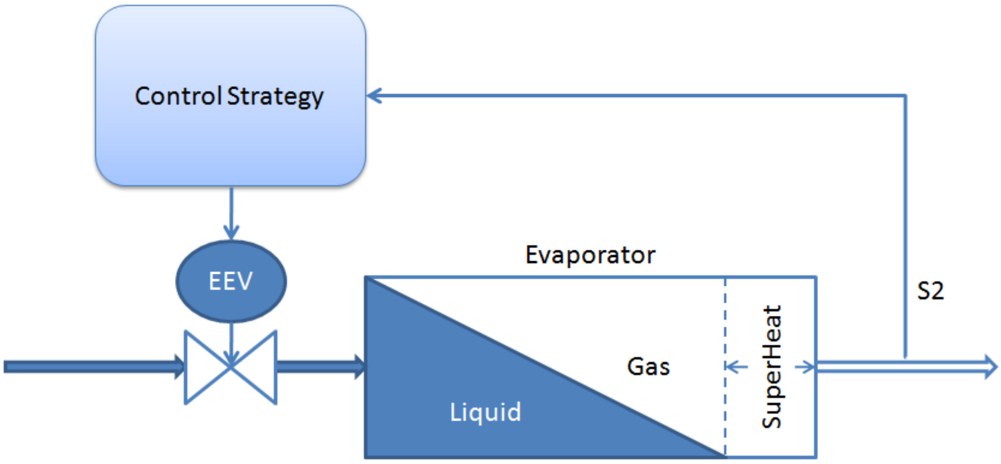
One Sensor Superheat Control - Variance based



Casper Lindholt Andersen

IRS2 - Spring 2011

May 1st – August 12th 2011

Det Teknisk-Naturvidenskabelige Fakultet

Aalborg University Esbjerg

8th semester

TITLE:

One Sensor Superheat Control of a refrigerant system - Variance based

PROJECT PERIOD:

IRS2, Spring 2011
May 1st – August 12th 2011

PROJECT GROUP:

DE6C-2-F11

GRUPPE MEMBER(S):

Casper Lindholt Andersen

SUPERVISORS:

Zhenyu Yang
Roozbeh Izadi-Zamanabadi

NUMBERS PRINTED: 3

NUMBER OF PAGES: 52

ABSTRACT:

The initial phases "ODSweep" and "ODStep" are both sucessfully tested on a live test unit at Danfoss. Where as the last phase "FinalStage" is only tested, sucessfully, in simulation.

"ODSweep" gathers suitable trigger values for the variance, and sets an appropriate initial reference point for the controller. "ODStep" provides the controller parameters, estimated around the operation point of the S2 temperature.

In simulation it is proven that the method is viable, thus proving that using the variance of the evaporator's output temperature and common knowledge about refrigeration system behaviour, is enough to control the SuperHeat, using the expansion valve, and optimize the refrigerant flow through the evaporator.

Preface

This report is made by Casper Lindholt Andersen, IRS-F10, at Aalborg University Esbjerg, in co-operation with Danfoss A/S AC-C Department, as documentation for the IRS2 project at 8th semester, Spring 2011.

The project theme is variance based control and the problem considered is "'Develop a strategy to control the Superheat of a refrigerant system, using only one sensor with variance as indication of Superheat level"'

The report's audience is mentors and other peers at IRS line. The report is written in an understandable language to ensure that the reader do not require considerable technical background. Some words however, will be explained by footnotes. The project meets the criteria which are listed in the curriculum.

The report is made based on my work at Danfoss in the period from the 6th of June to the 1st of July. During my employment at Danfoss I managed to create the algorithm and perform some tests on the live RAC unit. The tests coveres the inital phases "ODSweep" and "ODStep". As for the last phase "FinalStage", I managed to get there but with some controller issues. The controller was tested in a standalone code proving that it worked. Afterwards I recieved a simulink model, in this "ODSweep" and "FinalStage" has been tested, to provide results shwoing what the idea of a variance based refererence control is capable of. Chapter 3 will cover results collected on the live system and chapter 4 will cover the results gained through the simulink model.

The report refers to the sources and character references by entering the section with [#]. Information on sources can be found by lookup in the source list at the back of report.

This report includes a CD-ROM containing source code for the software, scientific articles containing the methods used, tests and simulations graphs and the report itself in PDF format.

