Mechanical Engineering Internship

Internship at Qirion
Internship Report

Author:

Koen Boonk S1384104

Duration: 14 weeks, 20 EC
Internal supervisor: dr. S. Hajimolana
External supervisor: F. Schut

June 13, 2019

Contents

Chapter 1

Summary

In this report I will describe my Internship at Qirion. The goal of this internship was to make in Simulink a thermal model of a gas boiler from Intergas. Furthermore Intergas delivered a test board which resembles the control unit of a gas boiler. The Simulink model should work with this test board together to form a complete gas boiler model. It is also described in this report how the connection between the Simulink part and the test board part has been constructed. This model can be used to test the intelligent heat control of Qirion called ISR. Since this model could not be verified due a delay with the sensors needed for the test setup this model cannot be used yet for it's original purpose. However it has been used to conduct some experiments regarding the maximum allowable temperature difference between the return temperature and the supply temperature. This temperature difference should be as high as possible and Intergas claimed it can be 40 degrees. In the end it was found that at low working temperatures, a supply temperature around and below 75 degrees Celsius, this difference can be achieved. At high working temperatures, in the range of 80-90 degrees Celsius for the supply temperature, this temperature difference was lower than 40 degrees Celsius.

Chapter 2

Introduction

Climate change is a serious issue right now. Back in 2015 the Paris Agreement was established which focused on reducing the global warming, reducing greenhouse emissions and reducing the use of fossil fuels. A lot of countries signed this agreement and among these countries are The Netherlands. This leads to an energy transition in the Netherlands where one of the goals is to no longer use gas. In 2030 the government will not use any gas from Groningen anymore and the goal for 2050 is to not use any gas at all anymore. So this is a long term goal however people need to base their decisions on this long term goal already. Keep the old, inefficient boiler because the heat pumps right now are to expensive or not efficient enough. Or buy a new boiler which is more efficient than the current boiler, but this new boiler may be forbidden within a few years. Some buildings cannot even use a heat pump since the isolation of the building is not good enough. This is mostly the case with old buildings.
For the people who need an intermediate solution Qirion developed ISR. ISR stands for Intelligente Stookregeling, which translates to intelligent heating control. The goal of ISR is to increase the efficiency of gas boilers and thus reduce the amount of gas used. This way it is possible to still invest in a new boiler right now and within a few years earn the investment back. A long term goal of the ISR project is to also use it for heat pumps but right now it only works with gas boilers.

Company Alliander

The company Alliander originated from Nuon. Nuon had to split the activities they were doing due a new law in 2006. From this parting Alliander originated. Right now Alliander consists of two big business units; Liander and Qirion. Liander is a public utility and electricity system operator. Qirion on the other hand is a company that designs new energy grids and changes current energy grids. One of the project they are doing right now is the heat network in Hengelo. They use the waste heat of AkzoNobel in Hengelo to supply heat to districts in Hengelo.

Figure 2.1: Business office in Duiven

In figure 2.1 one of the business offices where Qirion is stationed can be seen. I conducted my research in this office which is located in Duiven. This building is only a few years old and a lot of materials that are used for this building have been reused materials. Furthermore this building produces more energy than it consumes making it energy positive. There is also no gas used at all, this however made testing the model during the assignment a little bit harder.

Scrum Method

During this assignment it was decided to work with the scrum method. This was also new for my supervisor and product owner but they were eager to try it. The way it worked during the assignment is that the product owner would include everything he wanted to know in the product backlog. This product backlog would be looked at every two weeks when the sprint will be determined. The scrum master will use the items in the product backlog to determine the sprint backlog and when that is done the goals of the current sprint will be concluded and the persons responsible for each goal will be determined. To support the method Trello has been used. In figure 2.2 an overview can be seen how it looked like in Trello.

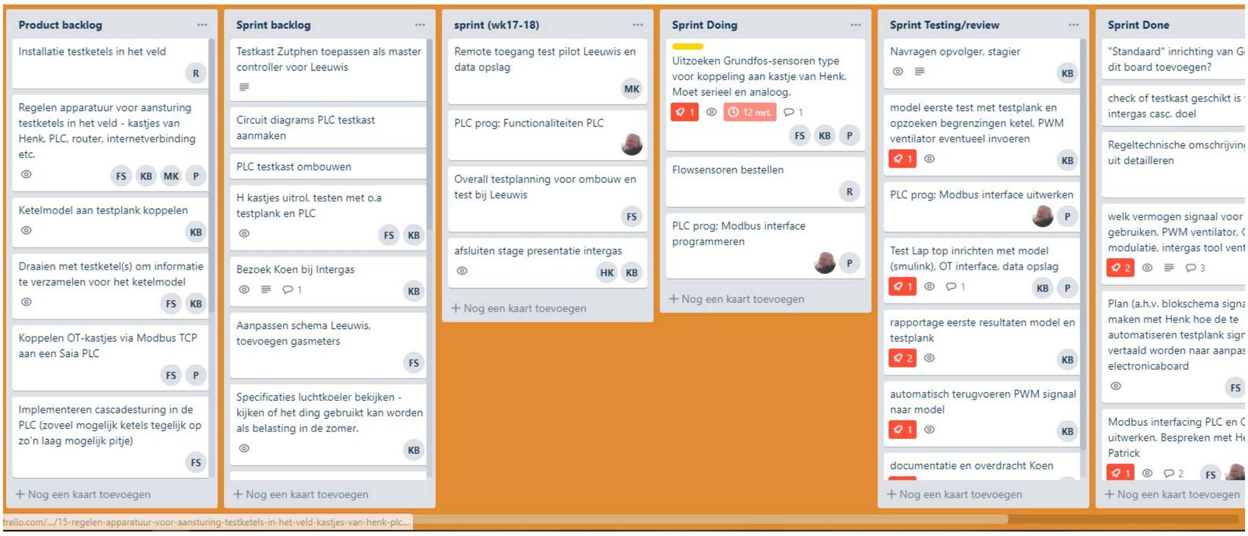


Figure 2.2: Overview Trello

Chapter 3

Problem Description

ISR is a way to improve the efficiency of gas boilers. One of the ways ISR tries to improve the efficiency of gas boilers is to increase the temperature difference between the supply and returning water. However most gas boilers do not permit a higher temperature difference than 20 degrees Celsius. This is to protect the heat exchanger in the gas boiler. But a goal of the ISR is to have a temperature difference around 40 or 50 degrees therefore a lot of gas boilers do not work as good as they could with ISR. When a company called Intergas announced that they developed a new heat exchanger which allowed a temperature difference of 40 degrees it sparked the interest of Qirion. The problem is that due trade secrecy there is no model for this specific gas boiler.

3.1 Description of the problem

The goal of this internship was to make a thermal model for a gas boiler designed by Intergas. The specific boiler where a model will be made of is a 36/48 HRE A Intergas boiler. The only thing that Intergas did deliver is a test board. On this board is the control unit of a gas boiler mounted. This test board can be seen in figure 3.1.

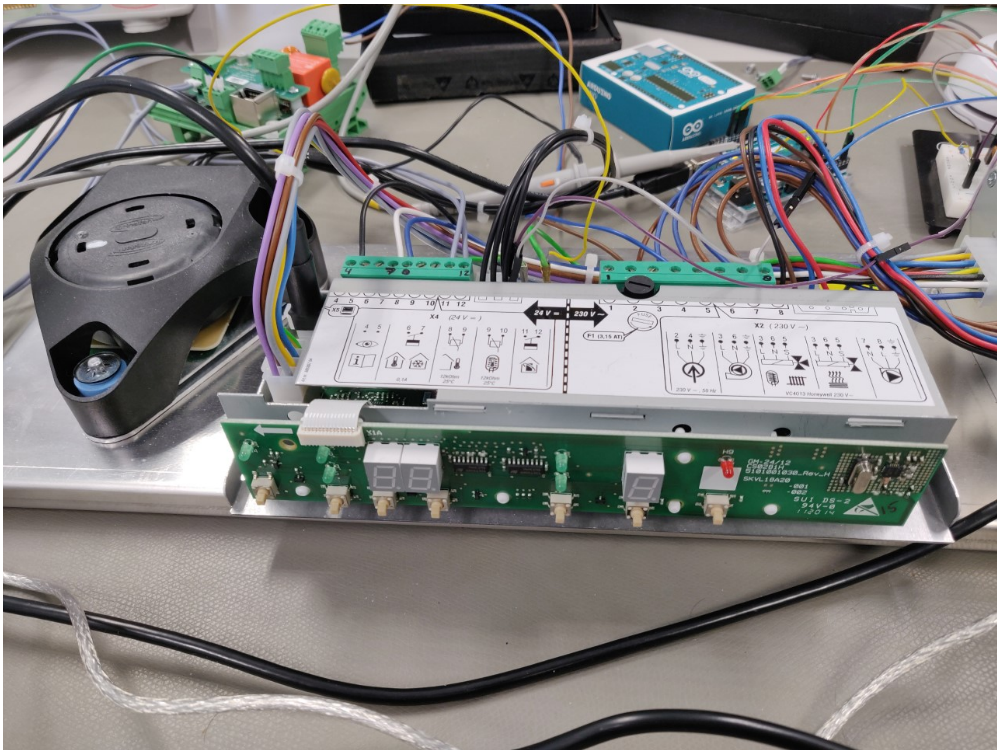


Figure 3.1: Test board of a Interhas 36/48 HRE A boiler

This control unit is like a real gas boiler and the display that is visible is the same display that is used for a functional boiler. Furthermore there is a fan attached which resembles the fan that is within a normal boiler. This fan also increases speed when the boiler increases power.

