Prediction of electricity consumption of heat pumps for use in an intelligent power-grid

Esben Holm
AAU Energy, Mechatronic Control Engineering
Final semester, Bachelor thesis

Project period: 5 November 2014 – 5 February 2015
Supervisor: Michael Møller Bech
ECTS: 15

Copies: 4
Pages, total: 61
Appendices: 17

Abstract

The Danish power-grid has to include more renewable energy. By 2050 the consumption has to be 100 % based on renewable energy. This means more wind-power, and other intermittent power producers will be incorporated in the grid. Since the energy-producers are not well-suited to change their production, the demand-side has to be more flexible in the future.

One way of making the demand more flexible is by gathering electricity consumers in a Virtual Power Plant (VPP) and control when they are using power and not. This study is focused on predicting the power of a VPP consisting of a pool of heat pumps.

A model has been developed, together with methods for estimating the parameters for the model, and an approach to evaluating the performance of the model. The model takes data from the "Control Your Heat Pump" platform to make the parameter-estimation. The report demonstrates how this model is deployed by using Matlab and YALMIP.

Preface

This report has been written during the period of my bachelor-project, from 5 November 2014 till 5 February 2015, at Aalborg University, Department of Energy Technology, with specialisation in Mechatronic Control Engineering. It gives an overview of the most important findings from the project, which was focused on developing a model for predicting the electricity consumption of heat pumps controlled in a VPP[1].

From 1 May – 20 October 2014 I was an intern at the company Neogrid Technologies, where I was working with developing and refining a VPP controller, which consisted of a pool of about 70 heat pumps actively under control. The development of this controller was part of a research project, where one of the stakeholders was Energinet.dk, the power-system responsible of Denmark. The overall objective was to develop and demonstrate a new solution for an intelligent power-grid. During my internship I found some difficulty with developing the VPP controller, which sparked the basis for this bachelor project.

The report is structured into three chapters and a conclusion. In chapter 1 a thorough description is given of the system at hand, starting with an overall picture of the power-grid, but quickly targeting into the VPP from the READY project, and the individual heating system in a house. The chapter ends with defining the problem statement, which supports the overview of the report.

In chapter 2 the model developed during the project is described, with an overview of how the parameters are estimated. This chapter refers to appendix B for some further elaboration on the methods used that do not directly add to the overview of the report.

In chapter 3 is considered a method for validating the model through old measurement data. The same approach may be used for evaluating the actual performance of the model, yet the implementation of the model into the VPP controller is not part of this report.

Through the report and in the appendices are given references to code in Matlab. These are considered demonstrations of how it would be written in Matlab and are minimum examples; it will not be meaningful to run these lines of code without the data. Some of the lines of mcode are making use of the optimisation toolbox YALMIP. I hope it is clear from the comments and description for the code, what the commands will do.[2]

[1] Virtual Power Plant

[2] Here I might have to put some disclaimer. Matlab is owned by MathWorks and YALMIP is developed by Johan Löfberg. I do not own any of these names, and have no intention of offending the rightful owners.

Contents

Abbreviations

BRP	Balance Responsible Party
COP	Coefficient of Performance (heat pump efficiency)
DMI	Danish Meteorological Institute
DSO	Distribution System Operator
DVI	Dansk Varmepumpe Industri A/S
GUI	Graphical User Interface
PSO	Public Service Obligations
PV	Photovoltaic, converting solar radiation to electricity
RMS	Root mean square
SDVP	Control Your Heat Pump
SGR	Smart Grid Ready
TSO	Transmission System Operator
VPP	Virtual Power Plant

1. Introduction

By 2050 Denmark will no longer be using fossil fuels, the energy consumption has to come 100 % from renewable sources [6]. To reach this point it is necessary to take smaller steps, slowly incorporating renewable energy in Denmark. Therefore the parliament has agreed on a 2020 energy agreement, to reach 35 % renewable energy in Denmark with 50 % of the electricity demand covered by wind energy.

Incorporating so many wind-turbines in the system, means that conventional power plants are shut down. Wind is known to be a very intermittent resource, which poses challenges in incorporating it in the current electric grid. Historically the electricity producers, the power plants, have had the task of balancing production with demand. If there was too little power in the grid, the power plants should produce more, and less if there was too much power. Thereby there will always be power from the socket when needed. There must always be balance in the electric grid, yet in the future it might be the consumers who have to adjust to the producers.

The flexibility of a power plant is relatively high compared to a wind turbine. It is possible to adjust the production by adjusting the amount of fuel you put into the power plant, whereas with a wind-turbine or a PV panel it is not possible to adjust the fuel input; we can not store wind or solar radiation. Therefore, when the electricity production becomes more intermittent, the need for flexibility on the demand-side becomes bigger.

Apart from putting up more renewable energy production, the demand itself should also move from fossil fuels to renewable ones. In private households where district heating is not installed, oil boilers are used to heat the house. These will in the coming years be replaced by boilers running on renewable fuels, or by heat pumps. Installing an electric heat pump will also make use of the extra electricity that will be produced from wind power. Though if the electricity consumption of these heat pumps is not controlled intelligently, and they consume power when there is no wind, they might do more damage than good for the total system.

Therefore, the 2020 agreement also focuses on the development of an intelligent electric grid in Denmark. Going back to 1998 the parliament already then thought about researching in the field of an intelligent electric grid, when they decided that the system-responsible Energinet.dk should use some of the PSO-money[1] on research. This lead to the foundation of ForskEL, a program to finance and support research on intelligent electric systems, among other areas. [7]

[1] PSO: Public Service Obligations.

1.1 READY project

When talking about intelligent electric systems, one of the main objectives is making the demand-side more flexible and controllable. This was the focus of the READY[2] project, which demonstrated a way to control a number of heat pumps as one pool. The objective was to keep the total power consumption of the pool of heat pumps at a certain reference level.

When several power-consumers are combined with a central controller, it is called a Virtual Power Plant (VPP). The VPP will get an input for the power-reference from a main operator, who perhaps also controls other VPPs, this is called an aggregator. On figure 1.1 is shown a simple diagram of how the aggregator sends a power plan for the next day to the VPP controller. The controller decides which heat pumps are turned on and off, determined from the data available for the VPP controller.

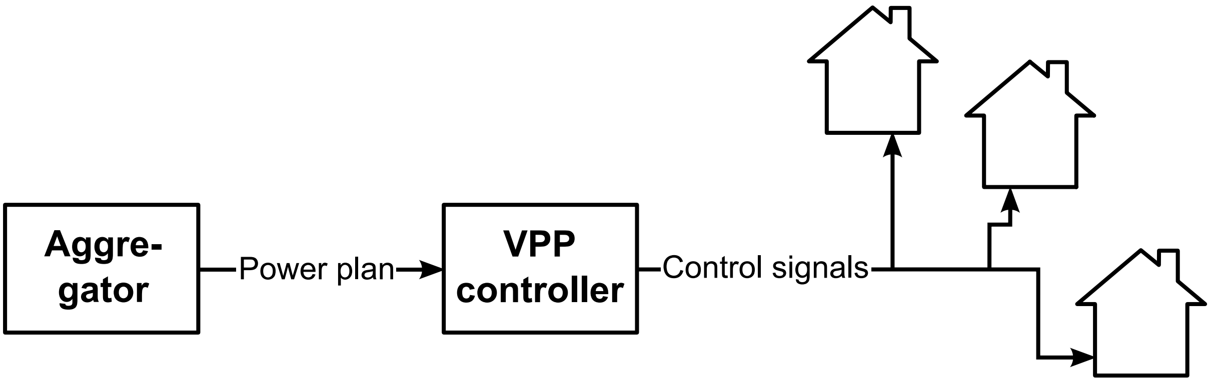

Figure 1.1: The aggregator determines how much power should be used. The VPP controller then determines how many and which heat pumps should be turned on. Naturally there are limits for the power plan, which are determined from measurement data from the houses.

The aggregator has to decide on how much power is going to be used by the VPP. This is typically done one day ahead, when the electricity price is settled for the next day. Of course there are certain limits that the power has to be within. It is not possible for the pool of heat pumps to use more power than the sum of all the heat pumps, just as it is not possible to use negative power (actually producing electricity to the grid). These are absolute limits.

There are also limits that varies from day to day and during the season. Depending if it is winter or summer, there is a bigger or smaller demand for heating, respectively. In the same sense, there are differences in the heat-demand depending on the weather from day to day. Even though it might be possible to go to the absolute limits, there should not be any reduction in comfort inside the houses. The inside temperature must be kept within a reasonable temperature-window, and there must always be hot water for the inhabitants.

Demonstrating how to make and the function of a VPP controller for heat pumps was the objective of the READY project, which was supported with PSO-money from the ForskEL programme.

1.1.1 VPP controller communication flows

The job of the aggregator is to choose and make the best power plan for the pool of heat pumps day-ahead. This means that in the morning the aggregator gathers all the information

[2]The official name of the project is *Smart Grid Ready VPP Controller for Heat Pumps*, in short READY.

available and wanted, and sets an hourly plan for the next days, each hour setting a certain desired amount of power. The plan is then sent to the VPP controller. In short the task of the controller is to determine how many and which heat pumps should be turned on and off.

Naturally, there are limits as to how much and how little power can be used by the pool of heat pumps. Therefore measurements and other data is gathered, not only for the VPP controller to determine the best execution, but also for the aggregator to make a reasonable plan for the pool. In figure 1.2 are shown the communication flows for the VPP controller. Since this was developed during a research project, the power-plan does not come directly from an aggregator, who are in it for the money. Yet the READY project was managed by the energy trading company Neas Energy, who wanted to use the VPP as part of their energy portfolio. The power plan is made from weather data, prediction of the need for heating for the houses and the spot price.

When the power-plan is made it is used together with measurement data from the households, to send a start/stop schedule to each heat pump. Each heat pump is installed with a box made by Liab that collects measurements and turns the heat pump on and off according to the schedule. This is also the box that sends measurement data back into a server, where it can be fetched from. A little more explanation on the Liab box is given in section 1.2.1.

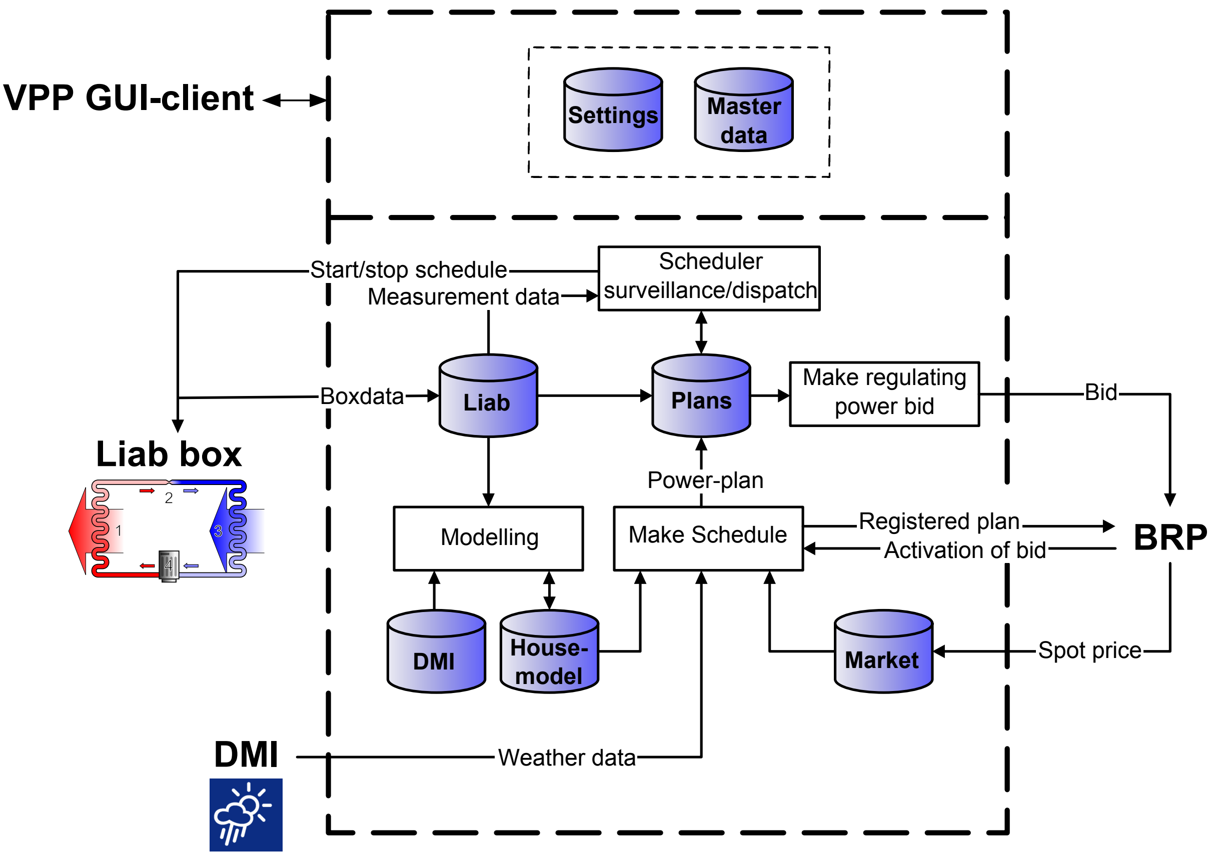

Figure 1.2: Diagram of communication flows in VPP controller. DMI is the *Danish Meteorological Institute* and BRP is the balance responsible party. The VPP GUI-client is the interface to input a power plan for the VPP to follow, this is supposed to be used by an aggregator.

Balance Responsible Party

Since the VPP has potential for quite quickly going up or down in electricity consumption, it can be used by a Balance Responsible Party (BRP). The main task of the BRP is to compensate for any difference during the day, compared to the day-ahead planning. There might have been a wind turbine park that bid in 10 MWh at 3 in the morning. If it only produces 7 MWh, the grid is out of balance by 3 MWh. Then the BRP must buy some extra power from a power plant, or make someone consume less power. In the case of the pool of heat pumps it would be either turning some of them off, or turning more on, depending on whether it is down-regulation or up-regulation, respectively. In the end this is controlled by the price of electricity within the day. The VPP controller calculates how much power can be turned on and off in the next hour, and sends this to the BRP, who then decides whether it should be used or not.

There are certain settings and master data for the VPP that should be set by an aggregator. This is not considered in this report, but is just to give a broad overview of the idea behind a VPP. The aggregator should have some easy-to-use interface, where settings on how the plan should be made can be entered. Perhaps also bidding strategy for regulating power, in case the aggregator also takes on the role as BRP, as in the case of Neas Energy.

1.1.2 Time-schedule on power-market

Elspot is the day-ahead power market for the Nordic and Baltic regions; every day power is traded to be delivered the next. In the morning buyers and sellers enter their bids and offers and how much power they can possibly provide or consume. At noon the auction closes and the prices are announced soon after. In the afternoon the trades are made and agreed by buyer and seller. The time-schedule of the Elspot is illustrated in figure 1.3.

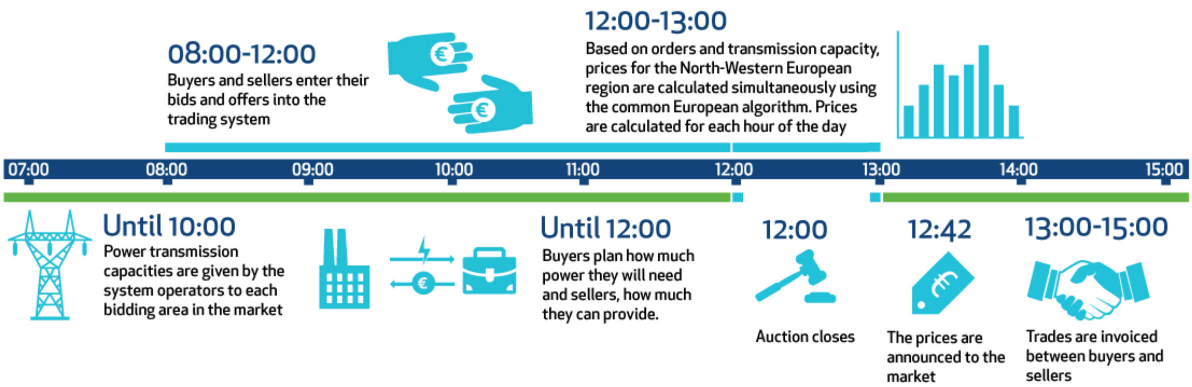

Figure 1.3: Time-schedule on the Nord Pool Spot market Elspot (the time is in CET) [4].

After this, when the day begins, there might be differences from the agreements made. Buyers might not be able to use the amount of power promised, or sellers can not provide the power promised for different reasons. Thus enters the regulating power market, where power trades are made intra-day.

If there is a need for more power in the grid, the price for power will go up for any actor who can provide the needed power on such a short notice. Likewise if there is too much power in the grid the price will go down, giving an incentive to use less power, or perhaps produce less.

In the power market there is value in knowing how much power a system can provide or consume, and in how flexible the power of the system is. One is making money on the day-ahead market, the other on the intra-day market. [4]

Time-schedule of VPP controller

The control algorithms of the VPP controller are executed in four different intervals: Every 24 hours, every hour, every five minutes and an interrupt-based execution.

Every day a model is calculated for the houses in the pool. It takes an average of all the houses and calculates a model for the past day and the fitness value of the model. The model with the best fit is used to create a power-plan for the next day, which is saved in a database. From this power-plan a schedule is made for each of the heat pumps.

The 24 hour time-schedule is shown in figure 1.4. The weather-data comes from DMI in this graphic. The measurement data is saved in the Liab-server, where it has been sent to the database. Neas is the TSO and BRP in this case, where the electricity price for the next day is fetched and the power-plan is submitted to them. In the VPP-server the algorithms for the controller is made, therefore a lot of data is used from the database, while the calculated model and schedule is sent back to the database.

