dynamics. The time varying nature of disturbances in this system makes this a reasonable result and may show that the disturbances are being handled correctly.

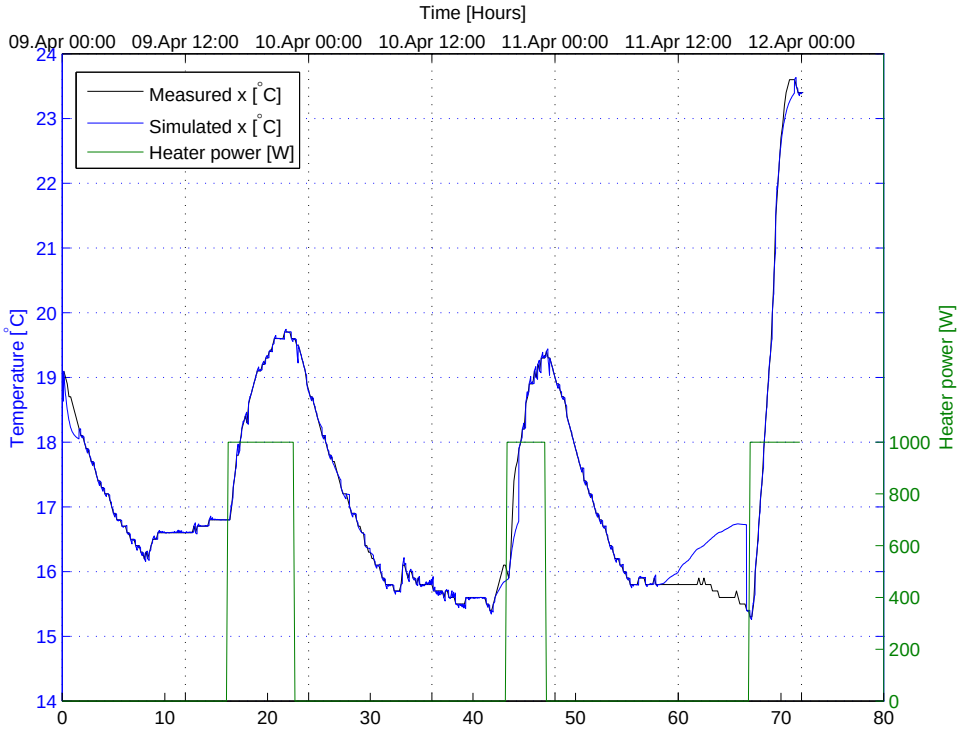

FIGURE 7.16: Experiment 4: Room 4: Living room

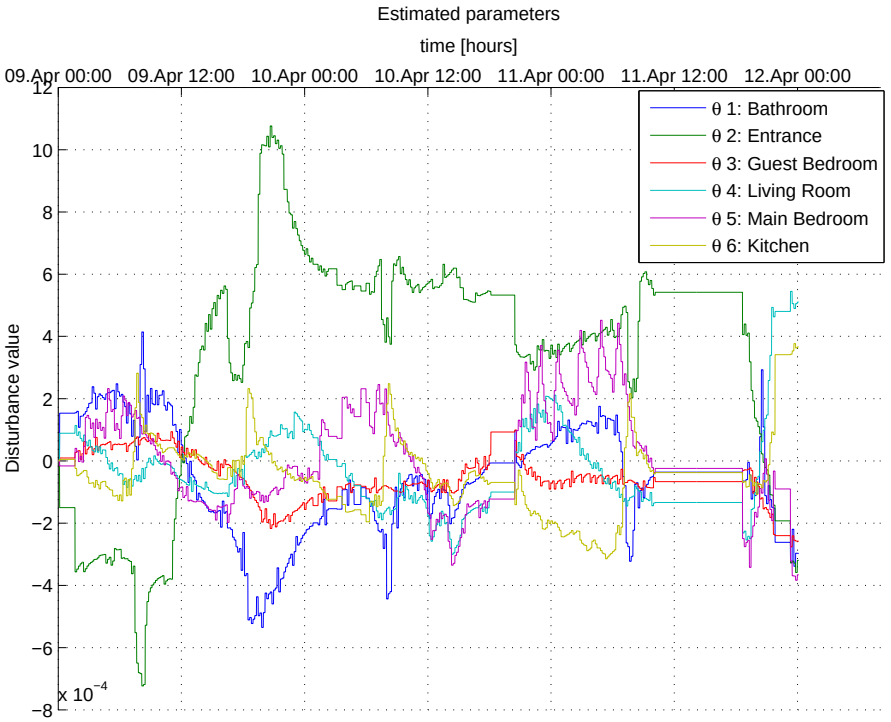

FIGURE 7.17: Experiment 4: Estimated coefficients

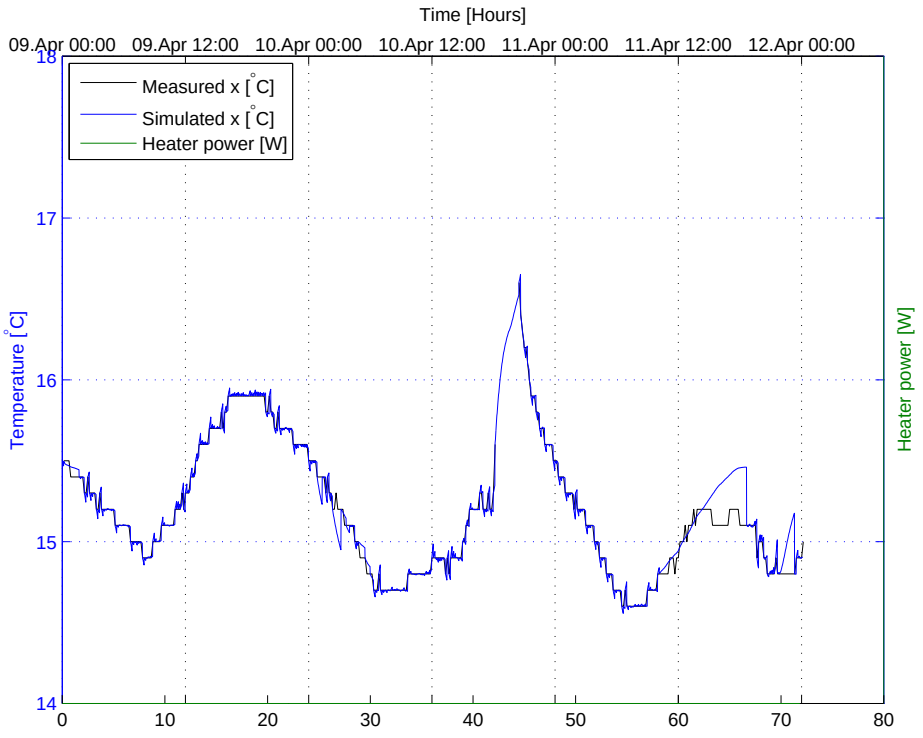

FIGURE 7.18: Experiment 4: Room 3: Guest bedroom

The plot from the guest bedroom Figure 7.18 shows the kalman filter estimate performing well when data is missing. It can be seen that on the evening of the 10th of April there is a period of missing data just as the building is heating up, the EKF seem to estimate the correct dynamics here ends up at roughly the same temperature as measured when the input is deemed valid again. This may show that with constant disturbances the EKF fulfills its purpose and predicts the temperature correctly. In the other rooms it can be seen that in this period the accuracy of the estimator is also high.

7.4 Computational time

The computational time of a single iteration is tested using Matlab-function `tic` and `toc`, the mean and maximum values is shown for three different operations:

- Update model with new parameter estimates
- Preform EKF-estimation
- Preform EKF-Filtering
- Total iteration time

Operation	**Update**		**Estimation**		**Filtering**		**Total**	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max
Value (Ts = 60s) $[ms]$	0.14	0.57	2.6	12.8	0.11	3.2	2.8	16.3

TABLE 7.7: Iteration time for estimator

Notice that the maximum operation times does not add up to the total time, this is because filtering is not performed every iteration, and the maximum total time is a iteration without filtering. To be applicable for real time implementation the EKF in this set-up will have to calculate ten estimates every minute, so the worst possible total computational time is $163ms$.

The computational time for offline estimation, Model-object construction and resampling is not considered, since these operations can be performed in a Matlab-environment and the model-object constructed can be exported to a real time system implemented in a arbitrary programming language.

7.5 General remarks

A few general remarks should be made before concluding the thesis.

First of all, in every experiment the model is found to be stable, observable and controllable. This is tested using Matlab-functions `eig()`, `ctrb()`, `obsv()` and `rank()`, see [40] for details.

It can be seen from the selection of dataset for estimation that there still may be unknown disturbances present, while it is not feasible to exclude all possible disturbances, some improvements to the experiment can be made, for instance by collecting data in a period without inhabitants in the building, compensating for different environmental disturbances like chilling effects from wind and heat effects from solar radiation.

Ideally the controller should be able to define temperature set-points for electrical ovens, and be able to receive feedback from the ovens regarding how much heat they are currently producing. This feedback can be done by introducing a power or current sensor in the electrical mains loop, for this thesis it is assumed that when energized the ovens power usage is equal to its nominal power rating. When a thermostat is introduced this will produce an inevitable deviation. In most simple oven thermostats the power is either on or of, and the amount of heat produced is controlled by turning the power on and of in intervals. The length of this interval is usually controlled by some kind of hysteresis (bang-bang) control, more advanced units may also use PID control. The heater sensors used in the case study are switches that energize the heaters, and provides no feedback as to what kind of power the heater is currently producing, this may account for small errors in the dynamics of the model.

The heater state of the Bathroom could not be measured in the case study, it is expected that including measurements from this room when estimating should improve performance.

The contribution of heat flow through the ceiling of the building is ignored, this from the argument that in most apartment buildings the temperature above the ceiling is close to equal to the room beneath it. In the case study this assumption may be incorrect, so the heat flow is treated as a contribution to the sum of disturbances. If measurements of temperature above the ceiling can be made the model can be expanded by including said temperatures as additional inputs.

The offline identified model is found to be stable by applying well known control theory methods but when using EKF the online estimated parameters could push the system over the edge of stability, this is not observed in testing, but if the EKF is to be implemented with a smaller estimation interval i.e. more estimation iterations between

filtering, this could become a problem. No formal stability proof estimator is presented, the stability property of this non-linear time variant system is simply tested empirically.

The precision of the temperature sensors $\pm 1^{\circ}C$ is relatively poor compared to industry standards. The same sensors are used for collecting the data for identification and comparing the model performance, so the precision of sensors are considered less important in this initial part of the project. When implementing a complete system the overall precision can be improved by applying more precise sensors, this may increase the cost of the system, so it would be sensible to balance necessary precision against cost of sensors.

The parameters estimated with EKF do not seem to converge to a single value, but rather follow the plant dynamics. A real value of the proposed estimated parameters is difficult to obtain, since they are a sum of all disturbances present. This is considered to be of less importance, since the EKF seem to perform well without the explicit knowledge of disturbance contributions. But if large and rapid changes occurs in the plant the EKF might not be able to react properly.

The parameters estimated online using EKF are only present in the state matrix: $\mathbf{A}$ of the system, so it may have difficulties to compensate for model errors and disturbances in the input matrix: $\mathbf{B}$. A possible improvement of the filter would be to add parameters to be estimated in the $\mathbf{B}$-matrix. This is straight forward according to KF-theory, but will add more states to the augmented state space system.

The modeling of outside-temperature as an input needs to be considered when using the EKF as an estimator for optimal control, outside temperature can not be controlled, so the use of these control inputs needs to be prevented. This can be done by weighing use of said control outputs heavily in the cost function of the controller.

Unmeasured disturbances are handled through parameters in the EKF, but even with a well tuned kalman filter, the performance can be improved by measuring the contribution from each disturbance. This can involve expanding the model to include measured disturbances. This expansion will increase the over all complexity of the system, in this case it is not just the complexity of the mathematical model, but also the amount of sensors that needs to be included. More sensors means that the system will be more expensive and cumbersome to install, the identification of parameters and tuning can also be more time consuming. Sensors that can expand the system can be environmental

sensors, that measure wind and sun radiation, sensors that detect human actions, for example sensors on windows and doors that signal if they are opened, power sensors for electrical appliances not primarily used for heating. Here a trade-off between complexity and precision needs to be considered.

Computational time of the system has been found to fit well within the constrains for the proposed sample interval, but when applying optimal control the computational time of the control algorithm add to total iteration time. Seeing how the EKF-estimator has a relatively low number of states and a low computational time it seems feasible to implement a controller within stated time constrains. If the system is to be implemented in real time on a micro-controller or SOC, the lower processing capabilities needs to be considered. Few of these systems can run Matlab, further more Matlab is considered slow compared to other low-level programming languages. It is probable that the system is to be implemented in one of the faster low-level programming languages, so while the processing power may be weaker, the language may be faster. Realizing this there is no need to expect computational time to be a problem for the remainder of the project.

Chapter 8

Conclusion and further work

This chapter summarizes the main results found in the case study, and points out possible improvements based on the discussion in the previous chapter.

	Bathroom		**Entrance**		**Guest bedroom**	
	mean	max	mean	max	mean	max
Experiment 1 [$°C$]	4.2	8.3	1.3	5.5	0.3	1.7
Experiment 2 [$°C$]	6.4	7.5	2.7	7.3	24.6	26.6
Experiment 3 [$°C$]	0.5	1.6	2.0	5.0	0.4	1.7
EKF	-	1.2	-	1.7	-	0.5

TABLE 8.1: All experiments: room 1-3: Model/estimator error

	Living room		**Main bedroom**		**Kitchen**	
	mean	max	mean	max	mean	max
Experiment 1 [$°C$]	1.8	9.5	1.7	3.8	1.3	5.0
Experiment 2 [$°C$]	2.4	7.0	6.2	8.6	5.7	7.7
Experiment 3 [$°C$]	1.1	4.4	1.0	2.8	0.5	2.0
EKF	-	1.2	-	1.9	-	1.1

TABLE 8.2: All experiments: room 4-6: Model/estimator error

From the results presented in the case study it is shown that a simple model for building thermal flow can be implemented and still have adequate performance for optimal control. By using a combination of calculation and offline estimation a model is derived, and the performance can be further improved by applying a EKF with state and parameter estimation, see Table 8.1 and Table 8.2 for a summary of the model errors found. The largest error observed is $\approx 2°C$ and occurs when data has been missing for a long time.

Thermal transmittance for the floor, external- and internal walls are calculated and differences of the materials in outer walls is compensated for in the offline estimation. Further more a parameter for the dispersion effect for heaters is estimated offline.

The collection of data in the case study is done with inhabitants present in the building, in order to collect a better set of data for estimation the inhabitants could be moved for the duration of data collection. This is expected to give a better conditioned data set, and in turn provide a more precise model.

In large rooms the model is shown to react to quickly to temperature input, this could be because of internal heat dynamics within the room itself. Proposed solution if this occurs is to separate the room into multiple states, where each state represents a certain zone of the room.

The EKF estimates a general disturbance parameter per state, no explicit knowledge of each disturbance seems to be necessary, however including measurements from disturbances in the system could improve the over all performance. Adding environmental sensors seems like the simplest and most efficient way of improving performance. Adding sensors measuring inhabitants behavior may also have a favorable effect.

Computational time of the proposed model with EKF is low and thus seemingly suitable for real time implementation. The model is of relatively low order, so the computational time of a optimal control algorithm using the proposed EKF as an estimator is also expected to be low.

Heater input to the model was measured as on and off, and the heaters assumed to provide heating power equal to their rated nominal power. Measuring the heater power using electrical power- or current sensors could improve the initial estimated model and have a very positive impact on model and estimator performance. Further more allowing the controller to assign set-points for heaters may be a central functionality in the

system to come.

It is observed that a large number of disturbances affects the input of the model rather than the states, by expanding the EKF to include parameter estimation in the input-matrix of the system some improvement may be made. This will however mean more states in the augmented system and therefore lower scalability.