Casper Lindholt Andersen ____________________

Contents

Introduction

Coolers come in many shapes and sizes, produced by many different manufactures each model with their own specific setup of refrigerant cycle. In supermarkets 2 main types of coolers are "Freezers" and "Display Case Coolers". These are in a one-to-many or many-to-many setup, meaning one or fewer (less) compressor(s) may be connected to several (more) cooling units (evaporators). In this report the target refrigerant systems is a Residential Air Condition (RAC) unit, which is a one-to-one setup, meaning one compressor is connected to one cooling unit (evaporator). Coolers in general use a lot of electricity so optimizing how they work is a good investment for any supermarket. A typical setup of the RAC unit is shown in figure 1.

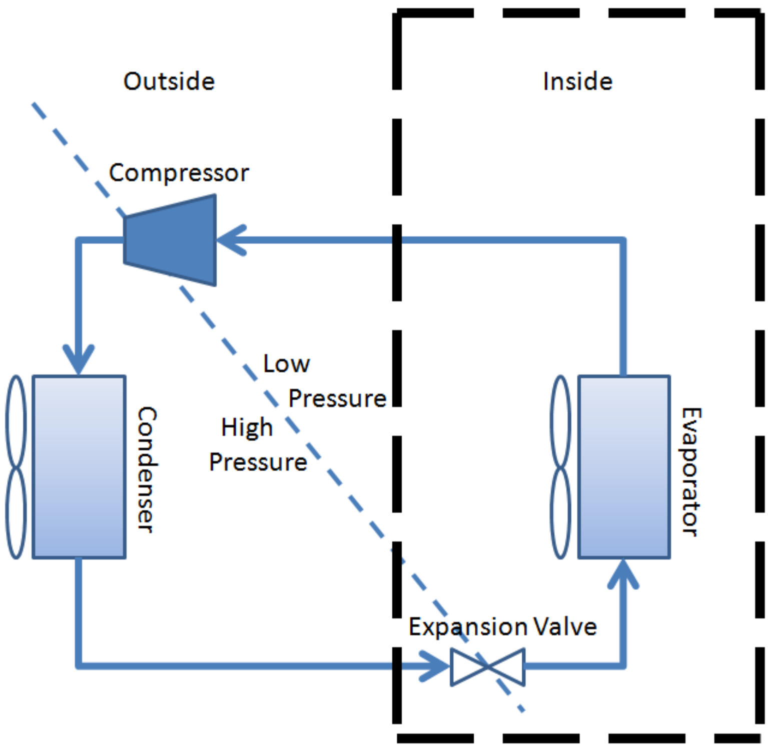


Figure 1: Illustration showing the standard setup of a Residential Air Condition.

The idea of controlling the efficiency of the evaporator is not new, in fact it's been done mechanically for some time, using a so called "Thermostatic Expansion Valve" (TXV). The TXV is actually a mechanical control loop using 3 pressures (P1, P2 and P3) as input, and an OD of the Expansion Valve as output. P1 is bulp pressure, P2 is evaporation pressure and P3 is spring pressure, P1 acts on one side of the diaphragm increasing the OD, P2 and P3 acts on the other side of the diaphragm, decreasing the OD. The bulp is placed on the outlet tube of the evaporator to indicate the pressure after evaporation. P1 and P2 is for stability while P3 is to regulate the actual SH [1].

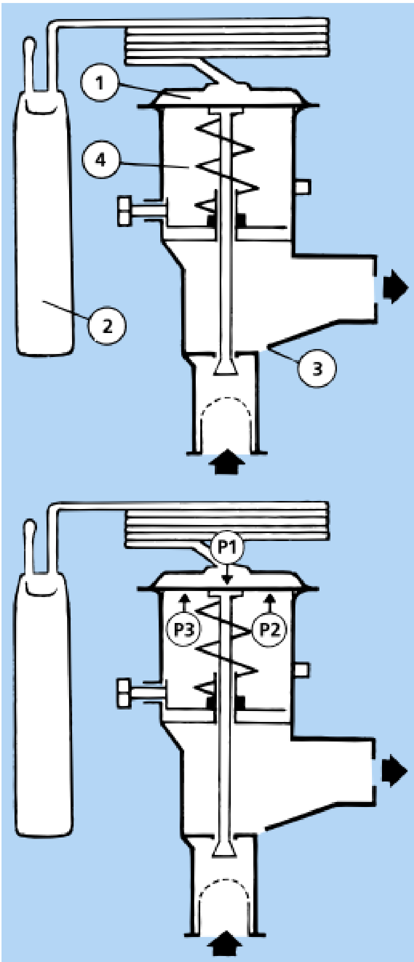


Figure 2: Overview of the Thermostatic Expansion Valve. 1: Thermostatic element, 2: bulp, 3: valve body, 4: spring. P1: bulp pressure, P2: evaporation pressure, P3: spring pressure.