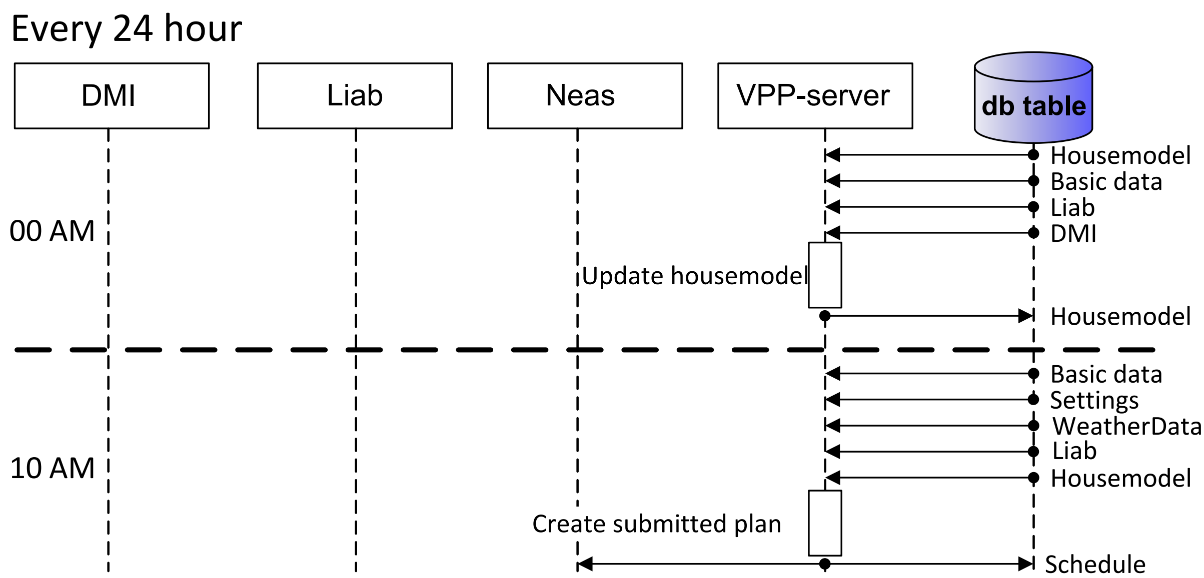

Figure 1.4: Execution made every day for the VPP.

Every hour weather-data from DMI and the spot-price from Neas are saved to the local database. The possible amount of power to turn up or down in the power-reference is calculated, saved in the database and sent to the BRP, in this case Neas. The current power-reference is fetched from the database to the VPP controller, which then calculates the schedule for each heat pump.

The execution for every five minutes is shown in figure 1.5, where the actors are the same. The arrows mark communication from one actor to the other. The blocks are calculations in the VPP controller.

As mentioned earlier the sample-time of the control loop is five minutes. Every five minutes measurement data from the Liab-server is saved to the local database and the dispatch of the heat pumps is executed. The power-reference for the current hour has been loaded and the

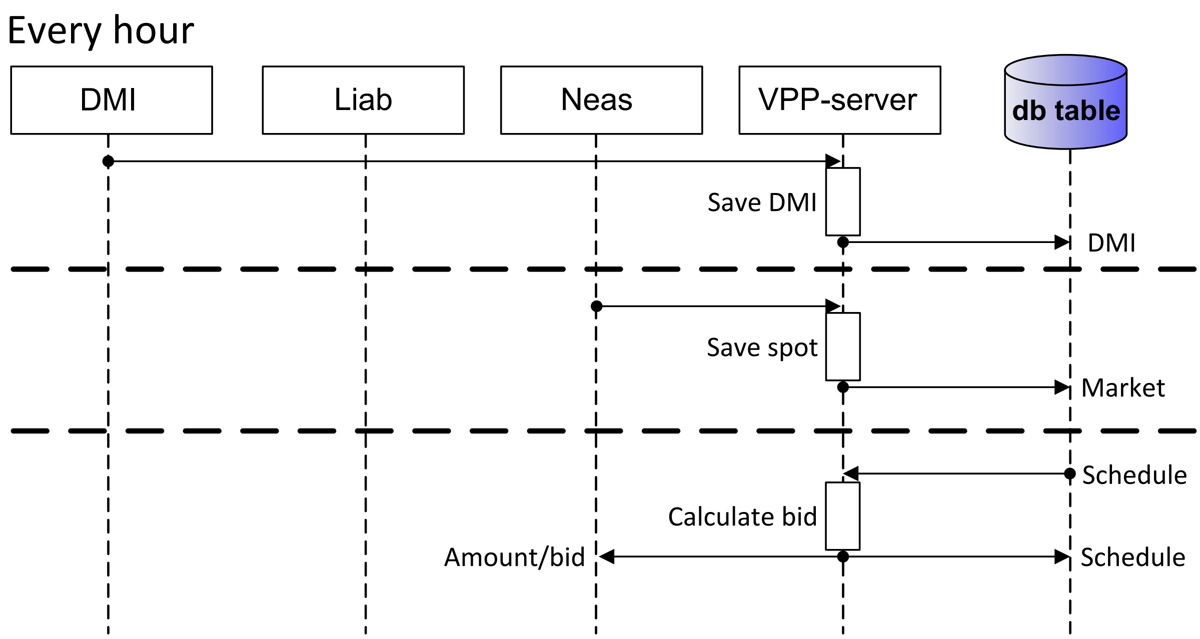

Figure 1.5: Execution made every hour for the VPP.

heat pumps to release or turn off are determined from the data available. Because the heat requirement is highest, the heat pumps in the coldest houses are released first.

In figure 1.5 is shown the execution done every five minutes. Data is loaded into the VPP-server where the dispatch of the heat pumps is executed.

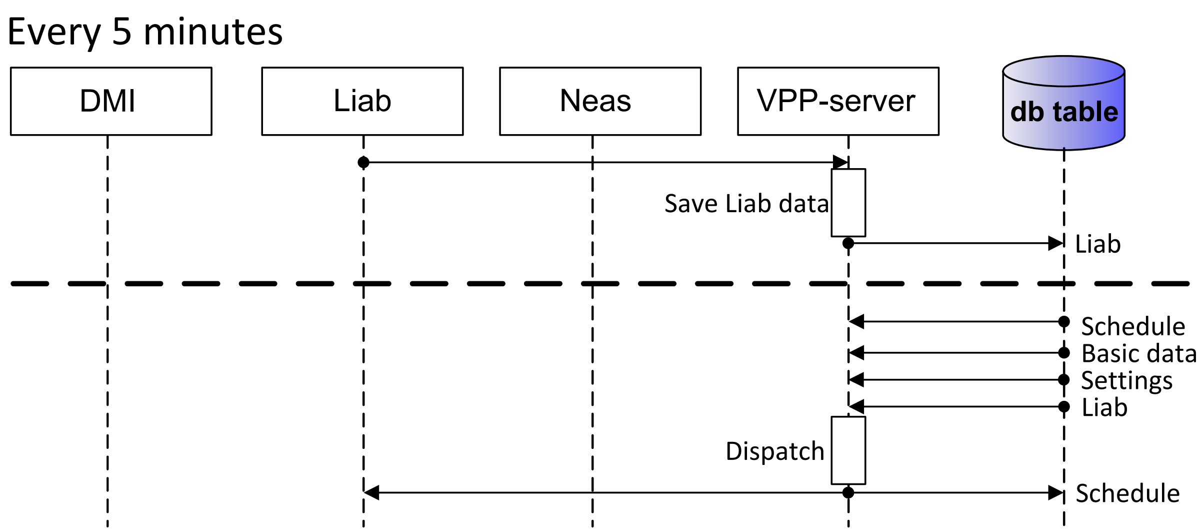

Figure 1.6: Execution made every five minutes for the VPP.

All the time the VPP controller is looking for any new data from the BRP (Neas), checking if there is activated regulating power. When there is a signal from the BRP, the VPP sets a new power-reference from the calculation of the regulating power bid made every hour. This will make the VPP controller turn on or off a number of heat pumps in order to reach the desired new power-reference. This is the interrupt-based execution for the VPP controller developed during the READY project.

1.1.3 Power-plan tracking

As mentioned in section 1.1 this project builds on the READY project, where a VPP controller was demonstrated. The basic idea of the VPP is to combine all the heat pumps in the pool, and control them on the basis of weather and price forecasts as if they were one big unit. It is the sum of the consumption that is interesting when looking into aggregated control.

A plot from the functioning VPP controller is shown in figure 1.7, where the sum of the measured power P_{act} is shown together with the power-reference P_{ref}. Bid is the potential for turning the power-reference up or down. For this period of time the reference will not be turned up, even if there is an activation signal from the BRP for down-regulation[3]. Instead the power can go down if up-regulation is activated. This happens around 7:00, where the green line turns purple in the graph; P_{reg} is the power-reference for this period. It is clear that the power goes down at the time the bid is activated, and it goes up again at the time it is no longer activated. This is how the interrupt-based execution of the regulating power is a part of the VPP controller.

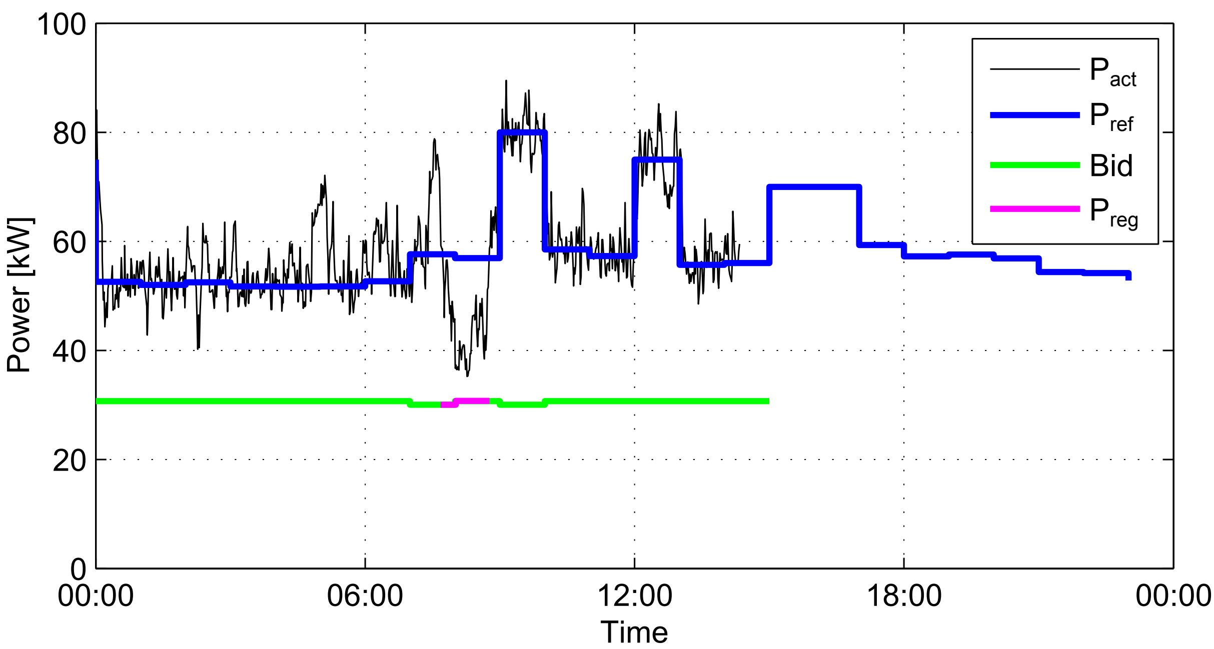

Figure 1.7: Sum of the measured power of the pool P_{act} with reference power P_{ref} and regulating-power bids. The activated regulating-power is marked with purple P_{reg}, Bid is the calculated possible power-reference for each hour.

From the graph it is obvious that the power-reference is not tracked precisely. It appears as disturbances in the measured power, while in fact it can be caused by many different factors. The primary reasons is that the heat pump can not be forced to start, and that there are certain constraints for how long the heat pump should be set free, and how long it can be shut off.

[3]When there is too much power in the grid, the price will go down. This is called down-regulation. Likewise if there is an acute need for more power, it is coined up-regulation.

Dispatch of control signals

When the heat pumps are under control, the relays are turned on and off depending on the power-reference. First the current power consumption is found from the sum of all the heat pump measurements. This is compared to the power-reference for the last cycle, which gives a power-error. Second, the power that the heat pump will use is estimated, since it is no guarantee the power consumption will be the same, quite contrary. It is estimated from the temperature inside the house compared to the desired comfort temperature $T_{i,set}$. This set point is found from a fit of the inside temperature for a representative month in the heating season. It is reasonable to guess that the set point is the mean of the inside temperature for a representative month, if the heat pump is working properly.

Third, the minimum power is taken as the estimated power of all the "control-locked" heat pumps. When a heat pump has been released by the controller, it should remain released for some time. Likewise the relay should remain pulled for some time after the heat pump has been shut off. These are the locked heat pumps, which gives the minimum estimated power.

Fourth, the desired amount of power is calculated from the power-reference minus the minimum power, considering the last error for correction. The last error is mainly used to make small adjustments to the estimated power. It is mainly the estimation function which provides the basis for determining which heat pumps should be released and which to shut off. The feedback does not play an enormous role, since the power-consumption is not directly related to the previous power.

Fifth, free heat pumps can be turned on or off depending on how far the minimum power is from the reference. The houses that are the coolest inside are prioritised so their heat pumps are turned on first. This is how the controller dispatches the power which is missing from the VPP.

1.2 System description

The pool of heat pumps is a part of the Control Your Heat Pump (SDVP)[4] platform. From 2010-2012 the first version of the SDVP project was running, which was focused on establishing a test-platform for heat pumps, where heating systems in about 300 houses can be monitored and controlled from local weather and electricity price forecasts [5].

The 300 houses had each a heat pump installed and used it for heating and hot water. As part of the SDVP project, they were upgraded with measurement equipment and a relay that can be remotely controlled. The remote controlled relay means that the heat pump can be forced to turn off, while it can only be allowed to run. When the heat pump is allowed to turn on it will do so only if the local controller inside the heat pump determines that there is a need for heating or hot water.

This upgrade makes the heat pumps so-called Smart Grid Ready (SGR), which covers the information from the measurements with the ability to control them. How this Smart Grid readiness is best used has been the focus of the READY project, and is what this bachelor project builds and improves on.

[4] In Danish: *Styr Din Varmepumpe* (SDVP).

1.2.1 The individual heat pump

Each house has installed a control box made by the company Liab[5]. The Liab box is the control unit that is accessible through SDVP. Each box collects measurements from the heat pump and the rest of the house and sends these each five minutes to the Liab-server, where it can be fetched by anyone who knows the codes. The communication also runs the other way each five minutes, to tell the Liab box to turn the heat pump on or off. This means that the time from receiving measurement data to sending a new command is at least five minutes. In case the connection can not be established at one update-time, the communication will be delayed to the next interval; ten minutes in all. Therefore the delay might be higher than the sample-time of five minutes.

In figure 1.8 the communication flows from and to the individual house is shown. A Liab box communicates with the Liab server, where it retrieves a set of scheduled commands, while it sends measurement data from the house to the server every five minutes. The measurements on the server can be retrieved by anyone with the codes. They are measured in the house and heat pump and collected by a Liab box that sends them to the server every five minutes.

The commands can be set to be executed at a certain time and the box can store several future commands in its schedule. An on-signal to the relay on the heat pump will mean the relay switches on, and the heat pump no longer gets power. An off-signal to the relay likewise means the relay switches off and the heat pump now gets power. This does not necessarily mean that it will turn on and use any power, but it is free to turn on. We say the heat pump is released.

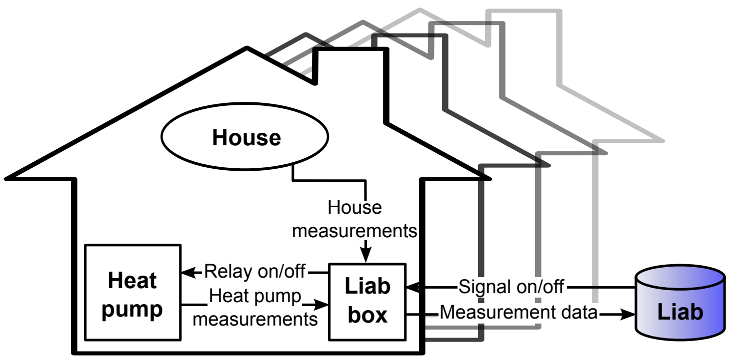

Figure 1.8: Drawing of communication flows between the individual houses and the Liab-server.

1.2.2 Measurements

Different measurements are coming from the heat pump and the house together with a time-stamp for each. These are written in table 1.1. Where the measurements are taken from the heating system is shown in figure 1.9.

There are seven temperature measurements from each house and heat pump. Not all of them are used in every heating system to control the heat pump. The inside and outside temperatures are measured by thermometers in the room and outside. The temperature of the hot-water is measured at the out-take, while the cold-water is measured at the intake to the tank. Likewise the temperatures of the heating is taken before the heat is sent to the radiators

[5] Linux In a Box (Liab). The box installed is based on the operating system Linux and is able to not only measure and switch a relay, but to do calculations too.

or similar, and the return temperature is taken before it runs through the heat pump again. The tank temperature T_{tank} is measured inside the tank.

There are two flow measurements. One that measures the flow of hot-water for taps and showers, and another that measures the flow through the heating of the house. For a house with radiant heating, $\dot{V}_h$ is the flow through the tubes under the floor. It is measured in the same position as the forward temperature, while the hot-water flow is measured at the cold-water intake to the tank.

The electricity consumption P_{hp} is the total electricity that the heat pump uses. This is measured by the electricity meter installed near the Liab box.

When controlling a heat pump the on/off signals to it are also saved with a time-stamp, from which it can be seen when the heat pump has been off and when it has been released.

Temperature	inside house	T_i
	outside	T_o
	forward	T_{fwd}
	return	T_{ret}
	cold water in	T_c
	hot water out	T_h
	hot-water in tank	T_{tank}
Flow	water for heating	$\dot{V}_h$
	hot water from tank	$\dot{V}_w$
Power	heat pump consumption	P_{hp}

Table 1.1: Measurement data sent from the individual house to the server.

1.2.3 Typical heating system

A typical heating system is shown in figure 1.10, where the heat pump cycle is in the middle, the heat source is to the left and the house is on the right. The heat is extracted from the heat source, and put into the house as heat for hot-water or for heating the house. Driving this heat transfer is the heat pump cycle, which uses electricity to run.

The heat pump cycle consists of a compressor pump, an expansion valve, a condenser and an evaporator. The cycle consists of four steps that the refrigerant goes through:

1. The evaporated refrigerant runs through a compressor pump, where the pressure is increased.
2. The gas provides its heat to the house through the condenser, thereby it shifts phase to liquid.
3. In the expansion valve the pressure is reduced, so the refrigerant becomes a mixed vapour.
4. When the fluid is cold it evaporates in the evaporator, where heat is extracted from the heat source. This could be the ground, the air or a lake where tubes are running through.