When missing data occurs in one of the inputs the filter must discard the whole input vector. If this can be prevented, by in some way handle a subset of the input vector while filtering the periods of missing data can be reduced, and better accuracy can be achieved.

Appendix A

Code documentation

A.1 Model

Properties	
Property name	**Description**
`SYS`	Identified state space system that is the basis of the model with properties corresponding to system matrices and sample time: `A,B,C,D,K,Ts`
`theta`	Identifiable parameters in the model
`noTheta`	Number of identifiable parameters in model
`A,B,C,D,K`	System matrices of the state space model augmented for parameter identification by `theta`
`J`	Jacobian function of the non-linear augmented system. Defined as array of Matlab-functions with call: `Aij = J{i,j}(Xi{:})` `Xi` is a cell array of numerical values of the estimate $\hat{\xi}$ `Aij` is the numeric value of the Jacobian matrix element `i,j` based on the estimate `Xi`
`Ts`	Sample time

Table A.1: Properties for object Model

Methods	
function name	**Description**

`Model(symFcn,SYS,NoTheta)`	Constructs a new instance of object Model based on the inputed state space system `SYS`, and the symbolic function `symFcn` that defines the relationship between the parameters to be estimated and the state space system.
	
	The `symFcn` is a handle to a function which can be called in two different ways:
	
	`[A,B,C,D,K] = symFcn(A,B,C,D,K,theta)`
	This call augments system matrices with the parameters to be estimated θ, the returned values are numerical.
	
	`[f,v] = symFcn(A,B,C,D,K,theta)`
	This call returns an array of non-linear symbolic functions that define each state in the augmented state space system $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \theta_k, \mathbf{u}_k)$ and a vector of symbolic variables `v` that defines the variables in the function. This function is the basis for calculating the Jacobians of the system.
	
	The state space model `SYS` is assumed to have the properties corresponding to the system matrices and sample time: `A,B,C,D,K,Ts` for efficency it is recomended to use a struct.

`updateParameters()`	Updates the state space parameters in the model based on the state space system `SYS`, and the property `theta`. It uses the symbolic function `symFcn` `[Am,Bm,Cm,Dm,Km] = symFcn(A,B,C,D,K,theta)` Where `Am,Bm,Cm,Dm,Km` is the augmented state space matrices and `A,B,C,D,K` is the properties of the state space system `SYS` inputed in the constructor, `theta` is a property of the object.
`jacobianInit()`	Initiates the J-property of the model which is the Jacobian matrix based on the symbolic function `symFcn`. It returns an array of Matlab-functions `J` to the object.

Table A.2: Methods for object Model

A.2 RoomTemperatureSet

Properties	
Property name	**Description**
`name`	String containing the name of the room.
`time`	Vector containing a numerical representation of the time and date for when samples where collected
`T`	Vector of temperatures in current room at time `time`
`u`	Vector containing the input value in current room at time `time`

Table A.3: Properties for object RoomTemperatureSet

Methods	
Function name	**Description**

`RoomTemperatureSet()`	Empty constructor, returns a instance with properties:
	`name = NULL`
	`T = []`
	`time = []`
	`u = -1`

`RoomTemperatureSet(name,u,time)`	Constructor for rooms without temperature measurement
`RoomTemperatureSet(name,u,time,T)`	Constructor for normal rooms

Table A.4: Methods for object RoomTemperatureSet

A.3 TemperatureSet

Properties	
Property name	**Description**
`rooms`	Vector of instances of `RoomTemperatureSet` each containing data for a single room.
`noInputs`	Number of inputs in dataset
`noStates`	Number of states in dataset
`DAT`	`iddata` time series object based on input and output, see Mathworks documentation center[40] for details about `iddata`-objects
`Ts`	Sampling time

Table A.5: Properties for object TemperatureSet

Methods	
Function name	**Description**
`TemperatureSet(filename,estimate,Ts)`	Constructs an instance of `temperatureSet` by reading using `readData()` from the excel-file named `filename`, the temperatures in the model are converted to Kelvin with the function `celsius2kelvin`, and the `iddata`-object `DAT` is initialized with the function `update()` `estimate` is a boolean defining if the dataset should be used for estimation or simulation. If it is to be used for estimation temperature data needs to be patched using the `patchMisData`-function, if it is to be used for simulation the temperature in the basement is set to the value of the first measurement from the basement.
`getInput(i)`	Returns a vector of the input in dataset at time `i`.
`getOutput(i)`	Returns a vector of the output in dataset at time `i`.
`update()`	Updates `iddata`-object `DAT` with the contents of the instances in `rooms`
`addRoom(j,room)`	Adds the `RoomTemperatureSet`-instance `room` to the vector `rooms` at index `j`
`readData(filename,estimate,Ts)`	Reads the data from the excel file named `filename` an example of the datafile layout for the case study is included in Figure 5.3 blank cells are interpeted as `NaN` patches missing input data with the `patchMisData`-function if the boolean `estimate` is `true` it also patches missing temperature data, id `estimate` is false then simulation is assumed, and the model sets the temperature value for the basement constant. `Ts` is sample time for the dataset.
`patchMisData(u)`	Patches missing data in vector `u` if values in any indexes of the vector is interpreted as `NaN` then the value for this index is the previous valid value in the vector. The function starts at index `1`. If there is no previous valid values the index is given the value: `-1`
`celsius2kelvin()`	Converts all temperatures in the set from Celsius to Kelvin.
`kelvin2celsius()`	Converts all temperatures in the set from Kelvin to Celsius.

A.4 EKF

Properties	
Property name	**Description**
`Kf`	Kalman filter gain
`Q`	Covariance matrix plant noise
`R`	Covariance matrix measurement noise
`P`	Estimation error covariance matrix
`N`	Order of filter
`I`	`eye`-matrix of size `N`

Table A.7: Properties for object EKF

Methods	
Function name	**Description**
`EKF(Tune)`	Constructs a kalman filter using the predefined struct of design matrices `Q` and `R` named `Tune`
`jacobian(J,xi)`	Calculates the numerical Jacobian matrix by calling the Matlab-functions in the matrix `J` with arguments based on current state estimate `xi`.
`estimate(A, B, K, xhat_k, uk)`	Returns a vector `xhat` that is the a priori state estimate based on the system matrices `A`,`B` and `K`, the previous estimate `xhat_k` and previous input `uk`. Calculates the numerical Jacobian matrix using the `jacobian()`-function and uses it to update the estimation error covariance matrix: `P` The previous estimate can be a-priori or a-posteriori
`filter(C,xhat_k,yk)`	Calculates Kalman gain `Kf` based on the covariance matrices and output matrix `C` filters the estimated state with current output `yk` recalculates the estimation error covariance matrix: `P`

Table A.8: Methods for object EKF

A.5 Support functions

Methods	
Filename	**Description**

`defParameters(type,thetaInit)`	Defines the names and initial values of parameters to be estimated offline
	
	Which names being used is defined by `type` the different names are:
	`lite1: Ui, Uo, Ub, Kh`
	`lite2: Kh`
	`lite3: K1, K2, K3, K4, K5, K6, Kh`
	
	`thetaInit` is the initial values of the parameters to be estimated
	
	The function returns a cell-matrix `par` with names of parameters in one row and the initial value in the other.

`odefun1(Ui,Uo,Ub,Kh,Ts)`
`odefun2(Kh,Ts)`
`odefun3(K1,K3,K4,K5,K6,Kh,Ts)`
	Case study specific function for defining the structure of a state space system to be estimated by `greyest` returns matrices: `A,B,C,D,K` defining the state space system. The input variables are parameters to be estimated.
	
	For defining other `odefun`-functions see: [40, idegrey]
	
	Ts is sampling time

`offlineEstimation(roomData,Ts,method)`	Estimates parameters for the state space system based on data from the `TemperatureSet`-object using different parameters and methods. Returns an `idgrey`-object [40, idegrey].
	
	`method` defines which method should be used, the alternatives are:
	
	`idgrey1` - Grey-box estimation with 4 parameters:
	`Ui,Uo,Ub,Kh`
	`idgrey2` - Grey-box estimation with 1 parameter:
	`Kh`
	`idgrey3`- Grey-box estimation with 6 parameter:
	`K1,K3,K4,K5,K6,Kh`

TABLE A.9: Support functions for system I

Methods	
Filename	**Description**
`[f,v] = M15Sym(A,B,C,D,K,theta)`	Case study specific function that defines the relationship between offline identified state space system, parameters to be estimated online and augmented state space system. The function returns either two vectors, where `f` is a vectors of symbolic functions and `v` is a vector of symbolic parameters both used for calculating the Jacobian
`[A,B,C,D,K] = M15Sym(A,B,C,D,K,theta)`	It can also return the augmented state space system matrices based on offline identified state space system and online estimated parameter value, in this configuration it is used to update the model to new online estimation parameter values
`runModel(SYS,roomData,mode)`	Runs a simulation of the state space system `SYS` on the data from the `TemperatureSet`-object `roomData`. `mode` defines which type of simulation that should be executed, for three inputs only `M` is valid.
`runModel(SYS,Mod,roomData,mode)`	Runs a simulation of either the state space system `SYS` or the `Model`-object on the `TemperatureSet`-object `roomData`. `mode` defines which type of simulation that should be executed. The alternatives are: `M` - Runs the state space system `SYS` without any filtering `CDKF` - Runs the state space system `SYS` with state estimation using a continuous discrete kalman filter. `CDEKF` - Runs the `Model`-object `Mod` with state and parameter estimation using a continuous discrete extended kalman filter.

TABLE A.10: Support functions for system II

Methods	
Filename	**Description**

`KFtuning()`	Returns a tuning-struct for EKF
	
	the resulting Struct has the following properties:
	`Q`
	`R`
	Corresponding to the EKF covariance matrices.

`plotResults(roomData,x,u,P)`	Plots results for experiment with proper formating and names. `roomData` is the `TemperatureSet`-object that has been simulated- `x` is a matrix of `M x N` containing the state estimate for every time instance. `u` is a matrix of `Q x N` containing the input value for every time instance. `P` is a number defining how many samples the kalman filter is estimating per filtering run. For discrete KF `P = 1`

