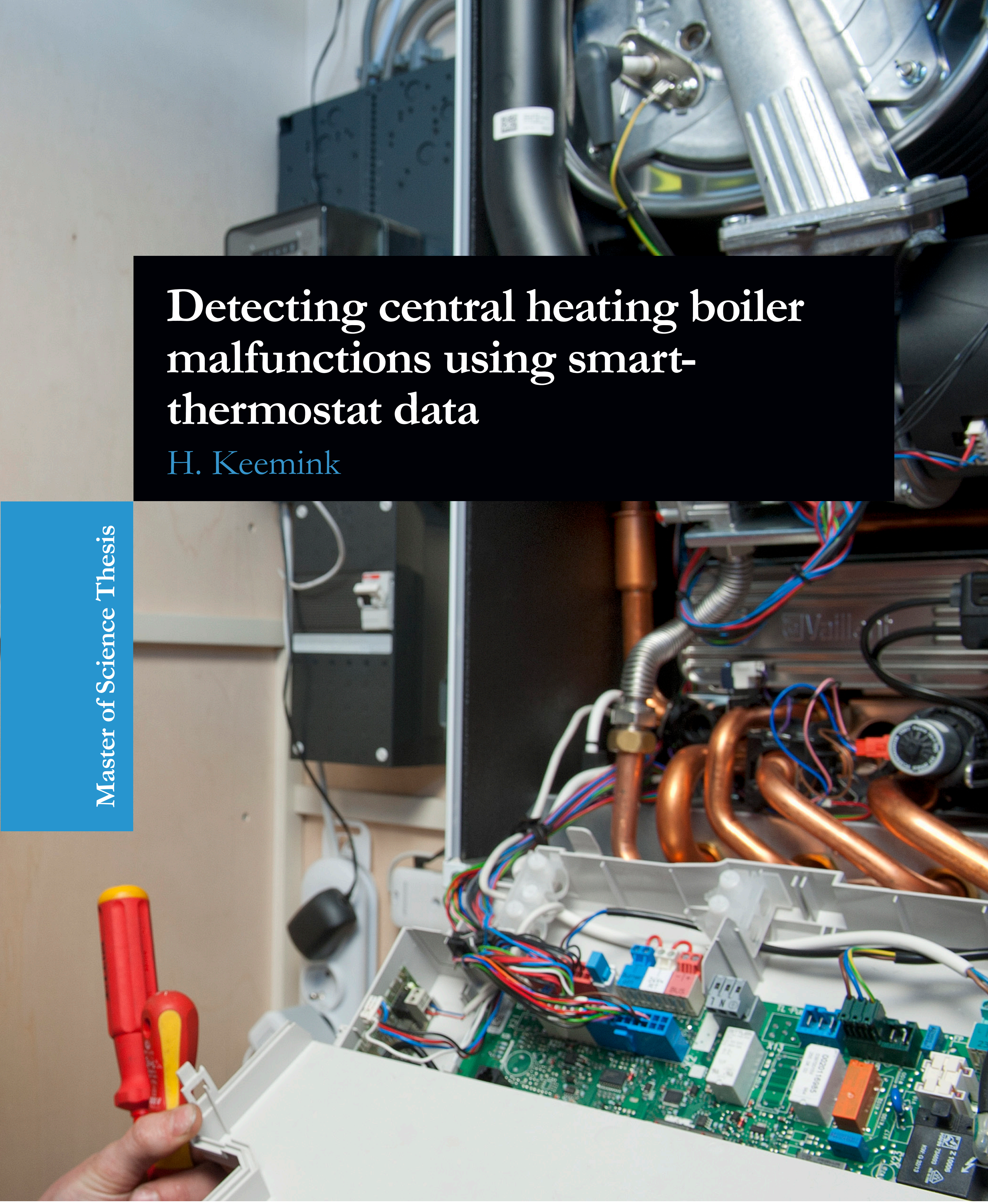

Detecting central heating boiler malfunctions using smart-thermostat data

H. Keemink

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Abstract

With the advent of smart thermostats like Toon®, detailed information about operation and usage of central heating boilers has become easily available. However, this information is not used in a systematic way by most companies including Eneco, and the wealth of information within is not available to the users.

This has a few drawbacks. For instance, when a mechanic is sent to repair a broken boiler, the mechanic has to rely on the data provided by the callcenter receiving the call from the customer. This data is often missing, incomplete or incorrect. This means that no reliable information about which parts to bring and how long the repair will take is available a priori. Secondly, some malfunctions could have been easily resolved by the end-user, for instance by refilling the system with water.

This research aims to provide a starting point in a systematic and automated approach in analysing the behaviour of the boiler by detecting malfunctions as they occur. To do so, a mathematical model of a house is designed. On this model an Extended Kalman Filter is built which monitors important parameters of the system in real-time. The estimated parameters can in future research be used as features in a more complete fault detection and identification scheme.

The filter has successfully been tested against simulated faults, and shows promising results when applied to real data.

Foreword

This research could not have been completed without the help of a group of people to whom I would like to express my gratitude. First I would like to thank my supervisor from the TU Delft: dr. ir. Jan-Willem van Wingerden, for being the most supportive and flexible supervisor one could ask for.

I would also like to thank ir. Tom Lemmens from Eneco for helping me structure this research, and for his help on managing the process of this research. Also from Eneco, I would like to thank dr. Alex Koutsman for always being there when I had technical or theoretical difficulties, and for getting me through the tough days that never ended. Furthermore, I would like to thank ir. Jan Willem van Gent for helping me explore the business value of this research, and getting me up to speed in the beginning of the project.

Apart from Tom, Alex, and Jan Willem who have acted as my supervisors from Eneco, I would like to thank the team at Eneco Innovations and Ventures, and especially ir. Jasper Müller, Dennis Ramondt, MSc and Mathias Veenman, MSc for welcoming me into the group and really making me feel at home. I would also like to thank dr. Fonger Ypma for giving me the opportunity to do this research at Eneco in the first place.

I would like to express special thanks to Sanne de Backker, MSc RC for her support throughout the project; she was there when I needed it most. Last, but not least, I would like to thank my family and friends for all the good times and much needed distractions during this research.

Amsterdam, The Netherlands — Huib Keemink
26 November 2016

Contents

Abstract i

Foreword iii

1 Introduction 1
1.1 Background 1
1.2 Problem statement 2
1.3 Methodological Approach 3
1.4 How to read this thesis 4

2 Central Heating Systems 5
2.1 Central heating boilers 5
2.2 Thermostats 7
2.2.1 Toon® 8
2.3 The OpenTherm standard 9
2.4 Available Data 10

3 Modeling 13
3.1 Modeling a heated house 13
3.1.1 Model structure 13
3.1.2 Validation 22
3.1.3 Model selection 24
3.2 Modeling most common central heating faults 26
3.2.1 Low water pressure 26
3.2.2 Stuck three-way valve 27
3.2.3 Gradual performance degradation 28
3.2.4 Closed radiators 29
3.2.5 Heating with the windows open 29
3.2.6 Summary 30

4 Online Parameter Estimation 31
4.1 Kalman filter 31
4.2 Extended Kalman filter 32
4.3 Using the Extended Kalman filter for parameter estimation . . 34

5 Results **37**
5.1 Tracking parameters without noise 38
5.2 Introducing noise to the model 39
5.3 Tracking time-varying parameters 40
5.3.1 Sudden loss of heating 40
5.3.2 Sudden increase in heat loss 42
5.3.3 Slow degradation of heating 44
5.3.4 Slow increase in heat loss 44
5.4 Applying the filter to data of a household 45
5.5 Effect of covariance windup 48
5.6 Summary 50

6 Conclusions & Recommendations **53**
6.1 Conclusions 53
6.2 Recommendations 54

Appendix A: Python base class **57**

Appendix B: Python EKF implementation **61**

Appendix C: Python house model **63**

Bibliography **65**

List of Figures

1.1 Screenshot of the recently launched BoilerIQ service 2
1.2 Structure of this thesis 4

2.1 Schematic overview of a condensing boiler[2] 6
2.2 Efficiency curve of a condensing boiler. Note how the effiency rises sharply above the dew point of the exhaust gasses[3] 7
2.3 The Toon® display. 8

3.1 Conduction and convection around a wall.[1] 15
3.2 Solar radiation throughout the year according to the KNMI, aggregated per month. 18
3.3 RC Diagram of heated house model 22
3.4 Comparison of simulation with real measured data of the chosen model 25

5.1 Schematic overview of parameter estimation setup using the Extended Kalman Filter (EKF). 37
5.2 Tracking two parameters: Q and β. 38
5.3 Room temperature with additive measurement noise ($\sigma = 0.05$) . 39
5.4 Tracking Q and β in the presence of additive measurement noise 40
5.5 Tracking Q and β when Q suddenly drops to zero. 41
5.6 Detail view of the sudden drop in Q. 41
5.7 Tracking Q and β when β suddenly increases significantly, without tuning the covariance matrices 42
5.8 Detail view of the sudden increase in β, without tuning the covariance matrices 42
5.9 Tracking Q and β when β suddenly increases significantly, with tuned covariance matrices for this specific failure 43
5.10 Detail view of the sudden increase in β, with tuned covariance matrices for this specific failure 43
5.11 Tracking Q and β when Q slowly drops to zero. 44

[2]Harish Satyavada and Simone Baldi. "A Novel Modelling Approach for Condensing Boilers Based on Hybrid Dynamical Systems". *Machines* 4, pp. 10–10, 2016.

[3]Simone Baldi et al. "Real-time monitoring energy efficiency and performance degradation of condensing boilers ".

5.12 Tracking Q and β when β slowly increases. 45
5.13 Tracking Q over a period of 10 days for a real household 46
5.14 Detail view of the dip in Q when the filter is applied to real data, showing a fault 47
5.15 Tracking the wall temperature over the course of ten days 48
5.16 Effect of lack of excitation on covariance 49
5.17 Effect of ad-hoc approach limiting the windup of covariance, resulting in more stable tracking of Q 50

List of Tables

2.1 Demonstration of faultcodes meaning different things for different types and brands 10
2.2 Statistics on selected variables in provided dataset 11

3.1 comparison of performance of different models 24
3.2 overview of parameters used in model. 26
3.3 Overview of different faults, where "HW" stands for Hot Water. 30

1 Introduction

1.1 Background

With the slow shift of the energy industry from fossil fuels to more sustainable sources of energy, the position of the energy providers is changing. Eneco - a major energy provider in the Netherlands - calls this “de nieuwe wereld” or “the new world” in english.

In this changing landscape, energy providers are looking for differentiators to keep customers loyal. Price can be one of these differentiators, and this is a tactic many new players on the market are using. For a big corporation with a relatively high overhead like Eneco, this is a not a viable strategy. For this reason Eneco is always on the lookout for new ways of binding customers.

Toon® is one of the differentiators Eneco is offering. With this smart thermostat, customers can save on their energy bill by cutting on their consumption. This might seem counterintuitive, because Eneco is effectively helping customers buy less of its product. However, by charging money for these services, Eneco is moving from a supplier of energy to a provider of services around energy.

With this in mind, it is in the best interest of Eneco to sell as many Toons as possible. However, thermostats by themselves are not very exciting. The better the thermostat, the fewer interactions the customer has with it. To keep the Toon® interesting for customers, additional services are offered. Most of these services are built around insight: “how much energy are you consuming compared to your neighbour?” or convenience: “turn your heating on (or off) remotely”.

Eneco has recently started adding a new service to its lineup called BoilerIQ. With BoilerIQ any faultcode transmitted by the boiler is registered and translated into an meaningful message on the Toon®. With these messages a mechanic can be requested via Eneco. The benefit from the side of the mechanic is that information about the boiler and its health is available, since the entire fault history of the boiler is available.

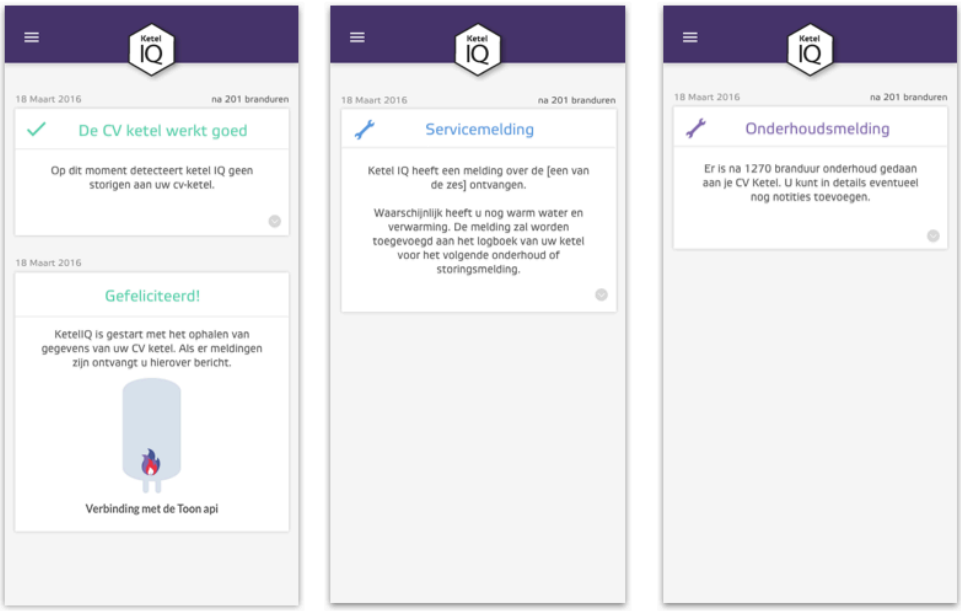

Figure 1.1: Screenshot of the recently launched BoilerIQ service

The benefit for the customer is that the hassle is reduced: calling a mechanic and taking a day off, only to find that the mechanic does not have the right parts in his van.

The service is built around fault codes transmitted over the OpenTherm protocol - one of the two protocols supported by Toon, see Section 2.3. Unfortunately only around 40% of boilers connected to Toon® is using the OpenTherm protocol, the rest is connected via the traditional On-Off method, where fault-codes are not available. Of those 40% communicating via OpenTherm, only around 30% of boilers actually transmits fault codes. *This means that currently, only roughly 12% of households can use this service.* [1]

1.2 Problem statement

As described in the previous section, services like BoilerIQ on Toon® that generate revenue and increase customer loyalty are valuable to Eneco. However, as it currently stands, the service does not reach most of the customer-base of Eneco.

The primary goal of this research is to create a fault-detection scheme which

[1] Numbers taken from Eneco documents and results of pilots run by Eneco

can be applied to all boilers, regardless of protocol, brand or type.

In addition to the primary goal, the following secondary objectives have been formulated.

1) Faults are to be detected over a longer period of time. Traditionally, mechanics have a few busy months when the weather starts getting colder during the autumn, as this is when residents notice their houses are not heated properly. By having an automated detection scheme, it is possible to spread repairs over a longer period of time, reducing costs.
2) Faults not noticed directly by residents should be detected. This can improve energy efficiency, for instance by notifying customers when the radiators in the living room are turned off and all heat ends up in the bathroom.

These lead to the following research question:

How can the most common faults in a domestic central heating system be detected using data from Toon® that is available for all households?

1.3 Methodological Approach

The data provided by Eneco does not contain reliably labelled faults due to how the OpenTherm standard is implemented (see Chapter 2), and the avaible data can not be used directly as a training set for classification algorithms. One option would have been to label the set by hand. However, faults are expected to be rare (on average, boilers break down once every three years, or 1000 working days versus 1 faulty day), and labelling by hand would require a lot of time of an expert, which is simply not available.

Therefore, a different approach is taken. First, a model of a house based on thermodynamic properties is built, which is then validated against the measured data (see Chapter 3). Then, well-known faults are introduced into this model, resulting in a labelled set of measurements that can be used to test the detection scheme. The faults are chosen so that they correspond to the most common faults in a central heating system, keeping ease of implementation in mind.

The detection scheme is chosen to be an online parameter estimator using an Extended Kalman filter (see Chapter 4) which tracks the heat given off by the radiators and the loss of heat to the outside, since these are important indicators of a fault. This approach has the added benefit of being easily implemented in Python, an open source programming language which is used within Eneco.

Finally, when the detection scheme is tested against simulated data (see Chapter 5), real data is used to show the effectiveness of the scheme on less-perfect data.