The control unit cannot measure anything since there is no real gas boiler attached to the control unit. For this reason Intergas made a box with software which can be seen in figure 3.2. This box has a lot of switches and potentiometers. There are only a few switches and potentiometers that are interesting for this assignment. These switches are the on/off switch and the flame switch although the last one should always be on auto which automatically controls the flame signal correctly. Furthermore the only two potentiometers that are interesting are the T-flow and T-return potentiometers. These two potentiometers determine the temperatures that the temperature sensors will receive. It should be noted that these sensors are located in the heat exchanger and thus not describe the temperature of the supply and return water. These sensors are named S1 for the T-Flow, which is located at the end of the heat exchanger and S2 for the T-return which is located at the beginning of the heat exchanger. With these two temperatures the boiler calculates the supply temperature. It is not known how the boiler calculates the supply temperature.

Figure 3.2: The software box of the test board of a Interhas 36/48 HRE A boiler

The last thing that was included with the test board was a program for any computer. This program is called IDS Intergas. This program showed the values the boiler receives from the software box, so every temperature, the fan speed and the pressure could be seen this way.
With the help of this test board a thermal model of the boiler needs to be made. The first goal of the internship was to make this thermal model and by using a test setup with real boilers this model could be verified. However due to the delay of the sensors which would also be placed in the test setup the model could not be verified since there was no test setup. Instead when the model was done several experiments with the model and the test board were conducted. The goal of these experiments was to determine if the boiler could indeed reach the high temperature differences as was promised. Since the test board does have a normal control unit and thus acts as a normal gas boiler it is a good way to check if no security measures would occur.

Chapter 4

Model

In this section the model will be explained as well as the connection with the test board. For the test board the way of creating a connection between the test board and the computer will be explained and for the model the most important parts will be briefly explained here. In appendix 7.2 it will be explained in detail. Making the model was a very big part of the assignment but it is assumed that explaining the method of getting to the model is not very important therefore only a brief explanation of the model will be added here. That way there will be more focus on the experiments done with the model.

4.1 Test Board

The internship assignment started off with understanding what signals were used for controlling the test board. The signals that are interesting are the ones generated by the potentiometers as well as the signals created by the control unit for the fan and pump. The pump signal is not needed for this assignment but my supervisor asked for this signal since it would be used for the ISR project in general.
Starting with the temperature signals, it was already given that the boiler normally uses 12k Ohm NTC resistors for the temperature sensors. Included in the install manual of the boiler was a table with temperatures measured by the S1 and S2 sensors and the corresponding resistance value [1]. To check if the potentiometers in the software box also were 12k Ohm pontentiometers a multimeter is used to check the resistance. Via the IDS Intergas the corresponding current temperature could also be checked. If the measured values agree with the values in the table the potentiometers have a resistance of 12k Ohm. The values agreed and thus it could be concluded that this test board also uses 12k Ohm NTC resistors.
Next are the fan and pump signals. It was also mentioned in the install manual that these signals are PWM signals[1]. However to use this signals it is necessary to know how long the pulse or pause times are. To measure the length of these times an oscilloscope was used. The total length of one period was 400 microseconds so the frequency of the signal was 2.5 kHz. Furthermore the voltage of the signal was 15.5V An interesting thing was that this PWM signal works the other way around, a smaller pulse meant a larger duty, but other that that the times of the pulses and pauses are known now which is enough information for these signals. An example of the scope images can be seen in figure 4.1.

4.1.1 Connecting the test board to the computer

The next step was to find a way to let Simulink communicate with the test board. The way this was solved is by using an Arduino. An Arduino is a micro controller designed for small projects which people can make by them self. Included is also a program were you can make models on your computer. However what makes the Arduino a good choice is that Simulink can communicate with the Arduino if the right add-ons are installed. This way the model software of Arduino was not used. Within Simulink there are several blocks that can be used to control the Arduino. Simple digital signals which have a 0 or 1 output, PWM signals and a SPI writeread block. Since the choice of the signal blocks belongs to the model part, this will explained in section 4.2. The physical part of connecting the computer and the test board will be explained here. By using a digital potentiometer it is possible to control the temperatures of the test board with a computer.

(a) fan at 20 per cent power (b) fan at maximum power

Figure 4.1: Fan PWM signals

The digital pontentiometer is a 10k Ohm resistor and has 8 bits resulting in 256 values. The reason for choosing a 10k Ohm resistance is the amount of values. An higher resistor with the same steps would decrease the accuracy since the amount of resistance per steps would be bigger and thus the temperature steps as well. Of course money also played a role and by using a 10k Ohm resistance in combination with normal resistors the most important temperatures could be reached which are 20 to 100 degrees Celsius. In figure 4.2 the setup can be seen. The Arduino is connected to a digital potentiometer which can send independently two different resistance values. This way only one potentiometer is needed for the two temperatures, S1 and S2.

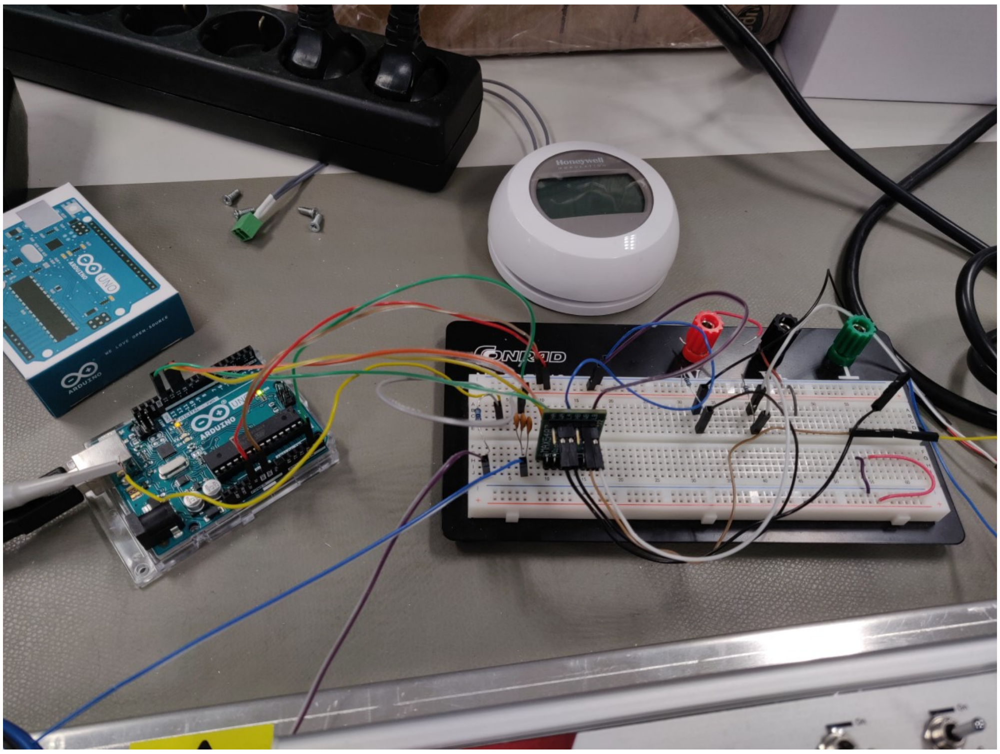


Figure 4.2: The Arduino and a digital potentiometer attached on a breadboard

By using a breadboard everything can be connected. The last thing that had to be done is to detach the physical potentiometer from the test board. By soldering the wires were detached from the physical potentiometer and attached to the digital potentiometer .In figure 4.3 the whole setup between the digital potentiometer and the software box can be seen. Now the temperatures can