Table A.11: Support functions for system III

Appendix B

Plots

B.1 Selection of dataset for estimation

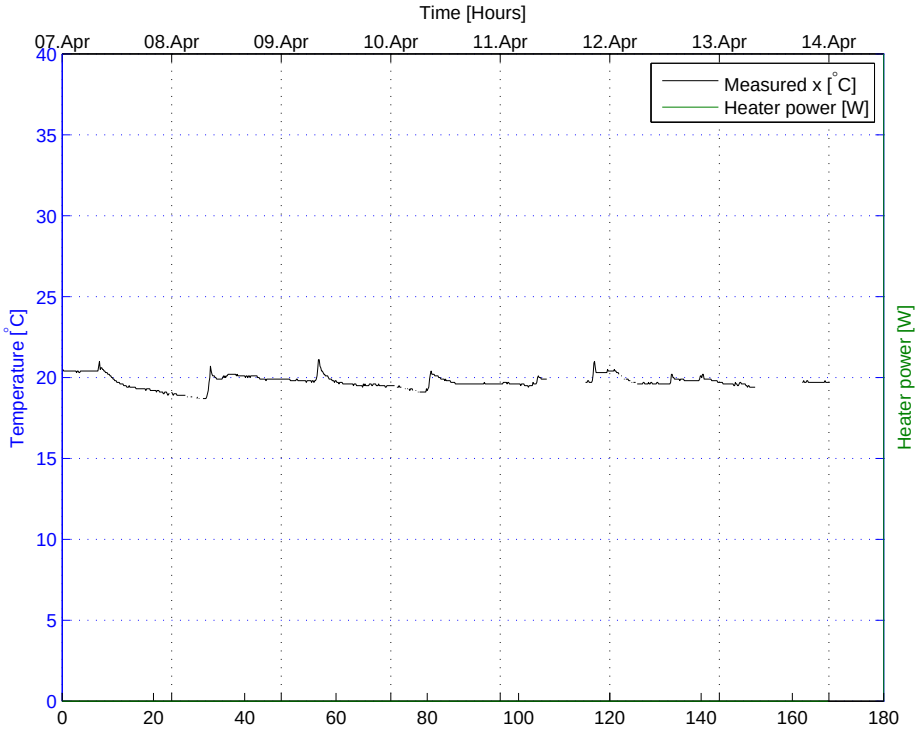

FIGURE B.1: Selection of dataset for estimation: Room 1: Bathroom

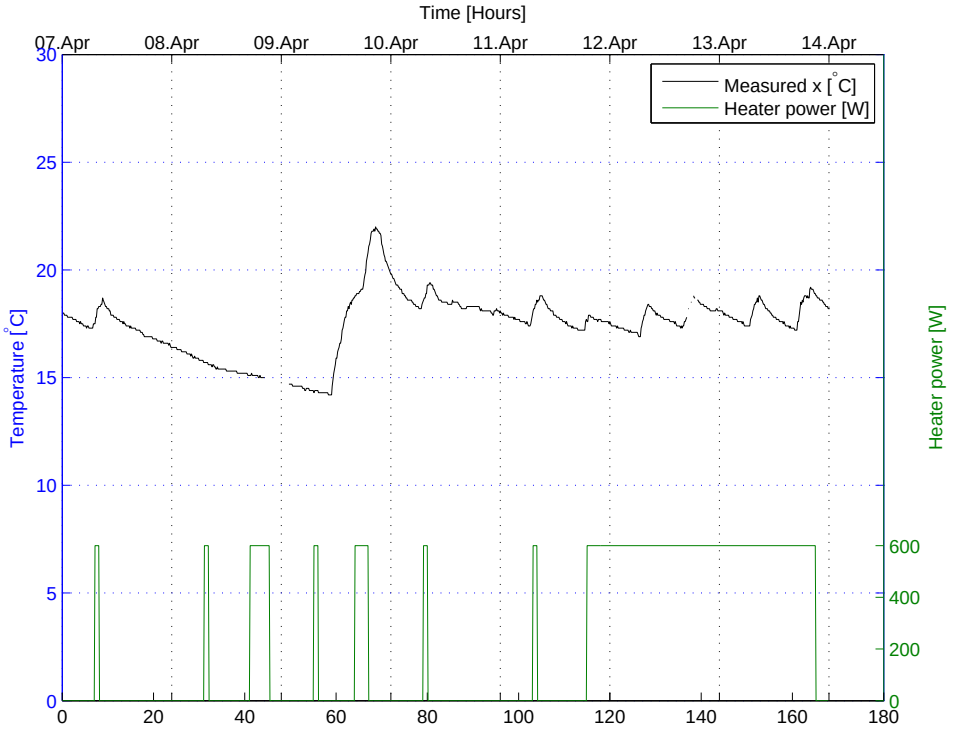

FIGURE B.2: Selection of dataset for estimation: Room 2: Entrance

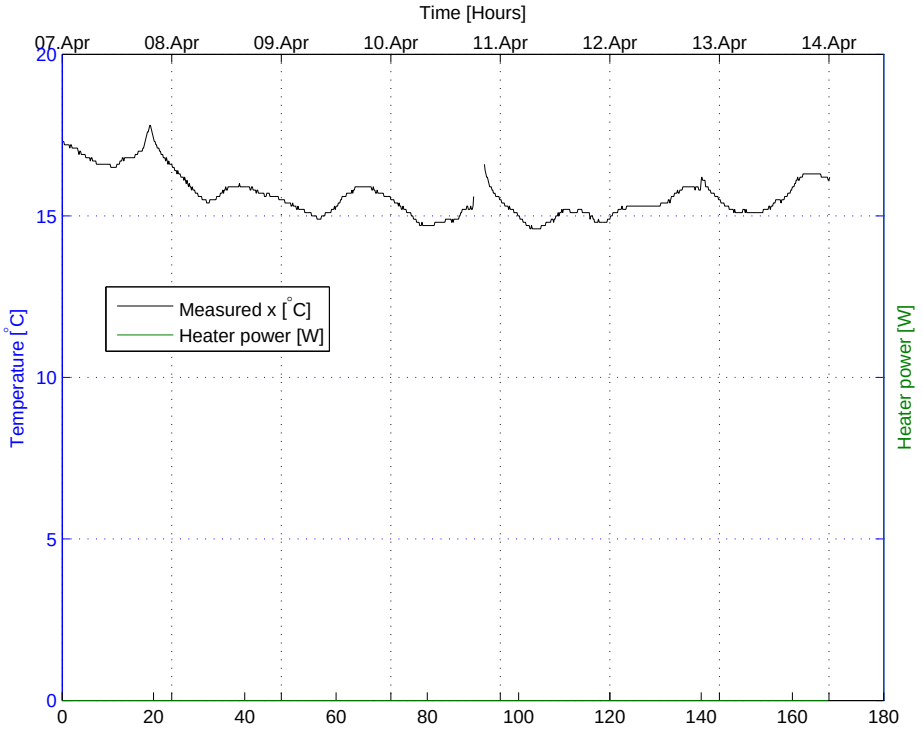

FIGURE B.3: Selection of dataset for estimation: Room 3: Guest bedroom

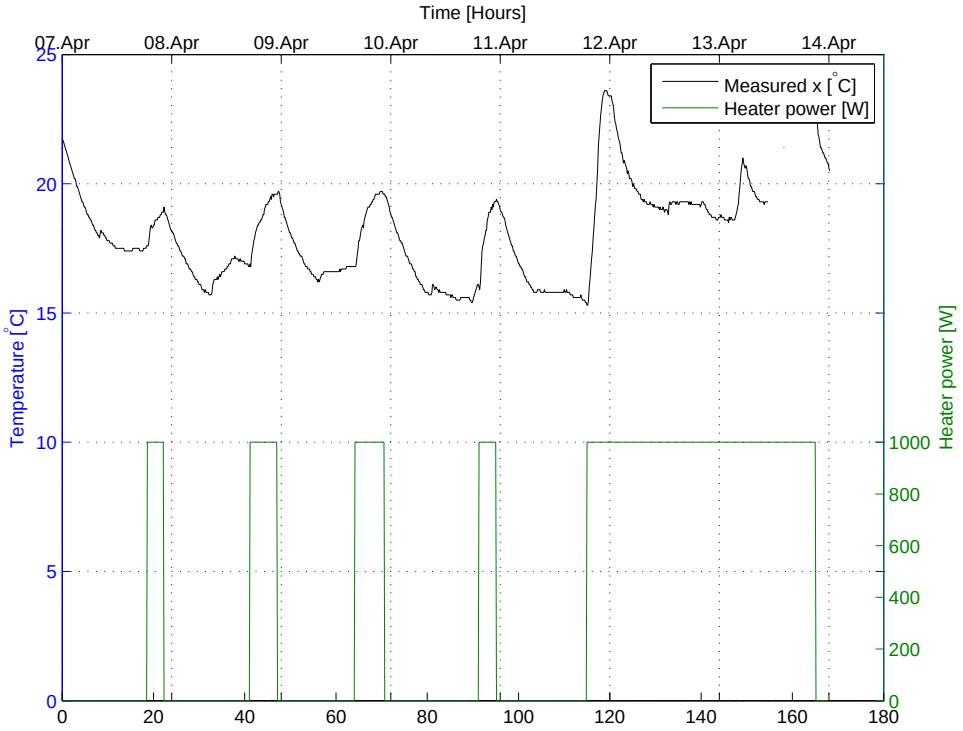

FIGURE B.4: Selection of dataset for estimation: Room 4: Living room

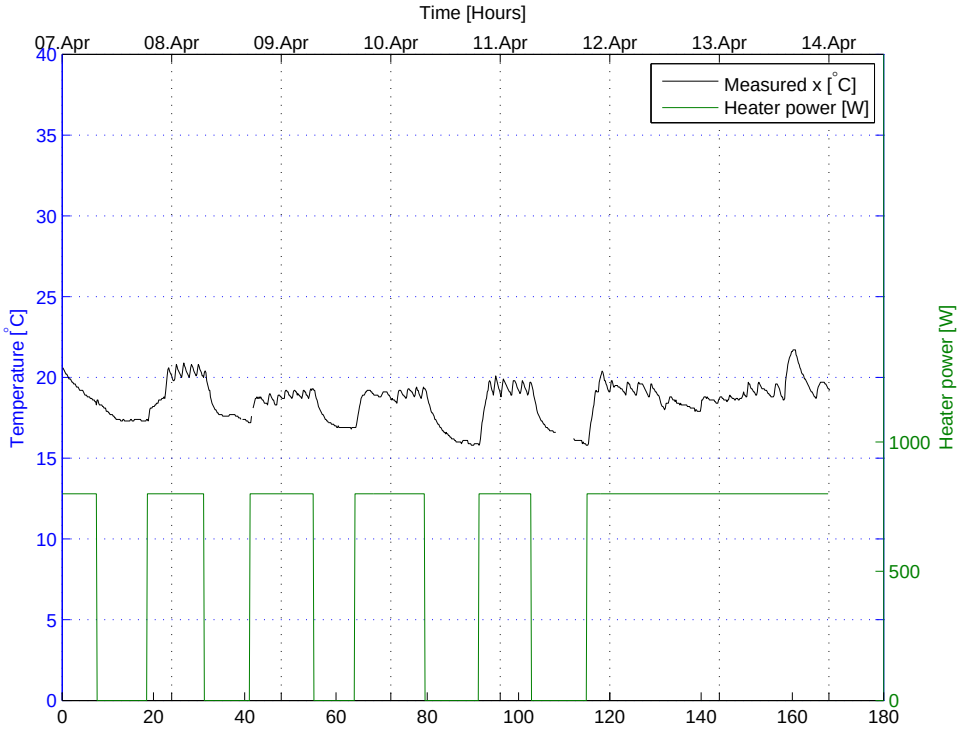

FIGURE B.5: Selection of dataset for estimation: Room 5: Main bedroom

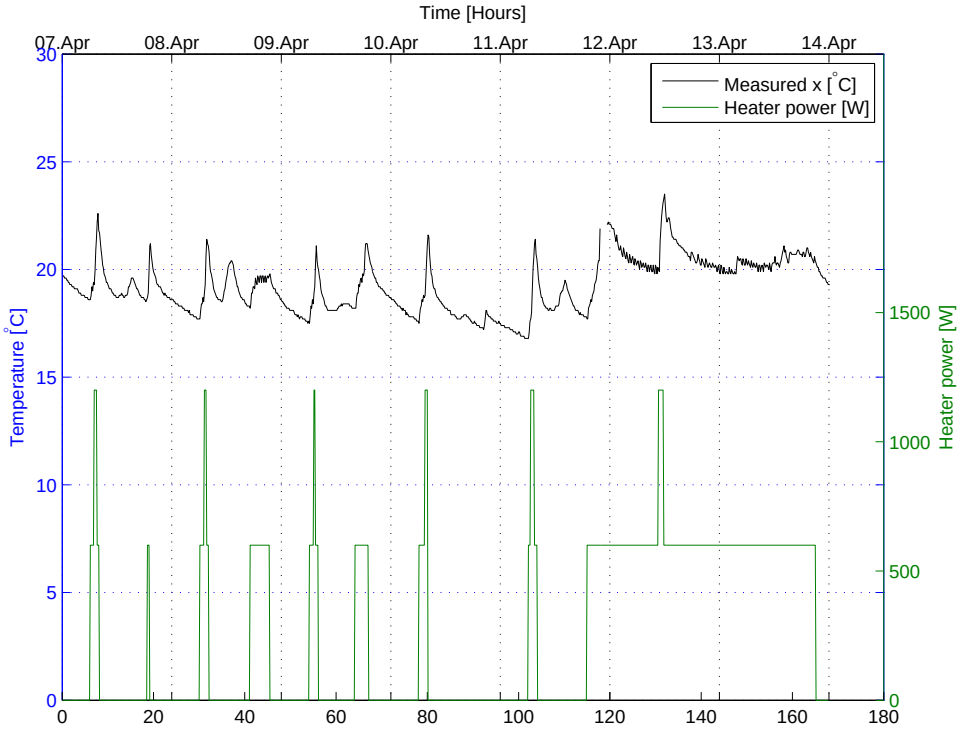

FIGURE B.6: Selection of dataset for estimation: Room 6: Kitchen

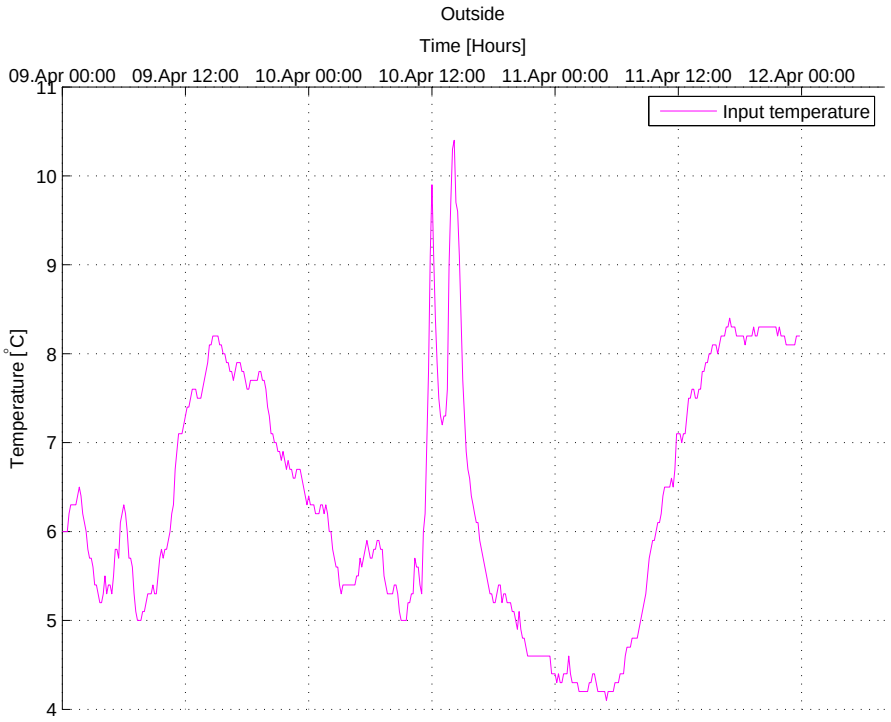

FIGURE B.7: Selection of dataset for estimation: Outside

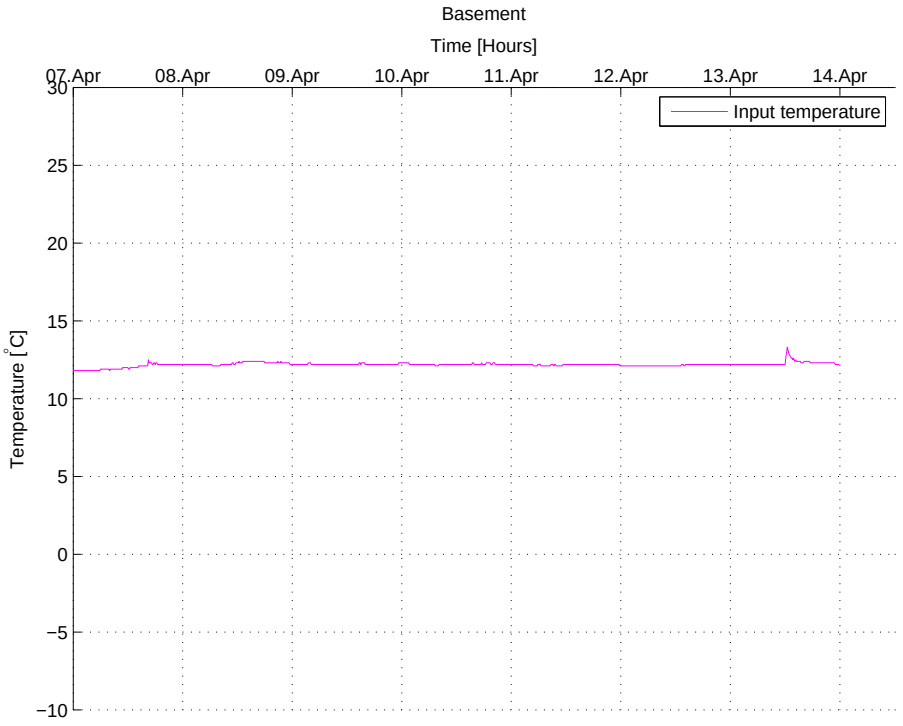

FIGURE B.8: Selection of dataset for estimation: Basement

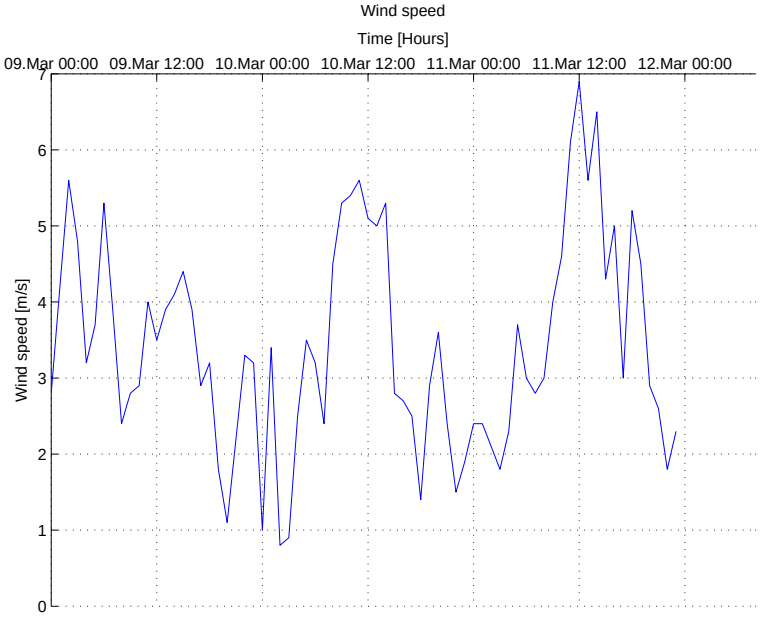

FIGURE B.9: Selection of dataset for estimation: Wind speed

B.2 Experiment 1

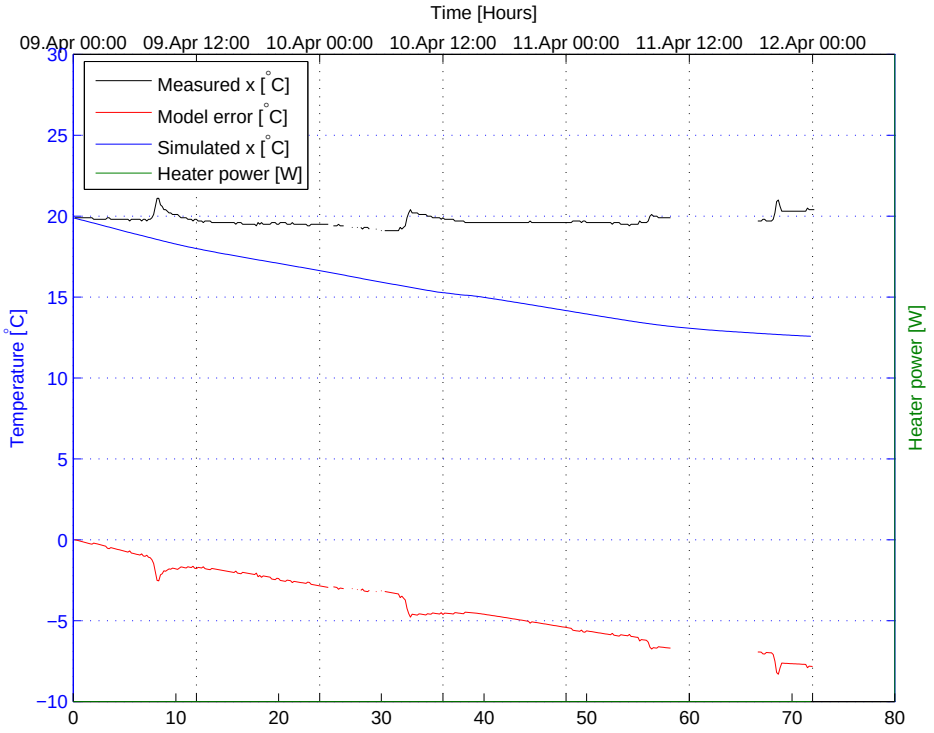

FIGURE B.10: Experiment 1: Room 1: Bathroom

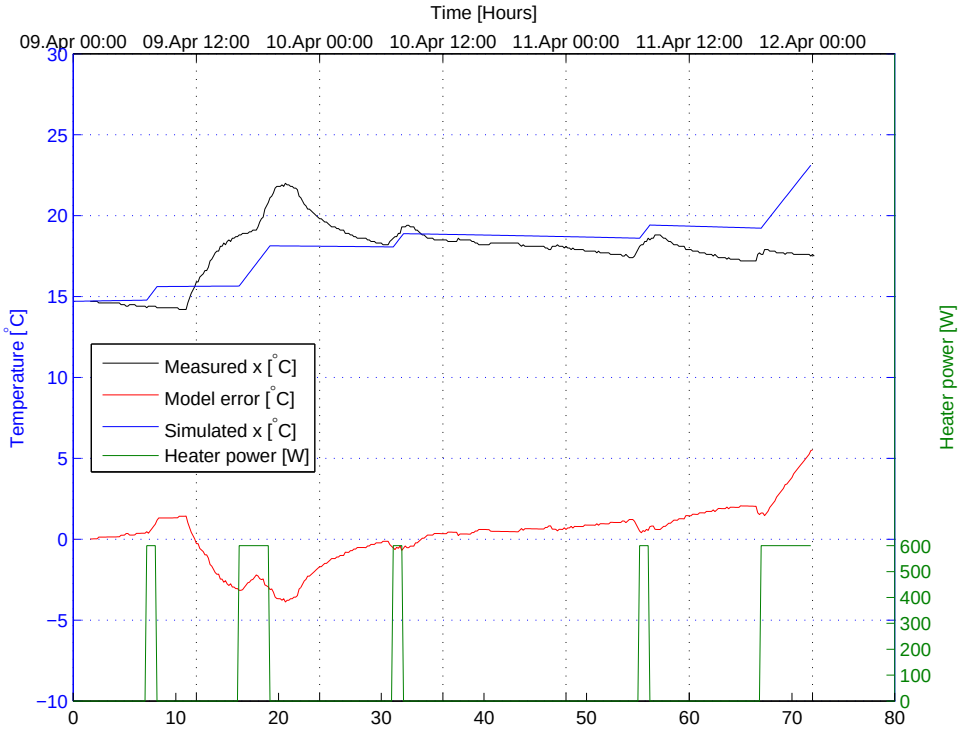

FIGURE B.11: Experiment 1: Room 2: Entrance

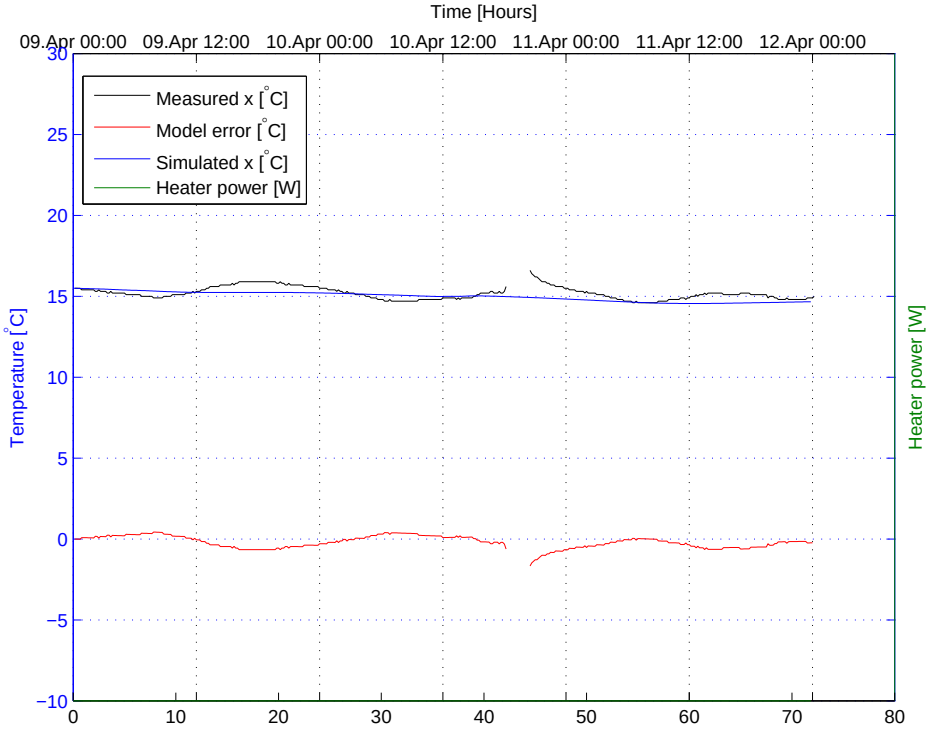

FIGURE B.12: Experiment 1: Room 3: Guest bedroom

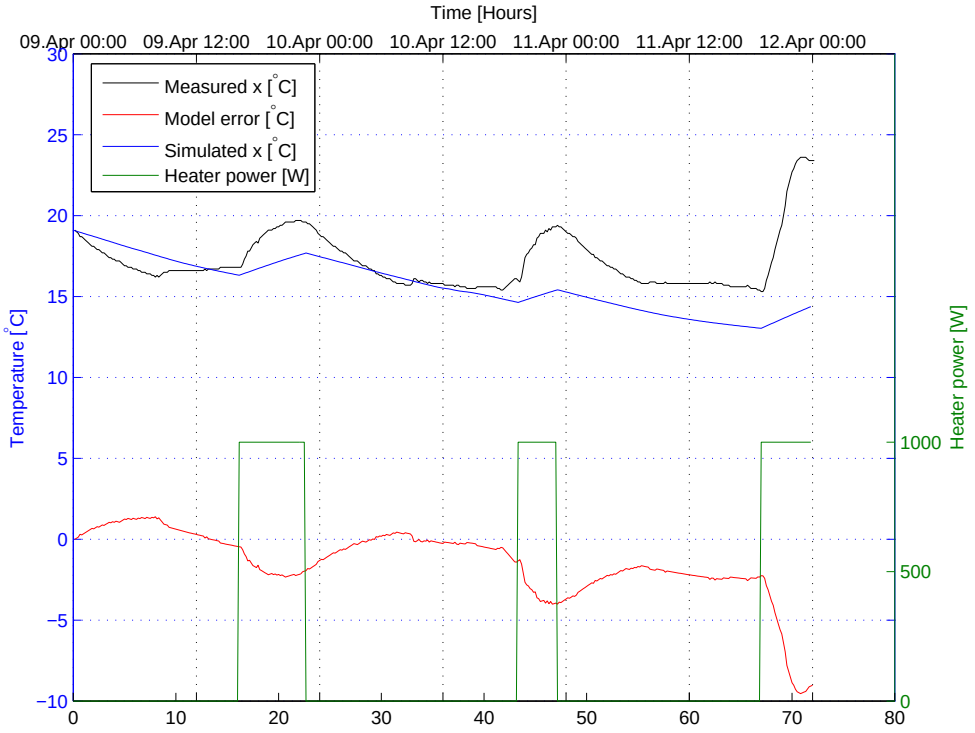

FIGURE B.13: Experiment 1: Room 4: Living room

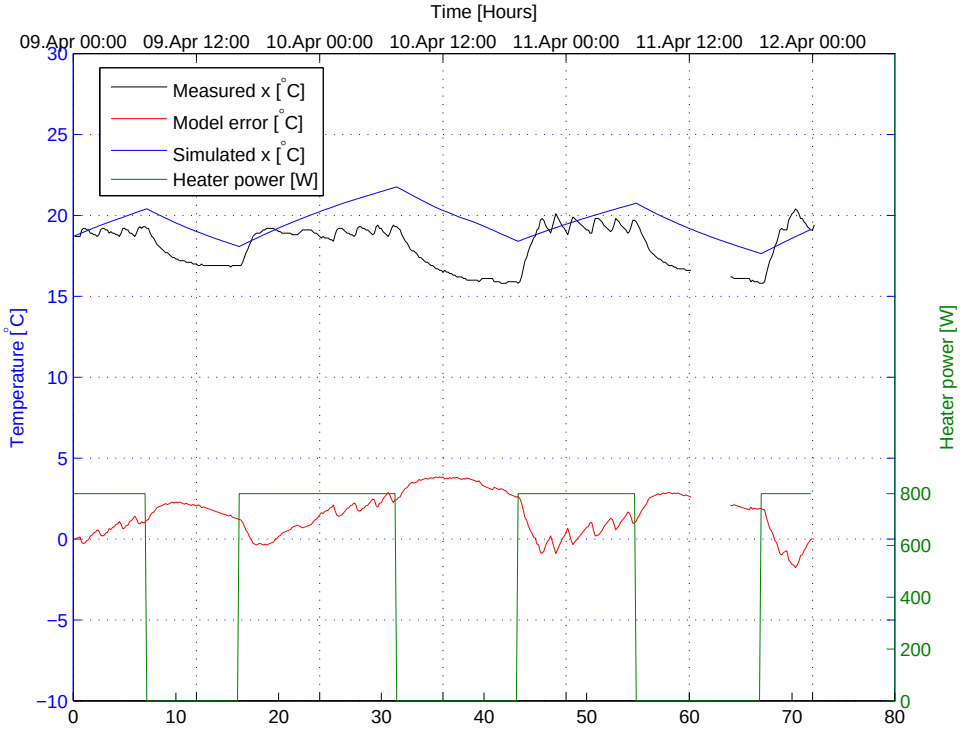

FIGURE B.14: Experiment 1: Room 5: Main bedroom

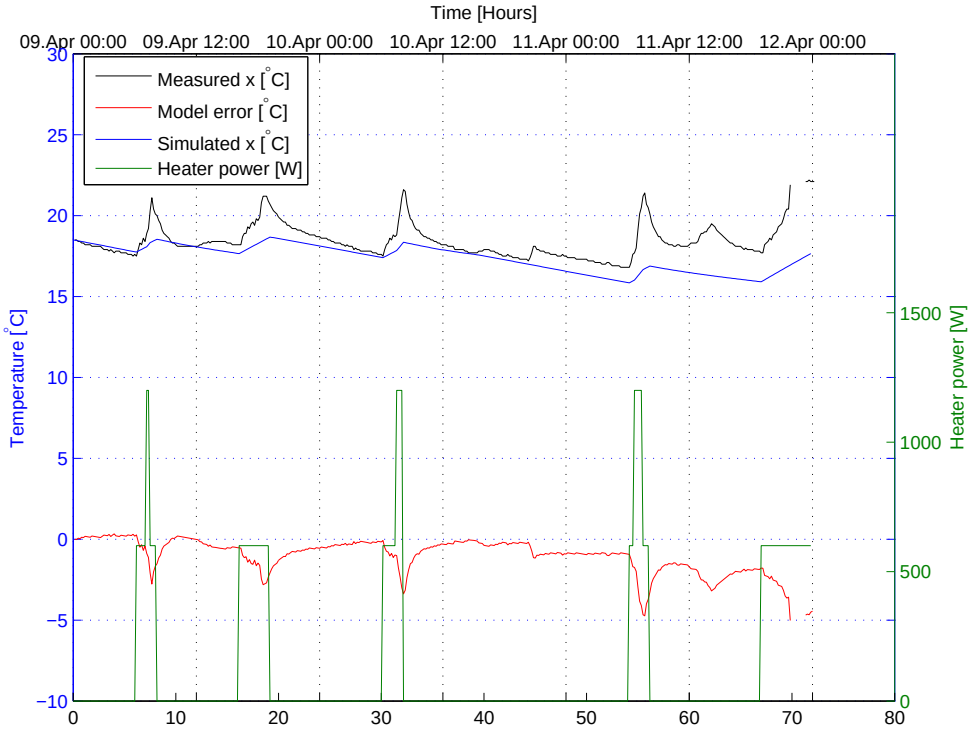

FIGURE B.15: Experiment 1: Room 6: Kitchen

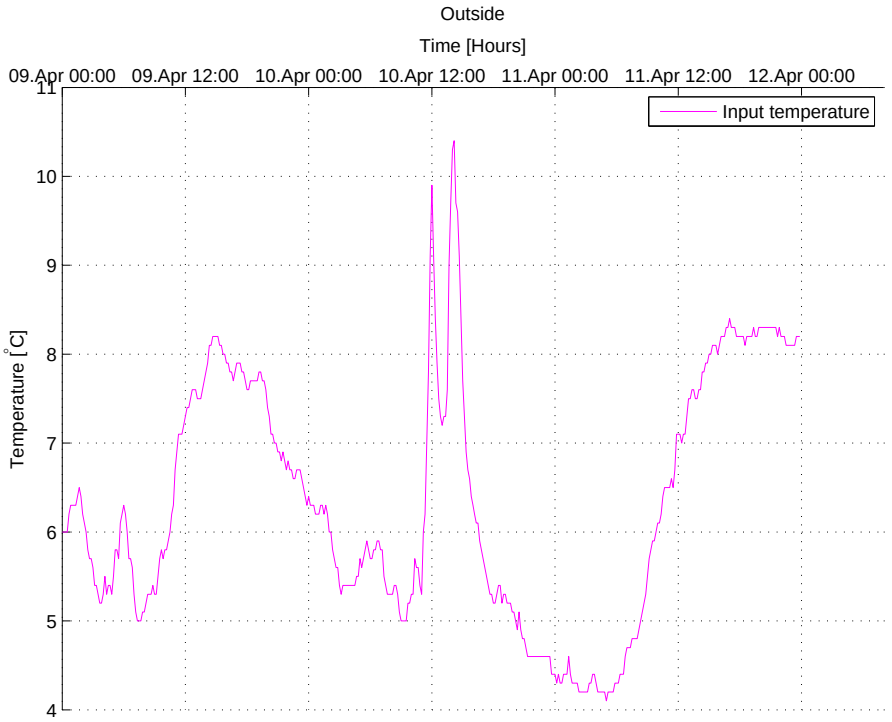

FIGURE B.16: Experiment 1: Outside

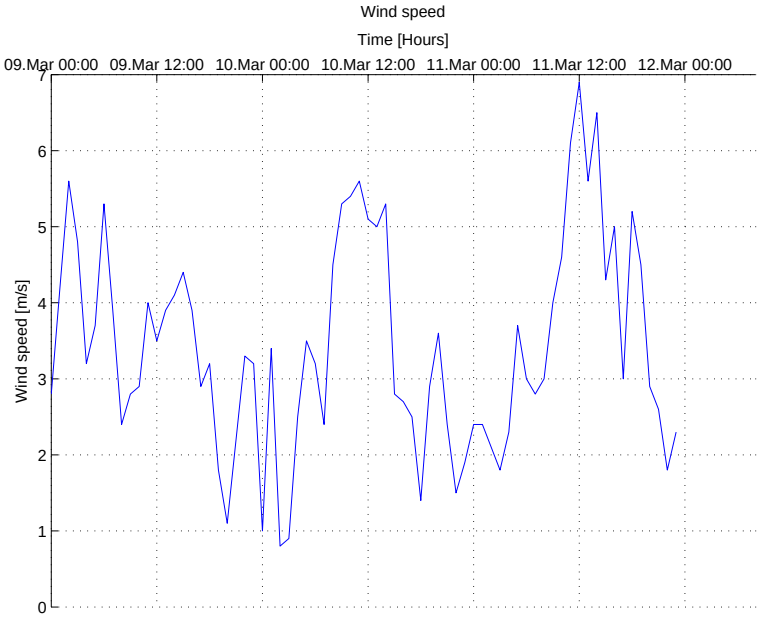

FIGURE B.17: Experiment 1: Wind speed

B.3 Experiment 2

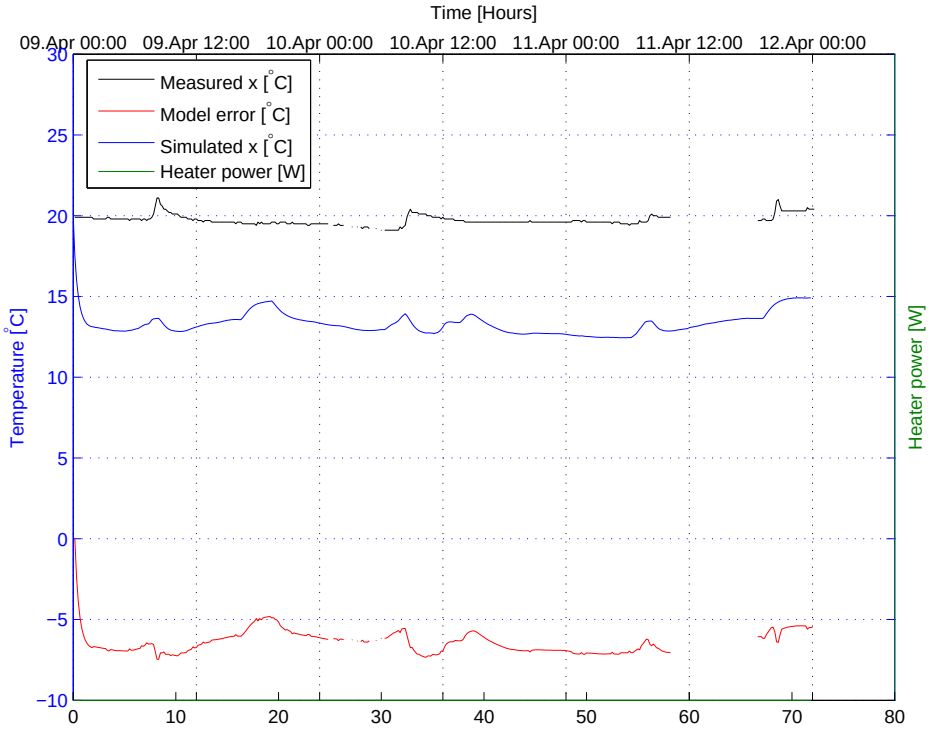

FIGURE B.18: Experiment 2: Room 1: Bathroom

FIGURE B.19: Experiment 2: Room 2: Entrance

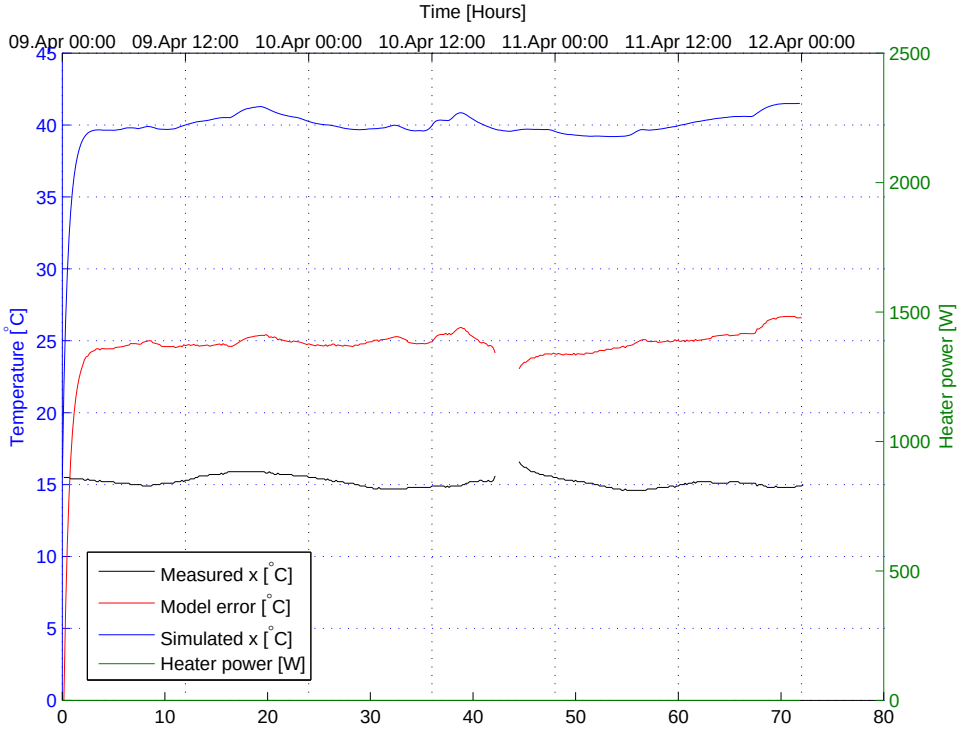

FIGURE B.20: Experiment 2: Room 3: Guest bedroom

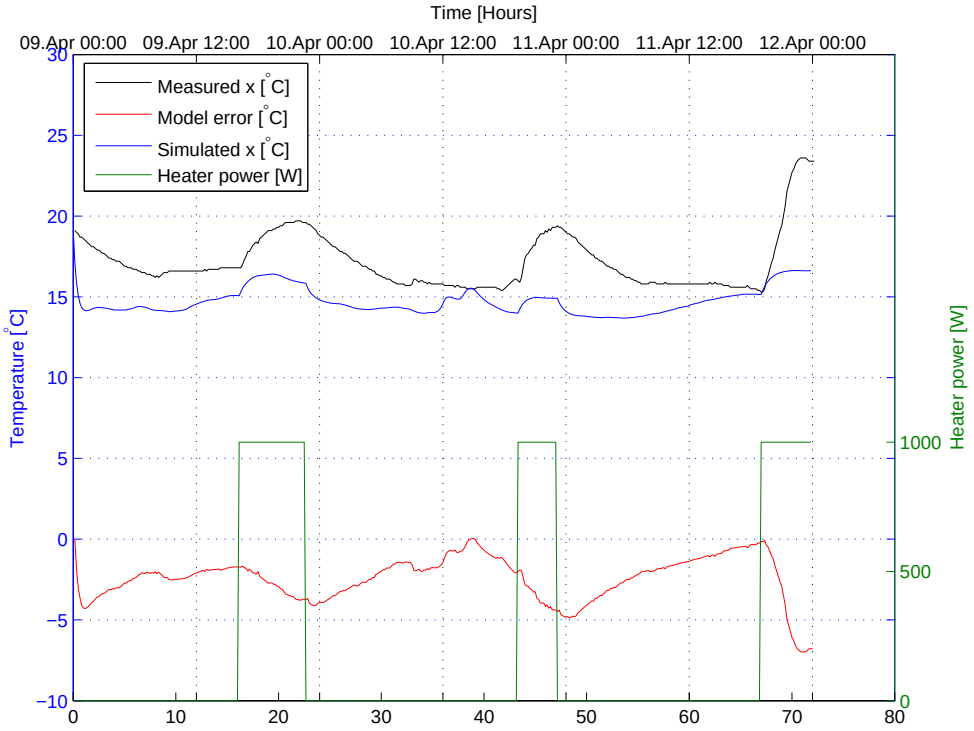

FIGURE B.21: Experiment 2: Room 4: Living room

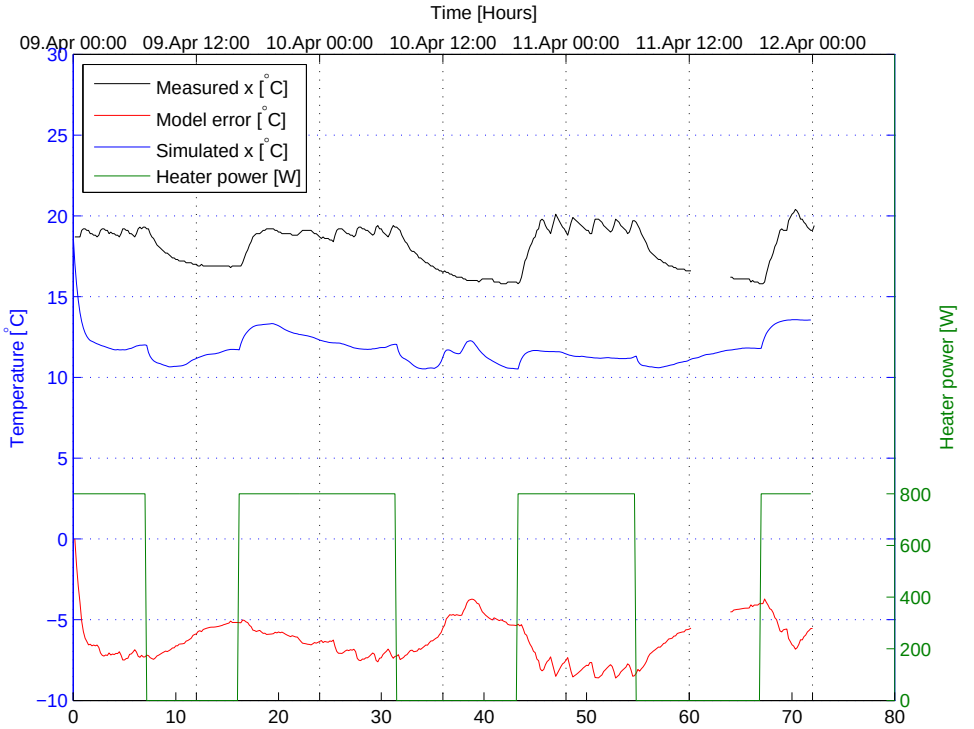

FIGURE B.22: Experiment 2: Room 5: Main bedroom

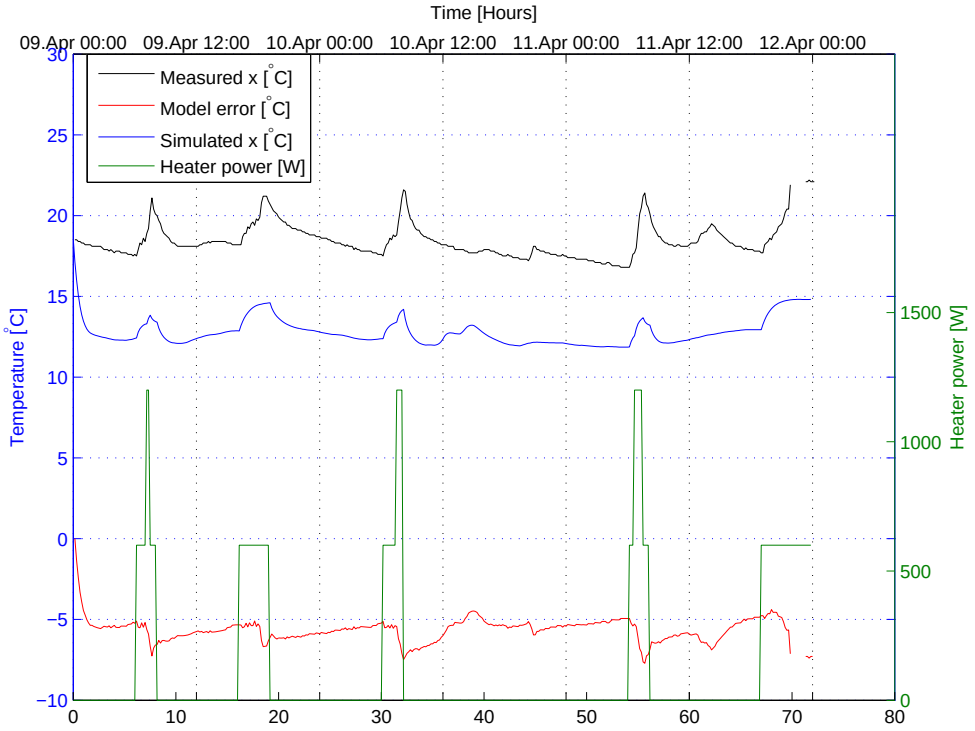

FIGURE B.23: Experiment 2: Room 6: Kitchen

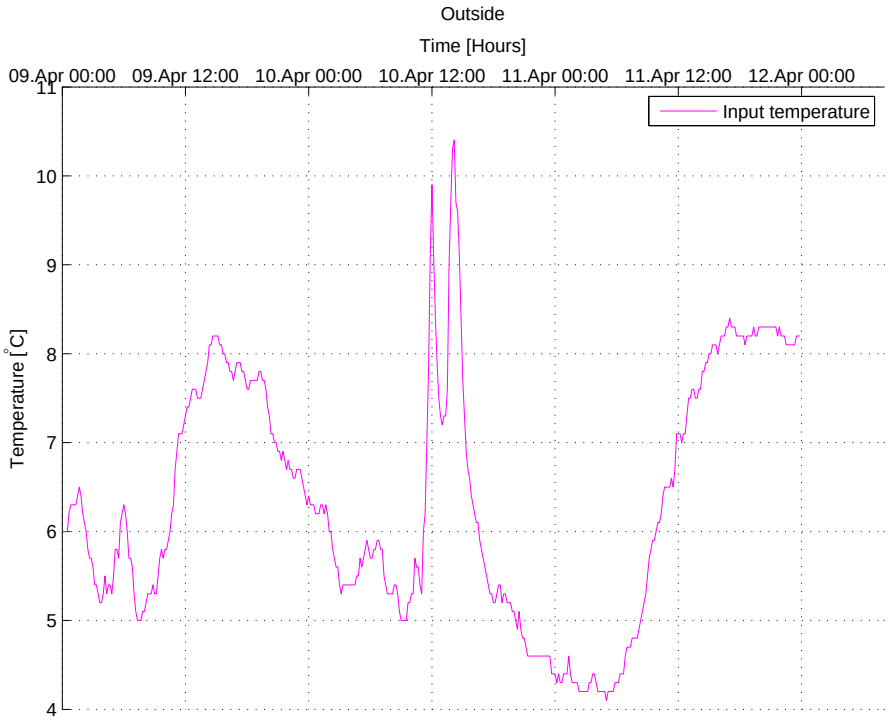

FIGURE B.24: Experiment 2: Outside