1.4 How to read this thesis

The main content of this thesis is split into four parts. First, what the available data is, how it is measured and what issues it has is described in Chapter 2: "Central Heating Boilers". The mathematical derivation of the model and common faults are described in Chapter 3: "Modeling". Then, a method of detecting these faults from the simulated data is explained in Chapter 4: "Online Parameter Estimation". Finally, the results of this scheme are discussed in Chapter 5: "Results".

To see how these chapters relate to each other, see Fig. 1.2.

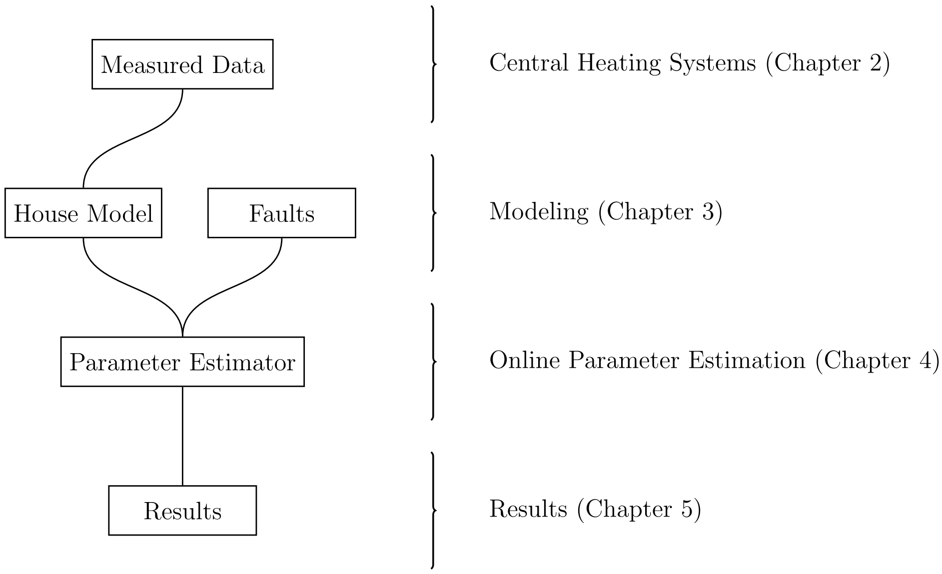

Figure 1.2: Structure of this thesis

2 Central Heating Systems

In this chapter the two most important aspects of a central heating system are described; boilers and thermostats. The goal of this chapter is to give some background into the different elements of the system, and to understand what kind of issues Eneco is running into while building smart applications for these central heating systems.

2.1 Central heating boilers

One way of heating a home is by using a central heating boiler. This boiler heats water, which is pumped through the home. The hot water gives off heat in the rooms via radiators, thereby heating the room. Other methods of heating the home are fireplaces, district- and block heating[1]. Toon® only works with central heating boilers, and other forms of heating will not be discussed in this thesis.

In the customerbase of Eneco, high efficiency condensing boilers are the most common. A schematic overview of a condensing boiler is given in Fig. 2.1.

In Fig. 2.1, the condensing part is the most interesting. After the hot combustion gasses are passed through the first heat exchanger, they are transferred to a second heat exchanger. The water in this heat exchanger is coming from the house directly, and has the purpose of cooling the gasses down below their dew point.

The reason this adds quite a bit of efficiency to the system is because the exhaust gasses from a non-condensing boiler are both hot and wet. When the water in the exhaust gas is removed via condensation, a portion of the energy remaining in the exhaust gasses can be extracted.

Note that the return water ("Cold water in" in Fig. 2.1) must be sufficiently cool in order for the condensation to occur. When the temperature of the re-

[1] Urban Persson and Sven Werner. "Heat distribution and the future competitiveness of district heating". *Applied Energy* 88, pp. 568–576, 2011.

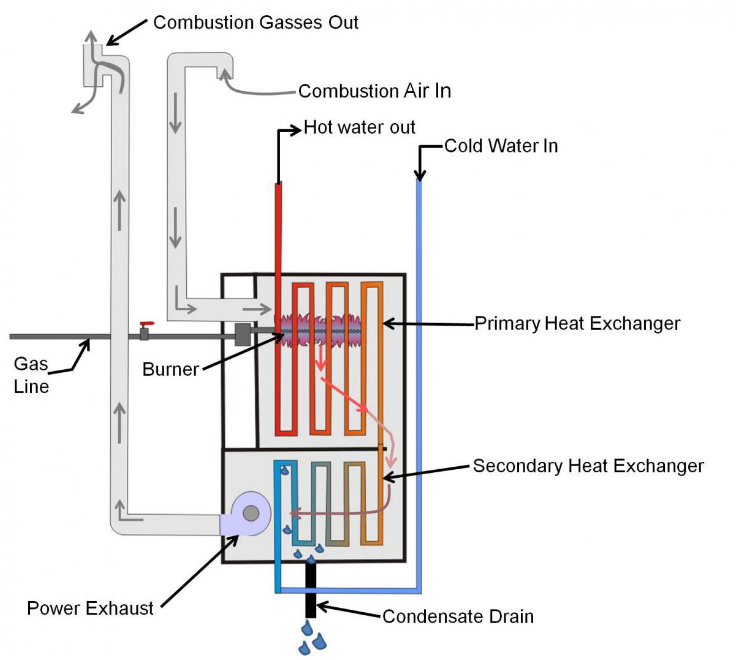

Figure 2.1: Schematic overview of a condensing boiler[2]

turn water is above the dew point, no condensation will occur, and the boiler will effectively be a non-condensing boiler.[3] The relation between return water and effiency is given in Fig. 2.2.

These boilers usually serve two purposes: to heat the house and to provide hot tap water. To prevent the formation of cavitating bubbles and calcium buildup, the boiler is a closed system. When the boiler is set to heat the house, hot water is pumped through the house, where it gives off heat via the radiators.[2]

When the boiler is set to provide hot tap water, the three way valve usually redirects the flow of hot water through a heat exchanger, which heats the tap water. Cooled down, the water in the closed loop is returned to the boiler, where the cycle starts again. More exotic systems where both tap and heating water are directly fed into the boiler exist, but a situation with a heat exchanger and a three way valve is most common.

[3] Simone Baldi et al. "Real-time monitoring energy efficiency and performance degradation of condensing boilers ".

[2] Harish Satyavada and Simone Baldi. "A Novel Modelling Approach for Condensing Boilers Based on Hybrid Dynamical Systems". *Machines* 4, pp. 10–10, 2016.

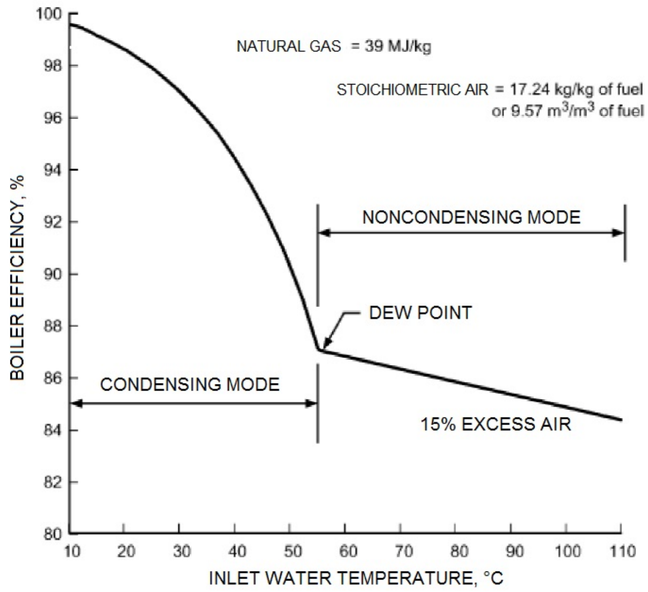

Figure 2.2: Efficiency curve of a condensing boiler. Note how the effiency rises sharply above the dew point of the exhaust gasses[3]

2.2 Thermostats

In order to have a comfortable living climate inside a house, thermostats are installed to control the temperature inside the house by regulating the heating. The thermostats installed vary from the famous "The Round" by Honeywell which is a completely mechanical device, to the touch screen controlled, internet connected thermostats like Toon® and Nest.

The system of a heated house is stable and slow, and a very simple controller is sufficient to control the temperature. An often used controller is an On-Off type controller. This controller usually has a lower and an upper bound set to the temperature. When the temperature crosses the lower bound fom below, the heating is turned off. When the temperature crosses the upper bound from above, heating is turned on. While this type of control is cheap and easy to implement, with the rise of high efficiency condensing boilers this kind of control is not desirable.

Why this kind of control is undesirable has two reasons. The first reason is efficiency. When an On-Off controller turns the boiler on, the boiler heats the water to its set temperature, normally around 70-80 degrees Celcius. Except for extremely cold conditions, this is usually too hot, and the water returns at a temperature well above the condensation point of the exhaust gasses.[4] This effectively rendering the high efficiency boiler a traditional boiler, as the exhaust gasses are not condensed, see Fig. 2.2.

Apart from energy considerations, controlling a house with a bang-bang controller inevitably results in over- and undershoots during the day. Compare the situation with a car, driving on the highway where the only options for speed control are full throttle or no throttle. To improve comfort, more gradual control is desirable.

2.2.1 Toon®

This research is based on data collected by Toon®, a smart thermostat developed by Quby and offered primarily by Eneco, a major Dutch utility. This thermostat offers a user friendly touchscreen interface (see Fig. 2.3), and various additional services traditional thermostats do not offer.

Figure 2.3: The Toon® display.

Toon® is an internet connected thermostat, which makes it possible to control the temperature in the house via the internet with the mobile app. This enables the user to come home early, and still arrive in a warm home by turning the heating on in transit.

[4]L Peeters et al. "Control of heating systems in residential buildings: Current practice". *Energy and Buildings* 40, pp. 1446–1455, 2008.

Because the Toon® is internet connected, it is also possible for Eneco to collect anonymous, aggregated usage data. This enables users to compare their usage with similar households, and can be used to propose a better heating schedule, to name a few possibilities. In previous research, it was found that by using Toon®, households can on average save between 5,1% and 6,1% on their annual gas bill[5], while at the same time increase comfort.

Toon® can be used on most modern central heating boilers, as most of these boilers support either the On-Off or OpenTherm protocol (see Section 2.3). However, for some newer services like BoilerIQ, an OpenTherm connection is required. Another reason to connect the Toon® via OpenTherm is that this protocol allows for more granular control than On-Off, as discussed before.

2.3 The OpenTherm standard

In Section 2.2 the benefits of having more gradual control was mentioned. One of the protocols for enabling this gradual control of the water temperature is the OpenTherm standard.[1] When the boiler supports it and the thermostat is wired correctly, this is the protocol that Toon® uses. Even though most modern boilers support OpenTherm, only 42% of all Toons is communicating with the boiler via OpenTherm. This gap is likely caused by older boilers, incorrect wiring or faulty settings.

OpenTherm is a digital standard, and allows for two-way communication between the boiler and the thermostat. The thermostat can request a certain water temperature, which the boiler will then regulate by controlling the flow of gas.

In contrast to On-Off thermostats, this means that OpenTherm connected thermostats can request a steady baseload, instead of switching on and off continuously. This results in stabler room temperature, less thermal stress in the boiler and pipes, and a higher efficiency.

The OpenTherm standard allows for two-way communication; the boiler can send information about faults back to the thermostat. Unfortunately, the standard for faults is under-specified, and this leads to a situation where fault-codes for different brands can mean completely different things. A few examples of this are listed in Table 2.1.

[5] Dennis Ramondt. "Savings from Smart Thermostats with Energy Displays". *Available at SSRN 2745144*, 2015.

[1] See `https://www.opentherm.eu`

Table 2.1: Demonstration of faultcodes meaning different things for different types and brands

Boiler	Code 1	Code 4
Intergas Kombi Kompakt HR22	Temperature too high	No flame signal
Agpo Domina C124e	No flame signal	N/A
AWB Thermo HR	Short circuit measured	Fault in controller

This means that a scheme based on faultcodes can only work when the brand and type of the boiler is known. Unfortunately, there is no flag in the OpenTherm standard which will convey this information, and with BoilerIQ it is required for Toon® to rely on information given by the customer.

2.4 Available Data

The data available for this thesis is an anonymous set of 545 Toon® customers participating in the pilot. The boilers in the set are both OpenTherm and On-Off connected. The data is provided for a 7 month period, starting at April 2015 and ending in October 2015. For each household, a set of 44 variables is provided. All of these variables are sampled once per minute.

Since the data spans the months from April to October, there are no real winter months in the set. At the time of this research, more data is not available. Fortunately, the set contains a range of cold days where residents are using their central heating systems.

Since the set is a mixed set of both OpenTherm and On-Off connected boilers, many of the variables populated by the OpenTherm connection are not set in On-Off boilers. And finally, because the implementation of the OpenTherm standard differs significantly between different manufacturers, not all variables are filled reliably even for OpenTherm boilers.

Table 2.2 shows a selected number of variables, together with how well the data is populated. An interesting point is that of the "boiler setpoint". This is the OpenTherm variable that is used to regulate the watertemperature of the water used for heating. For OpenTherm boilers, this variable takes a value between 0 and 80. For On-Off boilers, the variable takes a value that is either 0 or 1.

Of all the variables, only “room temperature”, “room setpoint” and “boiler setpoint” are reliably available for all households in the dataset. This is because these variables are either measured or set by Toon. The “room setpoint” is the setpoint set by the user, and this is what Toon® uses to calculate the “boiler setpoint”, which is essentially the control signal.

By only considering “room temperature” and “boiler setpoint” from Toon®, it is possible to completely take the Toon® and its control algorithms out of the loop, focussing only on the effect of the boiler on the room temperature.

Table 2.2: Statistics on selected variables in provided dataset

Variable	Data Points	Number of Toons
Room Setpoint	165M	545
Room Temperature	167M	545
Boiler Setpoint	164M	545

3 Modeling

In this chapter the mathematical and theoretical foundation of this thesis are explored. First, the model of the house is discussed, along with the simplifications and assumptions made. After, the most common faults in a central heating system are described, along with methods of modelling these faults.

3.1 Modeling a heated house

The model of the house will be used both to generate training data and as the basis of the Extended Kalman filter. In this section the choice of model structure is explained. Then, variations on this structure are described, along with a method of evaluating the performance of these variations.

3.1.1 Model structure

To model the thermal behaviour of a house, it is useful to look at the various elements of the house to see what role they play. Although the dimensions, materials and properties of the elements will differ from household to household, most houses share the following elements:

- Air
- Walls
- Floors and ceilings
- Furniture
- Windows
- Heating

The definition of all of these elements is taken very broadly, so that a house with a roof can be said to have the same elements as an appartment with no roof. This will make the equations simpler and more general, so that even if the building type is not know, the model is still applicable.

Air

First, let us consider the air in the home, since this is what we will be measuring and controlling the temperature of. Heat transfer within a medium such as air takes place via conduction, convection and radiation. This means there is no simple analytical solution which can accurately model the heat transfer through the air. However, using software like COMSOL[6], it is possible to do a numerical simulation which takes these effects into account.[7] The downside of this approach is that it takes enormous amounts of computing power, and it requires the geometry of the house to be well-known.

For simplicity, it is assumed that the air has a uniform temperature throughout the house, and that the measured temperature reacts to changes instantly.

Walls

Since walls behave differently when they face the outside than when they are between rooms, it makes sense to look at these two cases separately. First, the outside facing walls are considered.

Heat transfer through a wall is an example of conductive heat transfer. When the system is in steady state, the heat transfer from the inside to the outside of the wall is given by[8]:

$$\frac{\dot{Q}}{A} = \dot{q} = h(T_{in} - T_{out}) \tag{3.1}$$

Here, T_{in}, T_{out} are the (constant) temperatures inside and outside the wall, $\dot{q}$ is the heat transfer in Watt per m^2, $\dot{Q}$ is the total heat transfer and h is the heat transfer coefficient. A sketch of the situation where convection between a wall and the air is also taken into account is given in Fig. 3.1.

The assumption in the figure is that the wall is made from a single, uniform material. In reality, most walls consist of multiple layers with different properties. However, when the system is in steady state, the heat transfer is still described by (3.1).