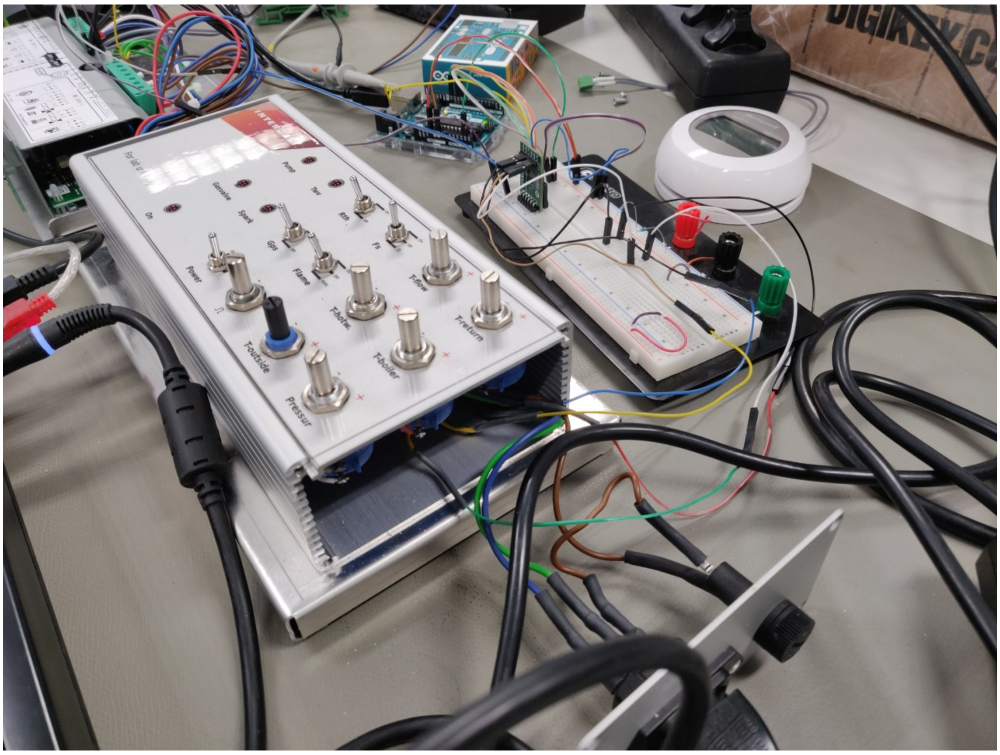


Figure 4.3: The connection between the breadboard and the software box

be controlled by any computer and every physical part to make this possible is connected. What is left is the power signal.
The output of the model, the temperatures, can be sent to the test board now but the input cannot be received yet. The PWM signal of the fan, which correspondents with the power of the boiler, is known. One problem is that there is no way for Arduino to receive PWM signals, it can only send PWM signals. So this signal was a bit trickier. Furthermore the PWM signal has a voltage of 15.5V. Since the Arduino can only handle signals of 0-5 volt this had to be lowered first. Someone helped me with the electrical part of this and what was done is using a voltage divider. This way the voltage was reduced below 5V so the Arduino could handle this signal.
Now that the voltage would be low enough the first attempt to receive the signal could be tried. This attempt made use of the digital input block. This block can only send 0's and 1's to the model. By counting the amount of 0's and 1's in a very short interval it should be possible to know what the duty of the fan would be. This is because when there is a pulse in the PWM signal it would sent a 1 to the model and when there was a pause in the PWM signal it would sent a 0 to the model. However since the signal has a period with a lenght of 400 microseconds the counting interval should be much smaller otherwise the different signals could not be recognized. This slowed the Simulink model tremendously and therefore this setup could not be used.
For the second attempt once again I got help. The same person as before told me to use a first order filter. This way the PWM signal would be changed to a certain amount of volts between 0-5V. Therefore fixing the two problems at once. The amount of volts is between the 0-5V range and this signal can be read by an analog input block in Simulink. The first order filter can also be seen in figure 4.2 on the left of the digital potentiometer in the breadboard. Every signal has been analyzed and with this information it has been made possible that the temperatures can be controller by the model and the power of the boiler can be received by the model.

4.2 Model

in figure 4.4 the whole model can be seen. On the far left and far right are the subsystems that communicate with the Arduino. The subsystem in the middle, called Ketel is the boiler. The inputs of this subsystem are the power of the boiler and the return temperature S2. The output that will be sent to the Arduino is the flow temperature S1. Furthermore S2 will also be sent to the Arduino but this temperature will be determined in the subsystem Radiator. The subsystem Radiator is the subsystem were the heat loss is programmed. This is not based on a real radiator since this was out of the scope of this research and thus every second just a large percentage of the heat will leave there. Lastly is the Sensor(beveiliging) + Controller subsystem. This subsystem controls the model when it is not connected to the test board. In here is a PID controller which controls the power of the boiler as well as the security measures that could be figured out.

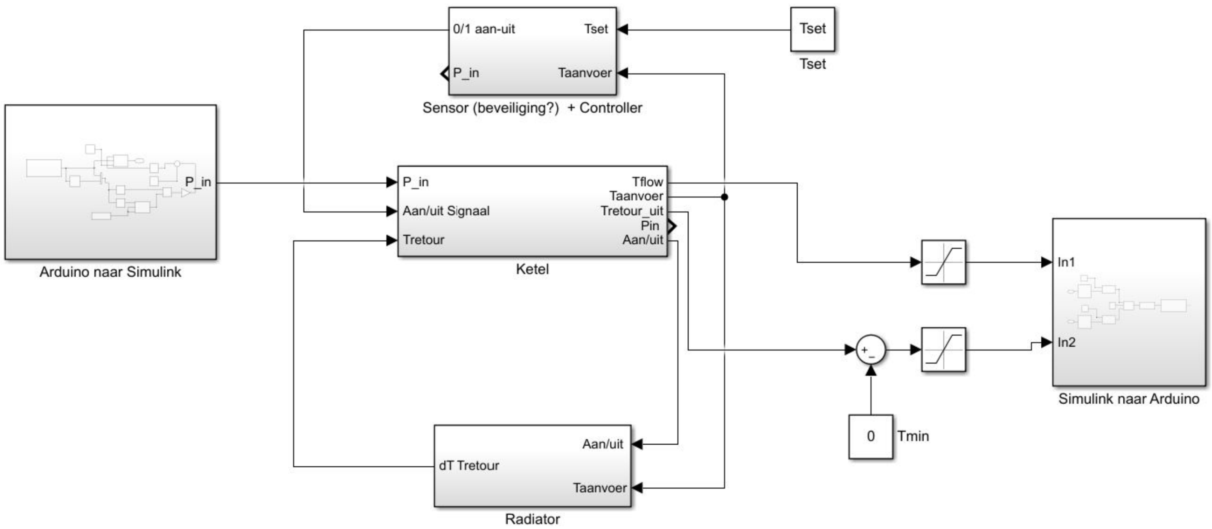
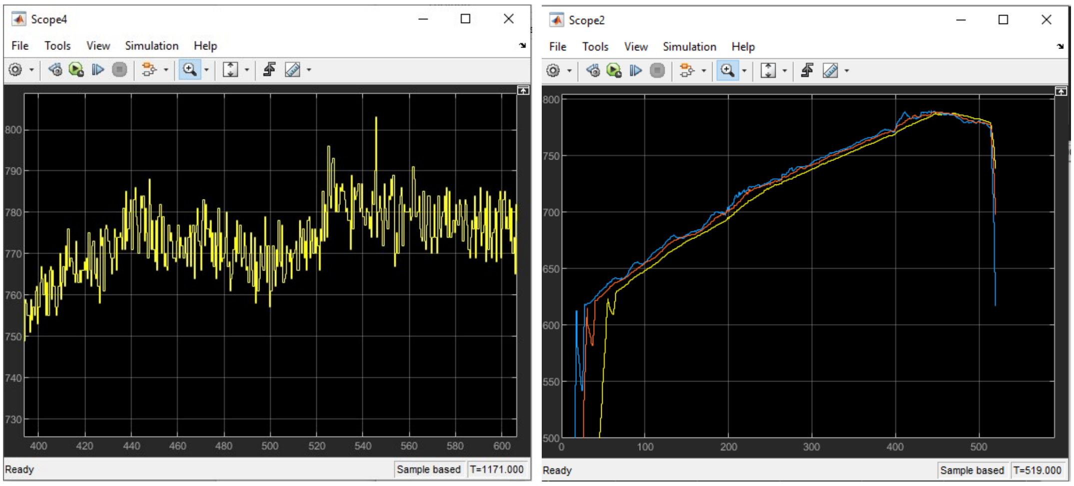


Figure 4.4: The Simulink model

4.2.1 Arduino Subsystems

First the Arduino subsystems will be briefly explained. Starting with the Arduino to Simulink subsystem. In this subsystem Simulink receives the power signal from the Arduino. This signal has been changed to a certain amount of volts as has been explained in sector 4.1.1. This amount of volts is received by the Arduino and since the Arduino has 10 bits this signal is converted to a value between the 0 and 1023. So in the end the Simulink model receives a value between the 0-1023. This should be enough steps to acquire a precise signal, however when the fan is turned on the amount of volts is at least 2.5V. Therefore a lot of these steps are skipped. Furthermore this signal jumps around a lot. What is meant by this can be clearly seen in figure 4.5(a). In this figure the signal Simulink recieves from the Arduino has been measured for 200 seconds and it can be seen that the signal is very shaky. This makes it a difficult signal to use. A physical solution to the noise could be using more resistors, stronger resistors or more condensers. This has been tried and solved it partially.

Another solution that has been tried to reduce the noise of the signal is using a moving average within Simulink. This moving average will also be used to reduce the noise. The results of the moving average can be seen in fig 4.5(b). In this scope 3 signals can be seen. The difference between this signals are the length of the moving averages. The red line has a length of 20 seconds, the blue line has a length of 50 seconds and the yellow line a length of 100 seconds. In the end the choose is based on the best combination of noise and delay. It can be seen for example that the



(a) PWM signal without anything to reduce the noise

(b) PWM signal with moving average of 20 (blue), 50 (red) and 100 (yellow) seconds

Figure 4.5: Converted PWM signals recieved by Simulink

yellow line increases quite late but has less noise. In the end it was chosen to use the 50 second moving average as it provided the best combined results.

The other Arduino subsystem, Simulink to Arduino, sends the generated temperatures to the Arduino. As said earlier the digital potentiometer works with 256 steps, so the temperature that needs to be sent to the Arduino has to be written in a value between 0-256. The block that has been used to sent the Arduino values is a SPI Writeread block. The read part of this block is not interesting though and therefore is left without a connection. This block requires a vector in uint8 where the first value of the vector determines what the digital potentiometer must do, found in its datasheet [2]. The second value is the earlier mentioned value for the temperature expressed in a value 0-256. For example the vector [01 50] means write for wiper register A the value 50. This means that S1 changes to the corresponding temperature for value 50. Lastly since the digital potentiometer is connected by one Arduino pin and thus one Simulink block, this block switches every half a second between S1 and S2 sending data.