The TXV is not a universal solution, different models exist for different systems and the spring geneally needs manualy finetuning after installation. The TXVs are cheap and reliable but they offer only one mode of control.

The focus of the project is to optimize the efficiency of a refrigerant system, by automatically constructing a PI controller to control the refrigerant flow, through the evaporator. Normally this is done, using 2 sensors, but in this case, only the outlet temperature sensor will be used. The motivation for using one sensor only, is to reduce the production cost of the unit, but alternatively also to provide a safety algorithm, able to take over in case the pressure sensor fails.

Chapter 1

Problem statement

1.1 Overview

The PI controller will be controlling the evaporator outlet temperature (S2), using an Electronic Expansion Valve's (EEV) Opening Degree (OD, 0-100%), to minimize the SuperHeat temperature (SH). SH is defined as the difference between the input(To) and output(S2) temperature of the refrigerant going through the evaporator. The SH is an indication of refrigerant overheat, or in other words, how much the refrigerant is heated past the state of it's dew point. The idea behind controlling the SH is then to maximize the amount of refrigeratn flowing through the evaporator, to increase efficiency of the cooling unit. It is however, really important that all refrigerant is vaporized, since liquid refrigerant entering the compressor unit could potentially damage it.

Figure 1.1 illustrates the sensors used on a typical evaporator in order to obtain SH measurement. The To sensor is a pressure sensor placed inside the evaporator, close to the inlet tube, which, based on the current pressure and knowledge of refrigerant type, estimates the temperature. The S2 sensor is a temperature sensor placed on the outlet tube of the evaporator. If they show the same temperature, refrigerant is flowing straight throught the evaporator without being vaporized (a flooded evaporator), and the SH will then show 0 degrees, the idea is to keep the SH as close to 0 degrees as possible, but without actually flooding the evaporator, since this can potentially damage compressors.

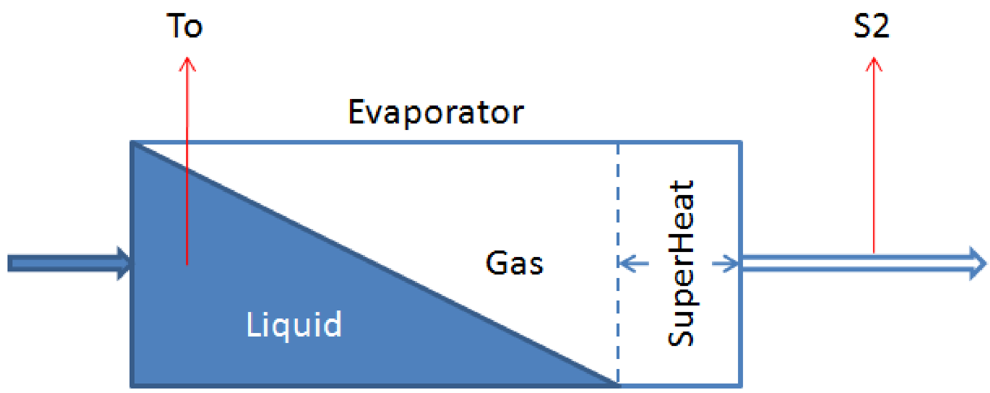


Figure 1.1: Illustration of an evaporator with normal sensor placement

Figure 1.2 illustrates the desired setup. When using the S2 sensor only, the SH is unknown. This makes it necessary to get a good understading, of the charactaristics a low SH will show on the S2 reading. Using this knowledge will allow the algorithm to control the S2 value, corresponding to a low SH.