The heat is extracted in the evaporator from some heat source, e.g. the ground. There are some tubes dug down in the ground, most commonly with a mix of water and glycol running

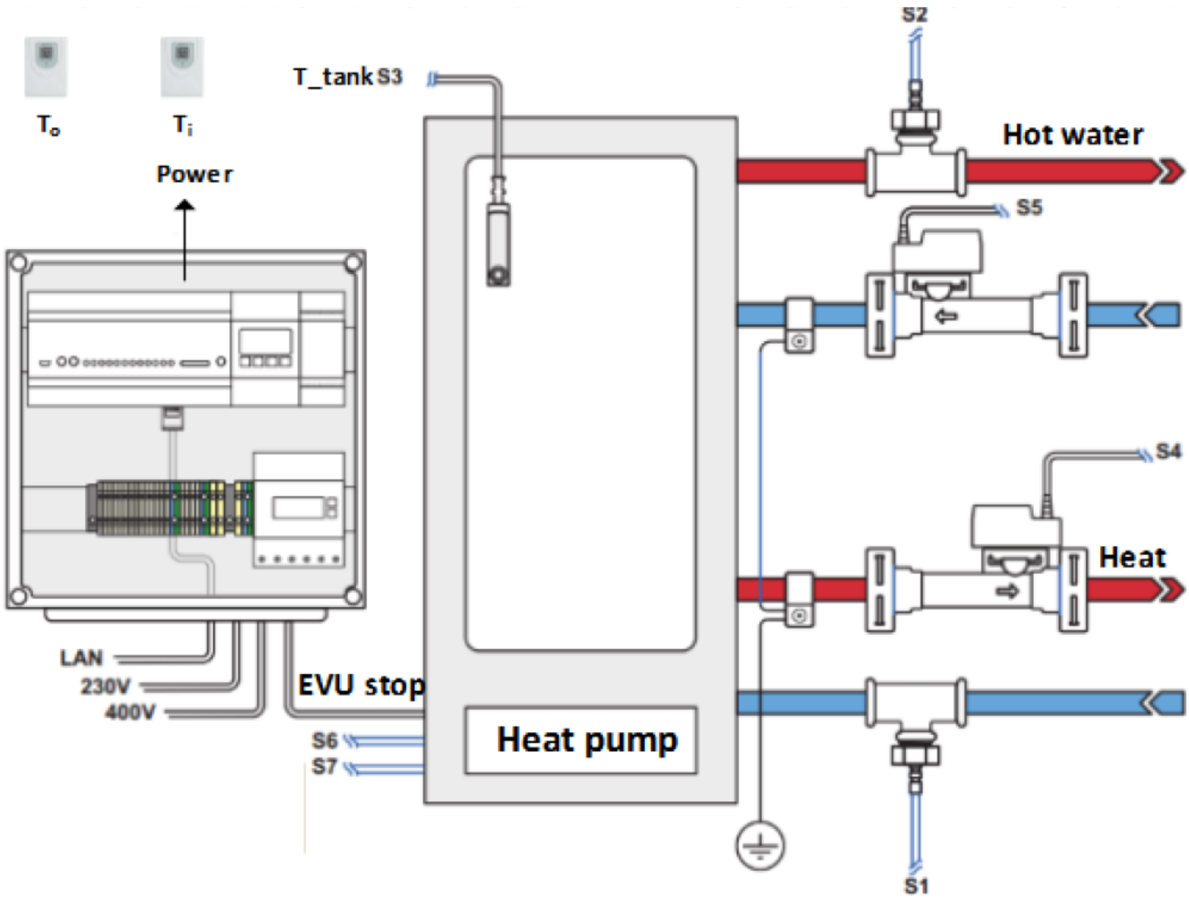

Figure 1.9: Illustration of where the measurements are taken from a heating system. In the middle is shown the heat pump with its hot-water tank. To the right are drawn the water-flows into and out of the heat pump. To the left is shown the Liab box and the electricity meter, where the heat pump is connected to. The Liab box is connected to the internet, so the data can be drawn out of it remotely.

through them. Therefore there is a circulation pump to make sure heat is available for the evaporator. The opposite happens at the condenser, where a circulation pump is installed to make sure the heat is being sucked out of the refrigerant, and into heating of the house or the hot-water. The sum of these two circulation pumps together with the compressor pump, is the measured electricity consumption P_{hp}.

Inside the house the heat pump determines whether the heating should go to the hot-water or to the house. This is actuated by the three-way valve. There might also in a few cases be some heating coming from an immersion heater. This is another electricity consumer, and in the few cases where it is used, P_{hp} will naturally go up quite drastically. The situations where an immersion heater might be turned on are considered in section 2.4.

Most of the heat pumps under control in the pool are on/off ones. This means the compressor pump, which is the main electricity consumer, is either running full speed or not at all. The circulation pumps will be speed-variable, controlled by the temperature of the fluid they are moving.

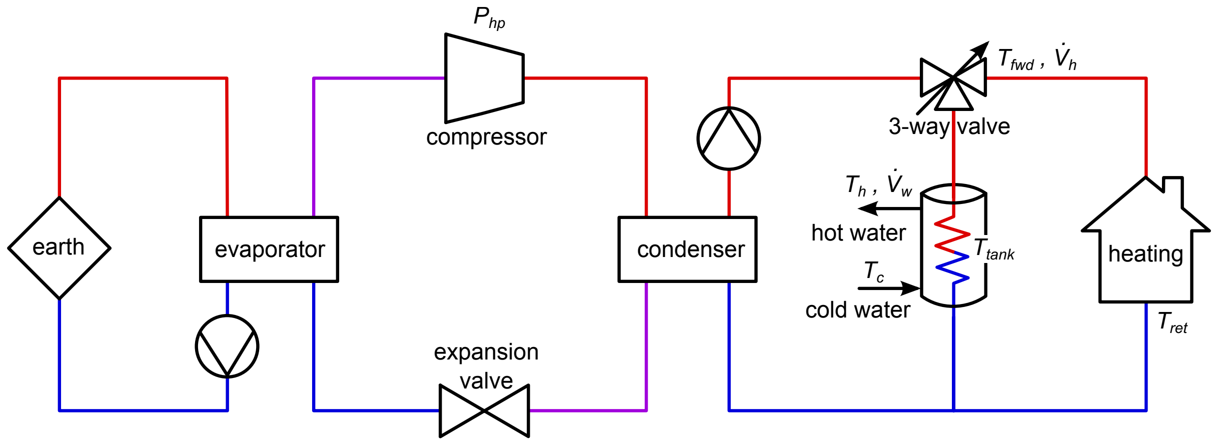

Figure 1.10: Diagram of the individual heat pump installed in each house with measurements placed. The heat pump cycle with its four components are drawn in the middle. The extracted heat is here illustrated coming from the ground. It could also be from the air or another source. The heat pump provides either heating for the house or hot water. Remind that the electricity consumed by the heat pump P_{hp} is the sum of the power for the compressor pump and the two circulation pumps.

1.3 Problem statement

Only if the power-reference can be tracked with a reasonable precision, is it possible to shift the power consumption in time, which is the objective for the pool of heat pumps. It is not the goal to use less electricity; the heat demand is the same, no matter if it is under control or not. The ultimate goal is to be able to shift the consumption in time, which opens up other opportunities. This is what is meant with intelligence in the power-grid.

The power could be used when electricity is cheap and saved when the price goes up, or the consumption could be used to balance the grid when there is an unforeseen unbalance within the day. Therefore, tracking the power-reference fast and precisely is the objective of a VPP.

As it was seen in figure 1.7, the actual power does not track the reference completely. Especially when there are shifts in the power-reference, the accumulated power vary substantially at some points. This could be explained by different effects. The important thing however, is that it could be prevented if the operation of each heat pump were predicted.

1.3.1 Electricity prediction

Since the pool is populated by on/off heat pumps, the electricity of each heat pump will be seesawing most of the time of operation. What happens is the power spikes up when a heat pump turns on, and likewise goes drastically down when it is turned off. Still the on/off operation might be used positively for aggregated control.

If the heat pump does not start when it is set free, it does not use the amount of power that was needed, and another heat pump must be set free. Yet, the first heat pump might start a few minutes later, which might give another problem.

If the heat pump starts that is good, it is what was wanted. Still it might turn off again after a few minutes, and then a new heat pump should be set free. This is not a problem in itself, but if the future consumed power was known before each heat pump was turned on, the process would be much faster and more precise.

One means of improving on the VPP controller of the pool of heat pumps, is to predict the electricity consumption of each individual heat pump.

1.3.2 Objective of the project

The estimator shortly described in section 1.1.3 is a simple function of the temperature inside the house. It does not depend on any other measurements or parameters. This power-estimation could be further developed by taking more measurements, as well as knowledge about the different heating systems into account.

This project seeks to develop a model that takes the state and operation of each heat pump into account, in order to ultimately predict the electricity consumption. The parameters for this model is found by utilising the measurement data available from SDVP.

The performance of the VPP is at the end of the day, evaluated on how much power is used hourly. Thus it stands to reason that the model should predict the power consumption for at least one hour ahead. Still, the heat pump is "control-locked" for 30 minutes, which makes it preferable that the prediction be at least 90 minutes ahead.

Thus the goal of the report is to formulate a model that predicts the electricity consumption of an on/off heat pump 90 minutes ahead, and to calculate the required parameters.

2. Model description

The heat pump model developed during this project will be presented in this chapter for each of the three points shortly described below. Most heat pumps have an immersion heater that is used at certain times. When and how this works is described in section 2.4 of the chapter.

The general model for an individual heat pump can be described as shown in figure 2.1. First the set point value is found for the forward temperature $T_{fwd,set}$, from the temperature settings in the heat pump. This is usually adjusted by a professional, who installs and inputs the settings based on the rest of the heating system (e.g. radiators or radiant heating).

Secondly $T_{fwd,set}$ is compared to the current forward temperature T_{fwd}, to determine whether the heat pump should start, stop or just continue what it is doing (running or not). This is determined from the trigger-limits, the forward temperature error e_{fwd} and the time. Time is important because it is used to avoid "silly-operation" where the heat pump is turned on and off many times. If it is turned on and off too often the heat pump is not really providing much heating, but uses a lot of electrical power, since the heat pump consumes an amount of power for each time it is turned on, before it starts drawing heat from the surroundings. Therefore the heat pump is limited to running a maximum number of times during an hour.

Thirdly the heat pump is either started, stopped or simply continues operation depending on the control signal u. From the operation the consumed power P_{hp} has some dynamic behaviour, as well as the temperature T_{fwd} used for determining the operation.

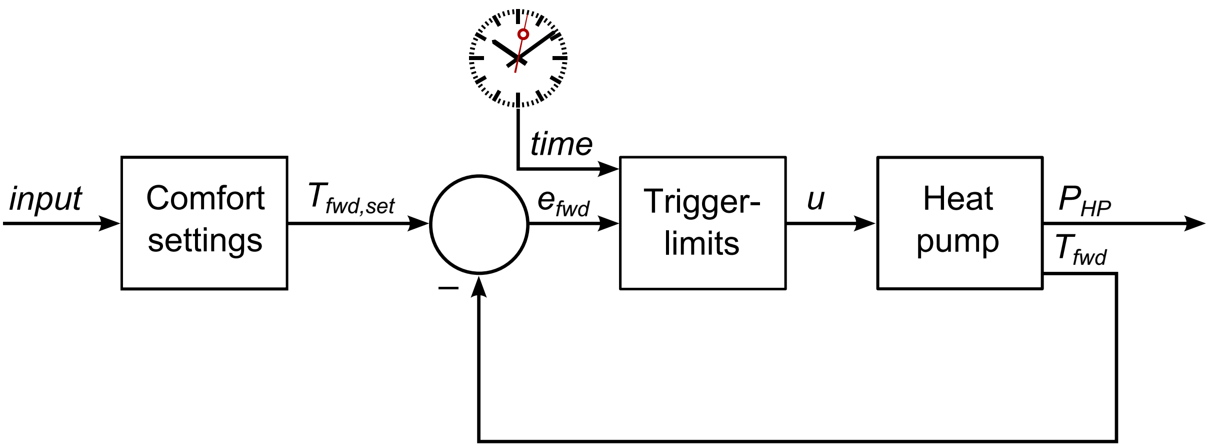

Figure 2.1: General heat pump model.

2.1 Forward temperature set point

The set point for the forward temperature $T_{fwd,set}$ can either be set manually to a fixed value, or it can be determined from some function of the outside temperature T_o.

The manual setting is simply a fixed set point throughout the year. The changing seasons and change in surroundings is not taken into any account, whereas when the set point is determined from the outside temperature there is some consideration about the environment. For the same reason this is the most used configuration of household heat pumps.

2.1.1 Determined from outside temperature

In heat pumps where $T_{fwd,set}$ is determined from T_o, the function can be described as a linear expression with a minimum and a maximum set point, $\check{T}_{fwd,set}$ and $\hat{T}_{fwd,set}$ respectively.

$$T_{fwd,set} = a\,T_o + b \quad , \quad \check{T}_{fwd,set} \leq T_{fwd,set} \leq \hat{T}_{fwd,set} \tag{2.1}$$

This is shown in figure 2.2, which shows an example of how the function could look like.

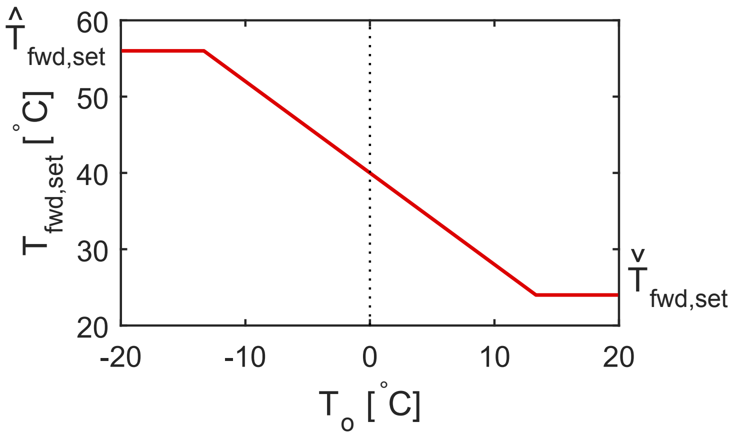

Figure 2.2: $T_{fwd,set}$ determined from outside temperature T_o.

2.2 Trigger-limits

If the forward temperature is too low, the heat pump should start running. If the forward temperature is too high, the heat pump should stop again. This can be reached in two different ways. Either simply by temperature-triggers or by an integral-control, where the error e_{fwd} is accumulated over time. In both cases the trigger-time where the heat pump has been turned on is important, and in both cases the window between the upper and the lower trigger-limit represents the operating area of the heat pump.

2.2.1 Time-triggers

A heat pump should not be turned on and off too often because it deteriorates the COP. A rule of thumb is that it should not have more than four starts during an hour, thus a heat pump should be locked from the time when it is turned on and 15 minutes forward. Therefore there is a timer in the heat pump that starts counting down each time it is turned on. When the heat pump is time-locked it does not matter if triggers for T_{fwd} want it to start. Yet if there is a need for hot water, it will turn on, time-locked or not.

Having the heat pump off for too long is not desirable either. Many houses have radiant heating, and the floor might seem a little cold if the heat pump is off for too long. Therefore there is a maximum allowed off-time for the heat pump before it will start again. This is usually three hours.

For the model, the last start and stop times will be saved. From this the time since the last start and the last stop can be calculated, which will determine if the heat pump is time-locked or will turn on because it has been off for three hours.

The trigger-times related to the current time is shown in figure 2.3. The time from last start to now is Δt_{on} and the time from last stop until now is Δt_{off}.

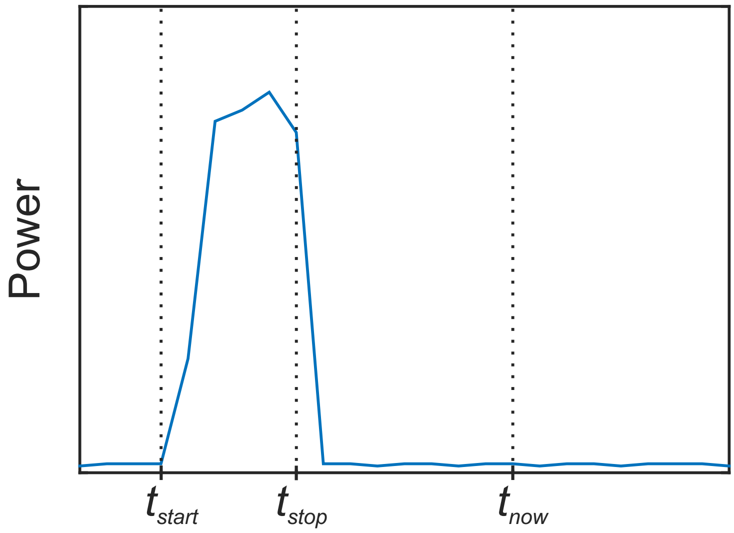

Figure 2.3: Trigger-times. Off-time is the time from t_{stop} to t_{now} and time since last turn on is from t_{start} to t_{now}.

$$\Delta t_{on} = t_{now} - t_{start}$$
$$\Delta t_{off} = t_{now} - t_{stop}$$

As mentioned there should at least be 15 minutes from one start to the next, unless there is a need for hot-water. Therefore the heat pump will only start because of heating for the house if $\Delta t_{on} \geq \Delta \check{t}_{on} = 15$ min. In case there is no need for heating for a period of time, the heat pump will turn on to keep the floor hot if $\Delta t_{off} \geq \Delta \hat{t}_{off}$.

2.2.2 Hot-water triggers

A comfortable inside temperature is not the only requirement that can make the heat pump start. Most of the heat pumps in the VPP also provides heating for the hot-water for the household. The comfort requirement here is that there should always be hot water available, and the temperature should not be below some limit $\check{T}_{tank}$. By integrating the hot-water flow $\dot{V}_w$ the use of hot water can be determined, which should not go over some limit $\hat{V}_w$ since the last heat pump stop.

So there are two hot-water triggers that will make the heat pump run:

$$\text{if } T_{tank} \leq \check{T}_{tank}$$
$$\text{or } \int_{t_{stop}}^{t_{now}} \dot{V}_w \, \mathrm{d}t \geq \hat{V}_w$$

2.2.3 Forward temperature triggers

The trigger for heating the house can either be determined directly from the temperature difference, or by the integral of this error. For both temperature-controlled and integral-controlled heat pumps, there are certain upper and lower trigger-limits.

Temperature triggers

When the heat pump is controlled by temperature-triggers, there is a fixed window, turning on and off the heat pump depending on e_{fwd}. It could look like the window shown in figure 2.4.