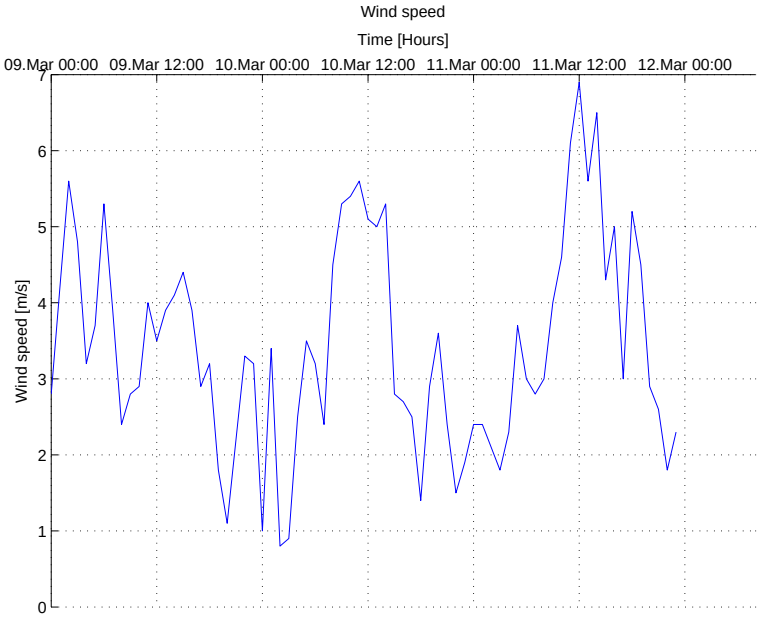

FIGURE B.25: Experiment 2: Wind speed

B.4 Experiment 3

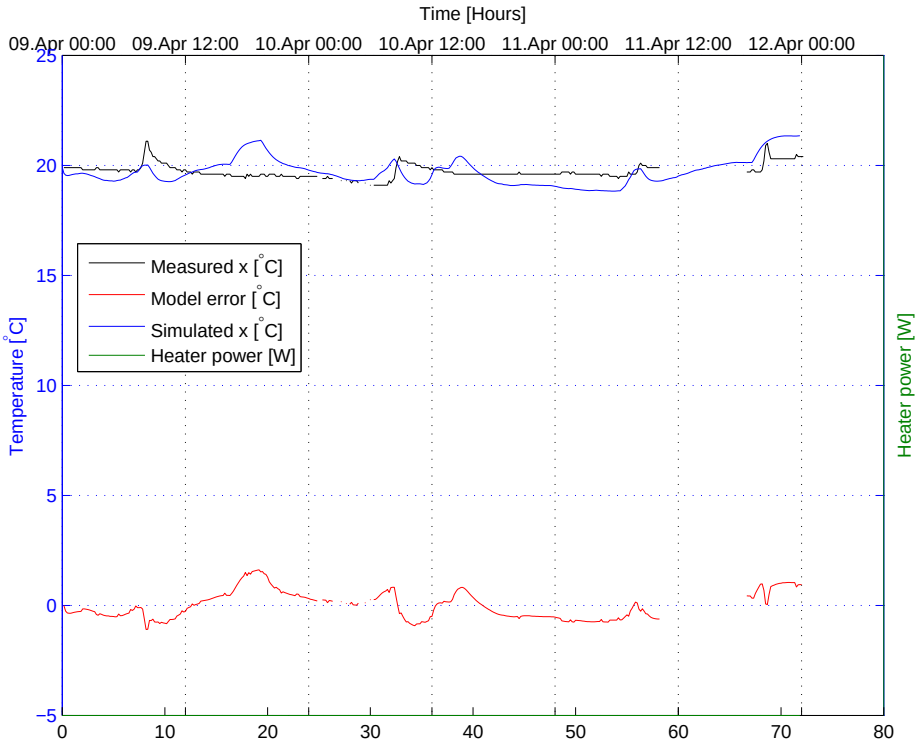

FIGURE B.26: Experiment 3: Room 1: Bathroom

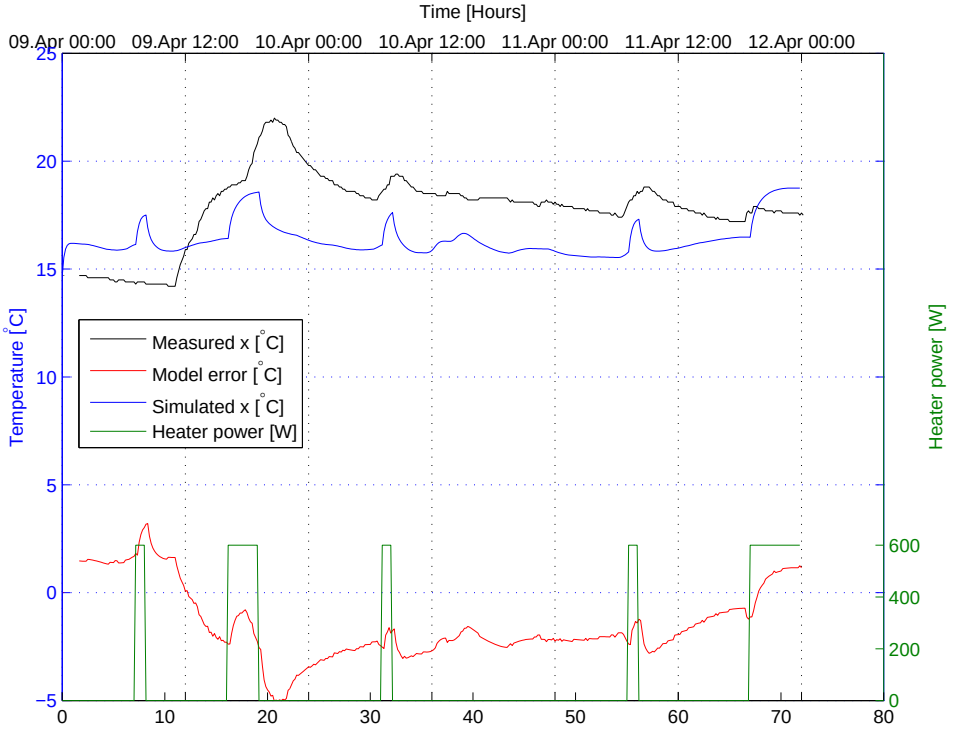

FIGURE B.27: Experiment 3: Room 2: Entrance

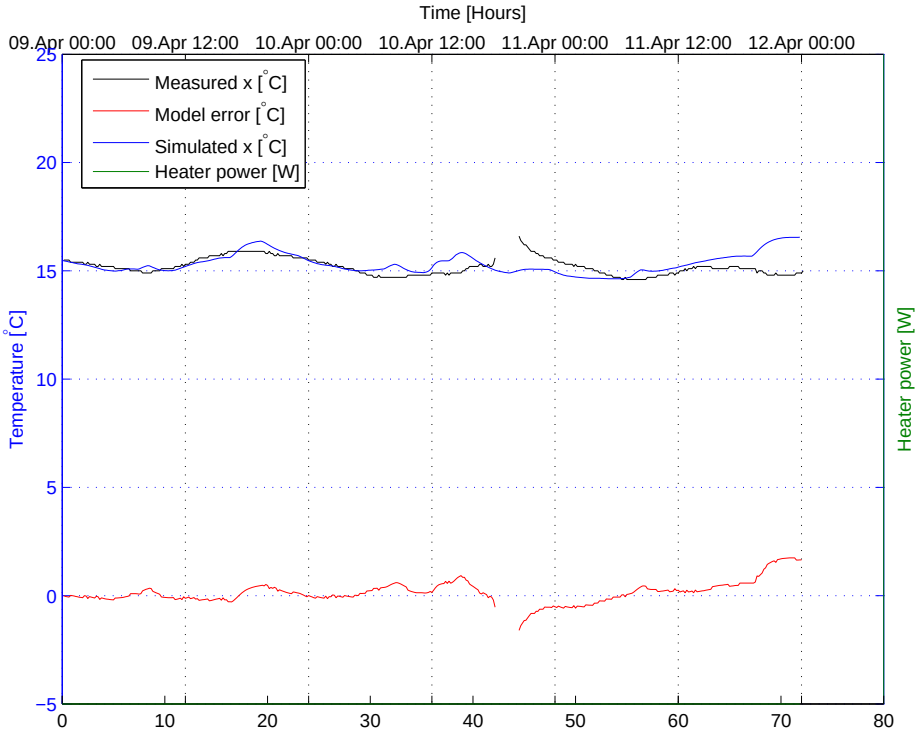

FIGURE B.28: Experiment 3: Room 3: Guest bedroom

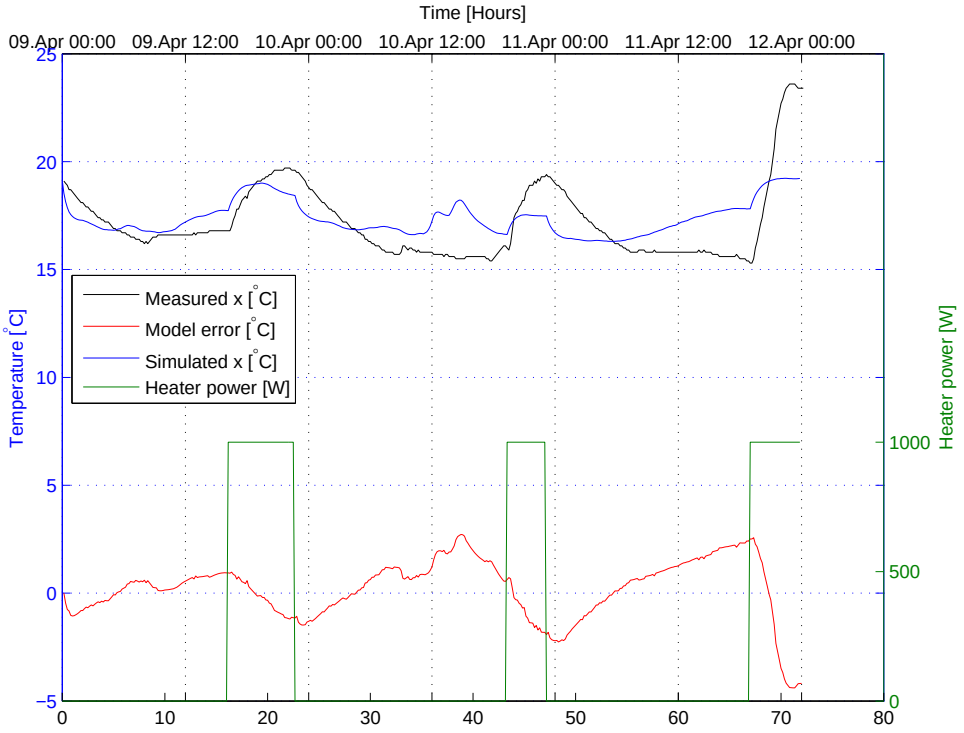

FIGURE B.29: Experiment 3: Room 4: Living room

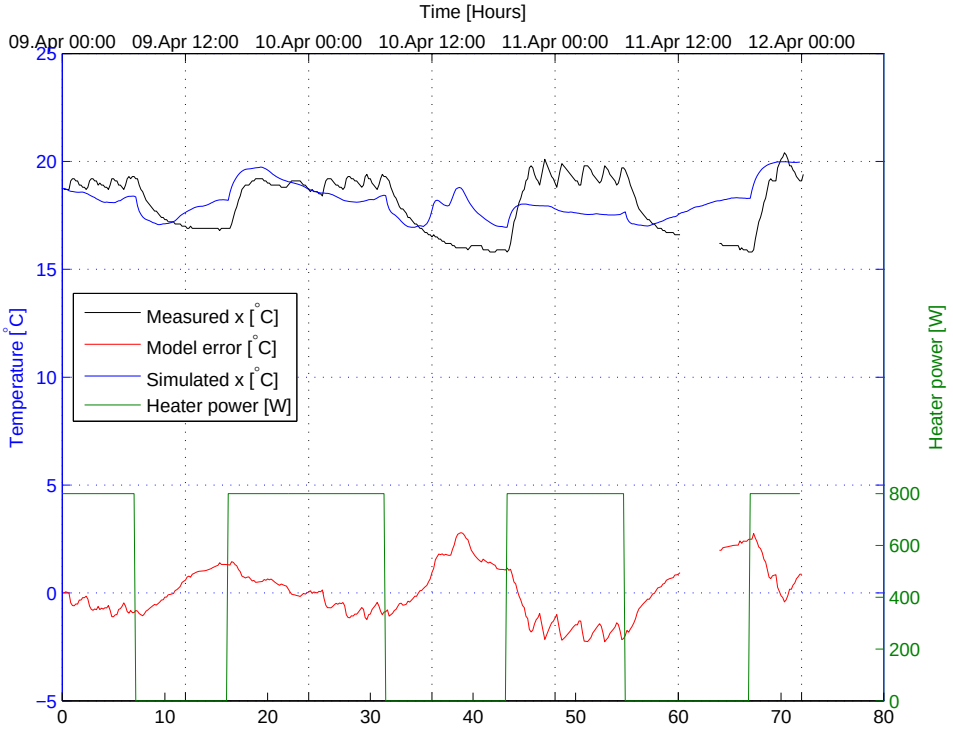

FIGURE B.30: Experiment 3: Room 5: Main bedroom

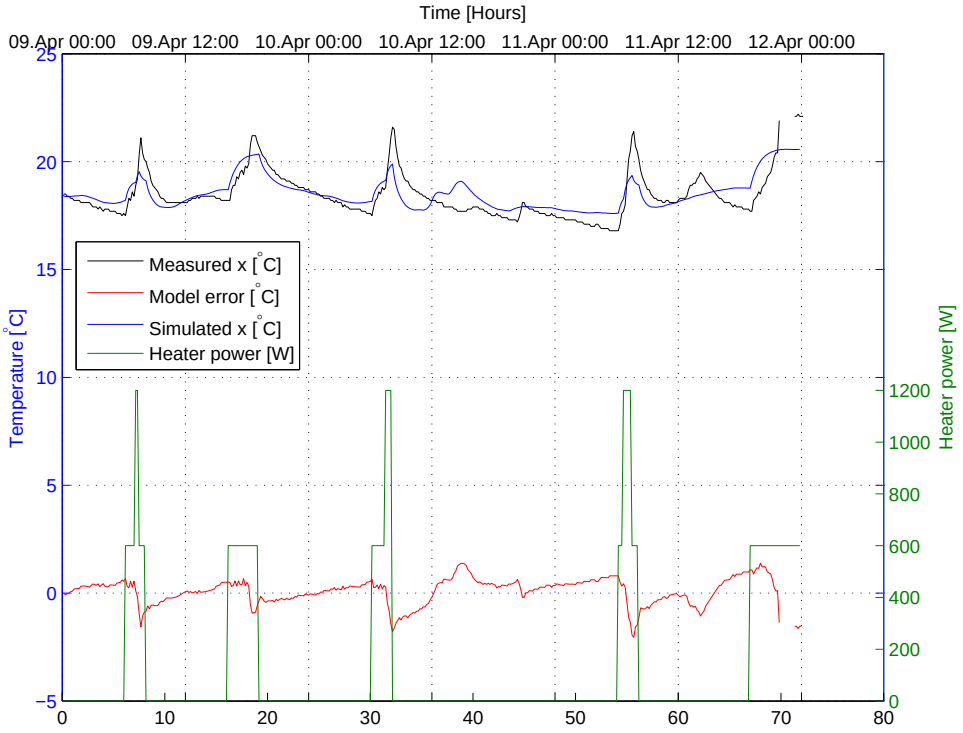

FIGURE B.31: Experiment 3: Room 6: Kitchen

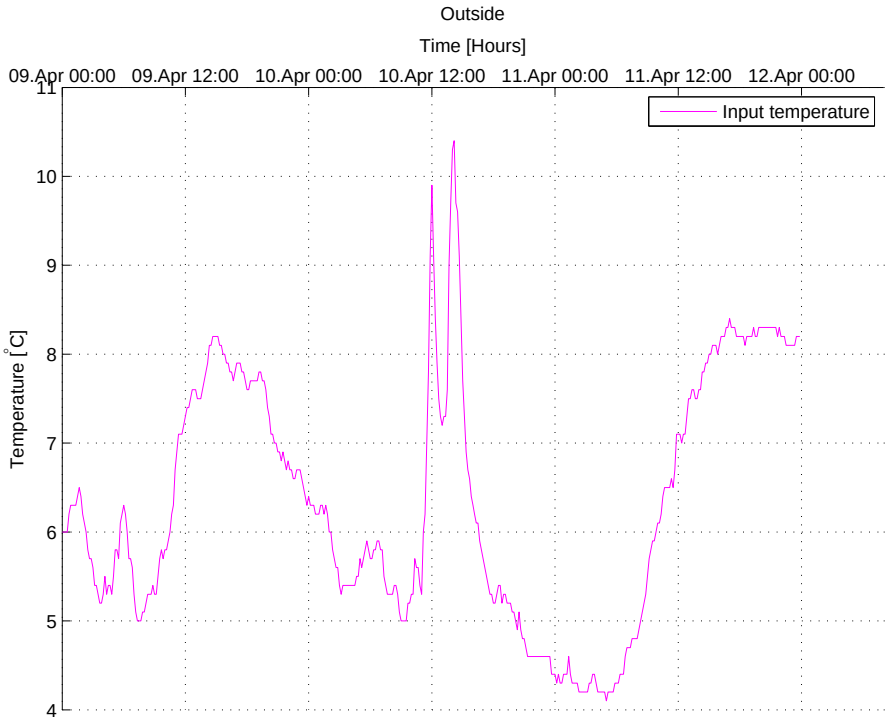

FIGURE B.32: Experiment 3: Outside

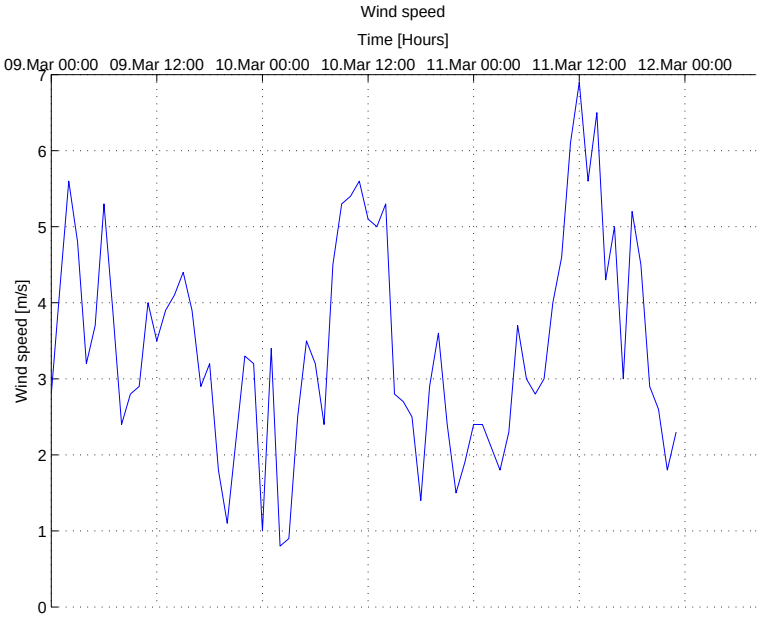

FIGURE B.33: Experiment 3: Wind speed

B.5 Kalman Filter test

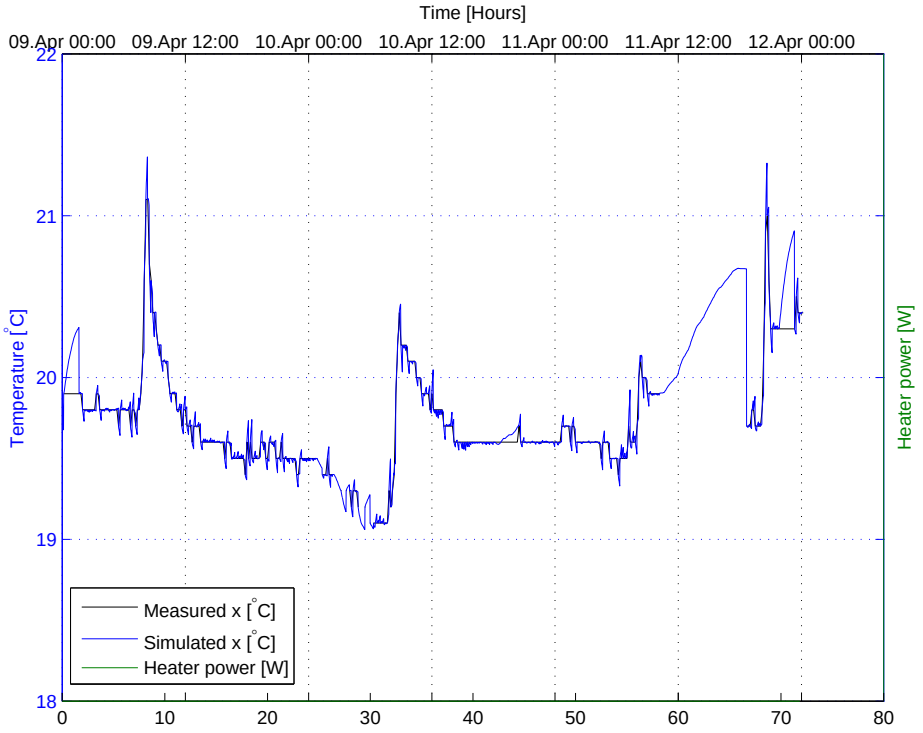

FIGURE B.34: Experiment 4: Room 1: Bathroom

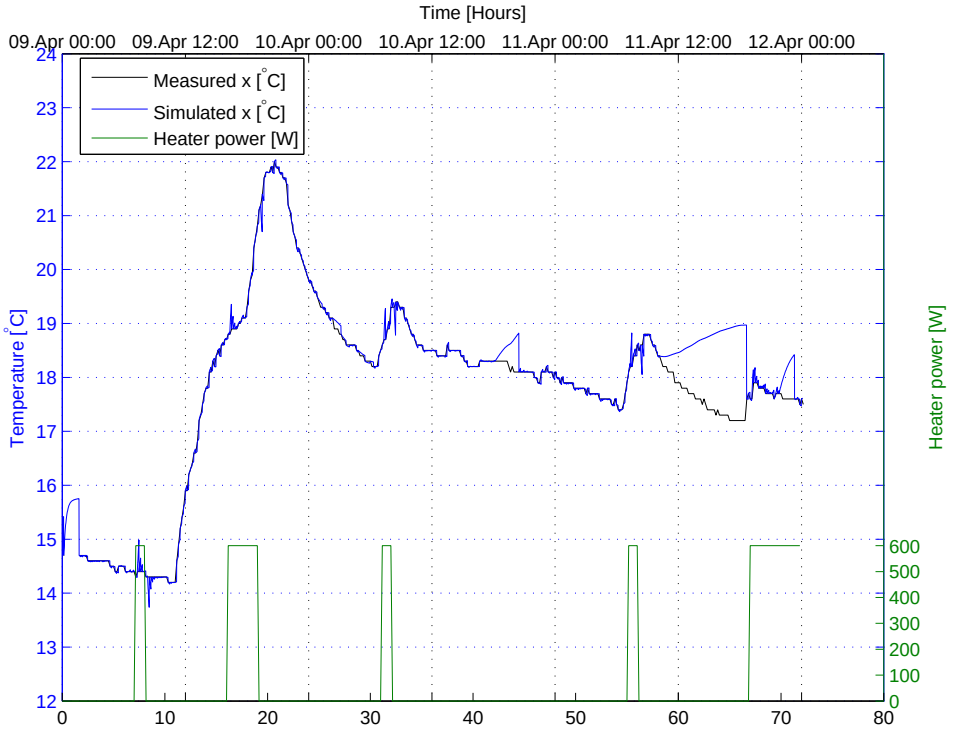

FIGURE B.35: Experiment 4: Room 2: Entrance

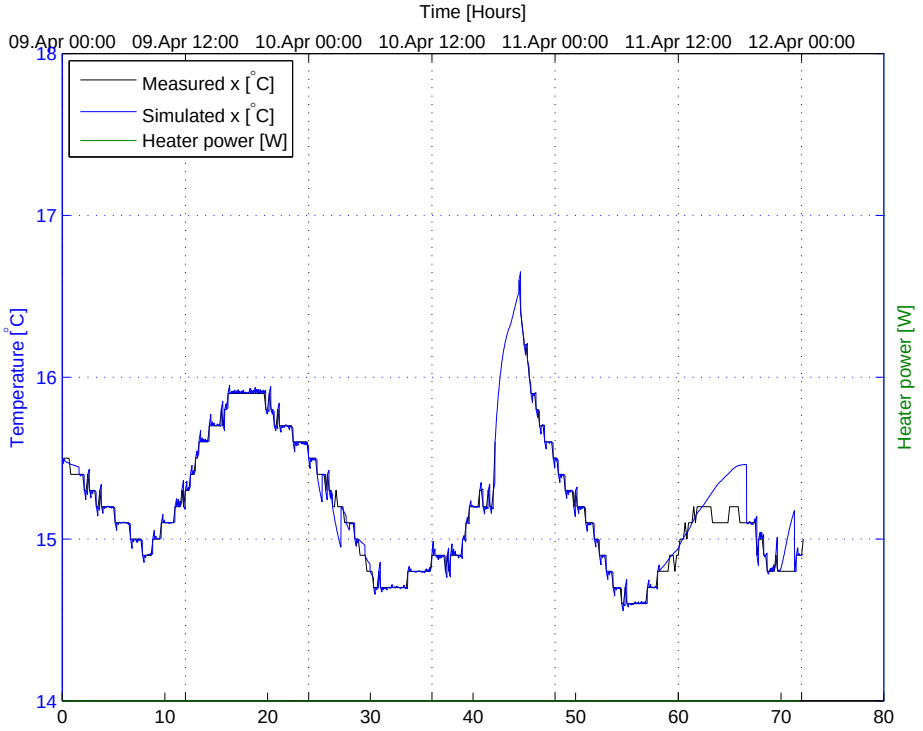

FIGURE B.36: Experiment 4: Room 3: Guest bedroom

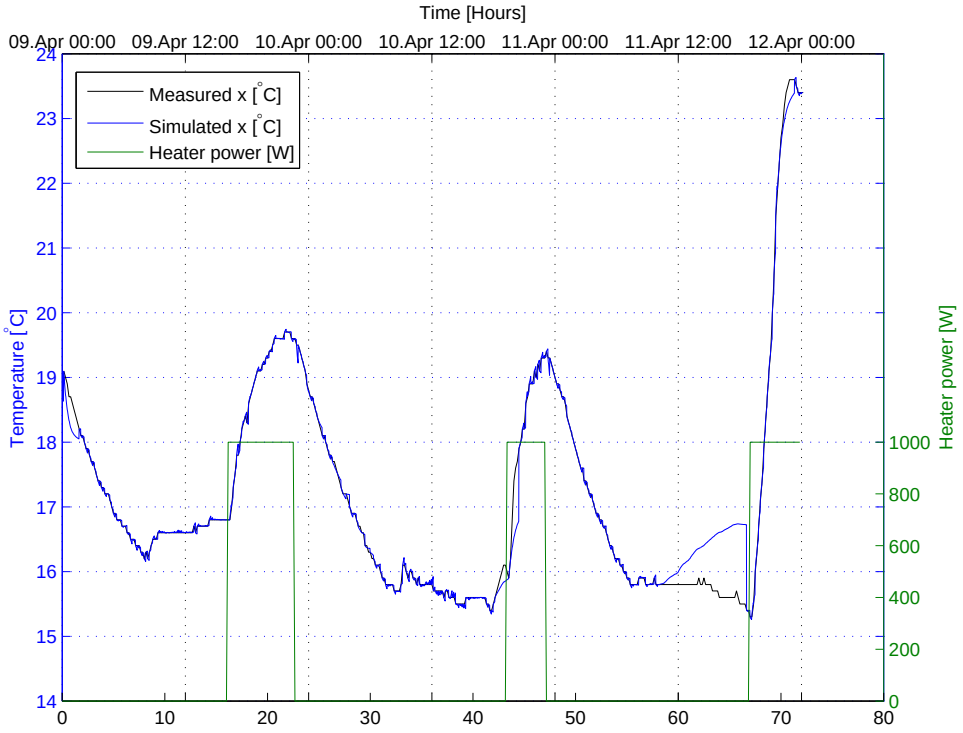

FIGURE B.37: Experiment 4: Room 4: Living room

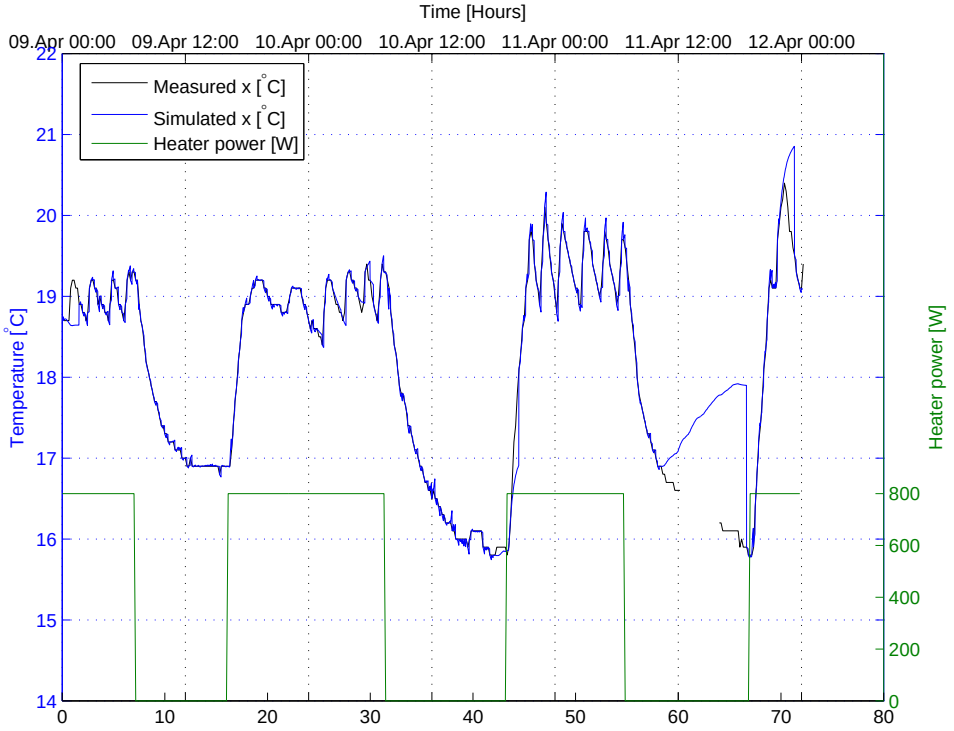

FIGURE B.38: Experiment 4: Room 5: Main bedroom

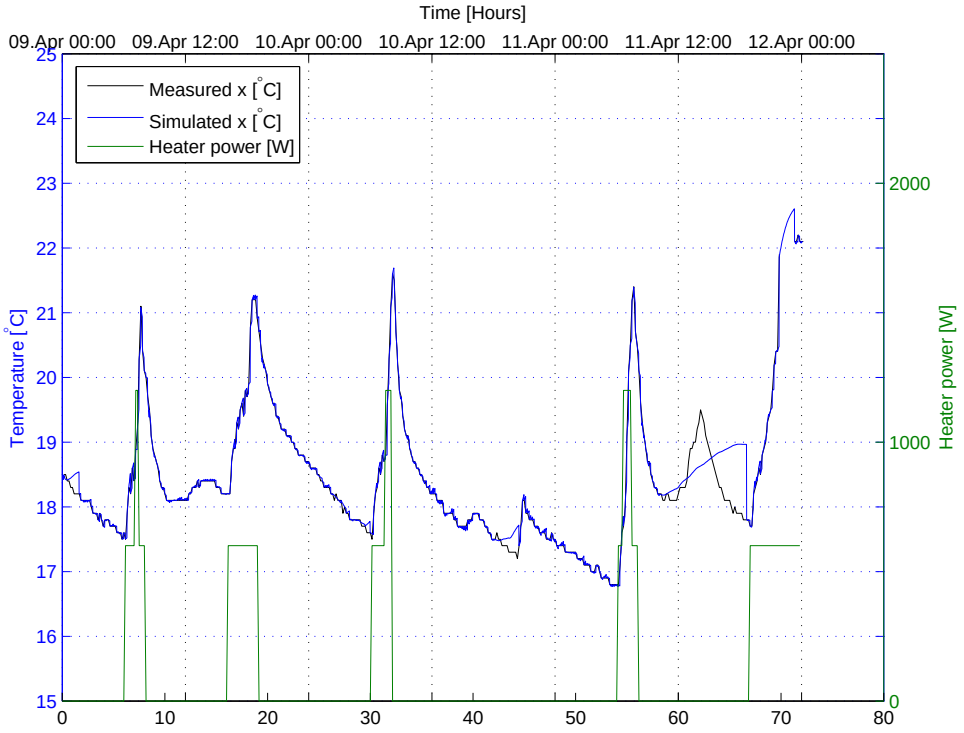

FIGURE B.39: Experiment 4: Room 6: Kitchen

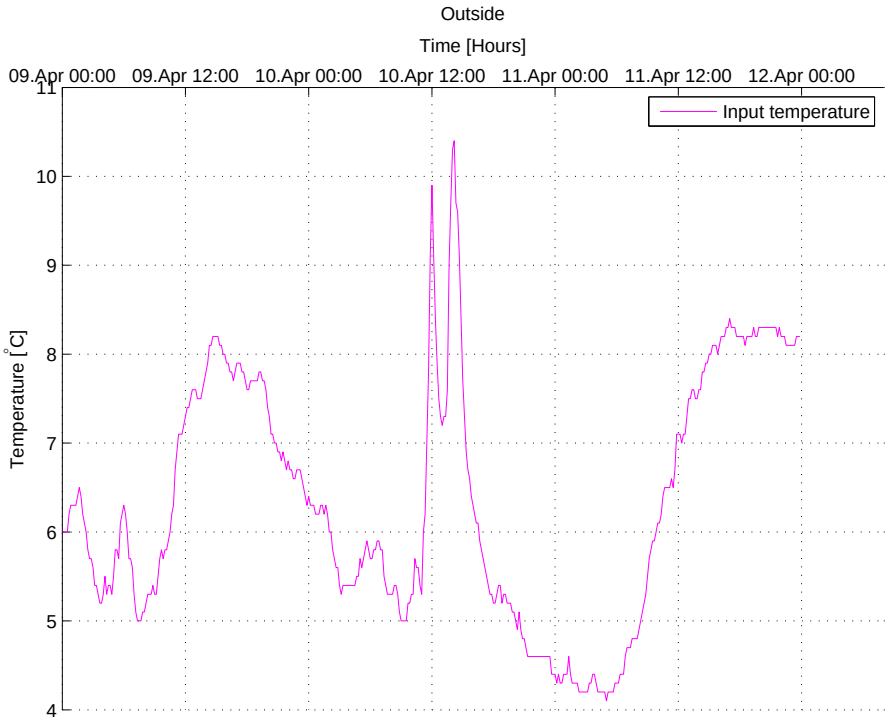

FIGURE B.40: Experiment 4: Outside

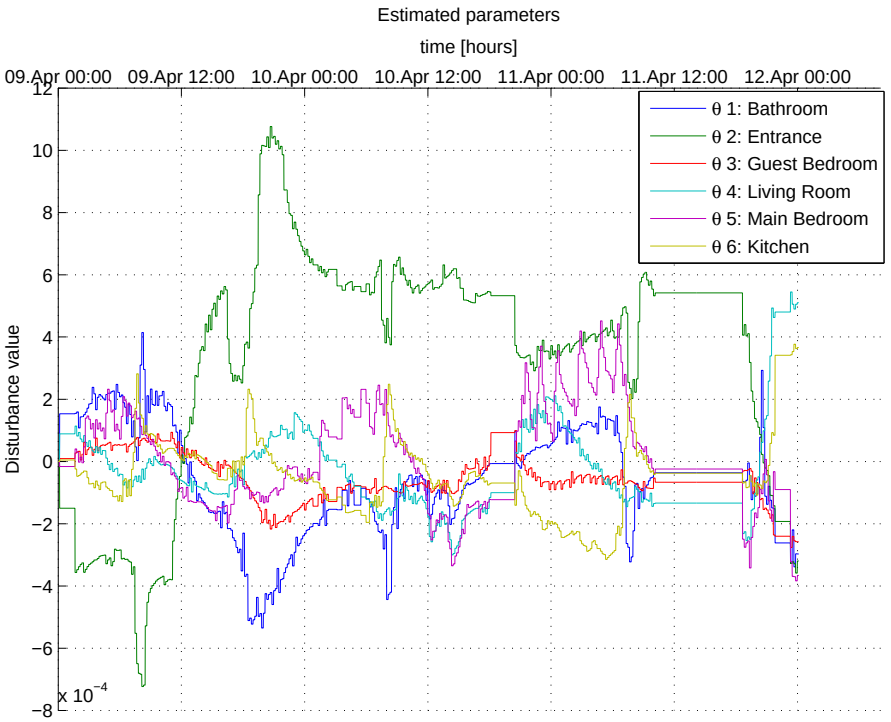

FIGURE B.41: Experiment 4: Estimated parameters

Bibliography

[1] eGotham objectives, January 2014. URL http://www.e-gotham.eu/index.php/e-gothamproj/objectives.

[2] I3RES concept page, January 2014. URL http://www.i3res.eu/v1/index.php/project2/concept2.

[3] Luis Pérez-Lombard, José Ortiz, and Christine Pout. A review on buildings energy consumption information. *Energy and Buildings*, 40(3):394–398, January 2008. ISSN 03787788. doi: 10.1016/j.enbuild.2007.03.007. URL http://linkinghub.elsevier.com/retrieve/pii/S0378778807001016.

[4] Paul A. Torcellini, M. Deru, B. Griffith, N. Long, S. Pless, R. Judkoff, and D. B. Crawley. *Lessons learned from field evaluation of six high-performance buildings*. National Renewable Energy Laboratory, 2006. URL http://www.nrel.gov/docs/fy04osti/36290.pdf.

[5] Frank P. Incropera, Adrienne S. Lavine, and David P. DeWitt. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, April 2011. ISBN 9780470501979.