4.2.2 Boiler subsystem

The other important subsystem is the boiler subsystem. In this subsystem there are a lot of smaller subsystems which contain all sorts of calculations to calculate the flow temperature S1 based on the power and the return temperature. The way this subsystem works is that it calculates the amount of heat is generated per second by the gas, taking efficiency into account. It then adds this heat to the return water every second also taking into account that the heat exchanger heats up. If there is no power, thus the brander is not burning, the flow temperature is the same temperature as the return temperature. The heat exchanger could also be the same temperature but this depends if it is already cooled down due convection. The boiler subsystem and each associated subsystem will be explained in appendix 7.2.
The only thing that will be highlighted here is the efficiency. It was already known that the efficiency depends on the return temperature, after all a lower return temperature means that more heat of the flue gasses can be used to heat the return water. Especially below 55 degrees Celsius improves the efficiency since the dew point of the flue gasses is around 55 degrees Celsius. Going below that temperature means the flue gasses change phase releasing a lot of latent heat. However later during the assignment it was figured out that the power of the boiler also played a

role. The heat exchanger is designed to work good when the power is 100 per cent. However the power will not all the time be 100 per cent, meaning the heat exchanger will be overdimensionized for lower power values. This will slightly increase the efficiency of the heat exchanger. This is also confirmed by literature [3]. From the literature also the bivariate polynomial for the efficiency was created. The polynomial found in the article did not seem to be correct and thus the figure in the article together with the curve fitting app have been used to create this polynomial. The efficiency curve from the article can also be seen in figure 4.6. This way a bivariate polynomial was created. However there was one more problem, Simulink could not use this function when it was saved in the workspace of Matlab. To solve this a table was created with fixed return temperature and power level size steps. By using the look-up block in Simulink with this table as input this was solved. Simulink searches for the corresponding value of the efficiency by using the two variables and interpolating if needed with help of the table.

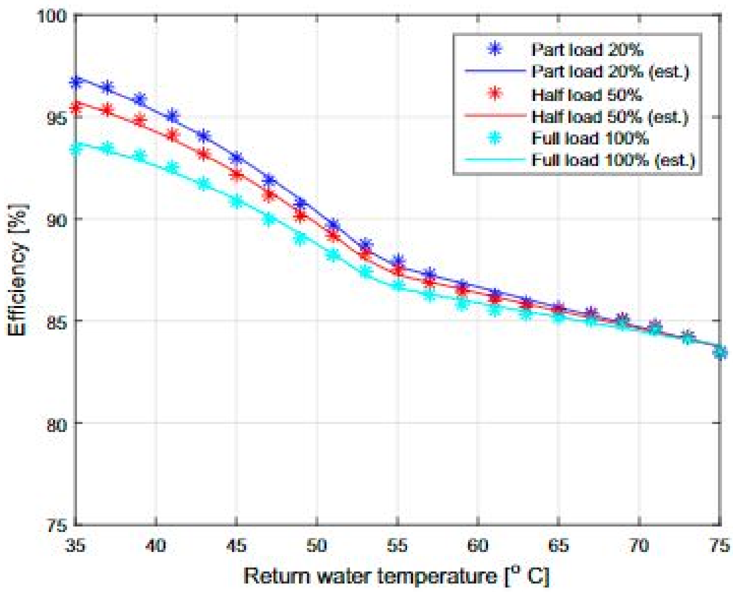


Figure 4.6: The efficiency curve under normal conditions [3]

4.2.3 OpenTherm control box

There is one more thing that has been done for the model, however it is a little bit a combination of a physical solution combined with Matlab. The company ExaTech made for Qirion a control box that uses the OpenTherm protocol. this control box can be seen in figure 4.7. With this control box it is possible to send OpenTherm commands with any computer to the test board. The reason this control box has been made is to control the power level of the test board. This was not possible and later in section 5 it will be explained why this was not possible. However what is still possible is to send other commands to the test board and the one command that is interesting for now is the set temperature. One problem was that this control box uses a software that has been designed by the company ExaTech and it would be preferred if this OpenTherm command and any further commands could be sent by Matlab or Simulink. In the end the technician that helped me previously also helped me to make a protocol that could be used by Matlab.

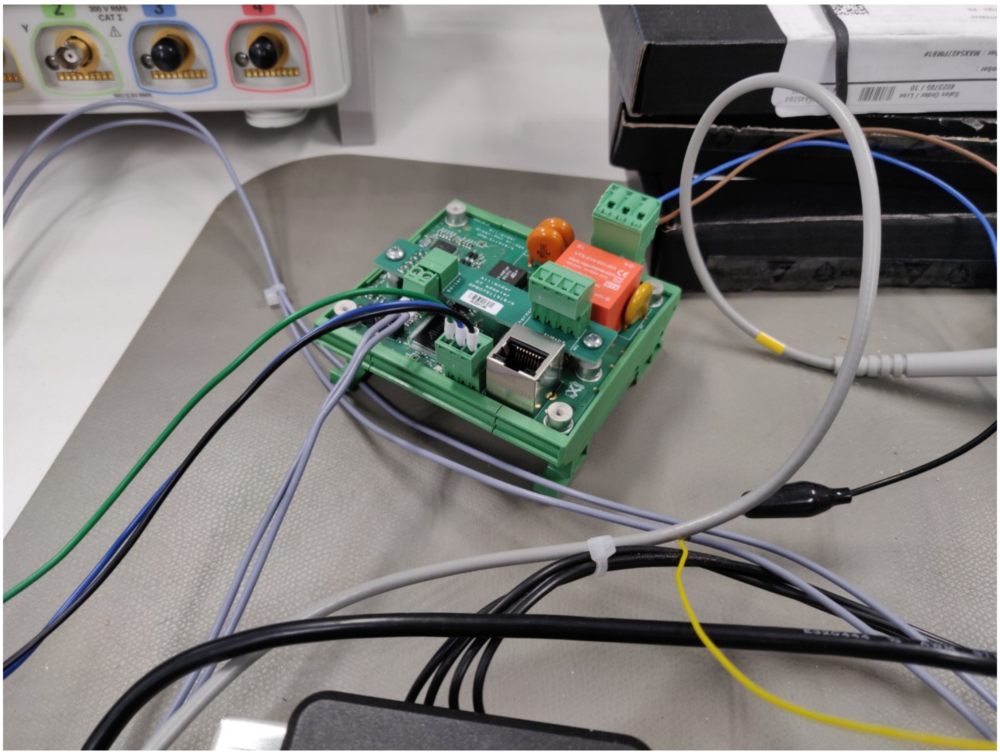


Figure 4.7: The OpenTherm control box made by ExaTech

Chapter 5

Experiments

In this chapter the experiments done with the model and the test board will be discussed. As told before it was not possible to verify the model which was the assignment in the first place. The assignment changed to do experiments regarding the security measures of the test board and thus of the gas boiler. These experiments were possible to do with a non verified model. First the goal of the experiments will be explained followed by the method. After that the results will be shown and discussed. More than one experiment has been done because of the results of previous experiments. Therefore the experiments will be explained in chronological order. The final experiment was done to see if the OpenTherm commands regarding the power can be used in combination with the OpenTherm control box. This experiment will be described at the end of this section.

5.1 Goal of the experiments

For ISR to work as good as possible it is required that gas boilers reach a high temperature difference at high working temperatures. For example the supply temperature is 80 degrees Celsius and the return temperature is 40 degrees Celsius. Most gas boiler have a security measures that prevents high temperature differences. Intergas claimed that their gas boiler can reach this temperature differences at high working temperatures. This is due their new heat exchanger design. The goal of the experiments is to see if the boiler can indeed reach the high temperature difference or that the test board will prevent this.

5.2 Research Method

For the first experiment the set temperature is set at 80 degrees Celsius. This temperature is the temperature that the gas boiler wants to reach. This set temperature is normally determined by the thermostat based on room temperature and asked temperature. This time it is set at its maximum by the earlier described ExaTech control box. This is done because of the large temperature difference. When the gas boiler reaches 99 percent of its power, which is its maximum, the return temperature is lowered to 30 and kept at 30. This is done by limiting the maximum temperature that is sent to the test board by Arduino. This limitation is put just before the signal is sent to the Arduino, this way the limit is not affecting the model on the computer. This is done so the model can still continue to calculate the temperature increase each second normally. Since the values of the model now do not correspond anymore with the values that are being send to the test board the IDS Intergas software needs to be used. By using this software all the values the test board receives can be seen as well as the supply temperature that the test board calculates.

In figure 5.1 the interface of the IDS can be seen. For this experiment the goal was to keep S2 at 30 Celsius degrees while Temp. aanvoer, which translates to supply temperature, needs to be at least 70 degrees Celsius but if it could go higher it would be even better. So for the first run it was tested what the maximum temperature difference could be.