When the system is not in steady state, the temperature profile in each of the segments is no longer linear, and the flow of heat entering the wall no longer equals the flow exiting the wall. This can be modeled by dividing the wall

[6] Comsol. *COMSOL Multiphysics: Version 3.3*. 2006.
[7] V Gerlich. "Modelling Of Heat Transfer In Buildings." *ECMS*, 2011.
[8] H E A Van den Akker and R F Mudde. "Fysische transportverschijnselen I"., 1998.

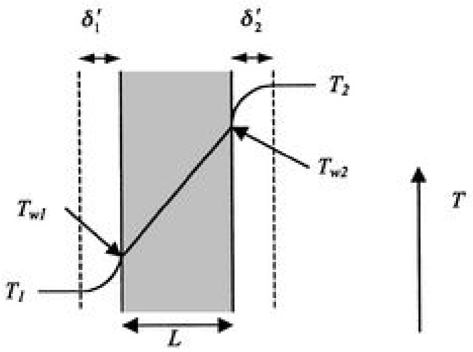

Figure 3.1: Conduction and convection around a wall.[1]

into layers with a certain heat capaciy, where each layer exchanges heat with its surrounding layers. Each layer is then cooling down or heating up until steady state is reached again. However, the number of parameters quickly explodes with this approach, and for simplicity it is assumed that the wall can be described as a single layer with a uniform temperature. The temperature of this single layer wall can be written as:[9]

$$C_{wall}\frac{dT_{wall}}{dt} = -\dot{Q} \tag{3.2}$$

Since the wall is in contact with both the inside and outside air, it exchanges heat with both, and equation (3.2) can be rewritten as:

$$C_{wall}\frac{dT_{wall}}{dt} = -h_{wall,out}(T_{wall} - T_{out}) - h_{wall,in}(T_{wall} - T_{in}) \tag{3.3}$$

Here, the wall has a heat capacity C_{wall} and heat transfer coefficients $h_{wall,out}$, $h_{wall,in}$ to the outside and inside, respectively. It is assumed that the temperature of all walls facing outward is uniform, regardless of orientation or position.

Apart from heat loss, walls can absorb energy from the sun causing them to heat up. Solar radiation is not measured by the Toon® directly, but satellite measurements are available via the KNMI (the Dutch metereology institute).

[9] Klaus Kaae Andersen, Henrik Madsen, and Lars H Hansen. "Modelling the heat dynamics of a building using stochastic differential equations". *Energy and Buildings* 31, pp. 13–24, 2000.

[1] Source: `http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/notes.html`

It is easy to imagine that with solar radiation local effects like orientation and shadows play a large role. However, because walls have a large total heat capacity compared to the air inside a house, it is assumed that these local effects even out over the course of a day.

The solar radiation is given in terms of power, and (3.4) describes the situation with an extra heating term. In the literature solar radiation is rarely taken into account with simple, generic models like the one used in this thesis. The assumption made above therefore needs to be verified before using it in the model. In subsection 3.1.2 four versions of the model are compared, of which two have this extra term and two do not.

$$C_{wall}\frac{dT_{wall}}{dt} = -h_{wall,out}(T_{wall} - T_{out}) - h_{wall,in}(T_{wall} - T_{in}) + Q_{solar} \quad (3.4)$$

Now that the outside walls are modeled, let us look at the internal walls. Here, internal walls refer to the walls that are not in contact with the outside air, like walls between rooms. Just like the outside facing walls, the heat transfer through the internal wall in steady state can be described by (3.1). However, with the assumption of uniform air temperature, this means that both sides of the wall are always at the same temperature, meaning that there is no heat transfer when the temperature remains unchanged.

However, when the system is not in steady state, meaning that the temperature of the air is different from the temperature of the wall, heat will flow. Like with the outside walls, it is assumed that the wall can be described as a single layer with a uniform temperature. The internal walls then transfer energy to the air via the following relation:

$$C_{wall_{internal}}\frac{dT_{wall_{internal}}}{dt} = -h_{wall_{internal}}(T_{wall_{internal}} - T_{in}) \quad (3.5)$$

By modeling the internal walls this way, they have effectively become a thermal mass, which can give off or take heat, effectively damping the system.

Floors, ceilings and furniture

When the different elements of a house were introduced, it was stated that the definition of each of the elements was to be taken very broadly. Why this is useful will become clear when dealing with floors, ceilings and furniture. The reason these are lumped together is because it is possible to model these as either inside or outside walls.

The reason behind this is simple: we have modeled an inside wall as a thermal mass, while we have modeled an outside wall as a thermal mass with an additional loss term. By using the arguments made for inside walls, it can be argued that furniture can be modeled as a simple thermal mass, too.

With floors it depends on the type of house if they should be modeled as an inside or outside wall. For an appartment with downstairs neighbours, the floor is likely heated from both sides. For a room on the ground floor, the bottom of the floor is in direct contact with the outside air, and should therefore be considered an outside wall. Fortunately, both elements are already modeled, and these differences in building types are reflected in the model parameters.

Windows

Windows are interesting to model, because unlike walls, they let radiation from the sun pass through. However, before considering solar radiation, we will first consider the basic thermodynamic properties of the windows.

Most windows in modern homes are double glazed windows. These windows are relatively good insulators, but unlike walls their thermal mass can be neglected.[10] This means that windows can be modeled as an extra heatloss term, where heat flows directly from the inside to the outside:

$$\dot{Q}_{window} = -h_{window}(T_{in} - T_{out})$$

As mentioned before, windows are interesting because the radiation from the sun passes through. The effect of solar radiation on the temperature of a house is unfortunately very difficult to model without detailed information about the orientation of the home, shadow created by obstacles, size of windows and their properties, and so on.

Therefore, in this thesis the direct effect of solar radiation coming through the windows is ignored. This has a direct effect on the performance of the model in sunnier months. However, by assuming that either the heating is not on during these months, or the sun is not yet shining when the heating is on, the model could still work for these periods.

Fortunately, during the winter months the effect of solar radiation is less pronounced, as can be seen from Fig. 3.2. When an effective surface area of a

[10] Léon Peter Bernard Marie Janssen and Marinus Maria Cornelis Gerardus Warmoeskerken. *Transport phenomena data companion*. 1997.

house of $10m^2$ is taken, the solar radiation contributes around $5kWh$ per day in June and July, while it contributes around $1kWh$ per day in December and January. To put these numbers into perspective, the energy content of a cubic meter of natural gas is approximately $10kWh$.[11] On average, an appartment in the Netherlands uses around $2m^3$ of gas per day for heating[2], which comes down to $20kWh$ per day.

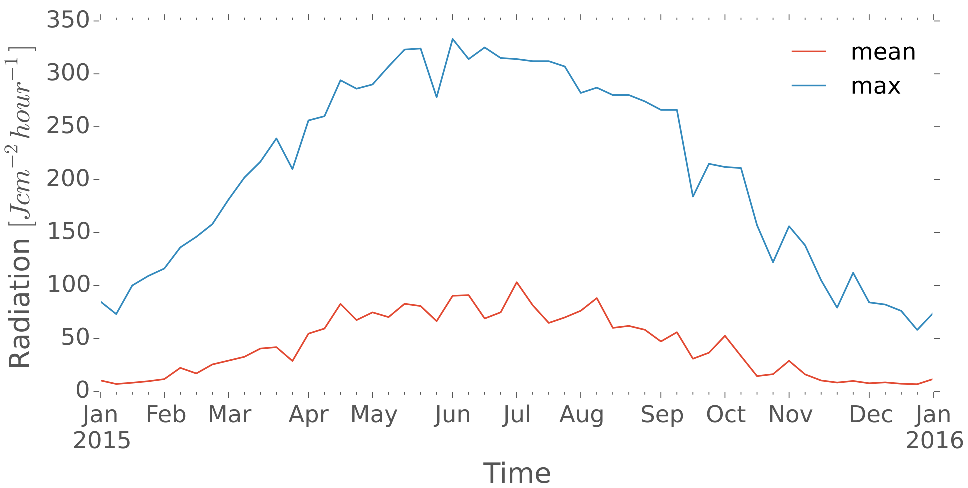

Figure 3.2: Solar radiation throughout the year according to the KNMI, aggregated per month.

Heating

Finally heating is considered. In the houses considered in this thesis, heating is provided by a central heating boiler and radiators. The boiler pumps hot water through the radiators, which give off the heat in the home.

The radiator can be modeled as a metal slab with a uniform temperature. However, unlike with the cases we have seen before, a large portion of the heat transfered from the radiator is in the form of radiation (hence the name).[8]

Heat radiation from a gray body can be approximated as:

$$\dot{Q} = \epsilon \sigma A T^4 \tag{3.6}$$

[11] David R Lide. *CRC Handbook of Chemistry and Physics, 85th Edition*. CRC Press, June 2004.
[2] Source: milieucentraal.nl

Here, $\epsilon\sigma$ is the emissivity factor, A is the surface of the body, and T is its temperature. This means that the heat flowing from the radiator to the air due to the radiation is:

$$\dot{Q}_{radiator,radiation} = \epsilon_{radiator}\sigma_{radiator}A_{radiator}T^4_{radiator} - \epsilon_{room}\sigma_{room}A_{room}T^4_{room}$$

The heat transfer between the water and the air through the metal is comparable to the transfer between out- and inside air through the walls, only with different constants. From the perspective of the radiator, the total heat transfer is:

$$\begin{aligned}\dot{Q}_{radiator} = &-\epsilon_{radiator}\sigma_{radiator}A_{radiator}T^4_{radiator} + \epsilon_{room}\sigma_{room}A_{room}T^4_{room}\\ &-h_{radiator,water}(T_{radiator} - T_{water})\\ &-h_{radiator,air}(T_{radiator} - T_{room})\end{aligned} \tag{3.7}$$

To get rid of the nonlinear term in (3.7), the system is linearized using the procedure from Mudde and van den Akker[8]. This results in:

$$\begin{aligned}\dot{Q}_{radiator} = &-h_{radiator,radiation}(T_{radiator} - T_{room})\\ &-h_{radiator,water}(T_{radiator} - T_{water})\\ &-h_{radiator,air}(T_{radiator} - T_{room})\end{aligned} \tag{3.8}$$

Where $h_{radiator,radiation}$ is a lumped parameter containing linearisation constants, material properties, etc. (3.8) can be simplified to:

$$\begin{aligned}\dot{Q}_{radiator} = &-h_{radiator,water}(T_{radiator} - T_{water})\\ &-h'_{radiator,air}(T_{radiator} - T_{room})\end{aligned} \tag{3.9}$$

Where $h'_{radiator,air}$ is the new heat transfer coefficient which handles both convection and linearised radiation. To simplify the relation further, it is assumed that the heat transfer coefficient from water to the radiator is much greater than the heat transfer coefficient from radiator to the room. Because of this, it is assumed that the water inside the radiator and the radiator have the same temperature.

$$\dot{Q}_{radiator} = -h'_{radiator,air}\left(T_{radiator} - T_{room}\right) \tag{3.10}$$

The temperature of the water coming from the boiler is regulated by the boiler. The water can not become cooler than room temperature, and not hotter than the setpoint in the boiler (usually $60 - 80°C$). This means that the difference can take values between 0 and $T_{max} - T_{room}$. By assuming that T_{max} is much higher than T_{room}, small variations in T_{room} can be neglected, and the heat-flow can be approximated to be linear over the entire range.

Now, because of this, the boiler effectively regulates this difference in temperature, and thus the flow of heat to the room. Assuming this regulator is a simple low-pass filter, the following relation holds:

$$\frac{P_{radiator}(s)}{U(s)} = \frac{P_{max}}{\tau s + 1} \tag{3.11}$$

Here τ is a time constant which determines the system behaviour, $P_{radiator}(s)$ is the Laplace transform of $P = \dot{Q}$, the heat flowing from the radiator to the room. U is the Laplace transform of the control signal from the thermostat, and P_{max} the heat given off when the radiators reach T_{max}. Because $u(t)$ takes values between 0 and 1, this extra scaling is neccesary.

From this relation, it is clear that the heatflow can be seen as a scaled and low-pass-filtered version of the input from the thermostat to the boiler. However, this assumes that the effect of the heat flowing from the radiators into the room is immediately measured by the thermostat, which is not realistic. To compensate for this effect, the radiator output can again be low-pass-filtered, resulting in a higher order low-pass filter.

Based on this result, different low-pass filters have been tested. A third order Bessel filter showed the best performance. This filter has a few desirable properties like minimal overshoot and ripple, none of which are expected in a heating system. The Bessel filter has no exact digital counterpart, but it was found that the approximated filter did not show significant degradation.

Combining the elements

Now that the heatflow from each of the elements is described, these flows can be combined into a single model taking all these effects into account. To do this, the flows are first rewritten to the perspective of the air, since this is what we will be controlling and measuring.

Because the elements are defined in terms of their heat flows, and these flows are independent, combining the elements is as simple as summing their flows into and from the inside air. However, this results in many different temperatures (furniture, internal walls, walls, etc) that need to be estimated, apart from their heat transfer coefficients. For simplicity, it is therefore assumed that these separate elements can be described by a single heat capacity.

The simplest way of doing this is by lumping all elements together. Because it likely contributes the most to the mass, we will refer to this mass as the wall. However, because all elements are lumped together it means that the wall should have different heat transfer coeffients going outside and in. This results in the following model of the home:

$$\begin{aligned}
C_{air}\frac{dT_{in}}{dt} =& - h_{window}(T_{in} - T_{out}) \\
& - h_{wall_{inside}}(T_{wall} - T_{in}) \\
& + Q_{heating} \cdot u_{filtered} \\
C_{wall}\frac{dT_{wall}}{dt} =& - h_{wall_{outside}}(T_{wall} - T_{out}) \\
& - h_{wall_{inside}}(T_{wall} - T_{in})
\end{aligned} \tag{3.12}$$

Here, $u_{filtered}$ is the low-pass-filtered control signal from the thermostat which runs from 0 to 1, and $Q_{heating}$ is the scaling factor described earlier. C_{air} is the heat capacity of the inside air, C_{wall} is the heat capacity of the combined elements, and h denotes the different heat transfer coefficients.

In the literature models such as these based on first principles are used often[12,13,14]. One of the benefits of such a model is that the parameters can be interpreted, since they have physical meaning. The model also lends itself very well to be linearized, because simply assuming the parameters are constant is enough to yield a linear model.

Since this model is essentially a model of energy storages and resistors, this model can be seen as an RC circuit, see Fig. 3.3. In this figure, the heat transfer coefficients are replaced for notational simplicity: $\beta = h_{wall_{inside}}$, $\bar{\beta} = h_{wall_{outside}}$, $\hat{\beta} = h_{window}$.

[12] J M Gordon and Y Zarmi. "Massive storage walls as passive solar heating elements: An analytic model". *Solar Energy* 27, pp. 349–355, 1981.

[13] Mario Vašak, Antonio Starčić, and Anita Martinčević. "Model predictive control of heating and cooling in a family house". *MIPRO*, pp. 739–743, 2011.

[14] Wesley J Cole et al. "Reduced-order residential home modeling for model predictive control". *Energy and Buildings* 74, pp. 69–77, 2014.

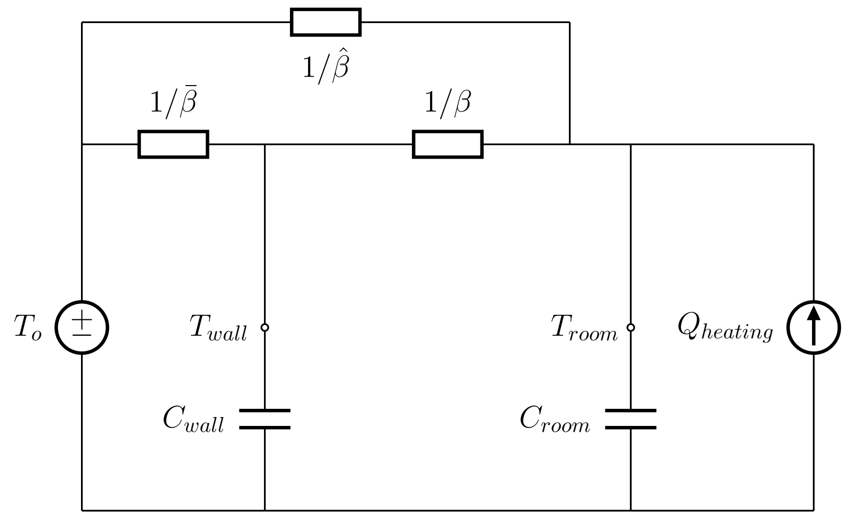

Figure 3.3: RC Diagram of heated house model

3.1.2 Validation

To test the accuracy of the model, the standard metrics of variance accounted for (VAF) and mean squared error (MSE) are used. The variance accounted for is defined as[15]:

$$\text{VAF} = \max\left(1 - \frac{\text{var}(y-\hat{y})}{\text{var}(y)}, 0\right) \cdot 100\% \tag{3.13}$$

Where y is the measured data, and $\hat{y}$ is the data provided by the model. The MSE is defined as:

$$\text{MSE} = \frac{(y-\hat{y})^2}{n}$$

Here, n is the length of the vectors y and $\hat{y}$. Again, y is the measured data, and $\hat{y}$ is the data provided by the model.