[6] CEN ISO. ISO 6946: Building components and building elements - thermal resistance and thermal transmittance - calculation method, 2007. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=40968.

[7] UNIEN ISO. ISO 10456: Building materials and products — hygrothermal properties — tabulated design values and procedures for determining declared and design thermal values. 2008. URL http://www.iuav.it/Ateneo1/docenti/design-e-a/docenti-st/Romagnoni-/materiali-/laaborator/UNI-EN-ISO-10456_20081.pdf.

[8] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, Inc., New York, NY, USA, 2nd edition, 1995. ISBN 0030602890.

[9] Samuel F. Fux, Araz Ashouri, Michael J. Benz, and Lino Guzzella. EKF based self-adaptive thermal model for a passive house. *Energy and Buildings*, 68:811–817, January 2014. ISSN 03787788. doi: 10.1016/j.enbuild.2012.06.016. URL http://linkinghub.elsevier.com/retrieve/pii/S0378778812003039.

[10] Radu Bălan, Joshua Cooper, Kuo-Ming Chao, Sergiu Stan, and Radu Donca. Parameter identification and model based predictive control of temperature inside a house. *Energy and Buildings*, 43(2-3):748–758, February 2011. ISSN 03787788. doi: 10.1016/j.enbuild.2010.10.023. URL http://linkinghub.elsevier.com/retrieve/pii/S0378778810003750.

[11] Yudong Ma, Anthony Kelman, Allan Daly, and Francesco Borrelli. Predictive control for energy efficient buildings with thermal storage: Modeling, stimulation, and experiments. *IEEE Control Systems*, 32(1):44–64, February 2012. ISSN 1066-033X. doi: 10.1109/MCS.2011.2172532. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6153586.

[12] Eva Zacekova, Samuel Privara, and Zdenek Vana. Model predictive control relevant identification using partial least squares for building modeling. In *Australian Control Conference (AUCC), 2011*, page 422–427. IEEE, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6114301.

[13] Jan Široký, Frauke Oldewurtel, Jiří Cigler, and Samuel Prívara. Experimental analysis of model predictive control for an energy efficient building heating system. *Applied Energy*, 88(9):3079–3087, September 2011. ISSN 03062619. doi: 10.1016/j.apenergy.2011.03.009. URL http://linkinghub.elsevier.com/retrieve/pii/S0306261911001668.

[14] David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. URL http://www.sciencedirect.com/science/article/pii/S0005109899002149.