When the error is positive it means that the temperature is too low, and the opposite for a negative error. Therefore when e_{fwd} goes above the upper limit e_U the heat pump will start. Likewise the heat pump will stop when e_{fwd} goes below the lower limit e_L. This way the temperature-triggers work as hard switches to determine the operation of the heat pump. As mentioned before the last start-time also plays a role for the operation.

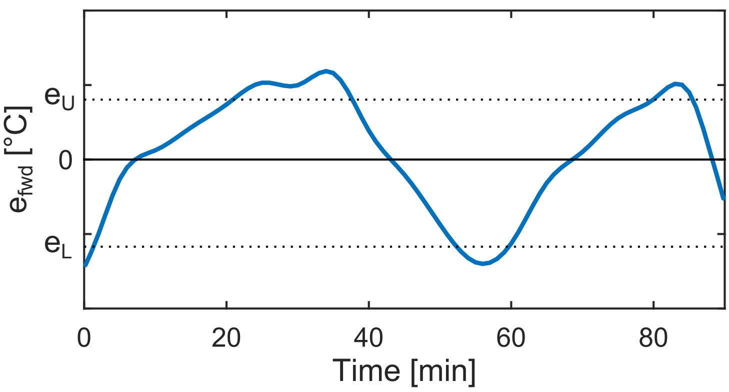

Figure 2.4: Temperature-triggers. If the error goes above e_U the heat pump will start. Likewise it will stop if the error goes below e_L.

Integral triggers

Some heat pumps are controlled from the integrated error I. Naturally, there has to be a high and a low limit to I, so the signal does not get out of proportions.

$$I = \int T_{fwd,set} - T_{fwd} \, \mathrm{d}t \tag{2.2}$$

Depending on the integral-error and Δt_{on} the heat pump will either start, stop or continue operation. Using the accumulated error makes the control of the heat pump slower, which mimics the thermal inertia of the house.

Figure 2.5 shows a diagram of the trigger-limits for a heat pump controlled by integral-triggers. The error signal e_{fwd} is here added to the accumulated error every minute, and saturates at certain high and lower limits producing the integral-signal I. From this signal the heat pump is controlled as it would be with temperature-triggers. There is an upper trigger that starts the heat pump, if it is not time-locked. And likewise there is a lower trigger that stops the heat pump as soon as I is low enough. In between, the heat pump operates as the last command u.

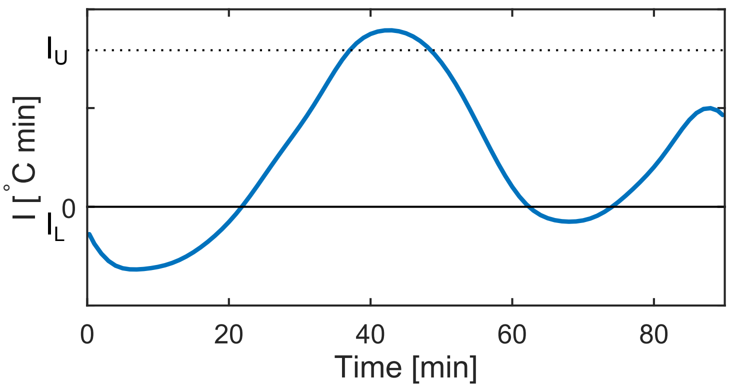

Figure 2.5: Integral-triggers. I is falling when e_{fwd} is negative, and the opposite when it is positive. When I is above I_U the heat pump will start and it will stop when I is below I_L (which in this case is 0).

2.3 Heat pump dynamics

From the time the heat pump gets a start or a stop signal it takes some time before the state of it is fully developed, there are some dynamics. Especially important for prediction of the power, is the dynamics for the forward temperature T_{fwd} and naturally the electricity P_{hp}.

From the data available from SDVP and experiments conducted with a higher sampling frequency, it stands to reason that the electricity dynamics are negligible. It does not take time enough for the electricity to develop to consider it, compared to the thermal inertia of the system. To read more on the experiments done, refer to appendix A.

Assuming that the temperature from the heat source and the inside temperature of the house are constant, the dynamics of the forward temperature can be described by equation (2.3)[1]. T_∞ is the temperature of the surroundings, which T_{fwd} is approaching. t is the time from start or stop, while τ is the time-constant for the temperature development. $T_{fwd}(0)$ is the temperature from start or stop.

$$\begin{aligned}\frac{T_{fwd}(t) - T_\infty}{T_{fwd}(0) - T_\infty} &= e^{-t/\tau} \\ T_{fwd}(t) &= T_\infty + \left(T_{fwd}(0) - T_\infty\right) e^{-t/\tau}\end{aligned} \tag{2.3}$$

For a start to a stop T_∞ is the temperature from the heat source T_s. As it is shown from table 1.1, this temperature is unknown. t is in this case the time from the start to the next stop, and $T_{fwd}(0)$ is the temperature at the time the heat pump is started.

[1]This equation comes from an energy balance of the heat-carrying fluid, assuming it is a lumped system. The assumption is that the water for heating is the same temperature.

For a stop to a start T_∞ is the temperature of the inside T_i, where the heat is transferred to. Likewise t is the time from the stop, and $T_{fwd}(0)$ is the temperature at stop-time.

Since the two processes of starting and stopping the heat pump are quite different, there might also be quite different time-constants for the two. Therefore it stands to reason that there should be a time-constant for each of the two. When the heat pump is running it will follow equation (2.4), while it will follow equation (2.5) when the heat pump is not running.

$$T_{fwd}(t) = T_s + (T_{fwd}(0) - T_s)\, e^{-t/\tau_{on}} \tag{2.4}$$

$$T_{fwd}(t) = T_i + (T_{fwd}(0) - T_i)\, e^{-t/\tau_{off}} \tag{2.5}$$

2.4 Immersion heater

An immersion heater is a simple resistance heater, hence all the electricity used goes to heating, as opposed to the regular heat pump cycle, where the COP can go above 1. This means the electricity consumption goes up whenever the immersion heater is put into use.

Most heat pumps have an immersion heater installed to run when there is need for extra heating. It can run in situations where T_{fwd} goes too far below the set point, thus adding an extra "emergency" trigger-limit above the upper triggers e_U and I_U.

Naturally, the immersion heater can also be used to heat the hot-water for the household. This might be just on times where the heat pump cycle can not keep up, or it might be a permanent setting, where the heater makes sure T_{tank} is always over some set point. Some heating systems are set to keep T_{tank} higher than it is reasonable to expect the heat pump cycle can provide. For the tank the immersion heater can also be set to heat the water once in a while to a high temperature to kill bacteria. It could be set to heat the hot-water tank to 70°C for ½ hour. This is to pasteurise the hot-water tank and keep the Legionella population down [2]. This is not configured in all installations. Usually it is quite easy to see the pattern from measurement data, e.g. every 14 days the tank-temperature goes much higher than what is usual, while the power spikes up to more than the heat pump is dimensioned for.

2.5 Model formulation

Either the heat pump is directly controlled by the temperature-difference between the set point and the measured temperature e_{fwd}, or it is controlled by the integral of this temperature-difference $I = \int e_{fwd}\, \mathrm{d}t$.

In both cases the model checks for five different variables to determine whether the heat pump will turn on, and one variable to determine when it will turn off again.

For an integral-controlled heat pump these variables are:

if $(\Delta t_{on} \geq 15 \;\&\; I \geq I_U)$

or $\Delta t_{off} \geq 180$

or $T_{tank} \leq \check{T}_{tank}$

or $V_w \geq \hat{V}_w$

start heat pump

elseif $I \leq I_L$

stop heat pump

For both types it is necessary to consider the time-triggers. If the heat pump is time-locked it will not start even if there is a need for more house heating. Only if there is a need for hot-water will the time-lock be overruled. Likewise it is not of importance what type of heat pump is considered, when it is set to run every three hours, even if there is no measured need. This mainly makes sense for a house with radiant heating.

For both types there are hot-water triggers that will turn the heat pump on. The tank-temperature and the amount of water in the tank will always turn the tank on. These can be hard to predict, since the temperature and the volume in the tank quickly can drop e.g. when someone in the house takes a shower. This can be from one sample to the next, which makes it too hard to predict from the same method as the heating of the house. Predicting the hot-water need is more a question of knowledge about usual behaviour and perhaps a sort of pattern-recognition of the hot-water use, which is not a focus of this report. Therefore the hot-water triggers is not part of the prediction model developed during this project. It is still important to consider when designing a VPP controller, and is already incorporated in the controller developed for the READY project. For this report however, it is not a focus-point since the prediction is quite different from predicting the heating of the house.

An immersion heater can be present in both types of heat pumps, and will turn on for the same reasons.

2.5.1 Block diagram

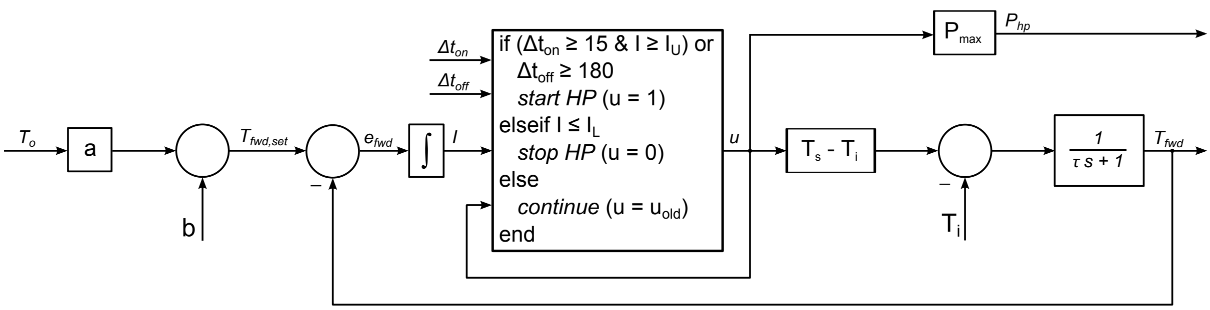

Figure 2.6: Diagram of model of integral-controlled heat pump.

2.6 Parameter-estimation

Now that the model is defined with variables, it is time to find a way to calculate these. They are calculated from the data available from SDVP, so

Parameters needed to be figured out:

- Time triggers: $\Delta\check{t}_{on}$ and $\Delta\hat{t}_{off}$.
- Forward temperature set point: a and b.
- Temperature triggers: e_U and e_L, from which $\check{T}_{fwd,set}$ and $\hat{T}_{fwd,set}$ are also estimated.
- Integral triggers: I_U and I_L.
- Forward temperature dynamics: T_s, τ_{on} and τ_{off}.

2.6.1 Time trigger-limits

The time trigger-limit $\Delta\check{t}_{on}$ is simply taken as 15 minutes.

The maximum allowed off-time should be found from the actual maximum off-time of the heat pump for a period of time, where there is not a big need for heating. Yet for the heat pumps under control in the pool, this value will not be meaningful, since the relay of the heat pump might have been pulled.

It must simply be assumed that at least for radiant heating systems $\Delta\hat{t}_{off}$ is three hours. This is the common value observed during summer.

2.6.2 Forward temperature set point

The constants a and b for $T_{fwd,set}$ is found from a linear fit of the T_{fwd} measurements for the past year. This is shown for a temperature-controlled heat pump in figure 2.7.

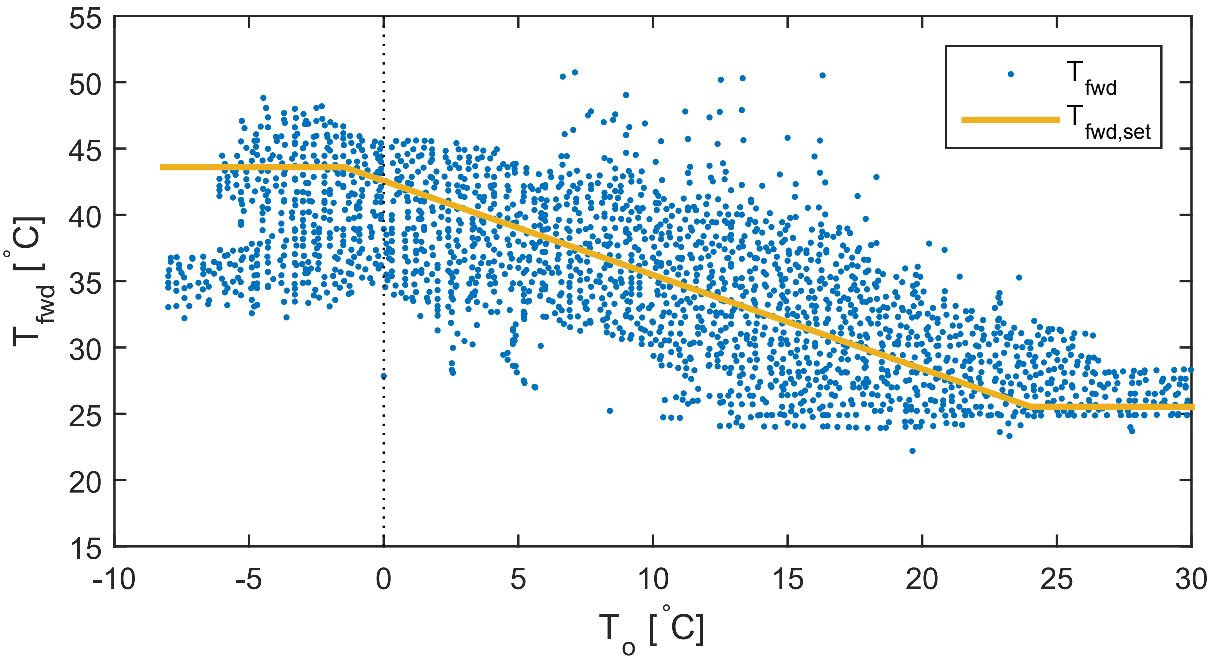

Figure 2.7: Forward temperature and set point plotted against T_o. The data is filtered because there are more than 10^5 data-points, which means the density close to the fit is not fully illustrated, yet the scattering is preserved. $a = -0.7$ and $b = 43$ for this graph, while $R^2 = 0.96$.

For a temperature-controlled heat pump this is a reasonable approach, since most of the measurements are within a ± 3°C of the linear fit. This gives an acceptable value for R^2 of the fit.

For an integral-controlled heat pump the plot looks a bit different. Around the fit the measurements are still more dense, but the scattering is much larger. This is because it is not controlled directly from the temperature difference, and therefore some time can pass before it will turn on or off. For this type of heat pump, the linear fit is not suitable for anything but a first guess.

The linear fit for an integral-controlled heat pump is shown in figure 2.8, which does not show a direct relation, though it is a bit exaggerated because the data has been filtered in order to fit the image in this report.

Instead of the linear-fit approach, $T_{fwd,set}$ for an integral-controlled heat pump is calculated by optimisation. This is described in appendix B.2.

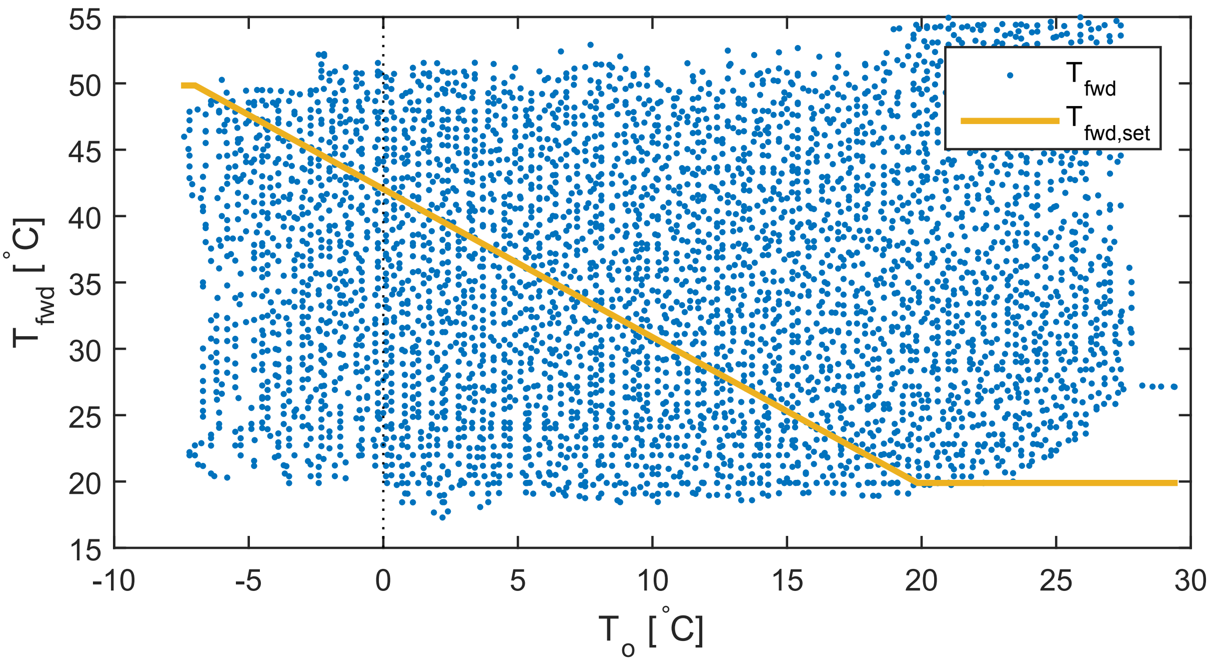

Figure 2.8: Forward temperature and set point plotted against T_o for an integral-controlled heat pump. Again the data has been filtered and mainly the scattering has been preserved. $a = -1.1$ and $b = 42$, while $R^2 = 0.71$ which also shows that this might not be a proper fit.

2.6.3 Temperature trigger-limits

The temperature trigger-limits are found from looking at T_{fwd} when the heat pump is started and stopped. By using the start and stop-times, the temperatures that will make the heat pump start and stop can be found. How these start and stop-times are found is described in appendix B.1.

The error in the temperature is found for starts and stops, and the mean of this error is taken as the trigger-limits e_U and e_L, respectively.

$$\begin{aligned} e_U &= \langle T_{fwd,set} - Tfwd(start) \rangle \\ e_L &= \langle T_{fwd,set} - Tfwd(stop) \rangle \end{aligned} \tag{2.6}$$

In figure 2.9 is shown the temperature set point with the upper and lower trigger-limits.

These are calculated to temperatures instead of error-triggers. It translates as written in equation (2.7).