The parameters of the model are found by running a simulation over two days and using an optimization algorithm to find the parameters that minimize the MSE.

[15]Jan-Willem Van Wingerden and Michel Verhaegen. "Subspace IDentification of MIMO LPV systems: The PBSID approach". *2008 47th IEEE Conference on Decision and Control*, pp. 4516–4521, 2008.

Because finding the parameters is a non-convex problem, the Levenbergh Marquardt algorithm is used in a multistart configuration. Because such a solution is not guaranteed to find the global minimum, poor fits - judged by MSE - are discarded.

Also, to make sure there is enough information about all the parameters in the data, only a subset of the data is taken into account. This subset is selected by the following requirements:

- The boiler must be heating the house for at least a total of 50 minutes in two days
- The temperature must drop below 18 degrees celcius at least once
- No holes in the data longer than 15 minutes may occur

The above results in a total of 3754 periods of two days that are suitable for optimization. To speed up the process, the optimization was run on a cluster of linux servers, with a total of 60 cores and 192GB ram. Even with all those cores available, running the optimization still took around a day for every version of the model.

Model variations

1. The simplest version, where direct heatloss through the windows is ignored;
2. The simple solar version. Windows are ignored, but there is an extra heating term Q_{solar} added to the wall to account for the solar radiation;
3. The windows version. This is the structure as described in the previous section and shown in Fig. 3.3;
4. The windows and solar version. Here, direct heatloss through the windows is not ignored, and solar radiation to the wall is added as an extra heating term Q_{solar}.

The results of the optimizations are given in Table 3.1. The numbers show that the "Windows" model performs the best on the dataset: it has the most good fits, and of those good fits there is a great number of fits with very high VAF (>95%).

Note that there is a large difference between the number of periods in the subset and the number of good fits. This can be caused by the fact that the optimization problem is non-convex, and a certain amount of luck is required to start within a basin of attraction of a good minimum.

Table 3.1: comparison of performance of different models

	Good fits	Fits with VAF > 95%	Fits with VAF > 90%
Simple	1050	489	748
Simple + Solar	989	351	582
Windows	1182	602	926
Windows + Solar	1054	426	646

Another caveat is that the selection criterium here is not the most thorough. By simply looking at the number of fits in each category, we are likely biasing against models that are difficult to fit, for instance because they have more parameters.

By comparing these numbers and looking at the plots resulting from these models, some understanding of the strenghts and weaknesses of each model is obtained. For example, it was found that the simplest model had trouble keeping the wall temperature constant over the period of a few days: good fits tended to have a steady decrease in wall temperature, resulting in a horrible fit when more than two days were simulated. When the fit was made over a longer period of time to alleviate this problem, the overall fit quality dropped drastically.

An interesting observation is that models using solar radiation as an extra input perform worse than the other models. The reason for this is probably that sunshine is a very local phenomenon, and taking the radiation levels from the Koninklijk Metereologisch Instituut (KNMI) is too broad a generalisation. It was found that the best fits for models with solar radiation taken into account were fits where the radiation profile had a shape similar to the temperature profile. On days where the two profiles did not look alike, the value of the solar parameter dropped drastically. This is an indication that the data on solar radiation provided by the KNMI does not add significant information to the model.

The model that was easiest to fit is the model with windows, as given in Fig. 3.3. To get a feeling for how good this model is, see Fig. 3.4, where the measured room temperature is plotted against the simulated wall and room temperatures.

3.1.3 Model selection

The structure as chosen is given in Fig. 3.3. Note how, compared to the literature[7,13], there is an extra loss term added to the model which models the loss

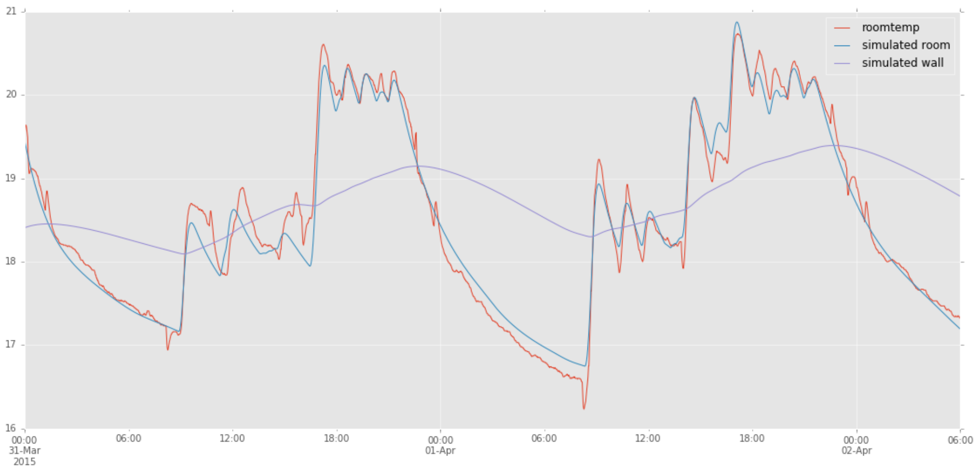

Figure 3.4: Comparison of simulation with real measured data of the chosen model

through windows. Also note that for simplicity the notation as in the figure is adapted, and that $Q = Q_{heating}$ and u is low-pass filtered with the Bessel filter described earlier.

Without loss of generality, the heat capacity of the inside air is set to 1. Since (3.12) is a continuous time model and the data is only available at fixed intervals (1 minute), it makes sense to discretize this model. To discretize, the forward euler method is used, yielding equation (3.14).

$$\begin{aligned} T_{room,k+1} &= T_{room,k} - t_s\beta(T_{room,k} - T_{wall,k}) \\ &\quad - t_s\hat{\beta}(T_{room,k} - T_{out,k}) + t_s Q u_k \\ T_{wall,k+1} &= T_{wall,k} - \frac{t_s\beta}{C_{wall}}(T_{wall,k} - T_{room,k}) \\ &\quad - \frac{t_s\bar{\beta}}{C_{wall}}(T_{wall,k} - T_{out,k}) \end{aligned} \tag{3.14}$$

Finally, this set of equations is transformed into a state-space model $\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k]$, $\boldsymbol{y}[k] = \boldsymbol{C}\boldsymbol{x}[k]$ where:

$$\begin{aligned}\begin{bmatrix} T_{room}[k+1] \\ T_{wall}[k+1] \end{bmatrix} &= \begin{bmatrix} 1 - ts(\beta + \hat{\beta}) & ts\beta \\ \frac{ts}{C_{wall}}\beta & 1 - \frac{ts}{C_{wall}}(\beta + \bar{\beta}) \end{bmatrix} \begin{bmatrix} T_{room}[k] \\ T_{wall}[k] \end{bmatrix} \\ &+ \begin{bmatrix} tsQ & ts\hat{\beta} \\ 0 & \frac{ts}{C_{wall}}\bar{\beta} \end{bmatrix} \begin{bmatrix} u & T_{out} \end{bmatrix} \\ y[k] &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} T_{room}[k] \\ T_{wall}[k] \end{bmatrix}\end{aligned} \tag{3.15}$$

The model depends on many parameters, for clarity an overview of the parameters and their physical interpretation is given in Table 3.2.

Table 3.2: overview of parameters used in model.

Parameter	Meaning
ts	Time between samples, in this case ts=60s
Q	Heating constant, indicates how much power is given off when the thermostat requests 100%
β	Heat transfer coefficient between thermal mass and inside air
$\bar{\beta}$	Heat transfer coefficient between thermal mass and outside air
$\hat{\beta}$	Heat transfer coefficient between inside and outside air
C_{wall}	Heat capacity of thermal mass

3.2 Modeling most common central heating faults

Now that the model for the house is selected, it is time to introduce some faults into the system. The faults as described in this section are some of the most common faults as described by Eneco technicians, together with faults technicians normally do not see, but which would be valuable to detect.

3.2.1 Low water pressure

The most common repair is a rather trivial one: low water pressure in the system. This is one of the few faults residents can actually solve themselves[16]

[16]Intergas. *Handleiding/instructie Kombi Kompakt HR-ketel Intergas*. Jan. 2011.

in a few simple steps, but it still shows up as one of the main causes of service requests.

Boilers measure the water pressure in the system, and when it drops below a certain point, the boiler automatically shuts off, resulting in a complete loss of control from the point of view of the thermostat. In OpenTherm connected boilers, the waterpressure can be communicated with the thermostat. With more primivitive “on-off” boilers, there is little warning; experts at Eneco do not expect to see a significant decrease in effiency because of decreasing water pressure.

This fault can be modeled by setting $Q = 0$ in the simulation.

3.2.2 Stuck three-way valve

In the Netherlands, a system which provides both domestic hot water and heating is most common[3]. In this system, there is a three-way-valve which determines the flow of hot water coming from the boiler.

When the valve is set to heating, hot water from the boiler flows through the house, where it gives off heat in the radiators. Cooled down, it returns to the boiler. When the valve is in the hot water position, hot water flows from the boiler through a heat exchanger. Again, it gives off heat and returns to the boiler. The system itself is a closed loop (ignoring leaks, which would trigger a low water pressure fault, eventually).

Sometimes the three-way valve gets stuck in the heating position, the hot water position or somewhere in between. Because different positions of the valve cause completely different behaviour, three different cases are looked at[4].

First we will look at what happens if the valve is stuck in the “central heating” position. Whenever the temperature in the heat exchanger drops below a threshold, or when there is a demand for hot water, the boiler will activate, pumping hot water through the house instead of the heat exchanger. However, because the heat exchanger will not heat up, the boiler will not turn off by itself, continuing to give off heat in the house.

When the valve is stuck in the “domestic hot water” position, the behaviour is the opposite. Whenever the thermostat requests hot water, it is pumped through the heat exchanger instead of the house - the exchanger quickly overheats, causing the boiler to shut itself off. This results in complete loss of

[3] According to data provided by Eneco

[4] As explained by Eneco technicians

control from the point of view of the thermostat, much like with the water pressure.

Whenever the valve is stuck somewhere in the middle, the behaviour becomes more complex and less predictable. To illustrate this, imagine the situation where the valve is stuck in a position where half of the water flows through the house, and half of the water flows through the heat exchanger. Whenever there is a demand for hot water, hot water is pumped through both the house and the heat exchanger. This results in an unexpected rise of temperature in the house, but since the exchanger is also heated, the heating should eventually stop.

Whenever the temperature in the house drops, the thermostat will demand hot water, which (again) will flow through both the house and the heat exchanger. After a while, very little heat will be given off in the heat exchanger. This causes the water returning to the boiler to be too hot, which will cause the boiler to reduce power or completely shut off. This situation is a lot harder to model than the two situations mentioned before, as the amount of loss of control depends on the heat given off in the house and the heat exchanger.

This means there are three different models for this kind of fault, of which the "heating" position is nearly impossible to simulate reliably. This type of fault is therefore left out of the scope of this research. The other two situations can be modeled as a sudden drop in Q, either to zero or to a value between 0 and $Q_{original}$.

3.2.3 Gradual performance degradation

Apart from sudden changes in the system like the faults we have seen before, some faults manifest themselves more slowly. One example is air slowly working itself into the radiators, causing only a part of the radiator to heat up, slowly decreasing the heat given off.

Most of the causes for slow degradation are subtle, and mechanics only see the systems once every one or two years. This means there is not much knowledge available on how these systems degrade, and if these degradations can be used to predict failures in the future.

However, detecting degradation is a first step in uncovering trends and correlation between faults. Without detection, more advanced schemes that try to predict when a boiler needs maintenance will not exist.

Introducing the degradations is simpler than finding out why they occur, and can be done by lowering Q, or increasing β slowly.

3.2.4 Closed radiators

The faults so far have been system faults. Another reason the central heating systen might not work as efficient as intended is caused by user error, and is caused by the user closing the radiators.

When a central heating installation is designed, the number of radiators and their size is chosen based on the expected power required in the room. After the radiators are installed, the flow of hot water needs to be balanced in the system, so that the amount of heat given off in the bathroom is in proportion to the heat given off in the living room. This is done by turning a small screw usually located at the bottom of the radiator.

Apart from this screw, there is usually a thermostatic knob on each radiator. This knob can be used to make adjustments in the amount of heat given off by each radiator. This can be useful if a room without a thermostat does not get warm quick enough.

Unfortunately, these knobs are often misused or forgotten; creating an imbalance in the system. What happens then is that a large portion of the heat ends up in a room that is not in continuous use, like an upstairs bathroom.

By continuously monitoring the heat given off in the room with the thermostat, changes to this knob could be detected. A model of this situation is a sudden drop of the heat given off in the room, so a sudden drop in Q. How low Q goes depends entirely on how bad the imbalance is, and how far the radiator has been turned down.

3.2.5 Heating with the windows open

The last fault we will look at is an inefficiency caused by user behaviour, like the previous fault. Since this is only a temporary inefficiency, intervention is usually not required. However, detecting opened windows or doors can add valuable information to the control algorithm. This information can then be used to either increase the power in an attempt to keep temperature stable, or to save energy by decreasing the setpoint for a while.

Finally, the detection can be coupled with other measurements like energy consumption to determine occupancy. This can then be used to improve the temperature programs that are configured on the thermostat.

By now it must be clear how this fault can be modeled. This is again a sudden increase, this time in the $\hat{\beta}$ parameter.

3.2.6 Summary

In this section, a number of common faults together with their proposed models are described. Since many of these faults share behaviour, the faults and their models are summarized in Table 3.3. Note that since no strict upper limit for most of the variable faults can be given, the entire possible range is given, where 100% corresponds to the fault not being noticeable.

Table 3.3: Overview of different faults, where "HW" stands for Hot Water.

Fault	Q	β	Timeframe
Low water pressure	0%	unchanged	immediate
Stuck 3-way valve (heating)	irregular	unchanged	immediate
Stuck 3-way valve (HW)	0%	unchanged	immediate
Stuck 3-way valve (in between)	0-100%	unchanged	immediate
Closed radiators	0-100%	unchanged	immediate
Performance Degradation	0-100%	increased	weeks-months
Heating with open windows	unchanged	increased	immediate

4 Online Parameter Estimation

In the previous section we looked at modeling central heating in a house, and a few common faults. We saw that by slowly or quickly changing a number of parameters in the model it is possible to obtain the faulty behaviour. In this chapter we will look at how we can find the parameters of the model from the measured values only. This chapter is split into three parts, in the first part the theoretical background of the Kalman filter is given, since it is the basis of the method used in this research. In the second part the Extended Kalman filter - a commonly used extension to the Kalman filter which can deal with nonlinear systems - is introduced. In the third part we will look at how we can use this Extended Kalman filter to estimate parameters of the model in realtime.

4.1 Kalman filter

Named after Rudolf E. Kálmán who published his paper "A New Approach to Linear Filtering and Prediction Problems" in 1960[17], the Kalman Filter has become one of the most important tools in the systems engineers' toolkit.

The main selling point of the Kalman Filter is that it can estimate unknown parameters even when the measurements are affected by statistical noise. It does so by combining previous measurements, and the resulting estimates are usually more accurate and reliable than those based on only a single measurement.

The algorithm works in two steps. In the first step - prediction - the Kalman Filter estimates the current state parameters and their uncertainties. Once the next measurement is available, the filter executes the second step. In this step, the filter combines the estimate with the measurement using a weighted average, giving a smaller weight to the more uncertain value.

[17] R E Kalman. "A New Approach to Linear Filtering and Prediction Problems". *Journal of Basic Engineering* 82, pp. 35–45, 1960.