[15] Bjarne Foss and Aksel N. Heirung. Merging optimization and control, September 2013. URL http://www.itk.ntnu.no/fag/fordypning/TK16-filer/Samling1_MPCnotat.pdf.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960. ISSN 0098-2202. doi: 10.1115/1.3662552. URL http://dx.doi.org/10.1115/1.3662552.

[17] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, June 2006. ISBN 9780470045336. URL http://www.google.no/books?hl=no&lr=&id=UiMVoP_7TZkC&oi=fnd&pg=PR3&dq=Optimal+State+Estimation:+Kalman,+H+Infinity,+and+Nonlinear+Ap-+proaches&ots=L-Gf2LMgth&sig=Zdzf58Yv6VDXsTDEwDWTJK208zQ&redir_esc=y#v=onepage&q&f=false.

[18] Giancarlo Marafioti. Enhanced model predictive control: Dual control approach and state estimation issues. 2010. URL http://www.diva-portal.org/smash/record.jsf?pid=diva2:379307.

[19] John Bagterp Jørgensen. A critical discussion of the continuous-discrete extended kalman filter. In *European Congress of Chemical Engineering-6*, 2007. URL http://www.nt.ntnu.no/users/skoge/prost/proceedings/ecce6_sep07/upload/3520.pdf.

[20] Fredrik Lindsten, Thomas B. Schön, and Lennart Svensson. A non-degenerate rao-blackwellised particle filter for estimating static parameters in dynamical models. In *IFAC Proceedings. 16th IFAC Symposium on System Identification*, volume 16, page 1149–1154, 2012.

[21] P. Li, V. Kadirkamanathan, and R. Goodall. Estimation of parameters in a linear state space model using a rao-blackwellised particle filter. *IEE Proceedings - Control Theory and Applications*, 151(6):727–738, November 2004. ISSN 1350-2379, 1359-7035. doi: 10.1049/ip-cta:20041008. URL http://digital-library.theiet.org/content/journals/10.1049/ip-cta_20041008.

[22] Samuel Privara, Jiri Cigler, Zdenek Vana, Frauke Oldewurtel, Carina Sagerschnig, and Eva Zacekova. Building modeling as a crucial part for building predictive control. *Energy and Buildings*, 56:8–22, January 2013. ISSN 0378-7788. doi:

10.1016/j.enbuild.2012.10.024. URL http://www.sciencedirect.com/science/article/pii/S0378778812005336.

[23] Drury B. Crawley, Jon W. Hand, Michaël Kummert, and Brent T. Griffith. Contrasting the capabilities of building energy performance simulation programs. *Building and Environment*, 43(4):661–673, April 2008. ISSN 03601323. doi: 10.1016/j.buildenv.2006.10.027. URL http://linkinghub.elsevier.com/retrieve/pii/S0360132306003234.