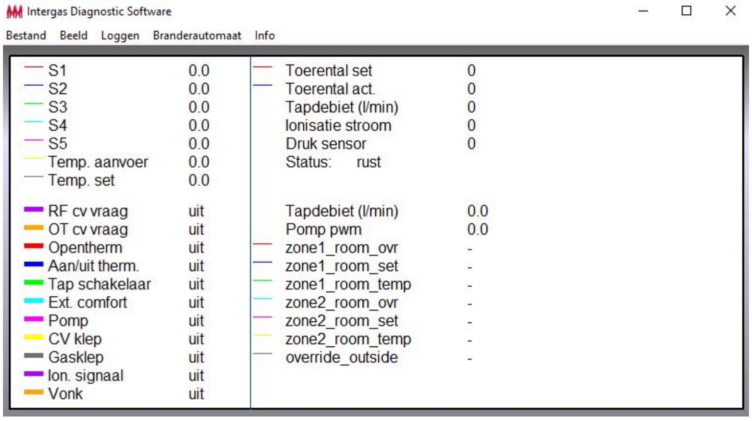


Figure 5.1: The Intergas Diagnostics Software (IDS)

5.3 Measurements and Results

The experiment started with letting the test board boiler go to maximum power which is 99 percent. When this maximum power was reached the return temperature was lowered to 30 degrees Celsius in the model and thus also send to the test board. Since the supply temperature is a function of S1 and S2 this one lowered as well. So the return temperature was set at 30 degrees Celsius and now the test board was left alone. The S1 value still slowly increases and thus the supply temperature as well until any security measures would interfere, if there are any.
When the supply temperature reached a temperature of 70 degrees Celsius there were no interventions by security measures so a temperature difference of 40 degrees was reached without a problem. This went on till a supply temperature of 75 degrees when the test board turned the burner unit off and went to mode 1 which is the stand-bye mode.

5.4 Conclusion, Discussion and further Research

So based on this first experiment it seems that there are no security measures for a temperature difference of 40 degrees. At a temperature difference of 45 degrees the burner indeed turned itself off. It did not however display an error. This means it is not completely sure if this was done as a protection by the test board or it was the on/off system. The on/off system could be the problem since the temperature difference was very large and the supply temperature was close to the set temperature it could have been that the gas boiler expected there was enough heat in the water to satisfy the asked heat. However when playing around earlier with the test board and testing the model it was found that when the supply temperature is 5 degrees Celsius above the set temperature the test board turns the burner off and it goes to stand-bye mode. So right now the supply temperature was 10 degrees lower than what is normally allowed. This decreases the change it was the on/off system that caused the burner to turn off Lastly it could also still be a security measure which is time related. It could be that a temperature difference of 40, or over 40, is too large but the test board waits a few minutes to see if it restores to an value that is allowed. So it can be concluded that a large temperature difference could be possible but cannot be said for sure. For now it seems at least that a higher temperature difference then 20 seems easily possible. Further experiments need be done to know why the test board did turn off the burner unit. Two researches will be done. During the first research that will be conducted the temperature difference will be kept around 40 degrees. This way it can be checked if there is a time related security measure. If the test board can run for a long time at this temperature difference it can be ruled out that there is a time related security measure. The second experiment that will be conducted is to increase the return temperature and thus the supply temperature as well while keeping the temperature difference around 40 degrees. This could give a conclusion concerning the on/off system.

5.5 Follow-up Research

As a result of the first experiment two new experiments will be done. The goals of these experiments have already been described in the last section and the method of research is mostly the same as has been described in section 5.2. So to prevent a lot of repetition these parts will not be described and instead the measurements and results will be shown.

5.5.1 Measurements and Results

The first thing to do again is let the test board increase the power to its maximum. When that was achieved the return temperature was lowered to 35 degrees Celsius. Furthermore the limit of the supply temperature has been set at 75 degrees Celsius. This is done so the temperature difference is 40 degrees and would be kept at 40 degrees. When the supply temperature reached a temperature of 75 Celsius degree the test board has been left alone to just continue to run. This was done to see if there was a time related security measure. After a long time the return and supply temperature were increased by one degree. So the return temperature was 36 degrees Celsius and the supply temperature 76 degrees Celsius and once again the test board was left alone to run for a long time. This process was repeated till the test board turned itself off at a supply temperature of 78 degrees Celsius.
In figure 5.2 the results can be seen. The first attempt went wrong so the first 1000 seconds should be ignored. After that it can be seen that the temperature difference has been kept at 40 degrees for a long time.

5.5.2 Conclusion, Discussion and further Research

So it can be concluded from this research that there is no security measure based on time. There has been a temperature difference of 40 degrees for a very long time. Still the test board turned of the burner when the supply temperature reached 78 degrees Celsius and not the asked set temperature of 80 degrees Celsius. So since the test board still does not allow the supply temperature to be above 5 degrees Celsius of the set temperature, what it normally does, there still seems that there is something preventing this.
In the end it can be concluded that there is no security measure involving passed time. When a high temperature difference works for the test board it works for a very long time. Further prove for this conclusion is when the combination of temperature difference and supply temperature became to high for the test board. The test board then goes to stand-bye mode immediately. It is unfortunately that this data was not collected and thus this cannot be seen in the figure. What can be seen is that the test board ran fine when the supply temperature was 77.26 degrees Celsius. It however is still not clear how the test board works with high temperature differences. It is now known that a temperature difference of 45 degrees is possible if the supply temperature is 75 degrees Celsius. Furthermore a temperature difference of 40 seems to high when the supply temperature reaches 78 degrees Celsius. For this problem there is another experiment needed. But this time around the way the temperature difference is created will be reversed.

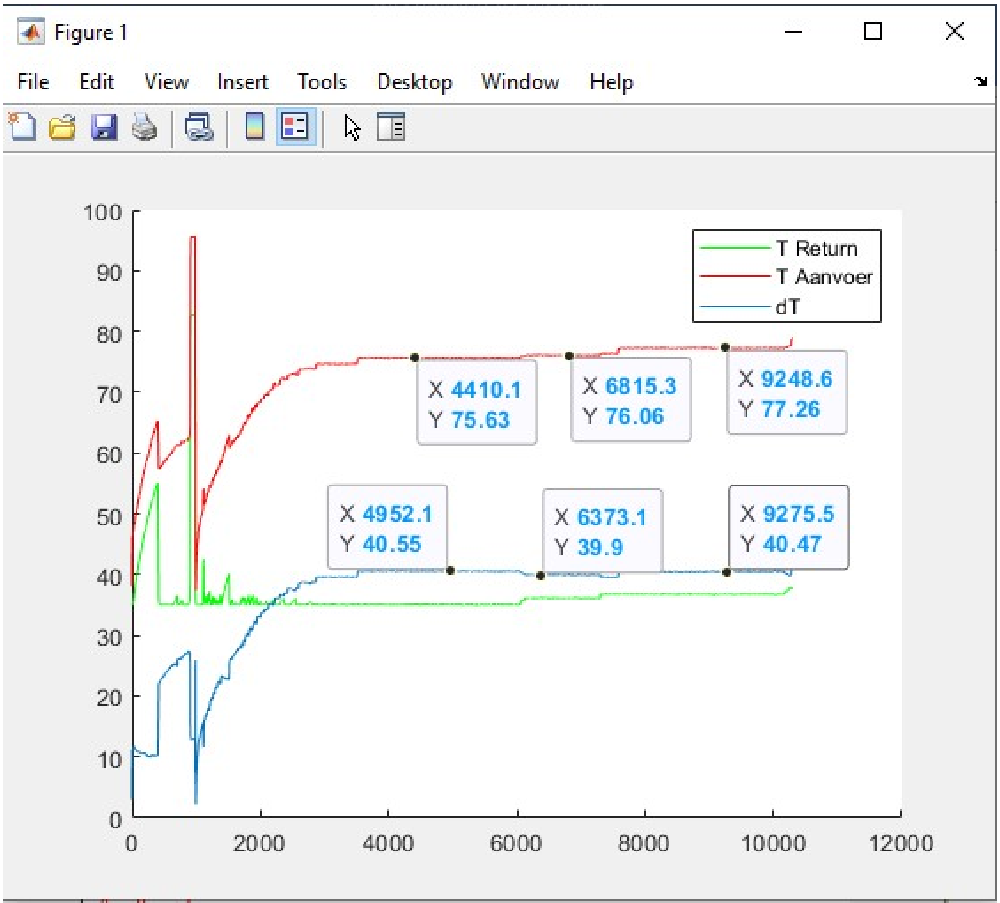


Figure 5.2: The results of the second experiment

5.6 Reversed Experiment

This time the return temperature will not be limited but the supply temperature will. The goal of this experiment is to find how the test board will react to any temperature difference at a high supply temperature. Also doing the experiment this way makes it so that it is no more guessing what the temperature difference could be. This time the exact temperature difference per supply temperature will be known and if a possible relation can be found. Furthermore during this experiment it has been found that the set temperature could be set at 90 degrees Celsius in stead of the earlier 80 degrees Celsius and this will will also be done to reach an even higher work temperature.

5.6.1 Research method

The experiment will mostly be the same again as the previous experiments. But this time the supply temperature will be fixed and then the return temperature will slowly be decreased. Furthermore this time many different supply temperatures will be tried. Every supply temperature will be checked twice, starting at a supply temperature around 89 degrees Celsius. After the two tests the supply temperature will be lowered by 2 degrees. This will be done until the supply temperature is 75 degrees Celsius.

5.6.2 Results

In table 5.1 the results can be viewed. Every value within the table is in degree Celsius. In the first column the supply temperature can be seen. In the end the first tried supply temperature is 89 instead of 90. It was hard to keep it around 90 degrees Celsius. What can be seen in this table in the last column that the temperature difference is way lower than 40 degrees when operating at high temperatures. Also when looking at the temperature differences and the corresponding supply temperatures it seems that there is a linear relation.

Table 5.1: Results of the reversed experiment, values are in degree Celsius

Tsupply	S2 (test 1)	S2 (test 2)	dT
89	64	64	25
87	59	59	28
85	54.8	54.8	30.2
83	49.5	49.4	33.5
81	46.6	46.5	34.4
79	41	41	38
77	36.8	36.8	40.2
75	30.1	30	44.9

In figure 5.3 the graph can be seen of the temperature differences by various supply temperatures. Here it can be seen that the relation seems linear. It is not exactly linear but that could be because of the small sample size per supply temperature.