This approach requires the algorithm to assign uncertainties to both the prediction and the measurement. The uncertainty in the measurement is a tuning parameter, which is usually fixed a priori by the engineer. The uncertainty in the prediction, however, is calculated by propagating the uncertainty in the state through the linear model.

The reason this can be done efficiently is that a normal distribution remains a normal distribution when transformed by a linear transformation. This means that if the uncertainties in the states of a linear system are assumed to be normal, they will remain normal, and only two parameters - the mean and variance - are required to describe their distributions completely.

With the background out of the way, let us look at the mathematical steps of the Kalman filter, and how the idea is implemented. Because it is the most widely used, and directly applicable to this research, we will only discuss the discrete version of the filter.

The first step of the algorithm is to make a prediction of the state $\boldsymbol{x}$ and its covariance $\boldsymbol{P}$:

$$\begin{aligned}
\hat{\boldsymbol{x}}[k|k-1] &= \boldsymbol{A}\hat{\boldsymbol{x}}[k-1|k-1] + \boldsymbol{B}\boldsymbol{u}[k-1] \\
\boldsymbol{P}[k|k-1] &= \boldsymbol{A}\boldsymbol{P}[k-1|k-1]\boldsymbol{A}^{\intercal} + \boldsymbol{Q}
\end{aligned}$$

When the measurement at k comes in, the update step takes place.

$$\begin{aligned}
\boldsymbol{e}[k] &= \boldsymbol{y}[k] - \boldsymbol{C}\hat{\boldsymbol{x}}[k|k-1] \\
\boldsymbol{K}[k] &= \boldsymbol{P}[k|k-1]\boldsymbol{C}^{\intercal}\left(\boldsymbol{C}\boldsymbol{P}[k|k-1]\boldsymbol{C}^{\intercal} + \boldsymbol{R}[k]\right)^{-1} \\
\hat{\boldsymbol{x}}[k|k] &= \hat{\boldsymbol{x}}[k|k-1] + \boldsymbol{K}[k]\boldsymbol{e}[k] \\
\boldsymbol{P}[k|k] &= (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C})\boldsymbol{P}[k|k-1]
\end{aligned}$$

The notation of $\hat{\boldsymbol{x}}[k|k-1]$ denotes the estimate of $\boldsymbol{x}, \boldsymbol{k}$ using knowledge up to $k-1$. Also, $\boldsymbol{A}$ and $\boldsymbol{C}$ are the standard discrete system matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are covariance matrices of the process and measurement noise, respectively.

4.2 Extended Kalman filter

However powerful the Kalman filter is, one of its limitations is that the system must be a linear system in state-space form. The Kalman Filter assumes the system matrices are known, and can not be used to estimate the model parameters directly. And it is these model parameters that are really the parameters of interest in a fault detection scheme.

Fortunately extensions of the standard Kalman filter exist that make it capable of estimating parameters for nonlinear systems. The simplest of these extensions is the Extended Kalman Filter, or EKF, which is also one of the most widely used.[18,19,20,21,22]

To understand the basics of the EKF, let us define a non-linear state space model, as follows:

$$\begin{aligned}\boldsymbol{x}[k] &= f(\boldsymbol{x}[k-1], \boldsymbol{u}[k-1]) + \boldsymbol{w}[k-1] \\ \boldsymbol{y}[k] &= h(\boldsymbol{x}[k]) + \boldsymbol{v}[k]\end{aligned}$$

Where $\boldsymbol{x}$ is the state vector, $\boldsymbol{y}, \boldsymbol{k}$ the measurement vector, and $\boldsymbol{w}$ and $\boldsymbol{v}$ are process and measurement noise, respectively. Finally, f and h are (possibly) non-linear functions.

Now, since f and h can be nonlinear functions, it is no longer possible to directly calculate the covariance of the outcome. However, we can approximate f and h by using their taylor expansion, and this is exactly what the EKF does; at every timestep, the two functions are linearized around their current value, and these linearized versions are used to calculate the new covariance matrices. However, it is important to note that the error term ($\boldsymbol{e}, \boldsymbol{k}$ in the standard Kalman filter) is calculated using f and h directly.

To see the difference between the EKF and the standard Kalman Filter, the EKF is given below.

$$\begin{aligned}\hat{\boldsymbol{x}}[k|k-1] &= f(\hat{\boldsymbol{x}}[k-1|k-1], \boldsymbol{u}[k-1]) \\ \boldsymbol{P}[k|k-1] &= \boldsymbol{F}\boldsymbol{P}[k-1|k-1]\boldsymbol{F}^{\intercal} + \boldsymbol{Q}\end{aligned}$$

[18] Hongwen He, Zhentong Liu, and Yin Hua. "Adaptive Extended Kalman Filter Based Fault Detection and Isolation for a Lithium-Ion Battery Pack". *Energy Procedia* 75, pp. 1950–1955, 2015.

[19] P Howlett, P Pudney, and X Vu. "Estimating Train Parameters with an Unscented Kalman Filter". 2004.

[20] Wei Xue and Ying qing Guo. "Application of Kalman Filters for the Fault Diagnoses of Aircraft Engine". In: *Kalman Filter*. Mar. 2014. Pp. 1–15.

[21] Heather H Lambert. *A simulation study of turbofan engine deterioration estimation using Kalman filtering techniques*. Tech. rep. June 1991.

[22] R Luppold et al. "Estimating in-flight engine performance variations using Kalman filter concepts". In: *25th Joint Propulsion Conference*. 1989.

Again, when the measurement at k comes in, the update step takes place.

$$\begin{aligned}
\boldsymbol{e}_[k] &= \boldsymbol{y}_[k] - h(\hat{\boldsymbol{x}}[k|k-1]) \\
\boldsymbol{K}_[k] &= \boldsymbol{P}[k|k-1]\boldsymbol{H}[k]^\intercal \left(\boldsymbol{H}[k]\boldsymbol{P}[k|k-1]\boldsymbol{H}[k]^\intercal + \boldsymbol{R}[k]\right)^{-1} \\
\hat{\boldsymbol{x}}[k|k] &= \hat{\boldsymbol{x}}[k|k-1] + \boldsymbol{K}[k]\boldsymbol{e}[k] \\
\boldsymbol{P}[k|k] &= (\boldsymbol{I} - \boldsymbol{K}[k]\boldsymbol{H}[k])\boldsymbol{P}[k|k-1]
\end{aligned}$$

Where:

$$\begin{aligned}
\boldsymbol{F}[k-1] &= \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}[k-1|k-1],\boldsymbol{u},\boldsymbol{k}} \\
\boldsymbol{H},\boldsymbol{k} &= \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}[x|k-1]}
\end{aligned}$$

Interestingly, note how there is no need for a (version of the) $\boldsymbol{B}$ matrix in this approach, since the control input is simply fed into the definition of $\boldsymbol{F}$. Also note how the two algorithms really are not that different. The only differences are in the use of $\boldsymbol{F}$ and $\boldsymbol{H}$ instead of $\boldsymbol{A}$ and $\boldsymbol{C}$, and how the new prediction is now done with a nonlinear model.

The EKF is a very powerful method, but since it relies on linearization, there are no longer any guarantees of optimality, like there are with the standard Kalman filter. In some cases - usually when the system is highly nonlinear - the EKF will not perform satisfactory and another extension will have to be chosen, like the Unscented Kalman Filter (UKF)[23,24], which is computationally more expensive.

Fortunately, in this case the model is already linear, and the Extended Kalman Filter can be used directly.

4.3 Using the Extended Kalman filter for parameter estimation

Now that we have seen how we can use the EKF to estimate the state in a nonlinear system, let us look at how we can use this to estimate parameters of the model of a house.

[23]E A Wan and R Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". *... 2000 AS-SPCC The IEEE 2000*, 2000.

[24]Mohammad Taghi Sabet, Pouria Sarhadi, and Mostafa Zarini. "Extended and Unscented Kalman filters for parameter estimation of an autonomous underwater vehicle". *Ocean Engineering* 91, pp. 329–339, 2014.

The model is given below:

$$
\begin{aligned}
T_{room,k+1} &= T_{room,k} - t_s\beta(T_{room,k} - T_{wall,k}) \\
&\quad - t_s\hat{\beta}(T_{room,k} - T_{out,k}) + t_s Q u_k \\
T_{wall,k+1} &= T_{wall,k} - \frac{t_s\beta}{C_{wall}}(T_{wall,k} - T_{room,k}) \\
&\quad - \frac{t_s\bar{\beta}}{C_{wall}}(T_{wall,k} - T_{out,k})
\end{aligned}
$$

Where $T_{room,k}$ means the temperature of the room at time k. t_s is the time constant, which in this case is set to one minute (60 seconds). Also note that the Q here stands for the heat given off in the room, and should not be confused with the $\boldsymbol{Q}[k]$ matrix in the EKF.

Now, to estimate the parameters of interest $(Q, \beta, \bar{\beta}, \hat{\beta}, C_{wall})$, we can add these parameters as states, augmenting the system. Since these parameters are assumed to be constant, the model becomes:

$$
\begin{aligned}
T_{room,k+1} &= T_{room,k} - t_s\beta_k(T_{room,k} - T_{wall,k}) \\
&\quad - t_s\hat{\beta}_k(T_{room,k} - T_{out,k}) + t_s Q u_k \\
T_{wall,k+1} &= T_{wall,k} - \frac{t_s\beta_k}{C_{wall,k}}(T_{wall,k} - T_{room,k}) \\
&\quad - \frac{t_s\bar{\beta}_k}{C_{wall,k}}(T_{wall,k} - T_{out,k}) \\
\beta_{k+1} &= \beta_k \\
\bar{\beta}_{k+1} &= \bar{\beta}_k \\
\hat{\beta}_{k+1} &= \hat{\beta}_k \\
C_{wall,k+1} &= C_{wall,k} \\
Q_{heating,k+1} &= Q_{heating,k}
\end{aligned}
$$

Where, again, the subscript k denote the value of the parameter at time k. This set of equations can be written as:

$$
\begin{aligned}
\boldsymbol{x}[k+1] &= f(\boldsymbol{x}[k], \boldsymbol{u}[k]) \\
\boldsymbol{y}[k] &= h(\boldsymbol{x}[k])
\end{aligned}
$$

with:

$$\boldsymbol{x} = \begin{bmatrix} T_{room} \\ T_{wall} \\ \beta \\ \bar{\beta} \\ \hat{\beta} \\ C_{wall} \\ Q \end{bmatrix}$$

With all this in place, implementing the EKF for parameter estimation is straightforward. We have everything we need to implement the EKF, except for the linearized equations:

$$\boldsymbol{F}[k-1] = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}[k-1|k-1],\boldsymbol{u},\boldsymbol{k}}$$
$$\boldsymbol{H},\boldsymbol{k} = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}[x|k-1]}$$

Calculating $\boldsymbol{F}$ shows one small issue: C_{wall} appears at the denominator of a fraction in $\boldsymbol{f}$. Taking the derivative of $\frac{1}{C_{wall}}$ to C_{wall} yields $\frac{1}{2C_{wall}^2}$ - so naively linearising here will not work. Fortunately, it is easy to see that $\frac{1}{C_{wall}}$ can simply be replaced with $C_{wall,inv}$ everywhere, solving this problem. The estimated value can then easily be transformed back by taking its inverse.

To make it easy to remove or add states and play with different variations of the model, the calculation of $\boldsymbol{F}$ and $\boldsymbol{H}$ is left to the computer. The code is listed in the appendices A-C.

5 Results

In this chapter the results of estimating system parameters using the Extended Kalman filter are shown. The goal of this reserach is to detect faults in the system, and these system parameters are a valuable indicator of these faults.

The estimator is first tested on a noise-free model to see if all parameters are tracked properly. A realistic amount of white noise is then added to mimic the less than perfect measurements in reality.

Various faults are then introduced into the model generating the data. First, sudden drops of heating and sudden jumps in heat-loss are introduced. These correspond to abrupt failures. Secondly, a slower degradation of the heating installation and insulation is introduced to the model. With these faults and degradations, the most common faults and sources of inefficiencies are covered.

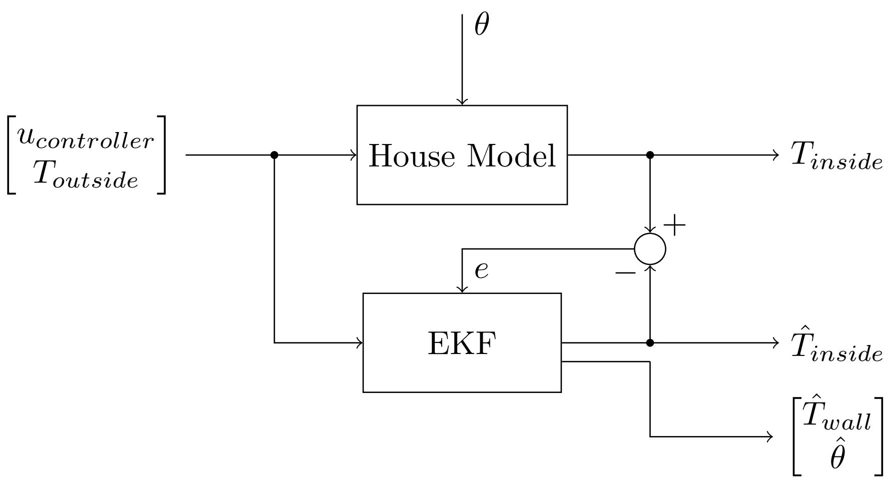

Figure 5.1: Schematic overview of parameter estimation setup using the Extended Kalman Filter (EKF).

In Fig. 5.1 the parameter estimation setup is shown. Here, θ denotes the real model parameters, and $\hat{\theta}$ are its estimates. All parameters and signals are time-varying, and the time indices are left out for simplicity. Note that only T_{inside} is measured, and therefore this is the only signal used in the feedback loop. In this setup, the faults are introduced by varying θ, and the goal is to use the Extended Kalman Filter to reconstruct these parameters from the available measurements.

Finally, the parameter estimator is used to estimate the parameters on measured data from a real household, putting the system to the test.

5.1 Tracking parameters without noise

The first step in finding out if the approach has any chance of success is to check if the parameters in the model can be found by the filter. To see if the filter works, the Kalman filter is initialized with different initial values than the boiler model used to generate the measured data.

There are many parameters of interest in the model, and plotting them all adds no real value. Therefore, only the parameters of interest are shown in the plots. However, it is important to note that all parameters of the model are tracked and estimated simultaneously, even when they are not plotted.

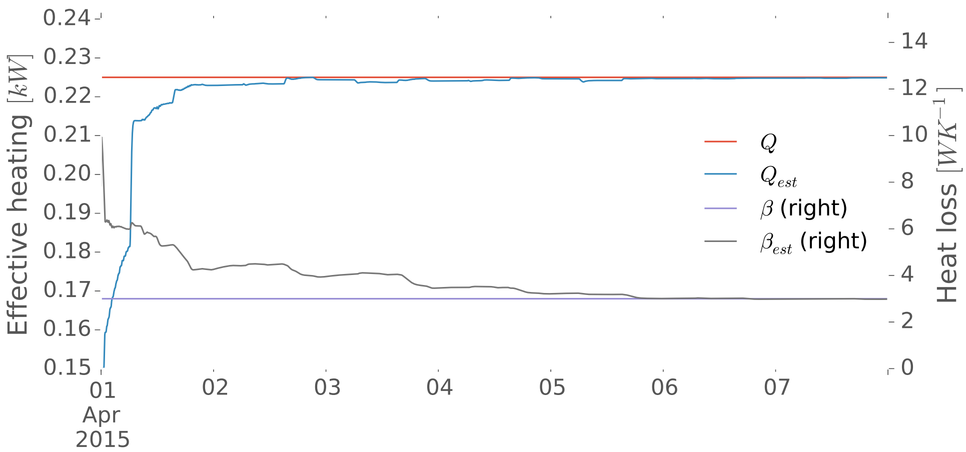

Figure 5.2: Tracking two parameters: Q and β.

In Fig. 5.2, Q and β and their estimates are shown for a simulated week in april. Both parameters are estimated correctly, without any bias, within the timeframe of a week. In the figures, Q and β are the real values of Q and β, as set in the model. Q_{est} and β_{est} are the values found by the Extended Kalman Filter.