[24] Samuel Privara, Zdenek Vana, Eva Zacekova, and Jiri Cigler. Building modeling: Selection of the most appropriate model for predictive control. *Energy and Buildings*, 55:341–350, December 2012. ISSN 0378-7788. doi: 10.1016/j.enbuild.2012.08.040. URL http://www.sciencedirect.com/science/article/pii/S0378778812004446.

[25] Peter van Overschee and Bart De Moor. *Subspace identification for linear systems*. January 1996. URL ftp://ftp.esat.kuleuven.ac.be/sista/markovsky/reports/book2x1.ps.gz.

[26] Samuel Privara, Jiri Cigler, Zdenek Vana, Lukas Ferkl, and Michael Sebek. Subspace identification of poorly excited industrial systems. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, page 4405–4410. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5717585.

[27] Michel Verhaegen and Patrick Dewilde. Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210, 1992. ISSN 0020-7179. doi: 10.1080/00207179208934363. URL http://www.tandfonline.com/doi/abs/10.1080/00207179208934363.

[28] Lennart Ljung. *System Identification Theory for User*. Prentice Hall, 2nd edition, 2009. ISBN 0-13-656695-2. URL http://een.iust.ac.ir/profs/Poshtan/Ljung%20L%20System%20Identification%20Theory%20for%20User.pdf.

[29] Siyu Wu and Jian-Qiao Sun. A physics-based linear parametric model of room temperature in office buildings. *Building and Environment*, 50:1–9, April 2012. ISSN 03601323. doi: 10.1016/j.buildenv.2011.10.005. URL http://linkinghub.elsevier.com/retrieve/pii/S0360132311003544.

[30] Jacob Chi-Man Yiu and Shengwei Wang. Multiple ARMAX modeling scheme for forecasting air conditioning system performance. *Energy Conversion and Management*, 48(8):2276–2285, August 2007. ISSN 0196-8904. doi: 10.1016/j.enconman.2007.03.018. URL http://www.sciencedirect.com/science/article/pii/S0196890407000854.

[31] Clara Verhelst, Filip Logist, Jan Van Impe, and Lieve Helsen. Study of the optimal control problem formulation for modulating air-to-water heat pumps connected to a residential floor heating system. *Energy and Buildings*, 45:43–53, February 2012. ISSN 03787788. doi: 10.1016/j.enbuild.2011.10.015. URL http://linkinghub.elsevier.com/retrieve/pii/S0378778811004592.

[32] D. Gyalistras, A. Fischlin, M. Morari, C. N. Jones, F. Oldewurtel, A. Parisio, F. Ullmann, C. Sagerschnig, and A. G. Gruner. Use of weather and occupancy forecasts for optimal building climate control. Technical report, Technical report, ETH Zürich, 2010. URL http://www.opticontrol.ethz.ch/docs/100929_OptiControl_PhaseI_FinalFlyer.pdf.