5.6.3 Conclusion and Discussion

Now it can be concluded that a high temperature difference will not be allowed for every supply temperature. Furthermore this also explains the results of previous experiments. At a return temperature of 30 the temperature difference is indeed 45 but the further the return temperature is increased the lower the allowed temperature difference becomes.

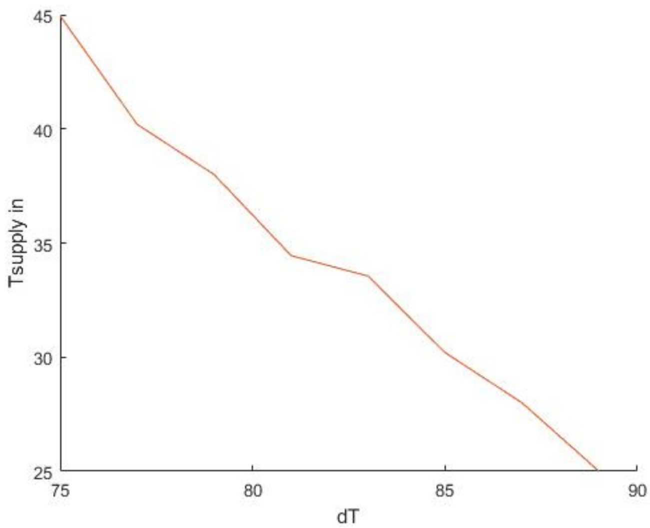


Figure 5.3: The temperature difference by various supply temperatures

Although the line looks a bit like a linear line it is not. There are few reasons that could contribute to this line. One of them is the small sample size, right now only 2 measurements have been done

per supply temperature. Although most of the times the S2 temperature was the same, it still would be more accurate when the sample size would be bigger. Another reason is the big step size of the digital potentiometers. Especially at high temperatures this step size can be around 0.5 degrees Celsius or even higher. That is also one of the reasons most S2 temperatures are the same for any supply temperature. The temperature would jump 0.5 degrees lower and crossed the allowable temperature difference.
Another thing that should be noted is that in this experiment and all the experiments before the temperature difference is between the supply temperature and the S2 temperature, so the temperature of the water at the entrance of the heat exchanger. This means that this water could already have been heated by the flue gasses. The exchange of heat between the flue gasses and water is namely before the heat exchanger that is fueled by the gas. So this means that the temperature difference could be larger than what is discovered here. The return water could already have been heated by the flue gasses and after that, it enters the heat exchanger and gets measured as S2. This difference depends a bit on the return temperature. As said earlier in section 4.2.2 the dew point of flue gasses is around 55 degrees Celsius. If the return temperature is way lower then 55 degrees Celsius a lot of latent heat already heats up the return temperature resulting in a higher S2 value. However if the return temperature is already quite high way, higher then the condensing temperature of the flue gasses, not a lot of heat will be transferred from the flue gasses to the water. Nevertheless the S2 temperature will be higher then the return temperature so the temperature difference between the return temperature and the supply temperature will be higher than what is shown in the table.

5.7 OpenTherm Commands

It is possible to send OpenTherm commands to the test board but it is also possible to receive values from a gas boiler with OpenTherm since some commands will be write and some will be read. Before my internship the company already figured out that it was not possible to use the OpenTherm commands to control the power modulation of the control box, but it might be interesting to use the received values to control the power modulation in the model.
This experiment was a quite simple experiment. What was done is to gather the data of the OpenTherm control box and the Arduino and compare these two signals when the gas boiler is running normally. There was one problem and that was that it was not possible to receive data of the ExaTech control box. So every 5 seconds the value was written down. In figure 5.4 the results can be seen. The red line is the signal that is received by the control box and the blue line is the signal received by the Arduino.

It is immediately clear that the signal received by the control box cannot be used in the model. The red line is always at 100 percent when the burner is on and the power is increasing slowly. The only moment the red line drops is when the boiler is almost at the set temperature and thus slowly reduces the power. That is the moment that the red line also drops quickly.
The problem described earlier was not a big problem in the end. The OpenTherm data is indeed way less accurate but it is clear the signal is just not usable. So the Opentherm control box will not be used to control the power neither will it be used to receive the power level. In the end the best way to receive the power level is the way it is done already by using the Arduino.

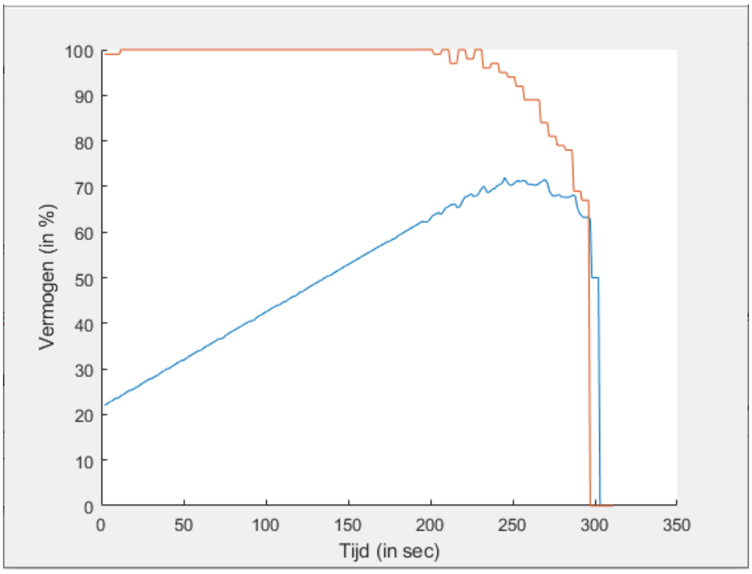


Figure 5.4: The received OpenTherm signal (red) versus the received Arduino signal (blue)

Chapter 6

Conclusion and Recommendations

A few conclusions can be drawn after this assignment. Since all the conclusions of the experiments can be read already in the previous sector they will only be described very shortly here. After a few experiments one conclusion is that the gas boiler does not allow a large temperature difference at high working temperatures. As a safety the gas boiler turns the burner off and goes to stand-bye mode. Furthermore the OpenTherm control box cannot be used to receive the power modulation.

There are also a few things that can be done to improve the model and the experiments as well. These things will be described here as well as a possible solution. The most straightforward recommendation would be that the model needs to be verified. This was not necessary for the experiments that have been conducted with the model but in the end if this model is needed for testing with the ISR it should be verified. The verification can be done when the test setup is installed. This test setup can also be used to improve the model. A few values have just been guessed right now, for example the heat transfer coefficient h. These values can be found using the test setup. But there are also other values like the surface area and the weight of the heat exchanger that have been guessed. A way to know these values is asking Intergas. More communication with Intergas could also be a good way to improve the model since it is their gas boiler and thus they have information about the gas boiler. They can also help with the model assuming they have their own models. This however was not possible in the beginning of the project since Intergas was cautious about giving information. This was mostly because of the secrecy regarding their new heat exchanger. But now that a model has been made it can be shown to Intergas and they might be willing to help with the model.
Something else that could be improved, or actually changed, is the temperature reach of the digital potentiometers. Currently the lowest allowed temperature is around 30 degrees Celsius. This is due the use of only the 10k Ohm digital potentiometer. There are more ways to change the temperature reach. One is to use a potentiometer with a larger resistance, this however lowers the accuracy. The other solution is to change the amount of digital potentiometers. It is not sure if this is possible within Simulink, it might not be possible to use multiple SPI Writeread blocks although a pin can be selected and there are enough pins left on the Arduino. The last way to change the temperature range is to add normal resistors. This way the amount of resistance that can be changed does not change but by adding a base resistence the minimum and maximum temperature decreases. This can also be seen in figure 6.1. If a minimum temperature of 20 degrees Celsius is desired the base load should be at least 4.7k Ohm. This however decreases the maximum temperature to 50 degrees Celsius. So this change should be based on what is the most important goal of the setup at that moment, although the second solution could be one which increases the workable amount of Ohm while not decreases the accuracy. But this one is the hardest one to apply.
Whatever is chosen to change the resistance the look up tables is Simulink should always be calibrated based on the situation. Right now the look-up tables are calibrated for the situation where the minimum resistance is 0 Ohm and the maximum is 10k Ohm. Furthermore the resistance can also be changed between this minimum and maximum, in other words there is no base resistance. So what the look-up tables do right now is to check which value should be send to the Arduino (0-255) based on the temperature and the amount of resistance. So if the model calculates a temperature of 30 degrees Celsius the look-up table has as output a value of 2 or 3 which the Arduino will use to control the digital potentiometer. This value is equal to 9.8k Ohm and thus the digital potentiometer will send this amount of resistance to the test board which recognizes this as 30

degrees Celsius. However when a base load resistance is applied, say 5k Ohm, and the look-up table is not changed it will still have an output of 2 or 3 when the calculated temperature is 30 degrees Celsius. This will result in a resistance of 9.8k send by the potentiometer resulting in a total resistance of 14.8k Ohm resulting in a temperature of 20 degrees Celsius at the test board.