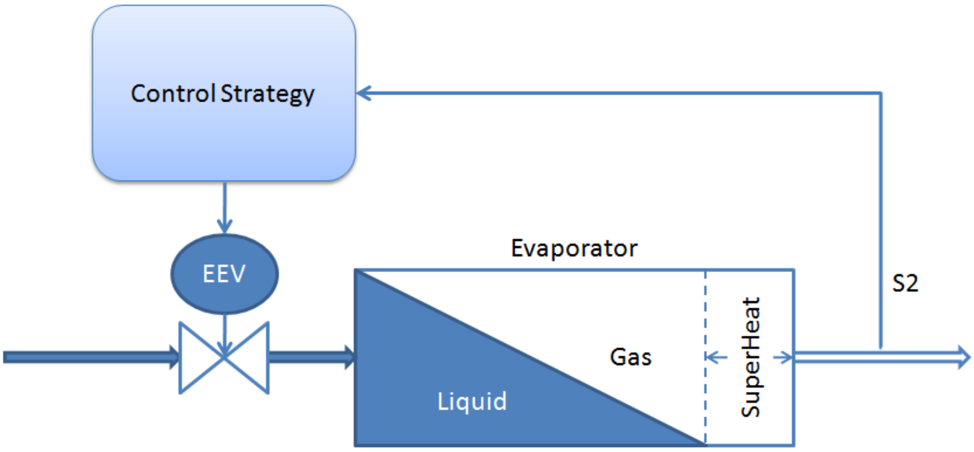


Figure 1.2: Illustration of an evaporator with only one sensor

1.2 Challenges

The main safety concern, when optimizing the refrigerant flow through the evaporator, is to avoid flooding it, as this can potentially damage the compressor. Normal SH controllers use a minimum allowed SH, to trigger a safety step in the algorithm, this general approach is illustrated in figure 1.3.

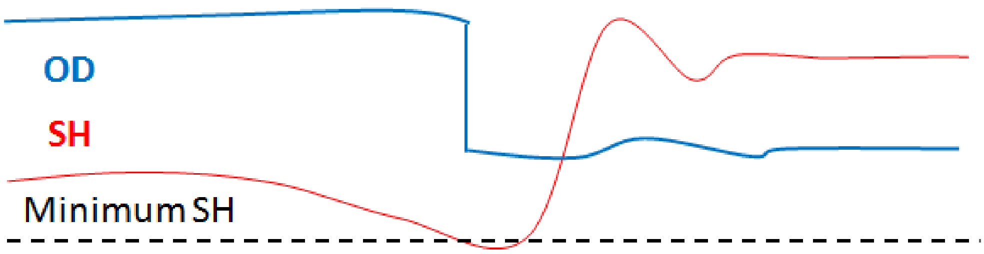


Figure 1.3: Low Superheat safety trigger that decreases the opening degree of the valve, in case the superheat becomes too low

Since the S2 sensor does not provide all information on the current level of superheat, another challenge is to be able to identify a low superheat, based on the behaviour of the S2 readings. Being able to identify a low superheat, based on the behaviour of S2 at one point in time, is a good start. However, considering that an air condition unit has different working conditions (most significantly night and day), it's likely that system behaviour will vary. If the algorithm only estimates a suitable S2 for the initial reference, then it might fail to deliver a proper result when the working conditions change. To overcome this challenge, the algorithm has to be able to adapt to changes in the working conditions.

- Find charactaristics in the behaviour of S2 that indicates low superheat

- Update the reference according to changes in the working conditions
- Be able to realize that the evaporator is critically close to being flooded, and take the necessary safety actions to counter it

Chapter 2

Variance Based Method

2.1 Introduction

As the SH gets closer to 0, the gain relation between OD and S2 increases, this phenomena is shown in figure 2.1. The figure shows the result of a test, where a stepwise increase of OD (+2.5 every 80 samples) is given to the EEV. Each step is marked with a vertical dashed black line, and at each step the current S2 is marked with a horisontal dotted black line. As seen on the figure the drop in S2 is largest in "'4"', as a result of the OD going from 52.5% to 55%. As OD is increased further, from 55% to 57.5%, there is no drop in S2. The fact that S2 does not drop further at this point, indicates that the evaporator is pushed to it's maximum capacity, in terms of vaporizing the refrigerant.