[33] M.J. Jiménez and H. Madsen. Models for describing the thermal characteristics of building components. *Building and Environment*, 43(2):152–162, February 2008. ISSN 03601323. doi: 10.1016/j.buildenv.2006.10.029. URL http://linkinghub.elsevier.com/retrieve/pii/S0360132306002952.

[34] Gilles Fraisse, Christelle Viardot, Olivier Lafabrie, and Gilbert Achard. Development of a simplified and accurate building model based on electrical analogy. *Energy and buildings*, 34(10):1017–1031, 2002. URL http://www.sciencedirect.com/science/article/pii/S0378778802000191.

[35] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, November 2007. ISBN 9780470191606.

[36] Thomas A. Louis. Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):226–233, January 1982. ISSN 0035-9246. URL http://www.jstor.org/stable/2345828.

[37] Niels Rode Kristensen, Henrik Madsen, and Sten Bay Jørgensen. Parameter estimation in stochastic grey-box models. *Automatica*, 40(2):225–237, February

2004. ISSN 0005-1098. doi: 10.1016/j.automatica.2003.10.001. URL http://www.sciencedirect.com/science/article/pii/S000510980300298X.

[38] Peder Bacher and Henrik Madsen. Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 43(7):1511–1522, July 2011. ISSN 0378-7788. doi: 10.1016/j.enbuild.2011.02.005. URL http://www.sciencedirect.com/science/article/pii/S0378778811000491.

[39] The Norwegian Meteorological Institute. B<locationforecast> - weather forecast for a specified place documentation for wether forcast API. URL http://api.met.no/weatherapi/locationforecast/1.8/documentation.

[40] MathWorks Inc. MATLAB documentation center - MathWorks nordic, 2014. URL http://www.mathworks.se/help/.

[41] Adrian Wills and Brett Ninness. On gradient-based search for multivariable system estimates. *IEEE Transactions on Automatic Control*, 53(1):298–306, February 2008. ISSN 0018-9286. doi: 10.1109/TAC.2007.914953. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4459815.

[42] GLAVA. Glava PROFF 35 product information, 2012. URL http://www.glava.no/sitefiles/1/dokumenter/ProdDok_1663_1.pdf.

[43] Norwegian Meteorological Institute. Weather statistics for trondheim (sør-trøndelag) – yr.no, 2014. URL http://www.yr.no/place/Norway/S%C3%B8r-Tr%C3%B8ndelag/Trondheim/Trondheim/statistics.html.