NTC 12kOhm

T [°C]	R [ohm]	T [°C]	R[ohm]	T [°C]	R[ohm]	T [°C]	R[ohm]	T [°C]	R[ohm]
		5	28600	30	9805	55	3863	80	1717
-15	76020	10	22800	35	8055	60	3253	85	1467
-10	58880	15	18300	40	6653	65	2752	90	1266
-5	45950	20	14770	45	5522	70	2337	95	1096
0	36130	25	12000	50	4609	75	1994	100	952

Figure 6.1: Table of the resistance values and the corresponding temperature values

Another problem of the model is that the Arduino cannot handle temperatures lower then 30 degrees Celsius. When going below that temperature the look-up table does not work anymore and sends a incorrect value to the Arduino which will result in a very high temperature. In the end this results in a boiler malfunction since it will think a sensor broke down. Right now the solution is to not allow a value lower then 30 degrees Celsius, however this is not realistic and limits the value of the model when used in combination of ISR. This problem has a lot to do with the previous mentioned increase of resistance and the solutions have also be mentioned already. The best solution to decrease the minimum temperature while also keeping a high temperature range is to increase the amount of workable resistance. Adding a base load will decrease the temperature range immensely when a low minimum temperature is required. This can be seen in figure 6.1, when a minimum temperature of 10 degrees Celsius is desired and the workable amount of resistance is 10k Ohm the base load should be 12.8k Ohm. This means that the maximum temperature is around 21 degrees Celsius and results in a temperature range of 11 degrees. So this is a bad solution at low temperatures and the solution should be increases the workable resistance by multiple 10k Ohm digital potentiometers or one digital potentiometer with a higher resistance.

Bibliography

[1] Intergas Verwarming BV. *Installatievoorschrift Kombi Kompakt HRE.* Intergas Verwarming BV, 2018.

[2] Maxim Integrated Product. Max5487/max5488/max5489 dual, linear-taper, digital potentiometers datasheet, 2010.

[3] O. Holub S. Baldi, T. Le Quang and P. Endel. *Real-time monitoring energy efficiency and performance degradation ofcondensing boilers.* Elsevier, 2017.

Chapter 7

Appendix

7.1 Reflection

One of the things that had to be done before this internship is setting up own personal learning goals. In this part these goals will be shown, discussed and reflected on. These personal goals will be reflected on by the STARL method one by one.

The first learning goal I wrote down is to experience how a project works in a company. What I meant by this goal is that I expect a project works a lot different in the real world compared to the fixed projects at the university. As expected the project was a lot different compared to a project on the university. First of all the amount of information available is a lot less. At the university there are always one or more courses combined with the project so you know already where you should look for information. For my project there was a lot less information. Of course it is known how a gas boiler in general works, but in my case due trade secrecy not a lot of details were available for me. In the beginning this was quite hard for me. I wanted to make the model pretty good already but when I tried to include as much as possible I found out that there were many unknowns all dependent on each other and really struggled what to do with that. Luckily I got help from my product owner who suggested I should start really easy and assume a lot of constants. This resulted in a very easy model but it worked and one by one I could improve some constants if possible. What I learned from this was that starting globally and with a lot of assumptions can be a good way to start a project, especially when the goals are not very detailed. Another thing that was different is that when dependent on other people or companies to deliver something it can take very long. During my internship it was already known in march which sensors we would need for the test setup, however at the end of my internship they just has been ordered. This was due various reasons but nevertheless it resulted in me not being able to verify my model with the test setup. This was not the only thing that was required from an external source. What I learned from this and what has also been told to me a lot during my internship when I needed to search contact is that you always start with contacting external sources. So if you get a list of things to do always do the things that require other people to do things first.

The second learning objective I had set as a goal is to proactive search contact with different stakeholders. This was a bit hard to do but nevertheless there are some cases when I had to search contact with people. By far my most important stakeholders were the product owner of the project and my supervisor. In case of my supervisor it was very easy since we started every day with a stand up since we worked with the scrum method. This required me to always tell me what I did last day, what I am going to do this day and what problems I have. The contact with the product owner was at least every two weeks as well since the sprints lasted for two weeks. I also had the same day as the sprint review an ISR meeting where my product owner was also present. Furthermore whenever I delivered a deliverable to my product owner we always called to walk through whatever I mailed to him. Where I needed to proactive search help was with the electrical engineering part of my internship. This was the part where I connected my laptop to the test board with the Arduino, digital potentiometers and first order filters. What I did is mail the person who could help me when he had time to help me. What was explained at the first goal also applied here, first search contact so a session could be planned and then plan the other objectives I can do alone around this session. I think this went fairly well, I planned any session quick and it hardly interfered with my other objectives.

The last goal I had set was to find out what I like to do and what not at a company and if Alliander is a company I would like to work for in the future. This assignment felt like a research, I had to find out how boilers work and specifically the Intergas boiler to make the model. Furthermore in the end with this model I conducted several experiments. I did not enjoy this assignment in general as much as I would have thought. I felt lost quite a few times but one of the reasons was that it was not clear to me what my goal exactly was. Partly was I to blame for this since I did not write a clear description of the assignment. Furthermore I did not have a clear and structured planning. Most of the time my planning was a bit vague. I knew my goals each day and for each sprint of two weeks but I did not organize these as good as I could. I am sure if I do this next time this will help me with the assignment, making it easier and more enjoyable.
However there were also things I enjoyed. I liked learning Simulink. Sometimes it was quite hard. Compared to Matlab I felt there is less information on Google. Luckily I knew some people who were willing to help me with Simulink. Also they would always explain to me what and why they did certain things when I asked them. In the end I am a lot better at Simulink. I also got help with applying different electrical components and in general a little bit electrical engineering, also learning things of electrical engineering of course. That is something I enjoyed, learning things that can help me for the assignment as well as in the future with other assignments. Furthermore I liked the multidisciplinary of this assignment forcing me to learn new things to bring the assignment to a good end.
Lastly the experiments. I liked doing these a bit, I do not mind them but I cannot see myself doing experiments my whole life. However without knowing what the result or answers to my problems would be was interesting. The only real research I did so far was for my Bachelor assignment however that research had already been conducted. So I knew a bit which results to expect. This time that was not the case so everything I found I had to figure out what it meant. Luckily I had some help here but that was quite interesting to do. What I find I should improve is finding good follow up researches. Sometimes I could not think of a good follow up research and needed help from my product owner. This is of course not a problem but I think I should and could improve this.
So to finalize this goal, there were certainly parts I liked about this assignment and I learned a lot from it but I am not sure this is what I want to do after my study. What would help is other work experiences, that way I should be able to make a good choice. Furthermore I am quite sure I would not like to work at Alliander, or more specifically at Qirion which was the department where I did my assignment. The people I talked to did a lot of work in automating processes. This is not something I am interested in although they also made a few heat networks which was one of the reasons I wanted to do an internship at Alliander. But based on what I saw so far I am not confident I would like to work at Alliander.

Analysis of external supervisor at Alliander

One of the things that was for all us new during this project was the use of the scrum method. My supervisor thought that this method worked well and I have to agree. Using the scrum method resulting in short goals every time in the foreseeable future. I think this works well for me since I am not very good at making clear schedules for a long time. This way there was a schedule for every two weeks and the schedule would be updated after that.
Another thing that caught my eye is that my communication was adequate/good. So good enough but not necessarily good. I also agree with this part. I am a bit a reserved person. That is also why one of my goals was to seek out proactively contact with my stakeholders to improve that point. I think that went quite good but I have to agree that I do not say everything all the time. It was mostly during the stand ups that I told what I had done and when I asked questions. Lastly

I got complimented that I tackled electrical problems although this was not my expertise. I think the electrical problems we faced were indeed solved quite good. Before the internship I did not know a lot of the electrical parts I used but it was interesting to learn. To combine the electrical part together with the modelling part and making it all work together was a good experience and also something I expected within a company. To combine multiple disciplines within a project and not like the university project where the focus is always on the courses associated with the project.

New SMART learning objectives for further professional development.

The last part of the reflection will be about stating new SMART learning objectives. Here new measurable goals will be stated.
One of the problems during this assignment was that I was often lost in the beginning. For my graduation assignment I want to make a clear assignment description as well as a planning. I plan on doing this a bit like the scrum method I used during this assignment because one big planning is too hard to make for such a large assignment. That is why I would like to work with sprints again. This way I can make more smaller schedules instead of one big planning.

Furthermore during my internship I had to use Simulink. I have never worked with Simulink before and learning it took some time but I think it can be a very clear and good way to make a model. Therefore I would like to use these skills in at least one part of my graduation project. This is of course only if my project allows it. Also I should not blindly use Simulink, if Matlab or another program would be easier or give better results I should use that.

Something I also did during my internship is keeping a diary. Someone told me to do this since after three months I would definitely forget some things I did. In the end I am happy I did this since it helped for sure. So that is why a goal of me is to do this again during my master thesis. I will not do this every day like I almost did during my internship since nine months is a long time to write something every day. That is why I want to write at least at the end of every week something in my diary. If needed I can of course write more often. This will mostly be needed when I conduct experiments. Keeping a diary is for me a good way to remember some thoughts I had about the experiments.

Lastly I want to improve my way of making reports mostly focused on the timing. During my internship I had to write quite some reports. I find writing reports one of the most boring parts of doing research however a good report is very important. A good report gives a nice overview of what you did during the experiments as well as the results. But during my internship I delayed sometimes this part of the experiments, already starting with a new experiment before finishing the report part of the last experiment. Sometimes this resulted in not completely remembering correctly what I did during an experiment. This is of course not desired and therefore during my thesis I want to to this the right way by writing an report immediately after the experiments.