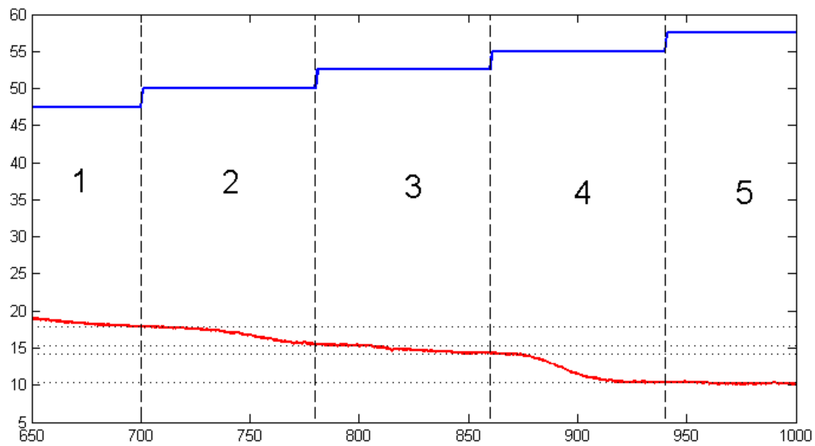


Figure 2.1: Test to show gain relation between OD(Blue) and S2(Red).

With the increased gain relation between OD and S2, a variance calculation based on S2 will show a result like in figure 2.2. As expected it shows a large value, following the highest drop in S2.

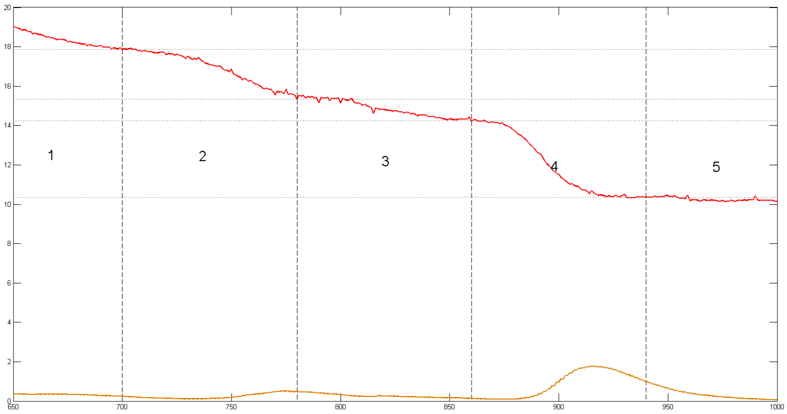


Figure 2.2: Test to show variance according to S2 affected by the stepwise OD increase.

A test is also performed with a continuous increase of OD, the result is show in figure 2.3 (the variance in the plot is multiplied by 10 for a better visual overview). The variance is a bit lower here than with the stepwise increase which is to be expected, but it still shows that the S2 is subject to a larger drop, as the evaporator reaches its maximum capacity.

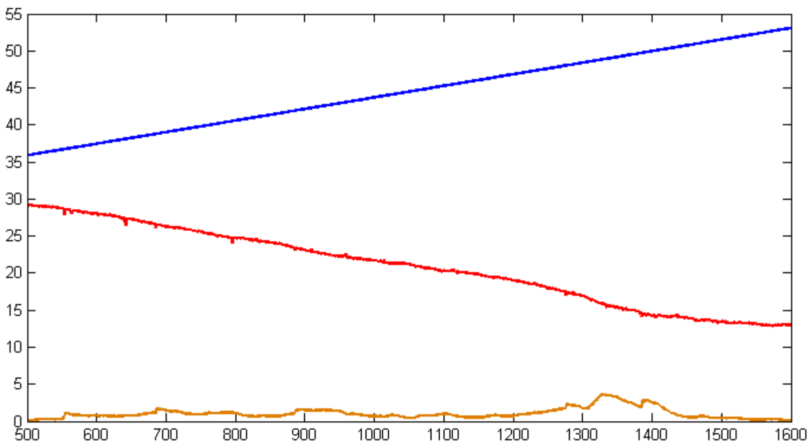


Figure 2.3: Test to show variance according to S2 affected by the continuous OD increase. The variance in this case is multiplied by a factor of 10.

2.2 Theory

2.2.1 Variance

The variance defines how far a value lies from the mean. In this case the mean value is calculated every sample weighted by the parameter "alpha" using this formula:

$$Mean(i) = alpha * Mean(i-1) + (1 - alpha) * S2(i)$$

The variance is then calculated with weightings "alpha" and "beta", using this formula:

$$Variance(i) = beta * Variance(i-1) + \frac{(1-beta)*(1-alpha)}{2} * (S2(i) - Mean(i-1))^2$$

The alpha parameter is set to 0.9, and to find a suitable weighting for beta a small test is performed, figure 2.4 shows variance plotted with 3 different values of beta. Since 0.9 shows a nice large peak just as S2 has it's steapest drop, this value is choosen for beta.