It is interesting to note that because β converges slower than Q, the effect of the mismatch in β can also be seen in Q. This is especially noticable around the 3rd of April, where Q had converged to the correct value, but is pulled downward by the incorrect value of β and other parameters.

5.2 Introducing noise to the model

Since Kalman filters are designed to deal with noisy data, and because real data is never free from noise, zero mean white noise is added to the output. In reality the noise on a system like this is not white, and likely not even zero mean. However, no reliable noise model is available, so zero mean white noise is the best we have.

The variance of the white noise is chosen so that the simulation is comparable to the real measurements from Toon, or slightly worse.

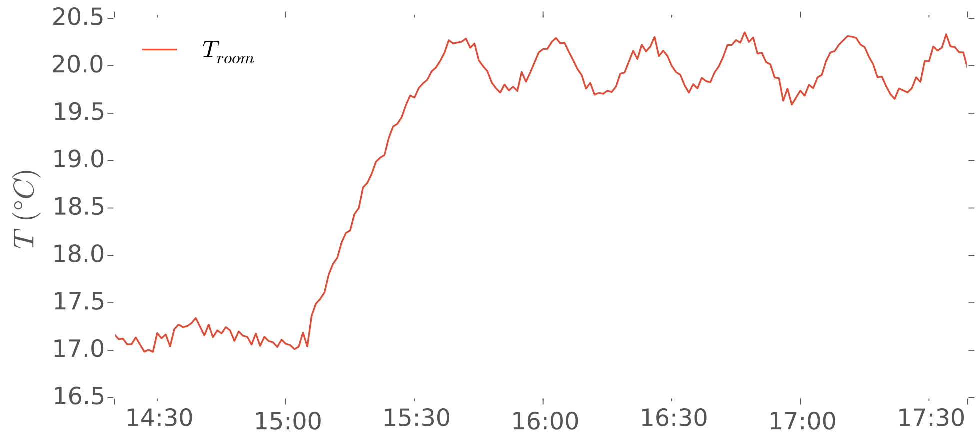

Figure 5.3: Room temperature with additive measurement noise ($\sigma = 0.05$)

In Fig. 5.3, the results of adding noise to the output is shown. Note that the overall pattern is less regular than that of the noise-free system, but that the overall trends are still easily seen by eye.

In Fig. 5.4 the effect of the noise on the estimates is shown. It is clear that the additive noise disturbs the estimates. However, even though the estimates themselves are more noisy, the overall behaviour is still correct. Both estimates for Q and β converge to their real values in roughly the same time as in the noise-free case.

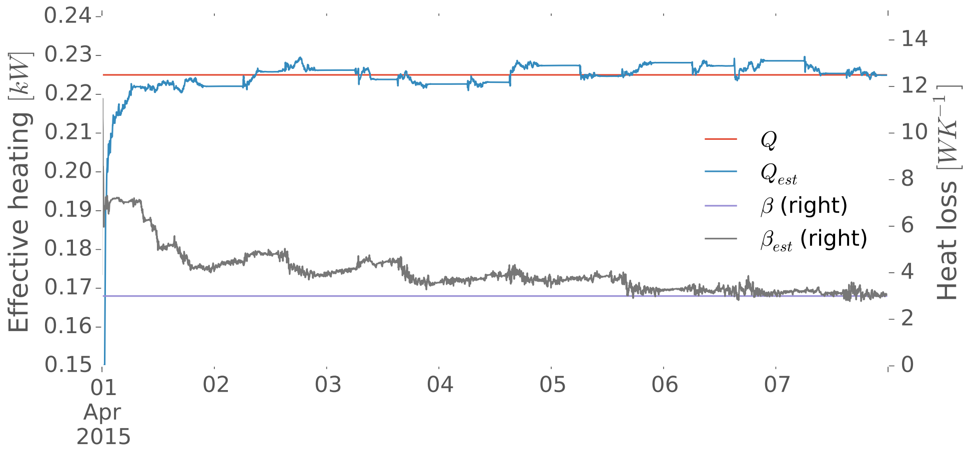

Figure 5.4: Tracking Q and β in the presence of additive measurement noise

5.3 Tracking time-varying parameters

In the previous two sections, it is shown that the EKF can correctly estimate the constant parameters of the model. Since the whole point of tracking these parameters and estimating their values online is to detect changes in their values corresponding to faults, the next step is to see if it is possible to track parameters that change over time.

First, two cases of sudden change are looked at. These situations correspond to a common failure like loss of water pressure and a common ineffiency like opened windows. Next, two cases of slow degradation are considered.

It is important to stress again that the goal here is to observe the performance of the filter in a controlled environment. Even though putting real measured data through it is more exciting, since the true state of the system is unknown there is no easy way to tell if the filter is working correctly.

5.3.1 Sudden loss of heating

As described in the section on modeling common faults, one of the most common faults is the loss of water pressure. This fault can be modeled by suddenly dropping Q to zero, resulting in complete loss of control from the point of view of the thermostat.

Here, Q is taken to be constant for a few days to give the filter time to converge before suddenly dropping to zero. All other parameters are kept constant for the entire duration. Also, since there is information about Q in the measured signals only when u is not zero, a convenient time for the failure is chosen - otherwise the filter would look slower than it really is.

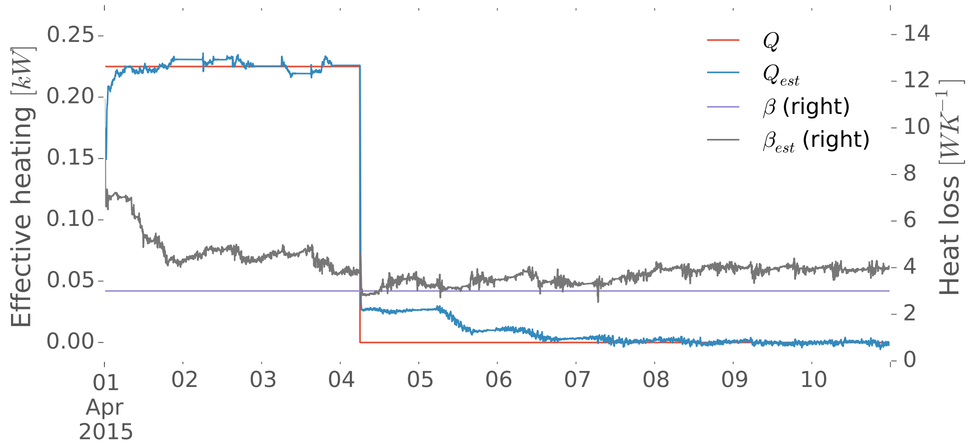

Figure 5.5: Tracking Q and β when Q suddenly drops to zero.

In Fig. 5.5 it is shown that the sudden drop in Q is picked up by the EKF almost instantly. Note how it seems to take a few more days for the filter to converge to the true value of zero.

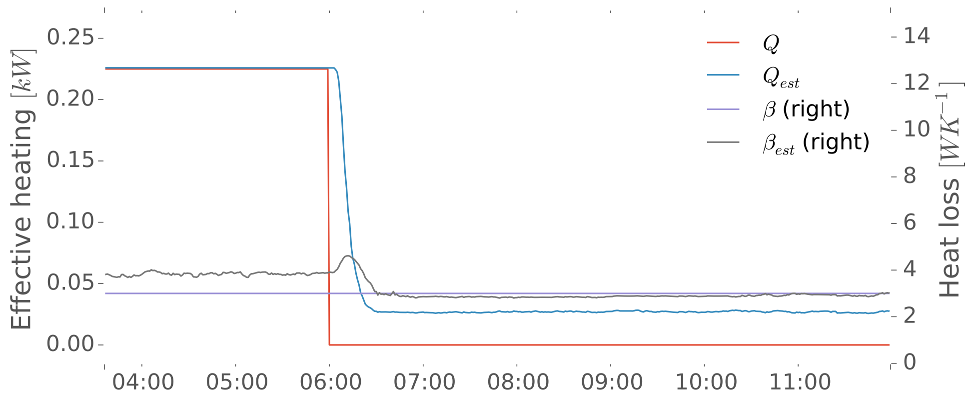

Figure 5.6: Detail view of the sudden drop in Q.

To get a better view of the error, in Fig. 5.6 a detail view of the same event is shown. Here, it shows the filter is able to pick up a value close to the true value within half an hour. However, it takes the filter much longer to finally converge to the correct value.

From this plot, it seems like β is not influenced by the sudden decrease in Q at all.

5.3.2 Sudden increase in heat loss

The next sudden change is that of a sudden heat loss. This corresponds to a resident opening a window or a door. This is not something that would usually require intervention, so it is important to be able to separate from a loss in heating control as described in the previous section.

Again, the algorithm is given a few days to converge before suddenly increasing β significantly. All other parameters are again kept constant.

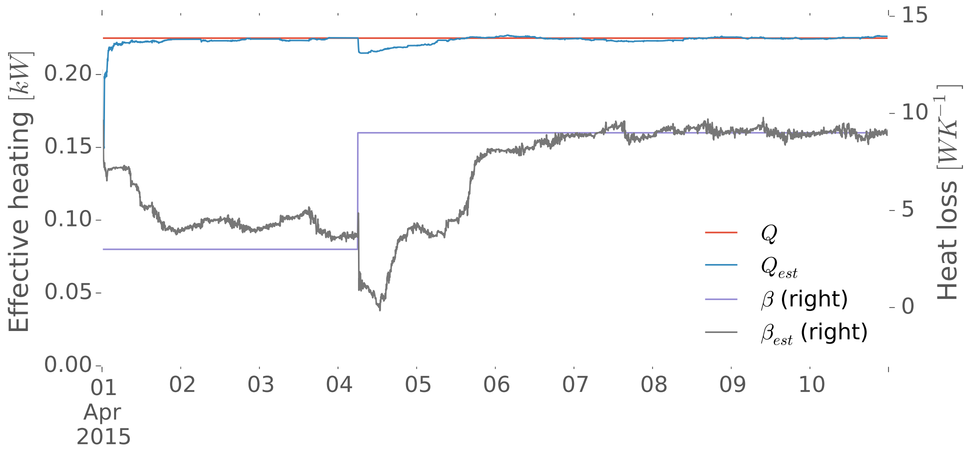

Figure 5.7: Tracking Q and β when β suddenly increases significantly, without tuning the covariance matrices

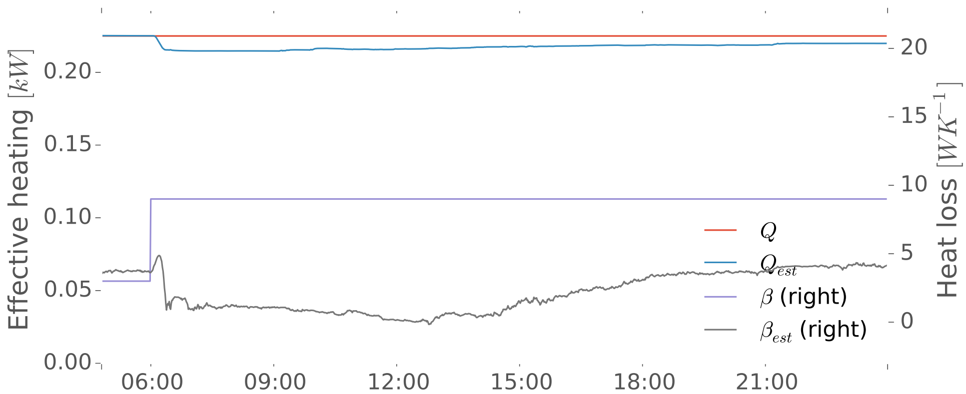

Figure 5.8: Detail view of the sudden increase in β, without tuning the covariance matrices

In Fig. 5.7 the result of this increase in β is shown. Clearly, the algorithm is struggling more than in the previous section. In the detail view of Fig. 5.8 it can be seen that there is a slight decrease in Q_{est} (from 0.15 to 0.14), but β_{est}

takes a long time to converge to its new value, and it even moves in the wrong direction first.

As mentioned, the covariance matrices for this case are set to the same values as for the detection of a drop in Q. In Fig. 5.9 the results of a simple tuning of these matrices is shown. Here, the uncertainty in β is increased, causing it to react faster.

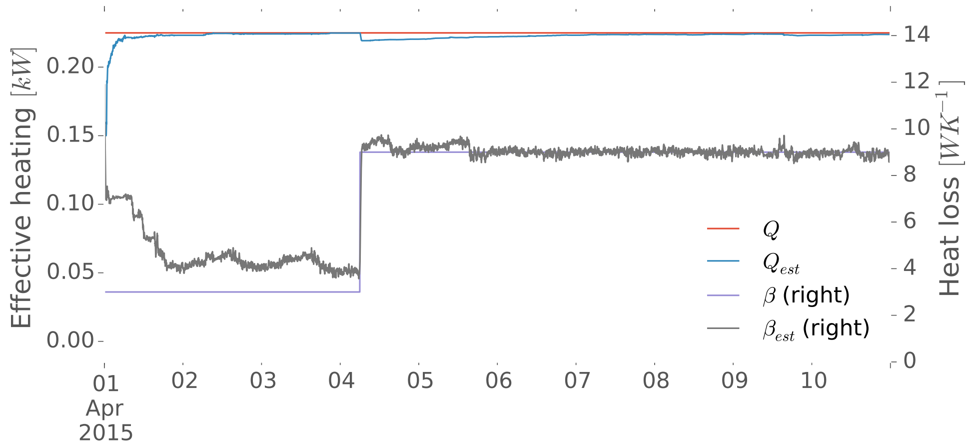

Figure 5.9: Tracking Q and β when β suddenly increases significantly, with tuned covariance matrices for this specific failure

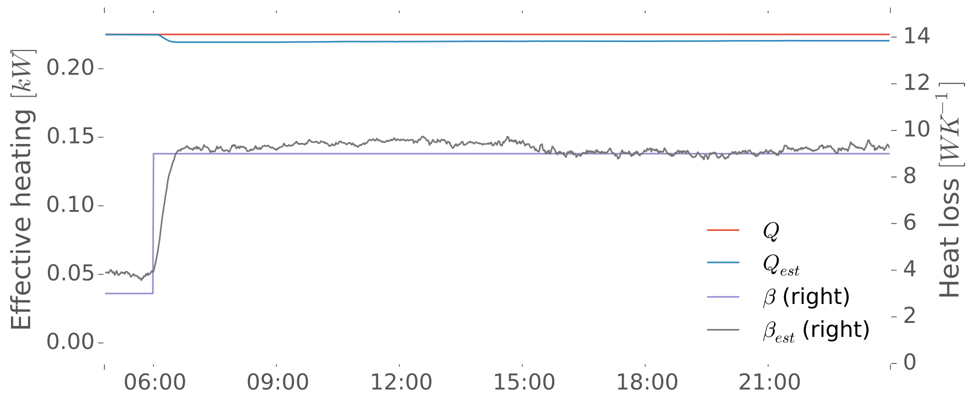

Figure 5.10: Detail view of the sudden increase in β, with tuned covariance matrices for this specific failure

From Fig. 5.9 it can be seen that the parameter is much better tracked than before tuning the covariance parameters.

5.3.3 Slow degradation of heating

So far we have looked at sudden changes in the behaviour of the system. If these changes are severe, the resident is likely to notice something is wrong and call a mechanic. With slow degradation, there is a timeframe in which the resident will not notice anything while the system is not performing optimally. The question now remains: is it possible to detect slow degradation of the system?

To answer this question the parameter Q is slowly decreased over a period of a month, while all other parameters are again kept constant. The covariance matrices are kept the same as in the situation with the sudden drop in heating.