As it is shown in the figure, the temperature window is about ± 2. This shows pretty well that this specific heat pump really is temperature-triggered, because the upper and lower trigger-limits are quite reasonable.

$$\begin{aligned} T_{fwd,U} &= T_{fwd,set} - e_L \\ T_{fwd,L} &= T_{fwd,set} - e_U \end{aligned} \tag{2.7}$$

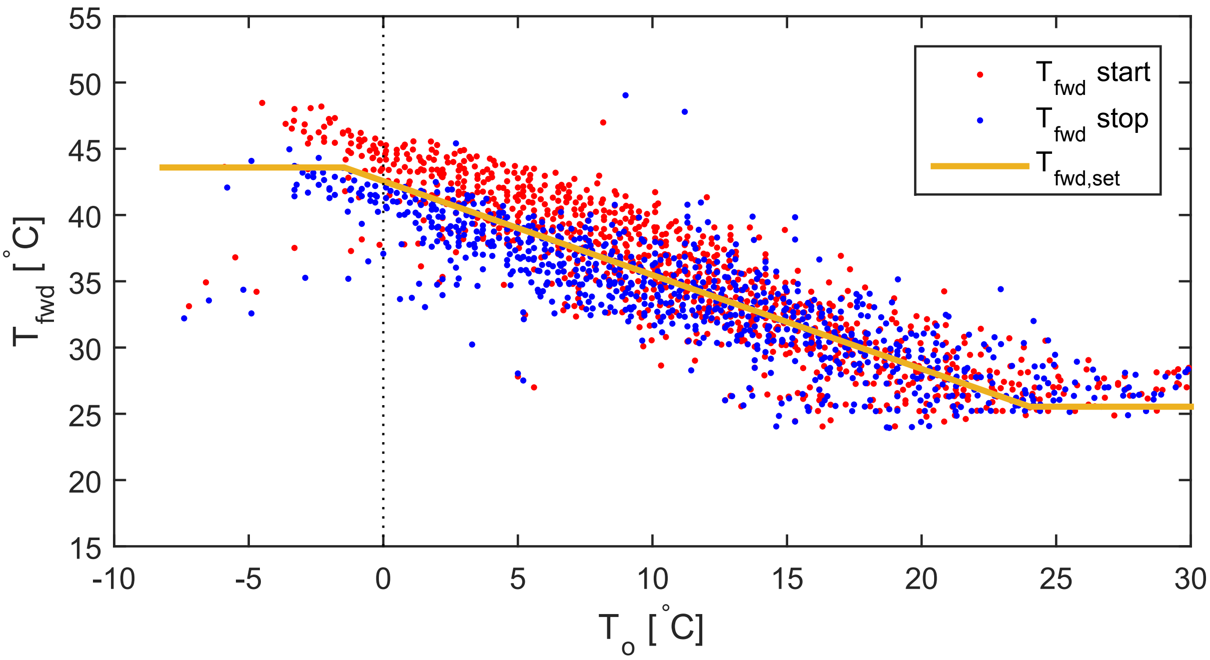

Figure 2.9: $T_{fwd,set}$ with upper and lower temperature trigger-limits. The data-points have been filtered, yet the scattering is preserved.

Set point maximum and minimum

When the trigger-limits have been found, the maximum and minimum of the set point is readily found. The lowest measured value of T_{fwd} must be the lowest temperature the heat pump will accept. By subtracting the lower trigger-limit from this value the minimum set point is found.

The same applies for the maximum set point, where the maximum measured temperature minus the upper limit is the value.

$$\begin{aligned} \check{T}_{fwd,set} &= \min(T_{fwd}) - e_L \\ \hat{T}_{fwd,set} &= \max(T_{fwd}) - e_U \end{aligned} \tag{2.8}$$

2.6.4 Integral trigger-limits

To determine the integral trigger-limits it has to be formulated to an optimisation problem. Though as pointed out in section 2.6.2 the calculated set point is probably not adequate for an integral-controlled heat pump. If the function for $T_{fwd,set}$ is unknown, it is not possible to find the trigger-limits, as is also shown in appendix B.2.

Instead the limits are guessed and can later be adjusted if it appears that the model is more accurate with other values. As a first guess is taken the default settings for a Danfoss heat pump: $I_U = 60\ °\mathrm{C} \cdot \mathrm{min}$ and $I_L = 0\ °\mathrm{C} \cdot \mathrm{min}$. [1]

2.6.5 Forward temperature dynamics

The dynamics of T_{fwd} could be found from measurements from the experiments described in appendix A. Yet it would be much better to find a method of estimating the dynamics of the actual heating systems in the pool under control. There might be differences in the inertia of the different heating systems, especially between houses with radiators and houses with radiant heating. The time-constant of a house with radiant heating should be higher, because there is more water running through the tubes, and possibly a higher resistance and capacity through the floor. Therefore it is important to find a way to estimate the three parameters of the T_{fwd} dynamics from the measurements available from SDVP.

To estimate the dynamics, it is suitable to look at a few starts and stops. Since the parameters might change substantially over a longer time-period, such as a few months, it will give a more accurate result to look at the latest measurements. Estimating parameters from starts and stops for the past 24 hours will be appropriate for the heating season.

The losses to the surroundings can be assumed constant, when looking at a short period of time. From this it is possible to estimate the three parameters from equations (2.4) and (2.5).

The time-constant for the stop to the next start τ_{off} can be found directly by isolating it from equation (2.5). To get τ_{off} over the whole period, the points from stop to start should be chosen.

$$
\begin{aligned}
\tau_{off} &= \frac{-t}{\ln\left(\frac{T_{fwd}(t)-T_i}{T_{fwd}(0)-T_i}\right)} \\
\tau_{off} &= \frac{-(t_{start}-t_{stop})}{\ln\left(\frac{T_{fwd,start}-\langle T_i\rangle}{T_{fwd,stop}-\langle T_i\rangle}\right)}
\end{aligned}
\tag{2.9}
$$

As mentioned in section 2.3 the time-constant for the start-dynamics might be quite different from τ_{off}. Therefore τ_{on} must also be found. For this case, the temperature of the heat source T_s is unknown, hence the same estimation-method can not be used.

Instead the issue is defined as an optimisation problem from equation (2.4), where the variables to find are the parameters T_s and τ_{on}. To read more on the formulation of the optimisation problem, refer to appendix B.

3. Model validation

In section 2.5 the model developed was described and illustrated in figure 2.6. In appendix C the Matlab code for the model is provided. To validate the suitability of the model developed, a certain performance indicator must be presented.

Also, some of the parameters that are calculated are not hard-written values, but estimates for the heat pump. Therefore the real parameters might be different from what they are estimated to. This may be a minor or a major issue. How big importance the parameter-estimation has on the performance of the model is an important factor for the robustness of the model. Therefore a sensitivity analysis is provided, taking some of the parameters and figuring out how sensitive the model is to changes in these.

3.1 Method of comparison

Since the main goal is to predict the power of the heat pump, it makes sense to validate the model from how well the simulated power P_{sim} corresponds to the measured power P_{hp}. A common way of calculating the fitness of the model, is by taking the RMS error of a simulation. The RMS error of the simulated power is calculated as written in equation (3.1).

$$P_{RMSE} = \sqrt{\frac{\sum\limits_{t=1}^{N} \left(P_{sim}(t) - P_{hp}(t)\right)^2}{N}} \tag{3.1}$$

This equation gives a number of how far the simulation typically is from the measurement. Though it does not make intuitively sense, unless the variation of the power is known. Therefore P_{RMSE} is divided by the difference between the maximum and minimum value of the measured power. This is called the normalised RMS error, given in equation (3.2). If P_{NRMSE} is zero, the simulation fits perfectly. If it is one, it will practically mean that the model is running absolute opposite of the measurement.

$$P_{NRMSE} = \frac{P_{RMSE}}{\max P_{hp} - \min P_{hp}} \tag{3.2}$$

P_{NRMSE} is a number that can be used to compare the different simulations to one another. Yet the number itself does not necessarily provide all there is to know about the fitness to reality. One simulation might be better than the other, even though they have the same NRMSE value. Plots of the simulated power and the measured power however, provides a more holistic illustration of the performance, yet without hard numbers. As with all statistics, the judgement should not be made from one parameter alone.

3.2 Simulation

Looking at an arbitrary time-period with measurement data, it is necessary to choose the right points to simulate from. When the simulation begins it should be initialised to some defined state. Most importantly is the integral-signal I which will determine if the heat pump is turned on or off. In between the starts and stops it can be hard to tell what the value of I is, since it only shows at these points.

Therefore the simulation should begin at starts and stops of the heat pump, which is derived from the power measurement, as described in appendix B.1. To be sure to simulate over more than one on/off cycle, a time-period of four hours is chosen. The longer the time-period is, the bigger should the error become, since no correction is implemented in the simulation. Yet as will be shown shortly, this time-period of three times the goal of the prediction does still give reasonable results.

The heat pump starts when the integral-error is equal to the upper trigger-limit, therefore it should be initialised as such $I(0) = I_U$. Likewise the heat pump stops when the integral-error is equal to the lower trigger-limit, and it should be initialised to $I(0) = I_L$. I is not adjusted at times when starts or stops are measured, though this would be a good idea when implementing the model in a controller.

The simulated temperature $T_{fwd,sim}$ will start at the measured value, and will either fall or rise as given by equation (2.3). Whenever a start is simulated the temperature will go up over time after Δt_{on} has been reset. Whenever there is a simulated stop the temperature goes down again, as a function of Δt_{off} after it has been reset. How this is written in Matlab is shown in appendix C.

3.3 Performance of model

In figure 3.1 is plotted the temperature and power measurement from a point where the heat pump starts. With the measurements are also plotted the simulated temperature and power with dashed lines. The NRMSE of the power is 0.28, which means the simulated power misses the measurement by an average or 28 %. Though from the plot it is seen that the start and stop instances are hit quite closely to P_{hp}, at least for the second and third on-cycle of the simulation. This figure is for a start of the heat pump where $I(0) = I_U = 60\ ^\circ\mathrm{C} \cdot \mathrm{min}$.

If the start-time together with the amount of time the heat pump will be on, is predicted this well for all the heat pumps under control in the VPP, there is a good chance it will make the aggregated control better.

In figure 3.2 a simulation run from when the heat pump stops is shown, where $I(0) = I_L = 0$. This simulation has a lower NRMSE, which means it fits better. Looking at the plot of the simulation it is also possible to see that the simulated power fits the measurement quite well. The simulated start-time and the on-time is close to the measurement.

3.4 Sensitivity to parameter-change

Since the parameter-estimations are not perfect, it is good to know which of the parameters the model is most sensitive to. This can give a hint to which of the parameters are most important spend some more effort on estimating.

$T_{fwd,set}$ is estimated from one start-time to the next for an integral-controlled heat pump.

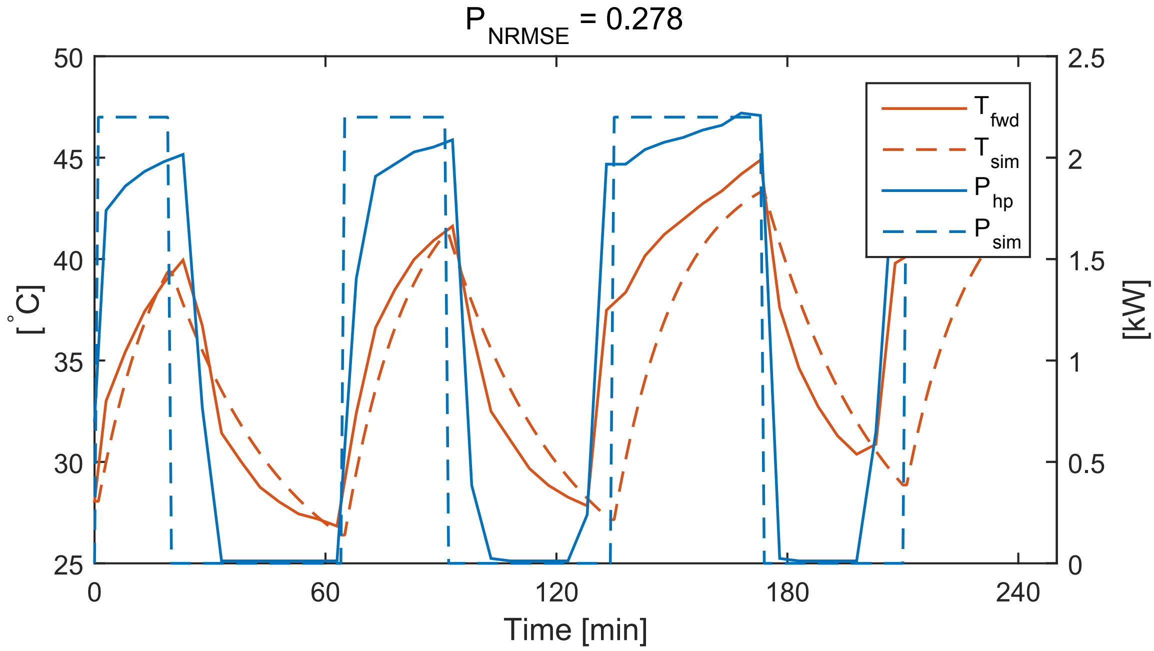

Figure 3.1: Simulation run from a start of the heat pump. T_{sim} is the simulated forward temperature, P_{sim} is the simulated power and the NRMSE is 0.28 for the four hours it is simulated over.

This might be a rough estimate since I probably is not reset for every start-times of the heat pump.

More interestingly however, is whether the values for the integral trigger-limits are very important for the model, since they are guessed from default settings for a heat pump. They should probably be adjusted for each heat pump, but in which manner will be covered shortly.

3.4.1 Forward temperature set point

An assumption for the simulations are that the outside temperature can be predicted quite well. $T_{fwd,set}$ depends on T_o and the set point varies during a day because of it. Therefore the simulation would be much worse at time-periods where the temperature outside changes. For the VPP controller considered throughout the report, the outside temperature is already known from estimates from a weather-service. It needs to be predicted no more than the same amount that the simulation is running.

In this sense it is already known that the model has some sensitivity towards the set point, and this has been adjusted for. How much the model will deteriorate from parameter-change of $T_{fwd,set}$ is covered here. As it is covered in section 2.6.2 the parameters a and b are subject to uncertainty, especially for an integral-controlled heat pump.

In figure 3.3 is shown a simulation where a is 5 % higher than the estimated value from the optimisation. It is shown for the simulation run from a heat pump start, since this was the one that was affected by the change.

From the plot it is seen that the on-time is a little bit shorter, and the starts are delayed by a few minutes. The reason why this simulation is affected by it, is that the outside temperature changes from 15–10 °C during the same time-period. The a-value of the set point has the biggest effect, when T_o is far from zero. Therefore the sensitivity to changes in a is higher in

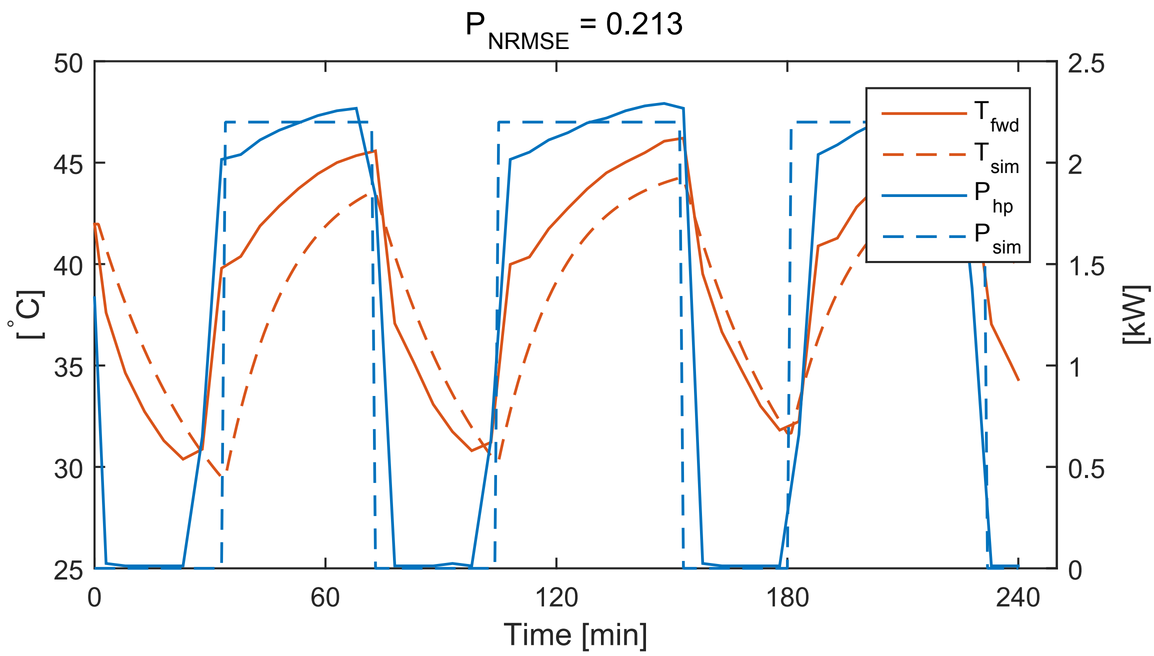

Figure 3.2: Simulation run from a stop of the heat pump. The NRMSE of the power is 0.21 for the four hours the simulation ran.

the summer, and not very high in the winter. This also makes it better to estimate the parameter from data from hotter days.

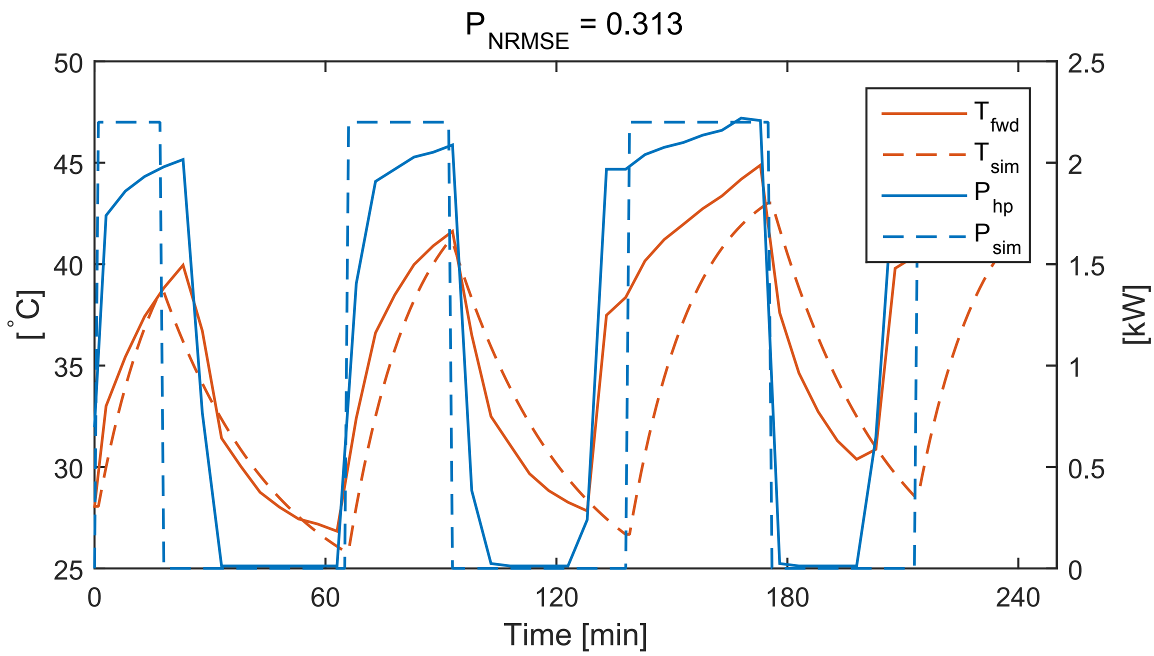

Figure 3.3: Simulation run from a start with a being 5 % higher than the estimated value from the optimisation.