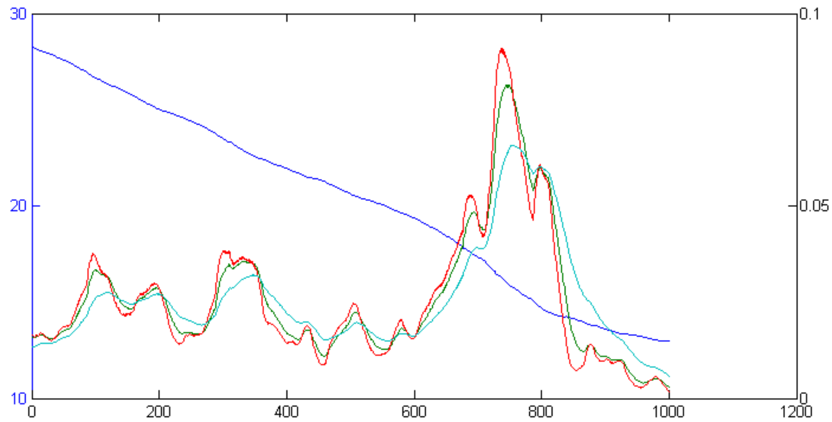


Figure 2.4: Test to find a suitable value for the beta weighting. Red: 0.90, Green: 0.95, Teal: 0.98.

2.2.2 PI Controller

A PI Controller is used to control the S2 temperature, based on a single step respones using "Lag" and "Slope" of the response according to the Ziegler-Nichols tuning principles [2]:

$$OD = OD + kp * (1 + \frac{1}{Ti*SampleTime}) * Error$$

The setup is illustrated in figure 2.5. As seen on the figure, the PI controller takes the Error between S2 and S2ref, while S2 is updated based on the variance reading of S2.

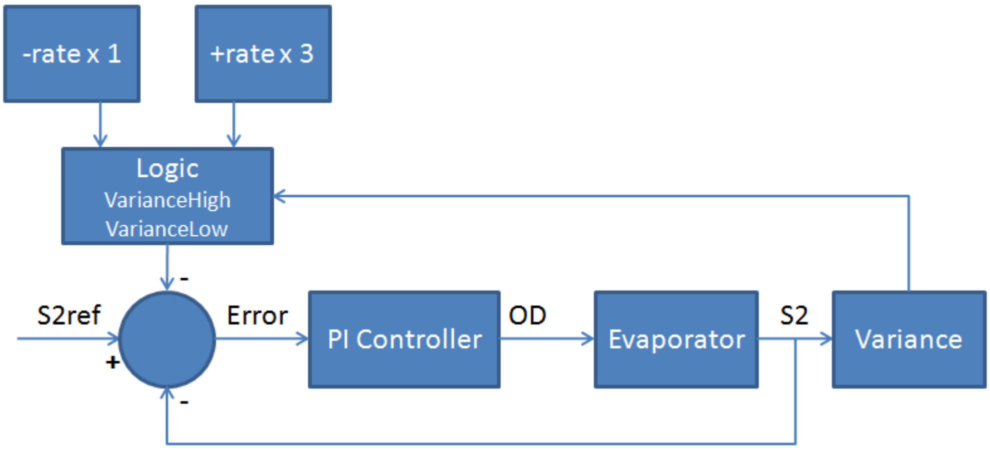


Figure 2.5: Block diagram for the controller setup. The "Logic" block descides, based on variance, if the S2 reference should decrease (slow) or increase (fast).

To calculate proper parameters for kp and Ti, a singel step response is performed once the algorithm recognizes that S2 indicates a low SH. At this point the delay of the response (Lag) is observed, and the Rate of Change (RoC) is being tracked to save it's maximum value (MaxRoC). Once the response is complete, the controller parameters "‘kp"’ and "‘Ti"’ are found. The proportional gain "‘kp"’ is found using MaxRoC and Lag, in this formula:

$$kp = \frac{0.9}{(-MaxROC)*Lag}$$

Using the most extreme value of RoC makes the controller fit the worst case gain relation between OD and S2. The integral time "‘Ti"’ is found using the Lag value in this formula:

$$Ti = Lag/0.3$$

The Step response happens right after the algorithm realizes that the S2 indicates a low SH. This is shown in figure 2.6.