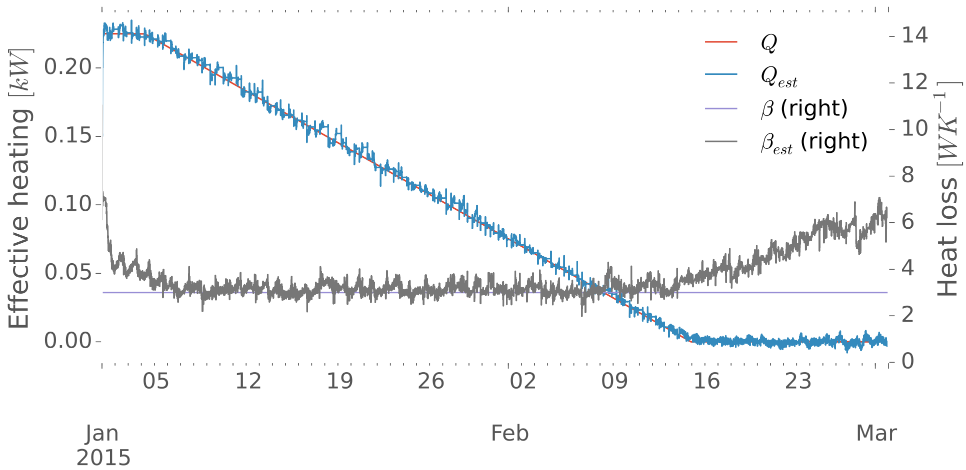

Figure 5.11: Tracking Q and β when Q slowly drops to zero.

In Fig. 5.11 it can be seen that the filter tracks the parameter over time nicely. When the heating has stopped almost completely, β_{est} starts to diverge, much like we saw when the heating stopped working suddenly.

5.3.4 Slow increase in heat loss

Here, the value of β is slowly increased over a period of a month, while all other parameters are kept constant. Also, the covariance matrices of the filter are the same as the ones used when tracking Q, and have thus not be tuned to respond to changes in β.

In Fig. 5.12 the results are shown. The filter is able to track the slowly varying value of β much better than when it varied quickly. This time, there is no confusion as to which parameter changed (Q or β).

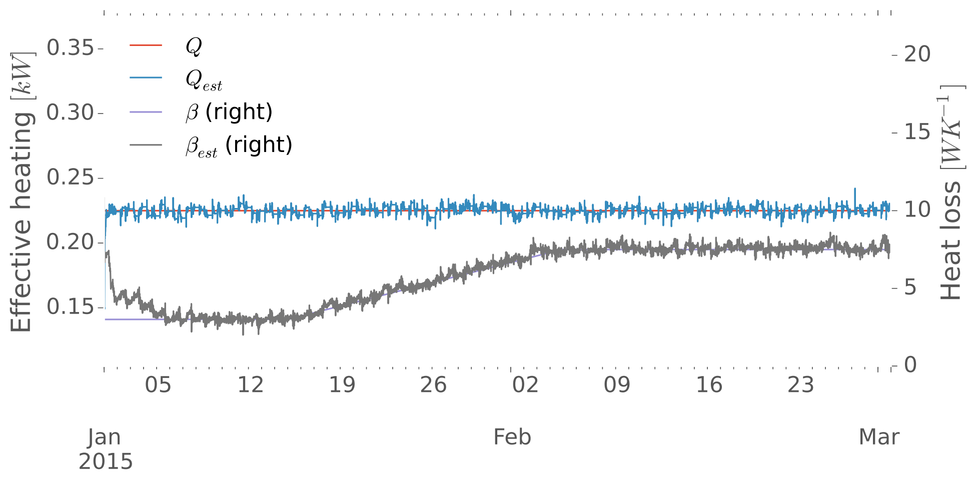

Figure 5.12: Tracking Q and β when β slowly increases.

Note that it might look as if the estimates for Q are more noisy than in the previous cases, but this is simply caused by a change in the axis - the scale is still the same.

5.4 Applying the filter to data of a household

In the previous sections we have seen it is possible to track the parameters of the model using the EKF. In the real world there are many more disturbances and imperfections than in the simulation run before, so it is interesting to see how the filter deals with this kind of imperfect data.

To keep the results interpretable, a nice household is selected, based on two criteria: the boiler must be controlled using the On-Off protocol, and the measurements must be nice. Nice here means there are no big gaps, outliers or erratic behaviour. All of those real-data problems must be of course be solved before implementing this algorithm, but are outside of the scope of this research.

A period of ten days at the beginning of april is selected. The length of ten days is chosen because it is the longest interval in which no big gaps occur in the data, and where it is sufficiently cold. The outside temperature is taken from the KNMI at "de Bilt[1]", because no detailed location information is available in the anonymized dataset. Also note that the input signal is low-pass filtered to mimic the slow response of the system.

[1]De Bilt is a typical choice as the average for the whole of Netherlands

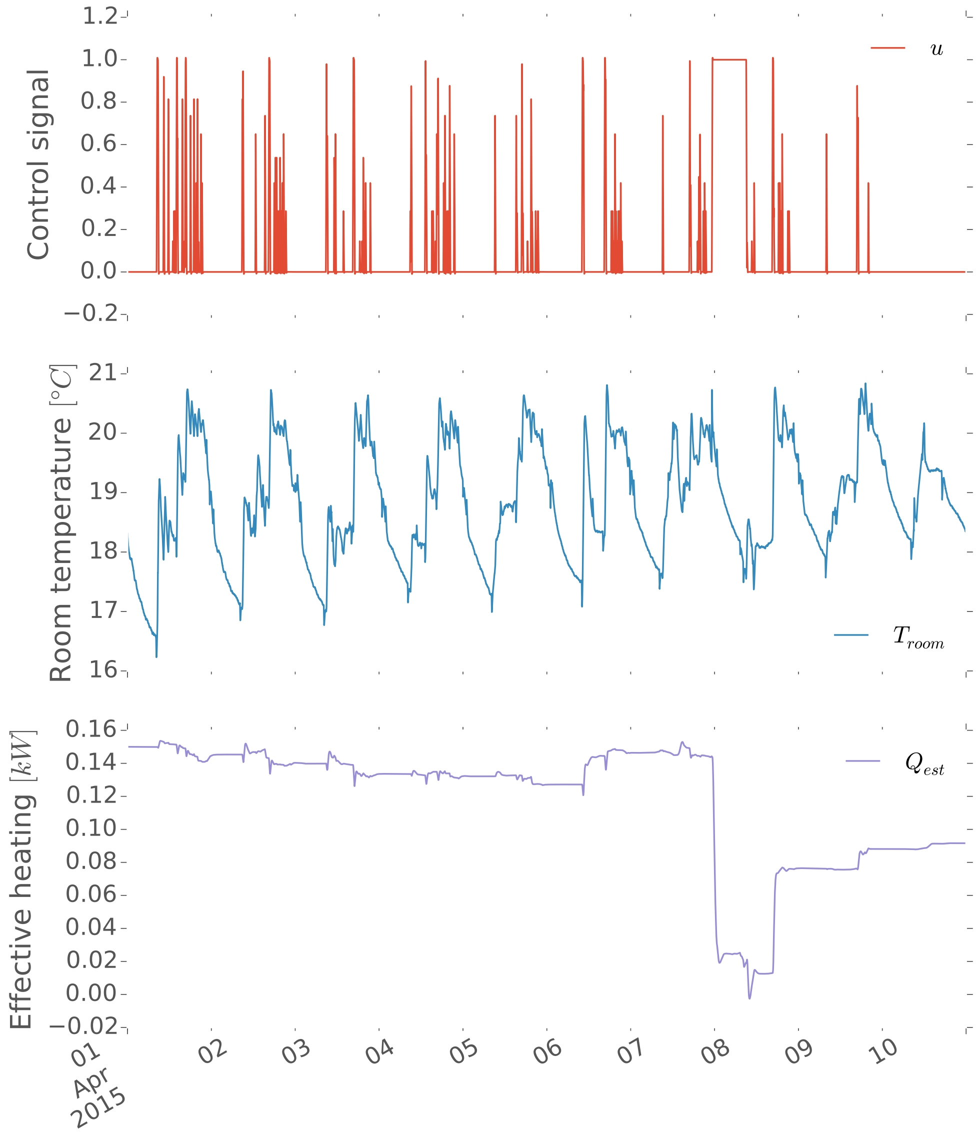

Figure 5.13: Tracking Q over a period of 10 days for a real household

In Fig. 5.13 the result of tracking the Q parameter over the course of ten days is shown. Here, u is the control signal from Toon® to the boiler, T_{room} is the measured room temperature, and Q_{est} is the estimated heating parameter. Interestingly, the value or Q_{est} converges to a value of 0.15, only to drop to almost zero rapidly near the end of the period. The drop seems to correspond to a long period of heating.

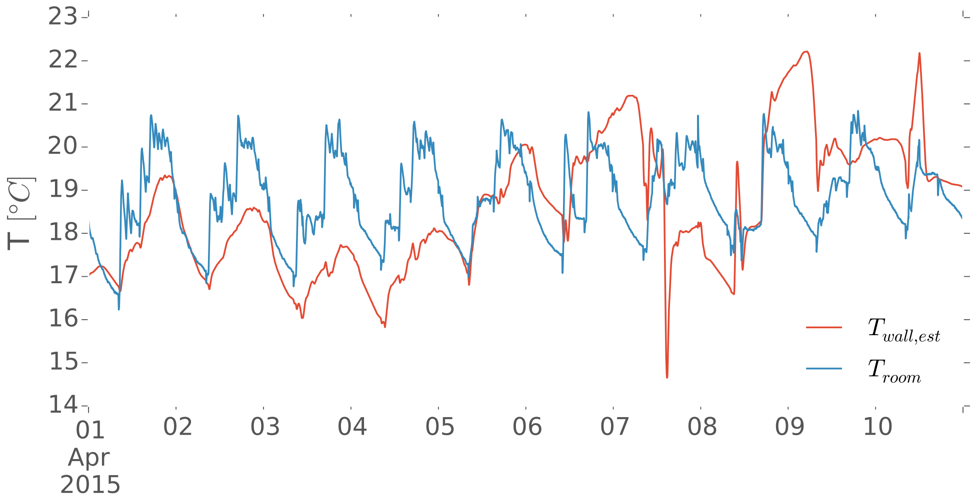

Figure 5.14: Detail view of the dip in Q when the filter is applied to real data, showing a fault

To see what is going on, in Fig. 5.14 a detailed view of the period around the 8th of april is shown. It is clearly visible that even though the Toon® requests full heating over the course of 8 hours, the temperature inside the house does not rise. In fact, the temperature is steadily dropping over this time period, which should not be possible when everyhing is working correctly.

Near the end of the period the behaviour of the system seems to go back to normal again, but the value of Q does not rise to 0.1 again. Unfortunately, it is not possible to see if the filter would pick up the value again later, due to large holes in the data after this period.

In Fig. 5.15 the wall temperature as estimated by the filter is shown. The first few days the estimation seems odd at times, but generally reasonable. However, near the 8th, the estimates are all over the place, jumping up and down.

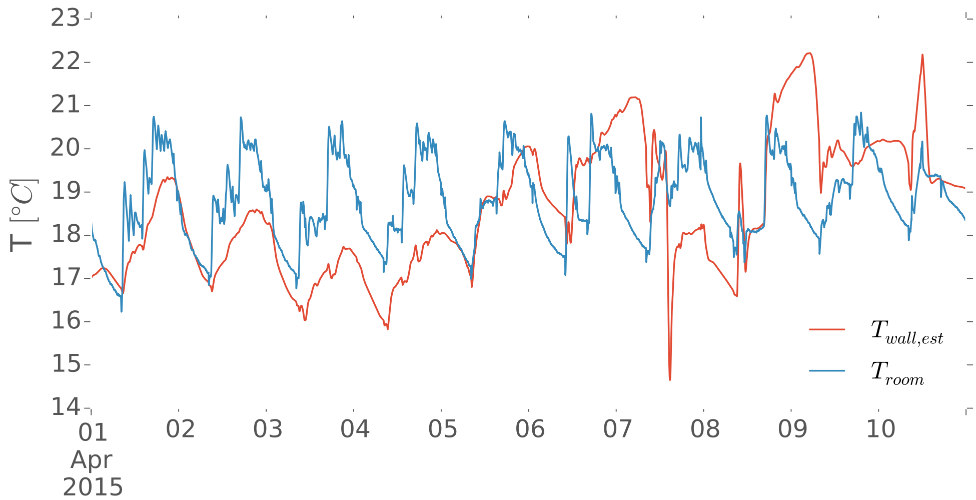

Figure 5.15: Tracking the wall temperature over the course of ten days

5.5 Effect of covariance windup

As discussed in the paper by Evestedt[25], when the input signal is not sufficiently exciting, the uncertainty in the state will grow at least linearly. This effect is called (covariance) windup, and results in unstable behaviour when the uncertainty grows too large.

In the model used in this research, it is possible for the heating to be inactive for long periods of time. When there is no heating, there is no information about Q present in the signal. This makes the estimator susceptible to the windup problem. The effect of the problem is shown in Fig. 5.16.

It can be seen that in the periods between heating, the entry in the covariance matrix that denotes the uncertainty in Q is growing linearly. The increase in uncertainty places heavier weight on the new measurements, which introduces peaks in the estimate of Q whenever new information is available.

The linear increase is mainly caused by the following equation in the predict step:

$$\boldsymbol{P}[k|k-1] = \boldsymbol{F}\boldsymbol{P}[k-1|k-1]\boldsymbol{F}^{\intercal} + \boldsymbol{Q}$$

[25] Magnus Evestedt and Alexander Medvedev. "Stationary behavior of an anti-windup scheme for recursive parameter estimation under lack of excitation". *Automatica* 42, pp. 151–157, 2006.

Here, $\boldsymbol{Q}$ is a tunable parameter of the Extended Kalman Filter, and the value of this matrix is usually constant.

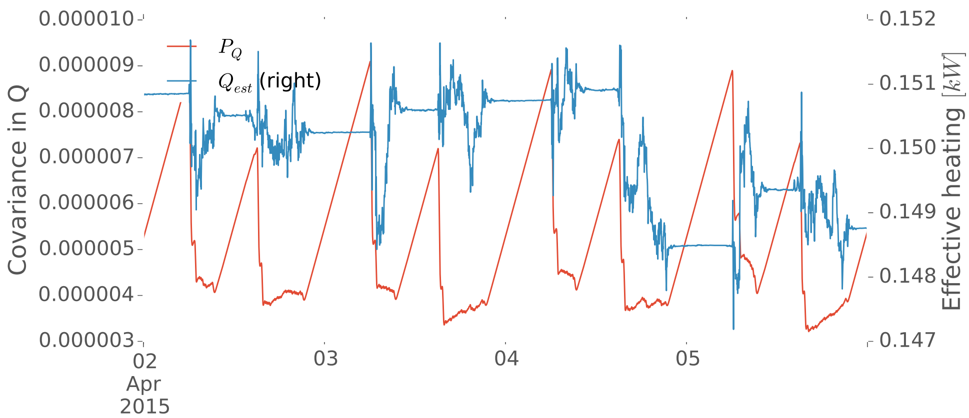

Figure 5.16: Effect of lack of excitation on covariance

A simple method of dealing with this problem is by artificially reducing the uncertainty in the model to zero when there is no information on Q available, effectively ignoring the measurements. This works for Q because there is a direct relationship between the input and Q. For more complex behaviour like β, a more elaborate method is required.

The scheme used here is implemented as follows: in the update step, the diagonal entry of the covariance matrix $\boldsymbol{Q}$ which corresponds to the uncertainty in the model parameter Q is multplied with $u^2[k]$. Here u is the low-pass-filtered control signal, which takes a value between 0 and 1. The result of this multiplication is that when there is no input signal, the uncertainty in the parameter remains constant. This new covariance matrix is named $\boldsymbol{Q}_{modified}$. By using this scheme, the update step mentioned earlier becomes:

$$\boldsymbol{P}[k|k-1] = \boldsymbol{F}\boldsymbol{P}[k-1|k-1]\boldsymbol{F}^{\mathsf{T}} + \boldsymbol{Q}_{modified}$$

In Figure 5.17 the result of this modification is shown. The linear increase of the covariance between the periods of heating have dissapeared, and the estimate for Q has far less overshoot whenever new information comes in (note the difference in scale on the y-axis for both values).