In figure 3.4 is shown a simulation when b has been multiplied by 1.05, making it 49 instead of 47. This change affects all simulation runs, and quite heavily so. The on-time is increased by 10–15 minutes for each on-cycle, and as seen in the plot this disturbs the rest of the prediction.

This parameter does not depend on the outside temperature and therefore there is no requirements to the data-set used when calculating it. a might change from one value to the other, if T_o in the data-set is close to zero, but b does not have this sensitivity towards the data chosen when calculating the parameter.

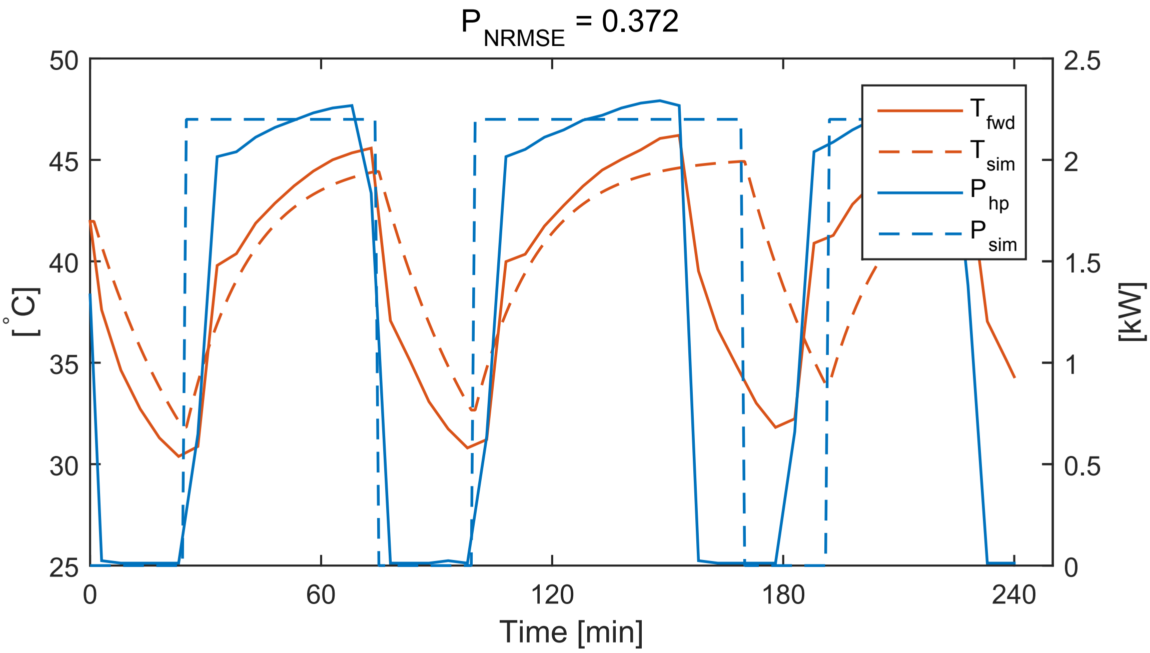

Figure 3.4: Simulation run from a stop where b is adjusted to 5 % higher than the estimated value.

3.4.2 Integral trigger-limits

In figure 3.5 is shown how the simulation will look if the upper trigger-limit is adjusted to 70 instead of 60. It appears that the simulated power is shifted in time, so that it is started later, and stopped later. The actual on-time is close to unchanged, yet it is running for a bit longer. The main change of operation is however the time-shift to a larger delay compared to measurements.

Changing the lower trigger-limit to a similar lower value $I_L = -10$ instead of 0, yields the same result. It would not make sense to include the plot of this, since they are so much alike. It appears that the main concern for the operation is the window between the trigger-limits, though this might not be the case if the limits are changed too heavily.

In figure 3.6 is shown a simulation run with a $10\,^{\circ}\mathrm{C} \cdot \mathrm{min}$ lower I_U. Here the simulated power is precipitated compared to the original value. The on-time is shorter and the simulation starts before the measurements.

Again it gives the same result making I_L 10 higher. It appears that the method for finding $T_{fwd,set}$ and guessing the integral trigger-limits works. Other heat pumps might have different trigger-limits, but they can be adjusted simply with one of the limits. This makes it considerably easier to make changes, when it does not matter which of the two is tuned.

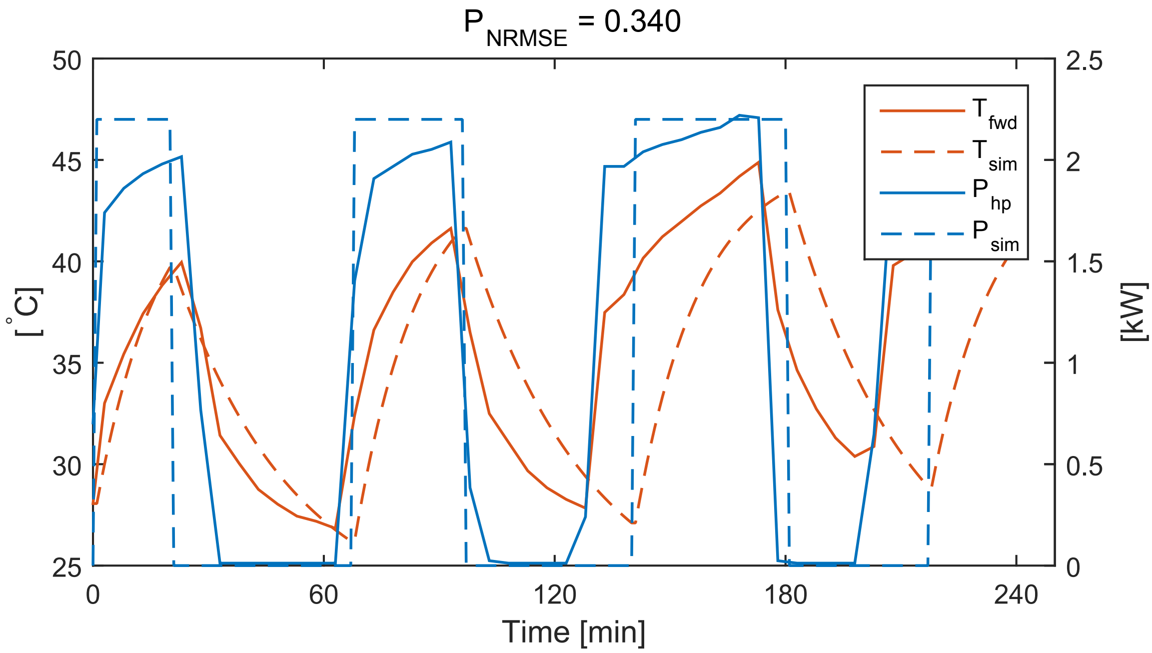

Figure 3.5: Simulation with $I_U = 70$ instead of 60. Obviously the simulated power is shifted in time, with an increasing delay.

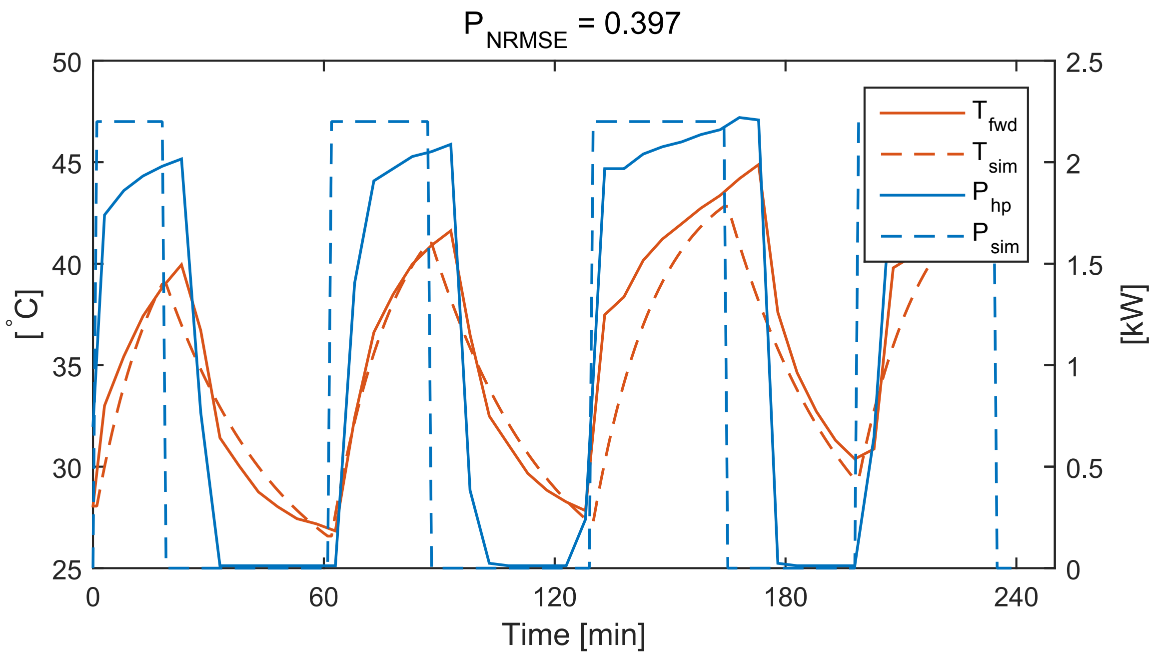

Figure 3.6: Simulation run with $I_U = 50$ instead of 60. Here the simulated power is shifted into shorter intervals.

4. Conclusion

The project was set out to find a way to predict the electricity consumption of an on/off heat pump before it is released, and a model that predicts the power four hours ahead is formulated during the report. With this the electricity can be predicted for use in a VPP, while the state of the individual heat pump is also estimated in the prediction model. By knowing this the overall performance of the VPP considered throughout the project can be improved to track the power-reference faster and more precisely.

The model incorporates data from the SDVP platform in order to estimate the necessary parameters for the model. Methods for calculating these parameters have been demonstrated and examples of how the code is written in Matlab is given, thus forming the basis for integrating the model in the VPP. It is meant to combine the findings presented in the report into a function-call that can execute as a part of the VPP.

Methods of evaluating the fitness of the model are given in chapter 3, where the NRMSE of the power is shown to be a way of evaluating the model. Especially when considering the sensitivity of the model towards the parameters, the NRMSE gives a quick pointer on how much the model changes from one value of the parameter to the next. Though this approach can not stand alone, but is combined with figures of plots for the simulations. There may be a better way of evaluating the simulations, comparing mainly the start-time and amount of running time.

To further develop the methods of this project, in particular the individual heat pump model should be scaled to a larger scale, where all heat pumps in the pool are to be simulated. The report sitting here is a product of another approach than making an aggregated model that is deployed on all the individual households. This project takes the opposite approach; making a model where the parameters are estimated from measurement data from each house. The next step is to take all the information these simulations can give, and sum them up to use in the VPP.

Concurrent with the increase of intermittent power generators, the demand for a more intelligent power-grid grows. Being able to predict the power and increase the flexibility of both the power generators and the power consumers, are at the core of the future intelligent grid. If it is possible to incorporate more intermittent power in the grid, it will be because the technology of shifting the consumption in time has improved. This project demonstrates a way of improving a VPP, by predicting the electricity consumption of the heat pumps that are part of the pool.

Bibliography

[1] Danfoss. *Vedligeholdelsesvejledning*. 2008.

[2] Den Centrale Afdeling for Sygehushygiejne. *Legionella i varmt brugsvand*. Statens Serum Institut, first edition, 2000.

[3] European Technology Platform for Electricity Networks of the Future. What is a smartgrid? `http://www.smartgrids.eu/ETPSmartGrids`, 2014. [Online; accessed 26 November 2014].

[4] Nord Pool Spot. Day-ahead market elspot. `http://www.nordpoolspot.com/TAS/Day-ahead-market-Elspot/`, 2014. [Online; accessed 3 February 2014].

[5] Nordjysk Elhandel, Aros Teknik, Neogrid Technologies, Aalborg University, Department of Electronic Systems, and EURISCO. *Fra vindkraft til varmepumper*. Energinet.dk, July 2012.

[6] Regeringen. *Energistrategi 2050*. Klima- og Energiministeriet, February 2011.

[7] Regeringen, Venstre, Dansk Folkeparti, Enhedslisten, and Det Konservative Folkeparti. Aftale om den danske energipolitik 2012-2020. `http://www.ens.dk/sites/ens.dk/files/politik/dansk-klima-energipolitik/politiske-aftaler-paa-energiomraadet/energiaftalen-22-marts-2012/Aftale_22-03-2012_FINAL_ren.doc.pdf`, March 2012.

Appendix A. Heat pump test set-up

Tests were conducted at the company DVI, where measurements were taken from a temperature-triggered on/off heat pump. The set-up was made to have a higher sampling frequency than the five minutes in SDVP. This was in order to find how much time it takes from the time the heat pump gets an on-signal till the power is more or less constant. Also it would be of interest to get a better idea of how the temperature T_{fwd} evolves over time, since this would make the model able to predict the state of the heat pump more precisely.

In short the dynamics of the heat pump was in focus, and especially the settling-time of the power P_{hp}. What was found is that the dynamics of the electricity for an on/off heat pump is not of importance. It settles quite quickly compared to what is possible to fetch from the data in SDVP, and more importantly it is much faster than the dynamics of the house with its thermal inertia.

Instead it is simply modelled as a gain P_{max} times the control-signal. T_{fwd} however is quite important to know the dynamics of. This was not done at the test facility at DVI, because this was only for one specific heat pump. For the model it is better to find a way to estimate the dynamics of each heating system from the measurement data available on SDVP.

A.1 Schematic of set-up

Basically a heat pump consists of four parts: A compressor pump, a condenser, an expansion valve and an evaporator. In the case of a geothermal heat pump the heat comes from the ground and is extracted in the evaporator. When the water is hot it will send the heat to either heating of the house or to heating up the hot-water. In both cases there is a heat exchanger.

A.2 Output data

In figure A.2 the typical measurement data drawn out from the experiments at DVI are shown. The power, the forward temperature and the temperature of the return water is plotted against time. The bubbles shown on the line of the power are the actual data-points, which are between 10-25 seconds apart.

From the plot in the figure it is not possible to determine any time-constant for the power dynamics; either it is off or it is on. If no dynamics can be found from measurements logged every 10-25 seconds, then it is negligible for a control algorithm with a sample-time of five minutes. Also considering the thermal inertia of the house makes the very fast dynamics of a heat pump unnecessary to consider.

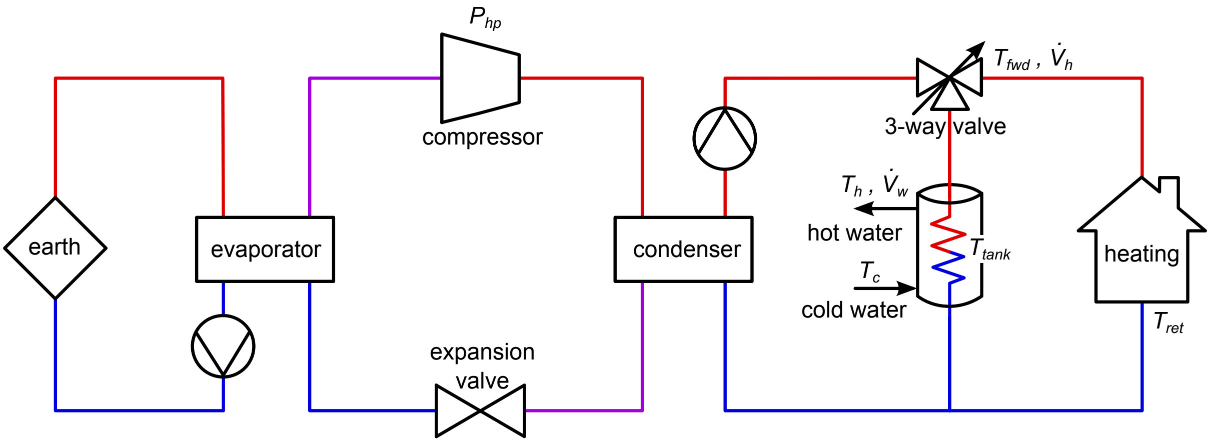

Figure A.1: Schematic of test set-up. It is the same as figure 1.10. The heat pump cycle with its four components are drawn in the middle. The heat source is the ground. The electricity consumed by the heat pump P_{hp} is the sum of the power for the compressor pump and the two circulation pumps.