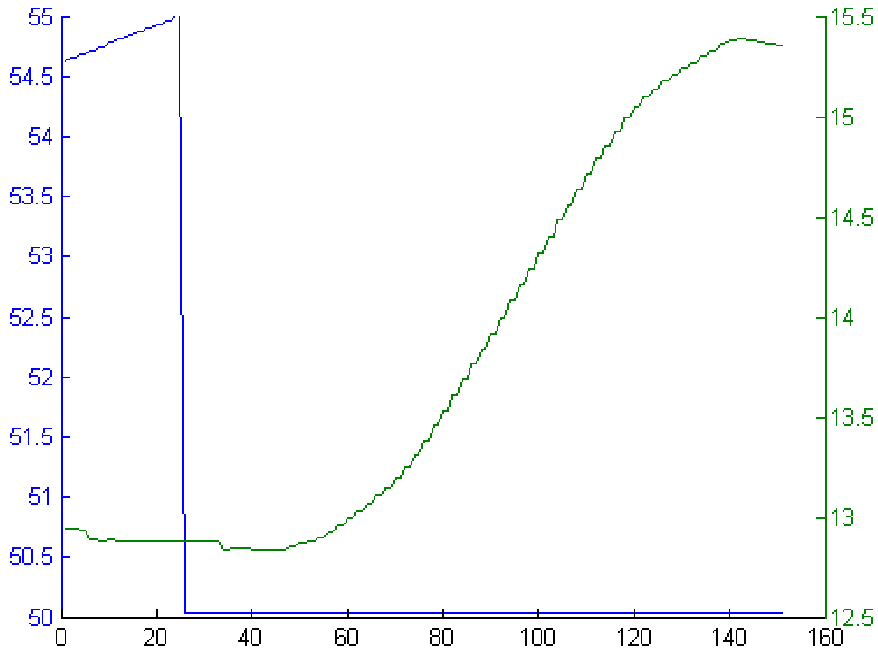


Figure 2.6: Step response of S2, after S2 indicates low SH.

2.2.3 Adaption

One of the main challenges using this method is the fact that the actual SH is unknown. To begin with the estimation is pretty good, but this is an estimation that SH is low at a given S2 at one point. At some other point the refrigerant system might be subject to other working conditions. To provide some sort of adaption, the variance is used to keep track of the system, to indicate if the system is calm or not calm. If the system is controlled by the PI controller and the variance indicates that the S2 is calm, it means theres room for improvement so the reference should be decreased. Likewise, if the variance indicates that the S2 is not calm, it indicates that the controller is pushed to it's maximum, and that the reference should be increased. This simple adaption ensures that the system and controller is always working at optimal settings, even thou working conditions change slightly.

2.3 Code

2.3.1 Algorithm Overview

The algorithm is split in 3 parts "ODSweep", "ODStep" and "FinalStage". These 3 parts serve different purposes, ODSweep and ODStep are initial phases to find a proper working point, variance trigger values and controller parameters. FinalStage is the phase where the controller is running, controlling S2 and correcting the reference based on variance trigger values. A complete overview is illustrated in figure 2.7. As seen on the figure, ODSweep finds a suitable S2 reference and a maximum variance value, ODStep collects Lag and MaxRoC to set controller parameters and in FinalStage the controller takes S2 to the reference and then the reference is changed according to the variance.