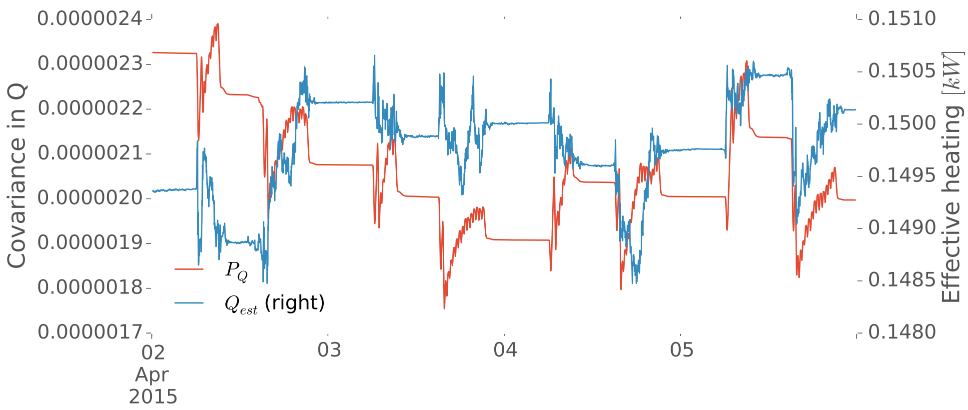

Figure 5.17: Effect of ad-hoc approach limiting the windup of covariance, resulting in more stable tracking of Q

5.6 Summary

In this chapter we have seen that the Extended Kalman filter can be used to find the parameters of a residential house model with central heating. The filter works reliably in the presence of white noise, and can track parameters that change over time.

The results also show that it is more reliable to track a slowly varying parameter than a sudden change. The system handled the slow degradation of the heating and the slow increase in heatloss without retuning, while the sudden increase in heatloss required a manual retuning to be correctly tracked.

In a way this makes sense: when the boiler is heating the house, the filter can not know where the sudden fault originates. Decreasing the incoming flow of heat has an effect comparable to an increase of the outgoing flow of heat. However, when there is a change in heating, these two faults can be separated.

Fortunately, despite the bad performance on sudden increases in β, it is possible to tune the system to provide a relative robust estimate for Q. This means that it is possible to detect faults in the heating, while ignoring faults in heatloss. Since increases in heatloss do usually not require an intervention, this is more than acceptable.

The filter is applied to the data from a real household somewhere in the Netherlands, where it tracked its parameters over the course of ten days. Even though not all parameters were estimated reliably, the Q parameter proved to be a good indicator of a fault by indicating a fault the author did not know was there.

In an attempt to minimize the effect of covariance windup, which can cause instability in the estimates when the input signals do not contain enough information, an anti-windup scheme is implemented. Despite its simplicity, the scheme works just as expected, and greatly reduces overshoot of the estimate when new information about Q comes in.

6 Conclusions & Recommendations

At this point the research is finished. The main findings are presented in this chapter. Finally, some recommendations for further research are presented.

6.1 Conclusions

To detect the most common faults in central heating systems, a model based on physical principles is built. Using this model, a parameter estimation scheme is designed, which is tasked to track parameters that are an indicator of the most common faults.

It is shown that by using an Extended Kalman Filter, it is possible to track these parameters of both the model and a real system over time. The Extended Kalman Filter is able to detect slow performance degradations of the system, as well as sudden changes in the model parameters. These parameters give valuable information about the state and functioning of the system, and can be used to detect common faults in a central heating system.

To conclude, the Extended Kalman filter shows great potential in being the foundation in a new detection scheme for central heating boiler malfunctions. A scheme which can be applied to every household with a Toon, regardless of boiler brand or type.

Limitations

When setting up the filter, there are a number of parameters that require tuning. Manual tuning has shown that the system is sensitive to the choice of initial conditions, with the wrong initial conditions convergence becomes slow. This is not desirable in a system which is meant to be deployed to hundreds of thousands of systems.

The noise that was added to the measurements is zero mean white noise, which is not realistic. In a normal household, deviations in temperature are likely caused by human activities like cooking and showering, or by operating devices that give off heat, such as dryers. All of these disturbances have in common that they are not zero mean but biased. Further research is needed to see what kind of effect these more realistic noise models have, and how the system can be improved to deal with them.

Finally, the anti-windup scheme as implemented for the heating parameter Q works fine for Q, but can not be used for other parameters directly. Since β has the tendency to drift when Qu is nearly zero for a while (which is either when there is a fault, or during warmer months), it appears the signal is not rich enough for β at times either, but no scheme is implemented yet for this parameter.

6.2 Recommendations

The first step in a detection scheme is designed in this research. However, tracking model parameters alone is not enough to automatically detect faults in a reliable manner. To make this research applicable, it is needed to build a detection model on the outcome of the filter.

At this moment a pilot is running at Eneco where a group of households are invited to share fault information. This means that the detection scheme can be trained on a real, labelled set, rather than simulated data. This also opens the possibility of using the algorithm to detect more elaborate faults, since more information on how these faults really look is available.

Tuning the covariance matrices is still a form of art, rather than science. To make it easier to deploy the scheme to a large group of households, further research should look into auto-tuning methods for the covariance.

Sometimes the estimate for parameters such as Q and β become negative. These kind of values can result in unstable behaviour, and should thus be avoided. There are papers suggesting it is possible to use constraints with the Kalman Filter[26,27,28], and this is something that should be looked at in further research.

[26] D Simon and T L Chia. "Kalman filtering with state equality constraints - IEEE Xplore Document". *IEEE transactions on Aerospace and ...*, 2002.

[27] D Simon. "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms". *Control Theory & Applications, IET* 4, pp. 1303–1318, 2010.

[28] D M Walker. "Parameter estimation using Kalman filters with constraints". *International Journal of Bifurcation and Chaos* 16, pp. 1067–1078, 2006.

Since the anti-windup scheme can not be used for other parameters than Q, a more advanced method should be adopted to prevent the divergence of other parameters when the input is not persistently exciting.

Since persistence of excitation can be a problem in this scheme, the controller could be outfitted with a "diagnostics" mode, where it creates a rich input signal. This rich signal will likely aid the detection and diagnosis of the fault considerably. Most faults are now detected (by residents) during the start of the colder season. By using this mode it would be possible to spread the detection of faults throughout the year.

Solar radiation is not taken into account in this research, as it was found that no constant value could reliably be found over the course of two days. This is likely caused by local effects such as shadows and the orientation of the house. Because it is shown that the EKF can track time-varying parameters, future research could be aimed at estimating the time-varying solar influence, thereby creating a model which also works in summer months.

Finally, the data available did not span the winter months. In future research it would be interesting to see how the performance of the Extended Kalman Filter differs for colder months, where the effect of heating is expected to be more pronounced.

To conclude, the results of this research are promising. This research could be the starting point of an exciting fault detection and identification scheme, which could potentially be deployed to all customers, as compared to the 11% of the customer base used now in the BoilerIQ application.

Appendix A: Python base class

```
from sympy import symbols, Matrix
from cached_property import cached_property
from sympy.utilities.autowrap import autowrap
import numpy


class DiscreteNonlinearModel(object):

    states = []
    inputs = []
    parameters = []
    Ts = 1

    def __init__(self):
        self.state = Matrix([0]*len(self.states))

    @cached_property
    def analytical_f(self):
        raise NotImplementedError()

    @cached_property
    def analytical_h(self):
        raise NotImplementedError()

    @cached_property
    def _lambda_f(self):
        v, f = self.analytical_f
        return autowrap(f.T, args=list(v))

    @cached_property
    def _lambda_F(self):
        v, f = self.analytical_f
        states = symbols(self.states)
        F = f.jacobian(states)
```

```
        return autowrap(F, args=list(v))

    @cached_property
    def _lambda_B(self):
        v, f = self.analytical_f
        inputs = symbols(self.inputs)
        F = f.jacobian(inputs)
        return autowrap(F, args=list(v))

    @cached_property
    def _lambda_h(self):
        v, h = self.analytical_h
        return autowrap(h.T, args=list(v))

    @cached_property
    def _lambda_H(self):
        v, h = self.analytical_h
        states = symbols(self.states)
        H = h.jacobian(states)
        return autowrap(H, args=list(v))

    def f(self, inputs):
        """x[k+1] = f(x[k], u[k])"""
        return self._lambda_f(*self.state.flatten(), *inputs, *self.params)

    def F(self, inputs):
        return self._lambda_F(*self.state.flatten(), *inputs, *self.params)

    def B(self, inputs):
        return self._lambda_B(*self.state.flatten(), *inputs, *self.params)

    def h(self, state, inputs):
        """y[k] = h(x[k], u[k])"""
        return self._lambda_h(*state.flatten(), *inputs, *self.params)

    def H(self, inputs):
        return self._lambda_H(*self.state.flatten(), *inputs, *self.params)

    def predict(self, inputs):
        state = self.advance(inputs)
        return self.h(state.flatten(), inputs), state

    def advance(self, inputs):
        self.state = self.f(inputs)
```

```
        return self.state
```

Appendix B: Python EKF implementation

```
import numpy as np
from sympy import eye


class ExtendedKalmanFilter(object):

    def __init__(self, model):
        self.model = model
        self.Q = np.eye(len(model.states))
        self.R = np.eye(len(model.outputs))
        self.P = 1e-3
        self.K = 0

    def predict(self, y, inputs):

        F = self.model.F(inputs)
        H = self.model.H(inputs)

        x_hat_k_m = self.model.f(inputs).T
        P_k_m = F.dot(self.P).dot(F.T) + self.Q

        K_k = np.divide(P_k_m.dot(H.T), H.dot(P_k_m).dot(H.T) + self.R)

        x_hat_k = x_hat_k_m + K_k.dot(y - self.model.h(x_hat_k_m, inputs))
        self.P = (np.eye(len(K_k)) - K_k.dot(H)).dot(P_k_m)

        self.model.state = x_hat_k
        self.K = K_k

        return self.model.h(x_hat_k, inputs), x_hat_k
```

Appendix C: Python house model

```python
from sympy import symbols, Matrix


class RealisticBoilerModel(DiscreteNonlinearModel):
    states = ['Tin', 'Twall', 'Q', 'beta_hat', 'beta', 'beta_bar', 'Cw']
    inputs = ['Tout', 'u']
    parameters = []
    outputs = ['Tin']

    def __init__(self):
        super().__init__()
        self.params = []


    @property
    def analytical_f(self):
        variables = symbols(self.states + self.inputs + self.parameters)
        Tin, Twall, Q, beta_hat, beta, beta_bar, Cw, Tout, u = variables
        return variables, Matrix([
                Tin - (self.Ts*beta)*(Tin-Twall) - (self.Ts*beta_hat)*(Tin-Tout) + self.Ts*Q*u,
                Twall - (self.Ts*Cw*beta)*(Twall-Tin) - (self.Ts*beta_bar*Cw)*(Twall-Tout),
                Q,
                beta_hat,
                beta,
                beta_bar,
                Cw
            ])

    @property
    def analytical_h(self):
        x, *v = symbols(self.states + self.inputs + self.parameters)
        return [x] + v, Matrix([x])
```

```
def linear_statespace(self):
    # Only the first two states are important in the linear statespace model
    # All other states here are only there for finding the parameters
    # Also, this model is linearised around u = 0
    u = np.zeros(len(self.inputs))
    A = self.F(u)[:2, :2]
    B = self.B(u)[:2]
    C = self.H(u)[:, :2]
    D = np.zeros([C.shape[0], B.shape[1]])
    return A, B, C, D
```

Bibliography

[1] Urban Persson and Sven Werner. "Heat distribution and the future competitiveness of district heating". *Applied Energy* 88, pp. 568–576, 2011. (Referenced on p. 5).

[2] Harish Satyavada and Simone Baldi. "A Novel Modelling Approach for Condensing Boilers Based on Hybrid Dynamical Systems". *Machines* 4, pp. 10–10, 2016. (Referenced on p. 6).

[3] Simone Baldi, Thuan Le Quang, Ondrej Holub, and Petr Endel. "Real-time monitoring energy efficiency and performance degradation of condensing boilers " (referenced on pp. 6, 7).

[4] L Peeters, J Van der Veken, H Hens, L Helsen, and W D'haeseleer. "Control of heating systems in residential buildings: Current practice". *Energy and Buildings* 40, pp. 1446–1455, 2008. (Referenced on p. 8).

[5] Dennis Ramondt. "Savings from Smart Thermostats with Energy Displays". *Available at SSRN 2745144*, 2015. (Referenced on p. 9).

[6] Comsol. *COMSOL Multiphysics: Version 3.3*. 2006. (Referenced on p. 14).

[7] V Gerlich. "Modelling Of Heat Transfer In Buildings." *ECMS*, 2011. (Referenced on pp. 14, 24).

[8] H E A Van den Akker and R F Mudde. "Fysische transportverschijnselen I". 1998. (Referenced on pp. 14, 18, 19).

[9] Klaus Kaae Andersen, Henrik Madsen, and Lars H Hansen. "Modelling the heat dynamics of a building using stochastic differential equations". *Energy and Buildings* 31, pp. 13–24, 2000. (Referenced on p. 15).

[10] Léon Peter Bernard Marie Janssen and Marinus Maria Cornelis Gerardus Warmoeskerken. *Transport phenomena data companion*. 1997. (Referenced on p. 17).

[11] David R Lide. *CRC Handbook of Chemistry and Physics, 85th Edition*. CRC Press, June 2004. (Referenced on p. 18).

[12] J M Gordon and Y Zarmi. "Massive storage walls as passive solar heating elements: An analytic model". *Solar Energy* 27, pp. 349–355, 1981. (Referenced on p. 21).

[13] Mario Vašak, Antonio Starčić, and Anita Martinčević. “Model predictive control of heating and cooling in a family house”. *MIPRO*, pp. 739–743, 2011. (Referenced on pp. 21, 24).

[14] Wesley J Cole, Kody M Powell, Elaine T Hale, and Thomas F Edgar. “Reduced-order residential home modeling for model predictive control”. *Energy and Buildings* 74, pp. 69–77, 2014. (Referenced on p. 21).

[15] Jan-Willem Van Wingerden and Michel Verhaegen. “Subspace IDentification of MIMO LPV systems: The PBSID approach”. *2008 47th IEEE Conference on Decision and Control*, pp. 4516–4521, 2008. (Referenced on p. 22).

[16] Intergas. *Handleiding/instructie Kombi Kompakt HR-ketel Intergas*. Jan. 2011 (referenced on p. 26).

[17] R E Kalman. “A New Approach to Linear Filtering and Prediction Problems”. *Journal of Basic Engineering* 82, pp. 35–45, 1960. (Referenced on p. 31).

[18] Hongwen He, Zhentong Liu, and Yin Hua. “Adaptive Extended Kalman Filter Based Fault Detection and Isolation for a Lithium-Ion Battery Pack”. *Energy Procedia* 75, pp. 1950–1955, 2015. (Referenced on p. 33).

[19] P Howlett, P Pudney, and X Vu. “Estimating Train Parameters with an Unscented Kalman Filter”. 2004 (referenced on p. 33).

[20] Wei Xue and Ying qing Guo. “Application of Kalman Filters for the Fault Diagnoses of Aircraft Engine”. In: *Kalman Filter*. Mar. 2014. Pp. 1–15. (Referenced on p. 33).

[21] Heather H Lambert. *A simulation study of turbofan engine deterioration estimation using Kalman filtering techniques*. Tech. rep. June 1991 (referenced on p. 33).

[22] R Luppold, J Roman, G Gallops, and L Kerr. “Estimating in-flight engine performance variations using Kalman filter concepts”. In: *25th Joint Propulsion Conference*. 1989. (Referenced on p. 33).

[23] E A Wan and R Van Der Merwe. “The unscented Kalman filter for nonlinear estimation”. *… 2000 AS-SPCC The IEEE 2000*, 2000. (Referenced on p. 34).

[24] Mohammad Taghi Sabet, Pouria Sarhadi, and Mostafa Zarini. “Extended and Unscented Kalman filters for parameter estimation of an autonomous underwater vehicle”. *Ocean Engineering* 91, pp. 329–339, 2014. (Referenced on p. 34).

[25] Magnus Evestedt and Alexander Medvedev. “Stationary behavior of an anti-windup scheme for recursive parameter estimation under lack of excitation”. *Automatica* 42, pp. 151–157, 2006. (Referenced on p. 48).

[26] D Simon and T L Chia. "Kalman filtering with state equality constraints - IEEE Xplore Document". *IEEE transactions on Aerospace and ...*, 2002. (Referenced on p. 54).

[27] D Simon. "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms". *Control Theory & Applications, IET* 4, pp. 1303–1318, 2010. (Referenced on p. 54).

[28] D M Walker. "Parameter estimation using Kalman filters with constraints". *International Journal of Bifurcation and Chaos* 16, pp. 1067–1078, 2006. (Referenced on p. 54).