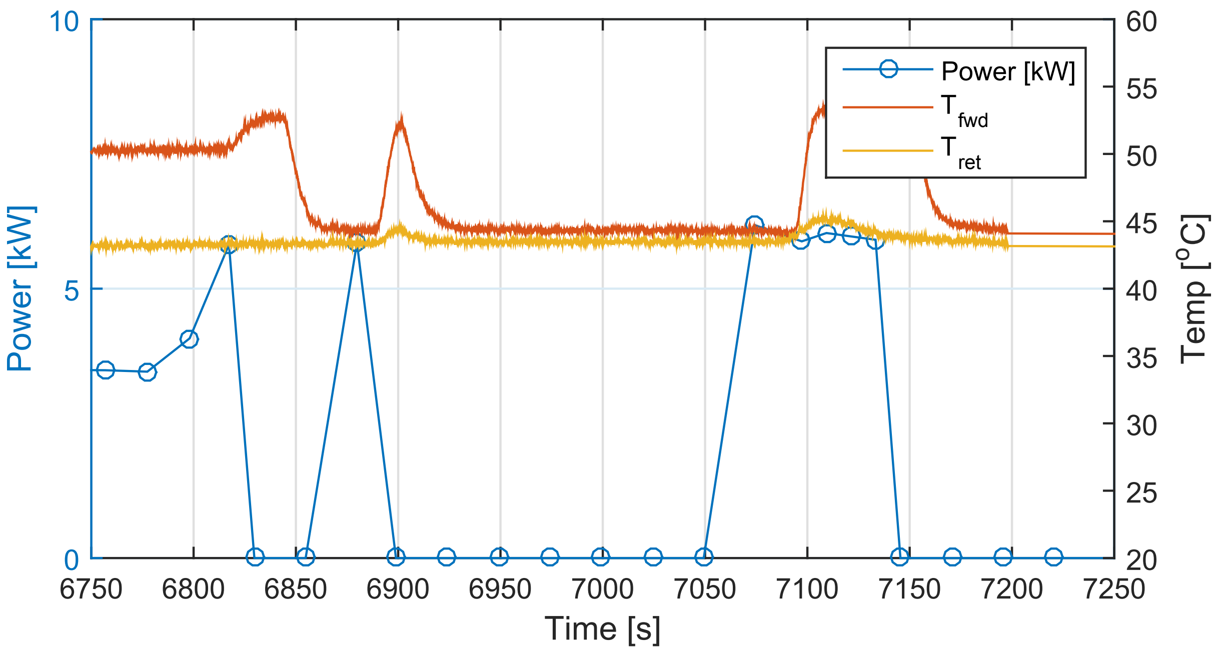

Figure A.2: Measurement data from experiments with heat pump at DVI. The o's on the power are the actual data-points.

A.3 LabVIEW

The software LabVIEW was used to acquire the measurements together with a NI-DAQ box. In figure A.3 is a screen-dump from the Block Diagram side of the program made to collect and store the data in a spreadsheet. The flow-meter was never able to provide meaningful data; it gave a lot of noise in the form of very short periods between each impulse, which would have meant thousands of litres of water every second.

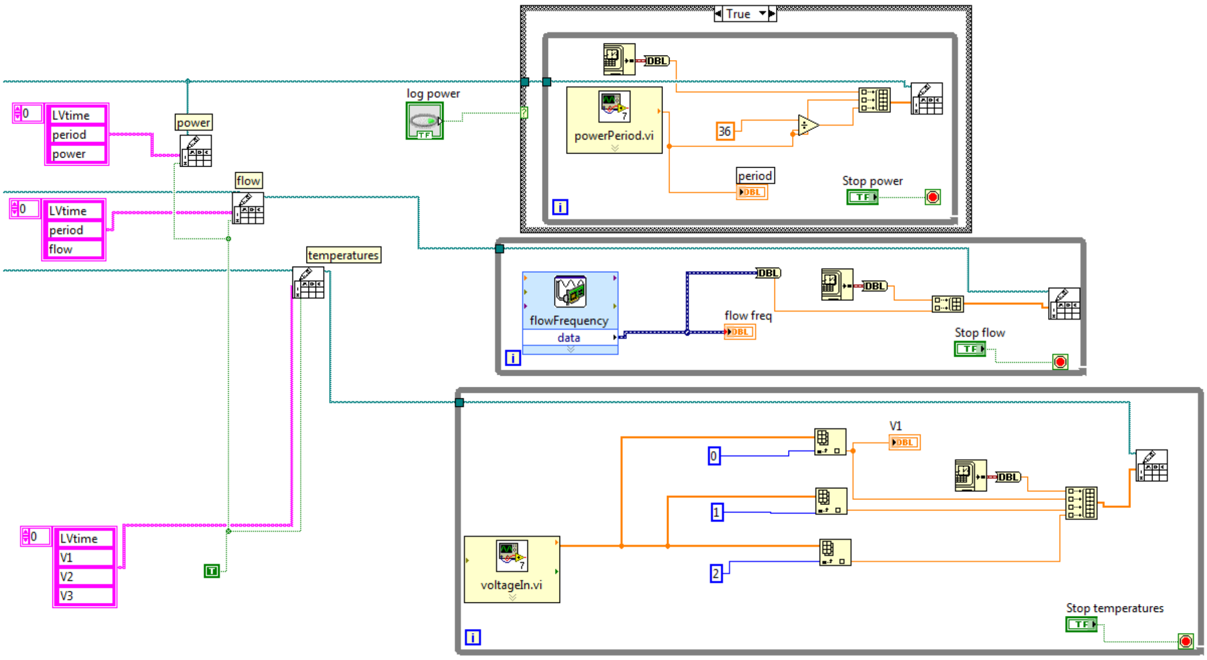

Figure A.3: Screen dump from LabVIEW program made to acquire data from the experiments.

Appendix B. Data manipulation

This appendix provides a short overview of the methods used on the data from the SDVP platform, and some explanation to why. This establishes the base of estimating the parameters to use in the model developed during this project.

B.1 start and stop times

First off it is important to know the actual points in time where each heat pump gets a start and a stop signal. To find these points the on/off signal u is first found from the power measurements P_{hp}. Whenever P_{hp} is above some limit $P_{lim} = 0.2\,\mathrm{kW}$, the signal is on $u = 1$. Likewise it is off whenever the power is below the limit $u = 0$.

The start and stop trigger-times are then found from finding the time where the signal changes from $0 \rightarrow 1$ and from $1 \rightarrow 0$, respectively. In figure B.1 is shown the power and the derived control-signal u, together with the assumed start and stop times for the heat pump.

These times t_{start} and t_{stop} can also be seen as the rise and fall-times of the power. They are used several times to figure out the different trigger-limits for time, temperature and hot-water tank level.

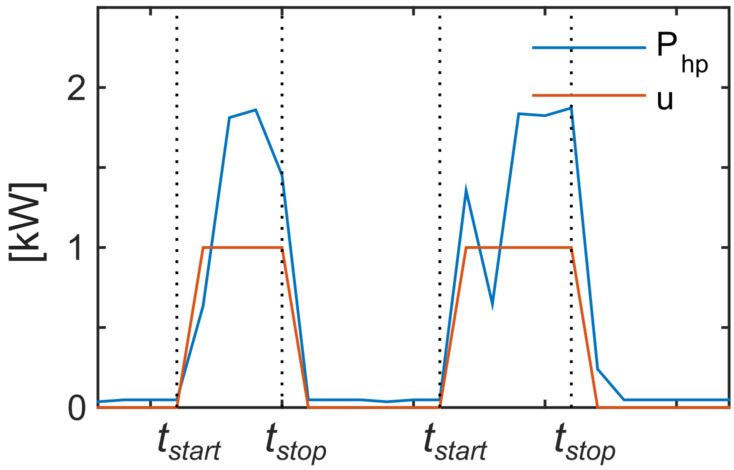

Figure B.1: Start and stop times.

Assuming the power measurements has been initialised as `Php`, the start and stop times can be found using Matlab. The following code outputs the control-signal as `u` and the start and stop-times as `i_start` and `i_stop`, respectively. These are logical arrays that can be used to find the start and stop points for the different measurements. To find the actual points in time, one writes `time(i_start)` or `time(i_stop)`.

The Matlab code would be:

```
Plim = 0.2; %[kW]
u = false(length(Php),1); %assumed on/off signal
u(Php>Plim) = 1;

%find HP start stop time
i_start = find(diff(u)==1);
i_stop = find(diff(u)==-1);

%begin with a start and end with a stop
if i_stop(1) < i_start(1)
    i_stop(1) = [];
end
if i_stop(end) < i_start(end)
    i_start(end) = [];
end

%remove start and stop in analysis where HP is blocked for a long time
ixx = find(i_start(2:end)-i_stop(1:end-1)>18); %find off-time > 1.5h
i_start(ixx+1) = [];
i_stop(ixx+1) = [];
```

B.2 Forward set point for integral-controlled heat pump

The same procedure for finding the linear fit of $T_{fwd,set}$ for a temperature-triggered heat pump, has been tested on an integral-triggered one. Though this would be harder to validate, since the trigger-limits are not evident from a T_o-T_{fwd} plot. In figure B.2 is shown the resulting graph of the forward temperatures. It is obvious that this heat pump is not controlled in the same manner as the temperature-triggered one. The temperature window would be at least ± 10, which is unreasonable.

A better way to find the temperature set point for an integral-triggered heat pump is by taking the integrated signal into account. The values of the trigger-limits must be the same, even though there might be other triggers either delaying or precipitating the starts and stops.

To find the integrated error-signal without knowing the actual set point nor the trigger-limits, the whole issue is formulated as an optimisation problem.

B.2.1 Data requirements

The first thing is to find a suitable period of time, where the heat pump can be assumed to be running mainly because of a need for heating in the house. Runtime caused by a need for hot-water or keeping the floor hot will only disturb the optimisation process. Therefore a day in April 2013 has been chosen, where it was quite cold. Another reason for choosing this time is that the variation in T_o is between 4–16 °C, which will give a better result than a small variation, since the set point depends on T_o. Also there should not be too far from a stop to the next start, since it might start just because of the maximum time off. Three hours is the absolute maximum, while a threshold of one hour is preferred.

In figure B.3 the electricity, forward- and outside temperature is illustrated for one heat pump on 16 April 2013. There are a lot of starts and stops in less than 24 hours, which makes it fast to solve the optimisation problem, while getting a reasonable result.

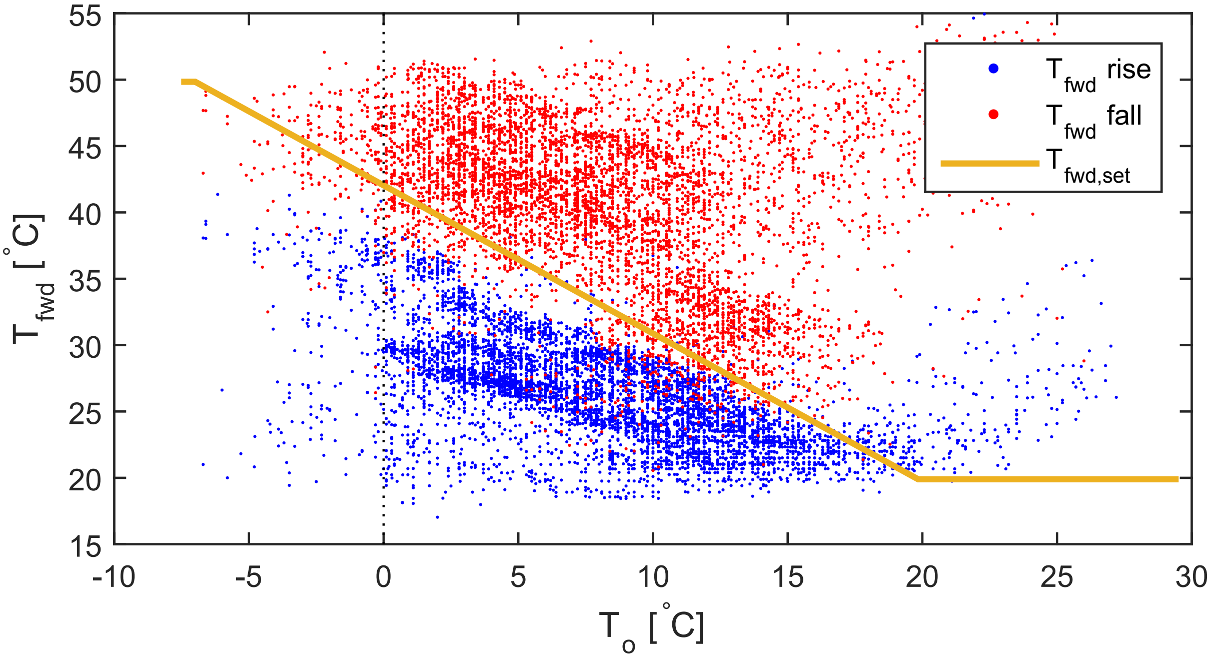

Figure B.2: Forward temperatures plotted against outside temperature for heat pump starts and stops. The set point is calculated from a fit of all the T_{fwd} measurements.

The start and stop-times are calculated as described previously in section B.1. This gives some very short starts and stops around 4:00 and 7:00, which can be seen from the power measurements. These downward-spikes comes from two samples that are zero, which is likely caused by missing measurement data or similar. The power is not zero anywhere else, instead it is 0.012 kW. The zero-data must be considered a mistake in the measurement equipment or similar. When these points are not considered, these "fake" starts and stops are thrown out of the data.

When the data is filtered, it will give a graph as shown in figure B.4. Now it is ready to optimise from the starts to the stops.

B.2.2 Optimisation

To formulate the set point as an optimisation problem, first the variables must be defined. From equation 2.2 the integrated error is known, yet it depends on a and b. To put it into an optimisation problem the measurements and b are accumulated. In Matlab this is written as the code below. The optimisation toolbox YALMIP has been used together with Matlab, so some of the code is YALMIP-specific, yet it should be understandable what each command does from the comments. In the code P_{hp}, T_{fwd} and T_o have been initialised.

As it can be seen the integrated error is reset for every start. This is to only look into the period from a start till the next stop.

Only three constraints are used for the problem, one of them being the definition of I. It is better to define use the cost-function instead of defining a lot of constraints, since they make the problem infeasible.

The curvature of I is part of the cost-function `objective`. When the heat pump is started, the curvature should be negative until it is stopped, while it should be positive from stop till start.

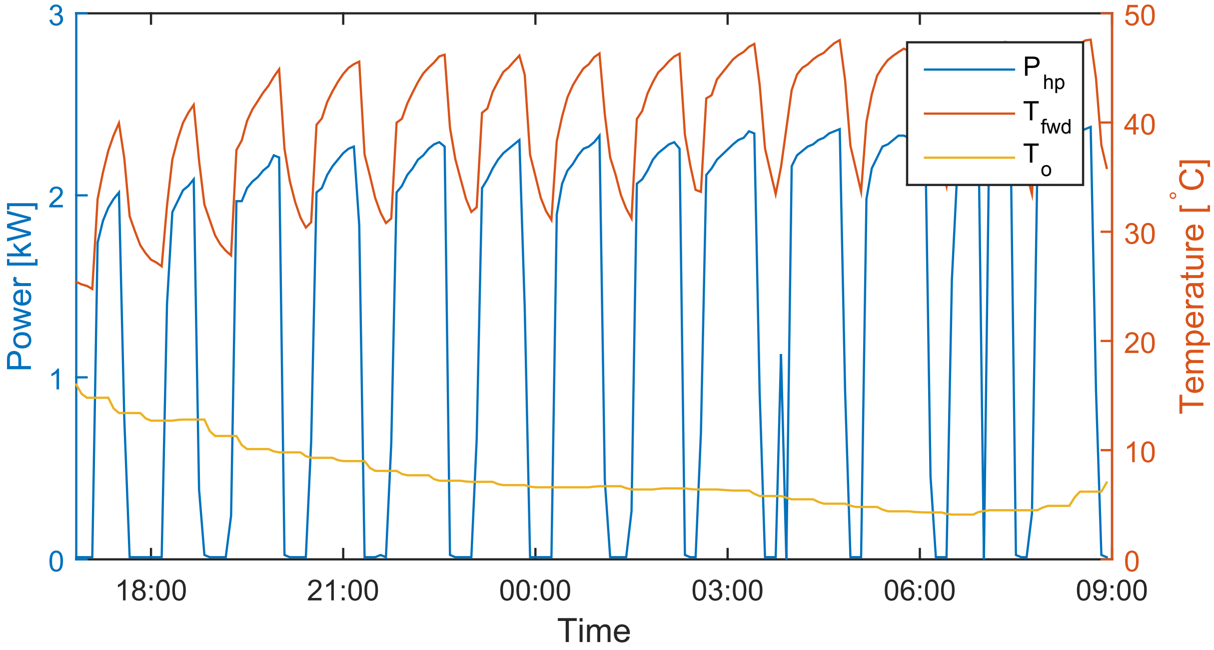

Figure B.3: Measurements from integral-triggered heat pump from 16 April 2013.

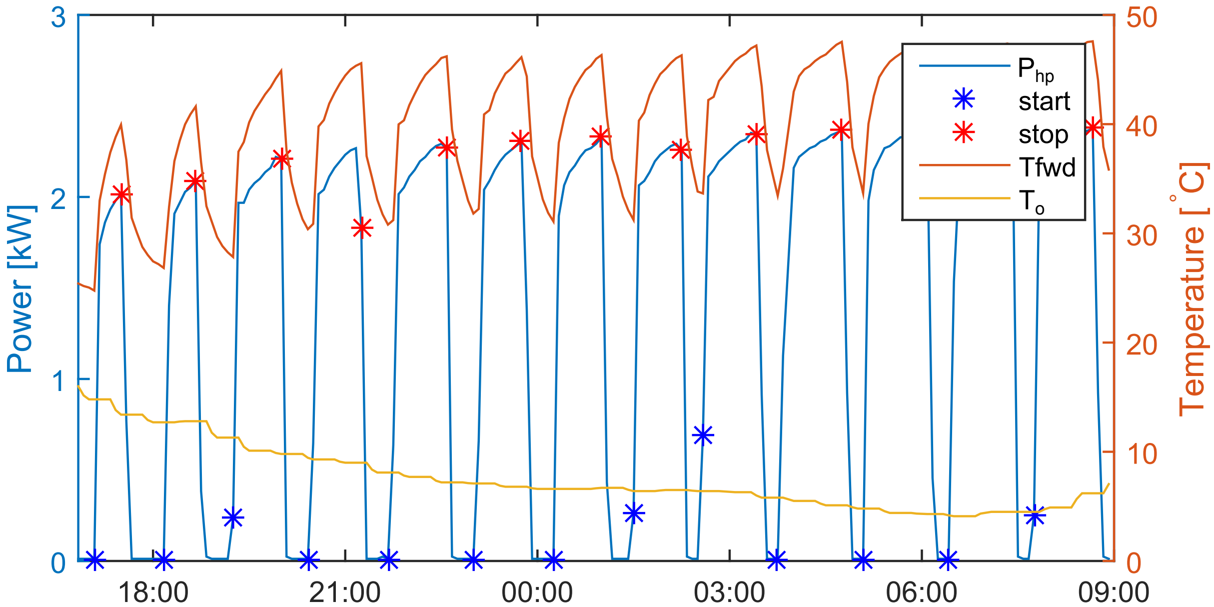

Figure B.4: Filtered measurement data for 16 April 2013 with starts and stops shown. Starts are marked with a blue * and stops are marked with a red *.

```
one_csum = zeros(1,length(Php)); %accumulated one
Tfwd_csum = one_csum; %accumulated Tfwd
To_csum = one_csum; %accumulated To
one_csum(1) = 1; %start value
Tfwd_csum(1) = Tfwd(1);
To_csum(1) = To(1);

for i = 2:length(Php)
    if ismember(i,i_start) %reset I for every start
        one_csum(i) = 1;
        Tfwd_csum(i) = Tfwd(i);
        To_csum(i) = To(i);
    else
        one_csum(i) = one_csum(i-1)+1;
        Tfwd_csum(i) = Tfwd_csum(i-1)+Tfwd(i);
        To_csum(i) = To_csum(i-1)+To(i);
    end
end

%Initialise variables
yalmip('clear') %erase old yalmip-variables

a = sdpvar;
b = sdpvar;
I = sdpvar(1,length(Php));

constraints = []; %init constraints
objective = 0; %init cost function
ysettings = sdpsettings('verbose',2); %solver settings

%Define constraints
constraints = [constraints I==5*(a*To_csum+b*one_csum-Tfwd_csum)]; ...
    %I=int(Tset-Tfwd)dt
constraints = [constraints a<0]; %negative a
constraints = [constraints b>0]; %positive b

%Define cost function
%curvature of I from start to stop must be negative
for n = 1:length(i_start)
    ddI = diff(diff(I(i_start(n):i_stop(n))));
    objective = objective + sum(max(ddI,0));
end

%curvature of I from stop to start must be positive
for n = 2:length(i_start)
    ddI = diff(diff(I(i_stop(n-1):i_start(n))));
    objective = objective + sum(max(-ddI,0));
end

objective = objective + std(I(i_start)) + std(I(i_stop)); %minimise I-variation

%Optimise
diagnostics = optimize(constraints,objective,ysettings);

%Output
I = value(I);
a = value(a);
b = value(b);
Tset = a*To+b;
```

B.2.3 Output

This does still not give the integral trigger-limits, because I is reset for each start. Yet from figure B.5 it appears as a reasonable approach for finding $T_{fwd,set}$. From the figure it is seen that the integral follows the curve that is assumed to be normal operation of a heat pump. The curvature of I is negative when the heat pump is running, making the integral go down, while it is positive when the heat pump is not running, building up the integral to the point where it turns on.