7.2 Simulink model

In this section all the important parts of the simulink model will be explained. First the boiler subsystem will be explained. Since the Arduino subsystems have already been described in sector 4.2.1 it will not be done again here. The Radiator and controller subsystems will be shown and briefly explained after the boiler subsystem.

7.2.1 Boiler Subsystem

In figure 7.2 the whole boiler subsystem can be seen. In this subsystem there are a lot of smaller subsystems each calculating a different part. On the left side the inputs are shown. These inputs are the power of the boiler, which is now a constant at 99 per cent, however can be connected to the Arduino when desired. The return temperature is the other input starting at 30 degrees Celsius. This is done because right now as explained the Arduino cannot handle temperatures below 30 degrees Celsius.

The first subsystem is the Berekening Qgas subsystem. In this subsystem The amount of energy within the gas in KW will be calcualted based on the temperature of the return water and the power of the gas boiler. This will be done with the following equation:

$$Q_{gas} = \Delta H_c^\circ \cdot \dot{m}_{gas} \cdot \eta \tag{7.1}$$

In this equation the ΔH_c° is the heat of combustion in kJ/m^3, the $\dot{m}_{gas}$ is the mass flow of gas in m^3/s and η is the efficiency which was already explained in section 4.2.2. This in the end gives the heat that can be delivered to the water every second.

The subsystem below the previously described subsystem Berekening Qgas is the subsysyem Berekening mwater. In this subsystem the flow of water in m^3/s is calculated based on the power level. It is assumed for simplicity that the pump cannot be controlled and works at the same modulation level as the power of the boiler. This was also found when operating the test board, however in the end it would be desired to operate these separately for the ISR. For now a linear line has been made for the water flow created by the pump. This line is based on the minimum and maximum flow which is known due the install manual. This line now gives the flow of water for every modulation level of the boiler. So it is also assumed that this line is linear, this can be checked when the test setup is working but for now this assumption has been made for simplicity.

The next subsystems all have to do with heating. Temp Ketel is the temperature of the heat exchanger, Qover = Qgas-Qwater is the amount of heat that is left to heat the water. What is meant by this will be explained shortly after. Lastly dT berekenen is the subsystems that determines the change in temperature.
So the way this is done is that the amount of heat per second is known from the previous subsystems. But since the heat exchanger also heats up this heat will not be solely used to heat up the water. Furthermore the water needs to heat up every time since the hot water loses it heat at the radiators and returns as cold water. The heat exchanger does not lose its heat that easily and it is assumed the only loss the heat exchager has is the convection to the outside world. For simplicity it is assumed the heat exchanger does not lose its heat to the cold water, instead the cold water is first heated to the same temperature of the heat exchanger and after that the remaining heat is calculated. From the remaining heat the $Q_{convection}$ is subtracted. If there is any heat remaining it will be used to heat the water and the heat exchanger together. This way the temperature of the boiler and the Tflow is determined.

The next subsystem is the Tflow Berekenen subsystem. In this subsystem the temperature of the flow is determined, so this is the temperature for S1 inside the heat exchanger. In this subsystem there are a lof of if statements, based on if the burner is on and if the pump is on and the different combinations. Right now three of the four if statements result in the same value for the temperature. In an earlier version of the model there were four different cases but it was changed. The four if statements are however kept, this is done since it might change in the future and right now the model is not influenced by these statements. The subsystem can be seen in figure 7.1.
Lastly in the boiler subsystem is another look-up table. This table determines the temperature of the supply water based on the flow and return temperature. This table has been based on an test where the test board has run for one hour and different temperatures for S1 and S2 have been tried. The resulting supply temperature as well as S1 and S2 data is used to make a bivariate polynomial where S1 and S2 are the variables. This supply temperature is only calculated for the model itself. This temperature is not send to the test board since the test board calculates this value itself.

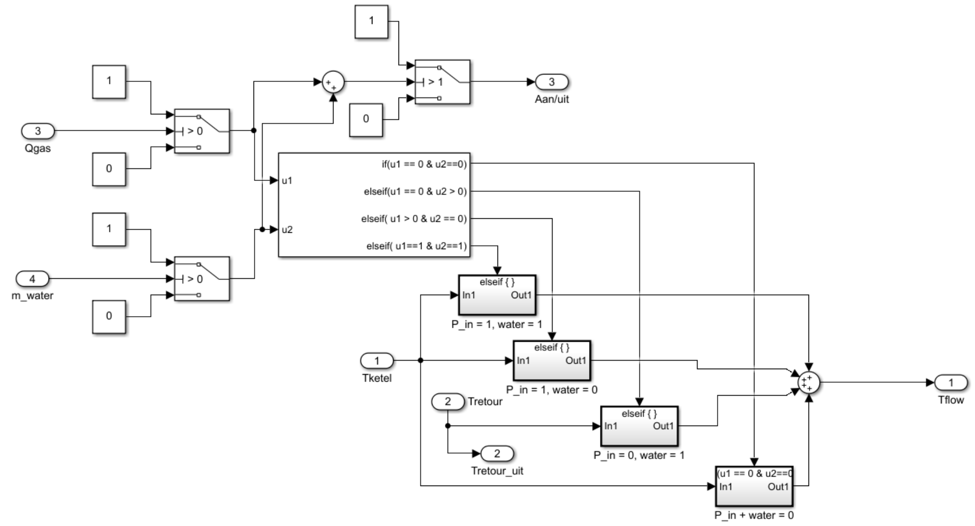


Figure 7.1: The subsystem that calculates Tflow

The radiator subsystem will be very briefly explained. In this subsystem the supply water which temperature is calculated in the boiler subsystem will lose its heat here. This is just a made up percentage which is only based on the supply temperature. If the supply temperature is high the heat loss is larger then when the supply temperature is low.
The last subsystem is the Sensor (beveiliging?) + controller subsystem. This subsystem is used when the model is not connected to the test board and it is still desired the model works like it is connected to the test board. The only two things this subsystem does right now is control the power of the boiler by using a PID controller. Furthermore it has an on/off system. This system checks every 5 minutes if the set temperature of the model is higher then the current supply temperature. The 5 minute timer was found when doing test with the test board. If the set temperature is higher then the supply temperautre this subsystem turns the burner on by using a switch, which is build in the model. If the burner is on the system checks every second if the set temperature is higher then the supply temperature. When the supply temperature becomes 5 degrees Celsius higher then the supply temperature the systems turns the burner off. The 5 degrees Celsius was also found during earlier tests but it was found in the later described experiments that this 5 degrees Celsius is not very accurate. The result of the later experiments have not yet been added to the model since there was not enough time left after the experiments.

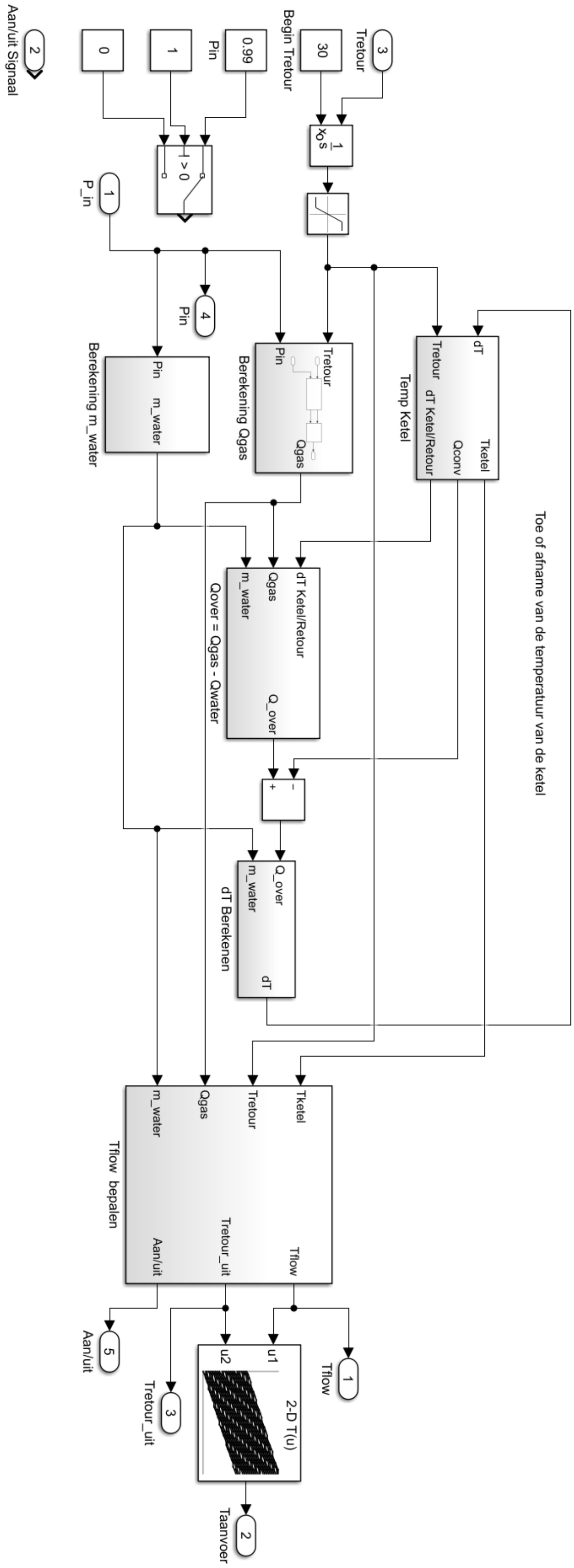


Figure 7.2: The boiler subsystem