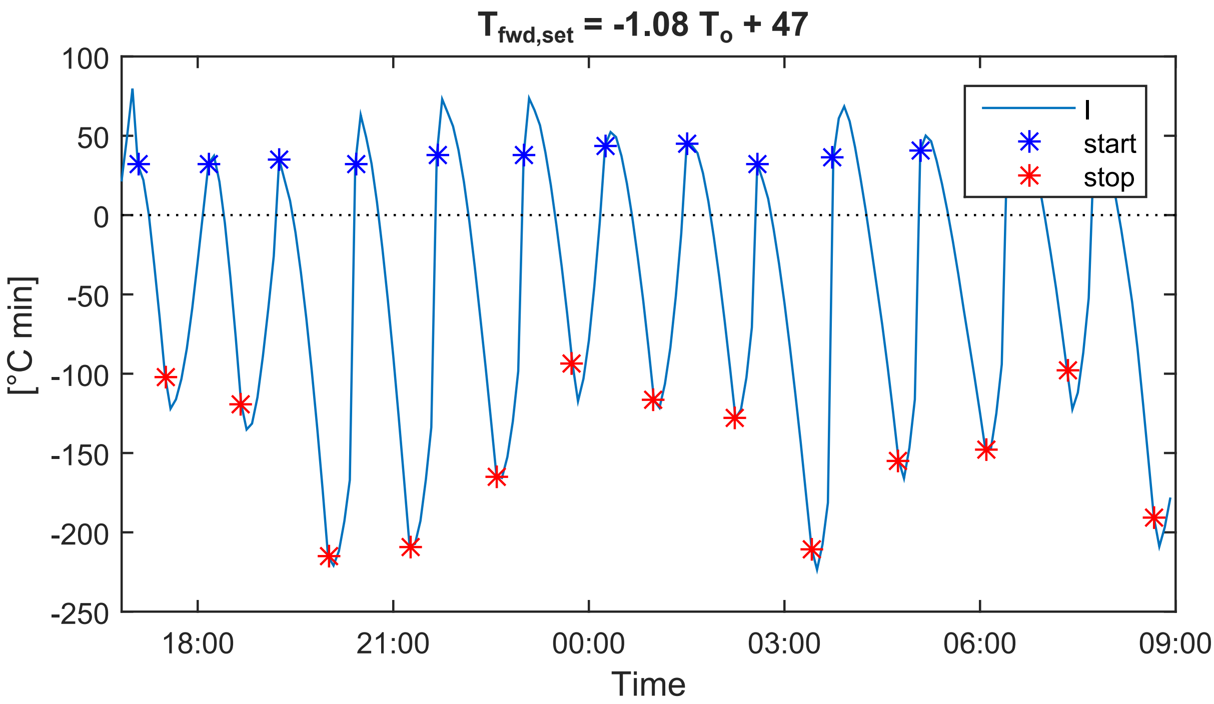

Figure B.5: The integrated signal with starts and stops against time. For this graph the set point is given by the equation $T_{fwd,set} = -1.08\,T_o + 47$.

B.3 Forward temperature dynamics

The dynamics of the start and the stop of the heat pump are described from equations (2.4) and (2.5), respectively. T_{fwd} is the temperature of the heat-carrying fluid, T_s is the temperature of the heat source and T_i is the inside temperature of the house. τ_{on} is the time-constant when the heat pump starts and τ_{off} is the time-constant when the heat pump stops. t is the time from either a start or a stop, and likewise $T_{fwd}(0)$ is the temperature at a start and a stop depending on the equation.

$$T_{fwd}(t) = T_s + (T_{fwd}(0) - T_s)\, e^{-t/\tau_{on}} \tag{2.4}$$

$$T_{fwd}(t) = T_i + (T_{fwd}(0) - T_i)\, e^{-t/\tau_{off}} \tag{2.5}$$

B.3.1 Adjusting start and stop times

In section B.1 it was shown how the start and stop times of the heat pump is found from P_{hp}. To get a more stable calculation of the heat pump dynamics, the start and stop times are "corrected" to fit to T_{fwd}; the closest minimum value is chosen for the start-time, while the closest maximum value is chosen for the stop-time. This is to compensate for that the temperature might continue up or down after the power has been turned off or on, which might be caused by inaccuracies in the measurement data, or an actual delay in the system.

The Matlab code for finding the adjusted start and stop times is shown below. For the start-time one sample before and one sample after is taken to find the minimum value for T_{fwd}. For the stop-time, the code looks two samples back and one sample ahead to find the maximum value for T_{fwd}.

```
%Make min and max Tfwd the real start and stop
for n = 1:length(i_start)
    m = i_start(n);
    if m > 1
        [~,imin] = min(Tfwd(m-1:m+1));
        i_start(n) = m+imin-2;
    end
    m = i_stop(n);
    if m > 2
        [~,imax] = max(Tfwd(m-2:m+1));
        i_stop(n) = m+imax-3;
    end
end
```

B.3.2 Heat pump stop

When the heat pump stops, the time-constant can be found from equation (2.5), solving for τ_{off}. Considering only the stop and start points, the equation can be expressed as in equation (2.9). By this equation the time-constant is calculated readily.

$$\tau_{off} = \frac{-(t_{start} - t_{stop})}{\ln\left(\frac{T_{fwd,start} - \langle T_i \rangle}{T_{fwd,stop} - \langle T_i \rangle}\right)} \tag{2.9}$$

B.3.3 Heat pump start

When the heat pump is started there are two parameters to estimate, both T_s and τ_{on}. This is done with optimisation. The Matlab code is shown below.

When T_{fwd} and the starts and stops have been loaded, the variables for the problem are initialised. For this optimisation problem is used a starting guess; otherwise it takes substantially longer to reach an optimal solution.

The constraints are that the parameters should be positive, and that the calculated temperature `Topt` should obey equation (2.4). Since the solver is not thrilled with calculating e^0, a small number has been added to the starting time (`(eps)` is the machine-epsilon Matlab is operating with).

```
yalmip('clear') %erase old yalmip-variables

%Initialise variables
Ts = sdpvar;
tau = sdpvar;
Topt = sdpvar(1,length(Tfwd));

assign(Ts,40); %starting guess
assign(tau,20); %starting guess

constraints = [];
objective = 0;
ysettings = sdpsettings;
ysettings.usex0 = 1; %use starting guess
ysettings.solver = 'Ipopt'; %choice of solver

%Constraints
constraints = [constraints Ts>0 tau>0];

for n = 1:length(i_start)
    vec = i_start(n):i_stop(n); %from start to stop
    constraints = [constraints ...
        Topt(vec)==Ts+(Tfwd(vec(1))-Ts)*exp(-(time(vec)-time(vec(1))+(eps))./tau)];
    objective = objective + sum(abs(Tfwd(i_stop(n))-Topt(i_stop(n)))); %get ...
        endpoint right
    objective = objective + 0.2*sum(abs(Tfwd(vec)-Topt(vec))); %get close to ...
        measurement
end

%Optimise
diagnostics = optimize(constraints,objective,ysettings);

%Output
Ts = value(Ts);
tau = value(tau);
Topt = value(Topt);
```

Running this snippet of code will provide the time-constant and the gain-constant T_s. How well this could fit to the measured data is shown in figure B.6, which is taken for the same heat pump and time-period as previously described. In this case $\tau_{on} = 18\ \text{min}^{-1}$ and $T_s = 48\ °\text{C}$.

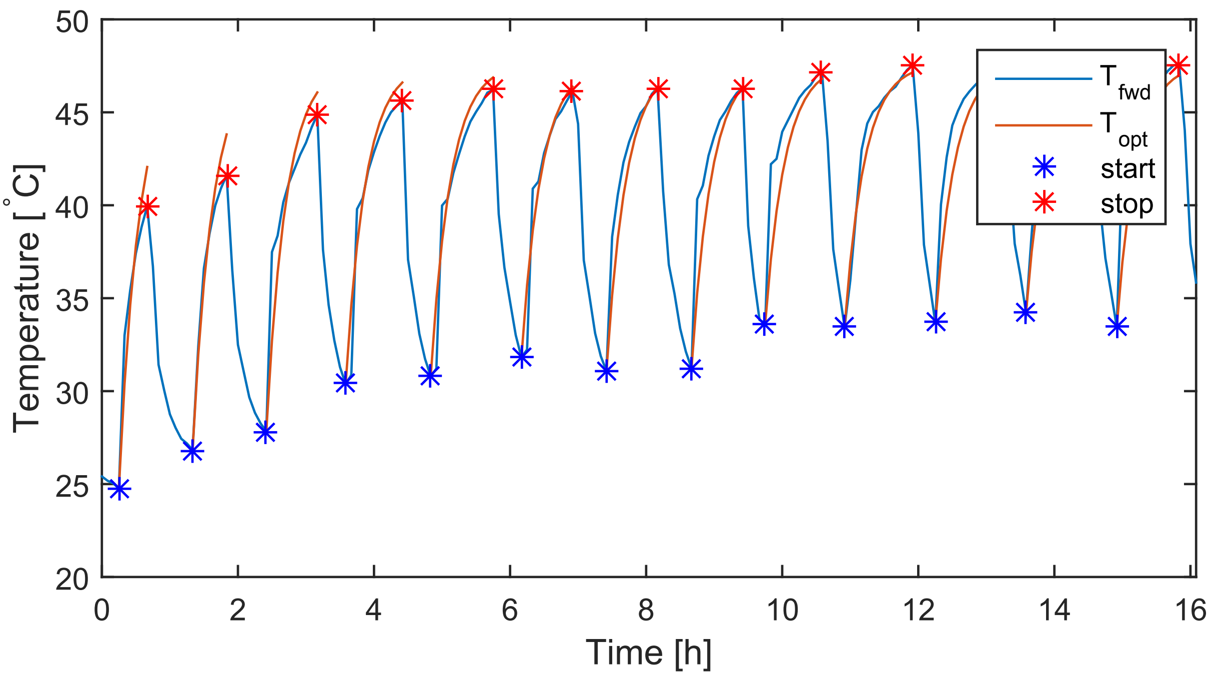

Figure B.6: Modelled temperature plotted with measured temperature. The blue * are starts and the red * are stops. For this graph $\tau_{on} = 18$ and $T_s = 48$.

Appendix C. Code for prediction model

This appendix contains the Matlab code written for the model of an integral-controlled heat pump. To simulate a temperature-controlled heat pump, one could simply switch the comments for line 110 and 111.

```
1  function [NRMSE] = Iprediction(boxdata)
2  %%% IPREDICTION simulates the operation of an integral-controlled heat pump.
3  %%% Outputs are plots of the simulation outputs with Tfwd_sim, Php_sim and
4  %%% NRMSE of Php_sim. boxdata is a struct with measurement data of the heat
5  %%% pump and calculated parameters such as time-constants, Tfwdset etc.
6
7  close all; clc
8  horizon = 240; %[min]
9  time = boxdata.time./60; %[min]
10
11 iu = 60;
12 il = 0;
13 Ttankmin = boxdata.tankmin;
14 wmax = boxdata.wmax;
15
16 %filter out Php=0
17 Php1 = boxdata.Php;
18 ix = Php1>0;
19 Php1 = Php1(ix);
20 %calculate control-signal
21 u1 = Php1*0; u1(Php1>0.2) = 1;
22
23 %time to simulate from
24 t_start = datenum('2013-04-16 14:50');
25 n = find(boxdata.time≥matlabtime2epochtimestamp(t_start),1,'first'); %index ...
       of starttime
26 t_sim1 = time(n):time(n)+240;
27 %adjust t_sim to start or stop
28 u = interp1(time(ix),double(u1),t_sim1,'nearest');
29 u = logical(u);
30 i_start = find(diff(u)==1);
31 i_stop = find(diff(u)==-1);
32 i_ss = sort([i_start i_stop]);
33
34 %simulate from each start and stop
35 NRMSE = 0*i_ss;
36 for m = 1:length(i_ss)
37     t_sim = t_sim1(i_ss(m)):t_sim1(i_ss(m))+horizon; %beginning point
38     Php = interp1(time(ix),Php1,t_sim);
39     Tfwd = interp1(time,boxdata.Tfwd,t_sim);
40     To = interp1(time,boxdata.To,t_sim);
41     Ti = interp1(time,boxdata.Ti,t_sim);
```

```
    u = interp1(time(ix),double(u1),t_sim,'nearest');
    u = logical(u);
    Ttank = interp1(time,boxdata.Ttank,t_sim(1)); %Ttank at beginning of simulation
    wacc = interp1(time,boxdata.wacc,t_sim(1)); %wacc at beginning

    %set point
    Tset = boxdata.a*To+boxdata.b;
    Tset = max(min(Tset,boxdata.Tfwdsetmax),boxdata.Tfwdsetmin);

    %dynamics
    Ts = boxdata.Ts;
    tau_on = boxdata.tau_on;
    tau_off = boxdata.tau_off;

    %time-triggers
    dt_on = 30; %last start is long ago
    dt_off = 0; %the heat pump has not been off for long

    I = 0*Tfwd; %simulated integral-signal
    %determine if simulation begins from start or stop
    if any(i_ss(m)==i_start)
        utemp = 1; %begins on
        I(1) = iu;
        fprintf('Start\n')
    elseif any(i_ss(m)==i_stop)
        utemp = 0; %begins off
        I(1) = il;
        fprintf('Stop\n')
    else
        error('There is something wrong with the starting point %i',i_ss(m));
    end
    % I(1) = I(1)+Tset(1)-Tfwd_sim(1); %update with first efwd

    % Tset_sim = Tset(1); %keep set point constant from beginning
    u_sim = 0*u;
    Tfwd_sim = 0*Tfwd;
    Tfwd_sim(1) = Tfwd(1);
    Tfwd_0 = Tfwd(1);
    for i = 2:length(t_sim)
        u_sim(i) = utemp;
        %resets
        if u_sim(i-1)-u_sim(i)==-1 %HP start
            Tfwd_0 = Tfwd_sim(i-1); %temperature at start
            dt_on = 0;
        elseif u_sim(i-1)-u_sim(i)==1 %HP stop
            Tfwd_0 = Tfwd_sim(i-1); %temperature at stop
            dt_off = 0;
        end

        %calculating Tfwd_sim
        if utemp %it is on
            Tinf = Ts; %temperature of heat source
            tau = tau_on; %time-constant
            dt_e = dt_on; %time since start
        elseif ~utemp
            Tinf = Ti(1); %temperature inside house
            tau = tau_off;
            dt_e = dt_off; %time since stop
        end
        Tfwd_sim(i) = Tinf+(Tfwd_0-Tinf)*exp(-dt_e/tau);
```

```
        dt_on = dt_on+1; %add one minute to time triggers
        dt_off = dt_off+1;

        %error and integral
        efwd = Tset(i)-Tfwd_sim(i);
        I(i) = I(i-1)+efwd;
        %determine if on or not
        % if (dt_on≥15 && efwd≥eu) || dt_off≥180 || ... %uncomment this for ... T-controlled heat pump
        if (dt_on≥15 && I(i)≥iu) || dt_off≥180 || ...
            Ttank≤Ttankmin || wacc≥wmax
            utemp = 1; %HP on
        elseif I(i)≤il
            utemp = 0; %HP off
        end
    end

    %RMS error of u
    N = length(u);
    E = u_sim-u;
    rss = E.^2;
    u_rmse = sqrt(sum(rss)/N);
    %normalised RMS error of Php
    Php_sim = u_sim*boxdata.maxpower;
    E = Php_sim-Php;
    rss = E.^2;
    P_nrmse = (sqrt(sum(rss)/N))/(max(Php)-min(Php));
    NRMSE(m) = P_nrmse;

    fprintf('P_nrmse for m: %i is \n %2.4f\n\n',m,P_nrmse)

    %plot figure
    t_sim = t_sim-t_sim(1); %make start time zero
    figure
    [hax,line1,line2] = plotyy(t_sim,[Tfwd;Tfwd_sim],t_sim,[Php;Php_sim]);
    legend('T_{fwd}','T_{sim}','P_{hp}','P_{sim}')
    xlabel('Time [min]')
    hax(1).YLabel.String = '[^\circC]';
    hax(2).YLabel.String = '[kW]';
    title(sprintf('P_{NRMSE} = %1.3f',P_nrmse))
end
```