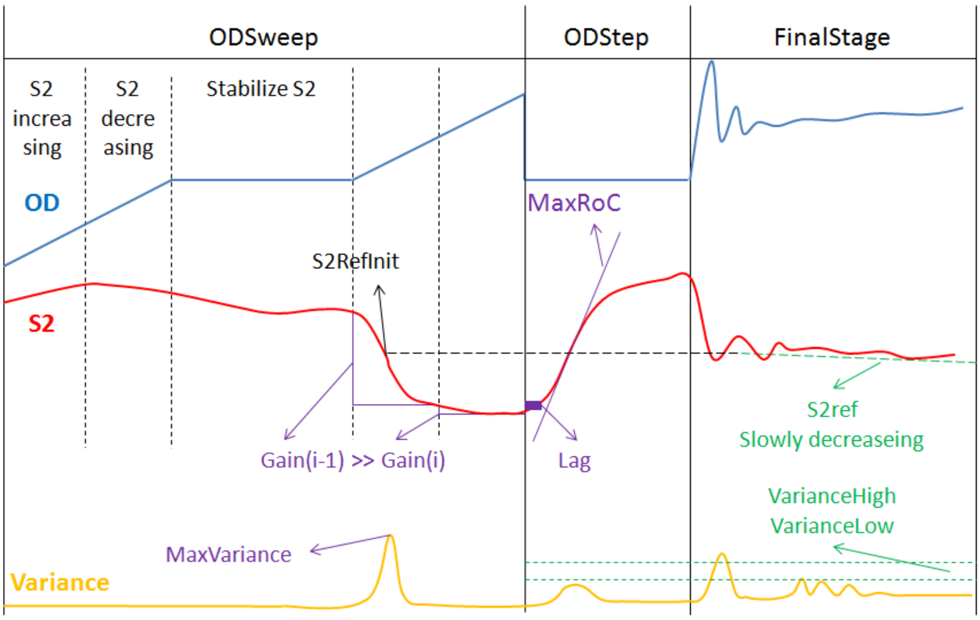


Figure 2.7: Complete overview of all 3 stages of the algorithm.

Furthermore, each stage is drawn in activity diagrams in the back of this report.

2.3.2 ODSweep

ODSweep is basically a sweep of the OD starting at OD=30%, and then increased at a rate of +2.5 over 80 samples. At startup the S2 temperature will increase as a reaction to suction from the compressor, lasting until refrigerant passes through the evaporator and reaches the S2 sensor. Once S2 starts to decrease, the OD is kept stable until S2 shows stability aswell, this is to avoid OD being increased ahead of the corresponding S2 value. After this stability-check the sweep continues, while keeping track of the gain relation (saving the maximum value as maxGain) between OD and S2, and the Variance of S2. At the maximum value of Variance, $S2_{ref}$ (initial S2 controller reference) and MaxVariance is set. Once the gain relation is much lower than maxGain, S2 is considered to reflect a low SH, and at this point the algorithm moves to ODStep. To sum it up, ODSweep collects the following:

- MaxVariance
- VarianceHigh
- VarianceLow
- $S2_{ref}$

2.3.3 ODStep

ODStep consists of a step in the OD of -5, and then the following S2 response. This response is used to set the PI controller parameters "kp" and "Ti", based on Lag (the delay from input to change in output), and RoC (Rate of Change in S2 during response slope). Once S2 has peaked, the algorithm sets "‘kp"’ and "‘Ti"’ values and moves to FinalStage. To sum it up, ODStep collects the following:

- kp
- Ti

2.3.4 FinalStage

FinalStage is the actual control mode. At this point the controller parameters and the initial reference are all set. In this mode the algorithm starts out at a safe S2 reference and waits for variance to show that the S2 reading has calmed down. Once the variance indicates a calm system, the S2 reference is slowly decreased until the variance increases past the VarianceHigh trigger value, indicating that SH is getting close to 0 degrees. This triggers a rapid increase in the S2 reference, to avoid flooding, the increase lasts until variance value drops below VarianceLow, then the S2 reference decreases once again.

When the S2 variance is continuously pushed from calm (variance lower than VarianceLow) to not calm (variance higher than VarianceHigh) it ensures that the system and the controller is constantly working at maximum capacity. However, the system behaviour sometimes changes, making the controller unfit for the system, in this case a safety mechanism is needed, one that will trigger a new estimation of the controller paramenters, to make it fit once again.

2.3.4.1 Safety

The main concern is to avoid the evaporator from flooding (SH $\approx$ 0), since this could potentially bring harm to the compressor. To avoid this, the reference is increased at a faster rate once variance is higher than VarianceHigh. Furthermore, if the variance is constantly higher than variance high, it implies that the controller is unable to control the system at the current level of S2 temperature, thus a step increase of the S2 reference is performed. Everytime this step is executed, the variable "Steps" is increases by 1, should this ever reach 10, the algorithm will realize that the controller simply isn't fit for the system at all, thus triggering a complete re-estimation of the controller parameters.

Chapter 3

Results Live System

3.1 The Live test system setup

The live test setup corresponds to a Residential Air Condition unit. The system is a one-to-one setup, with one compressor and one evaporator. The compressor and the condenser is located outside to release heat, and the evaporator is located inside to absorb heat (thus provide cooling for the inside). The setup is shown in figure 3.1.

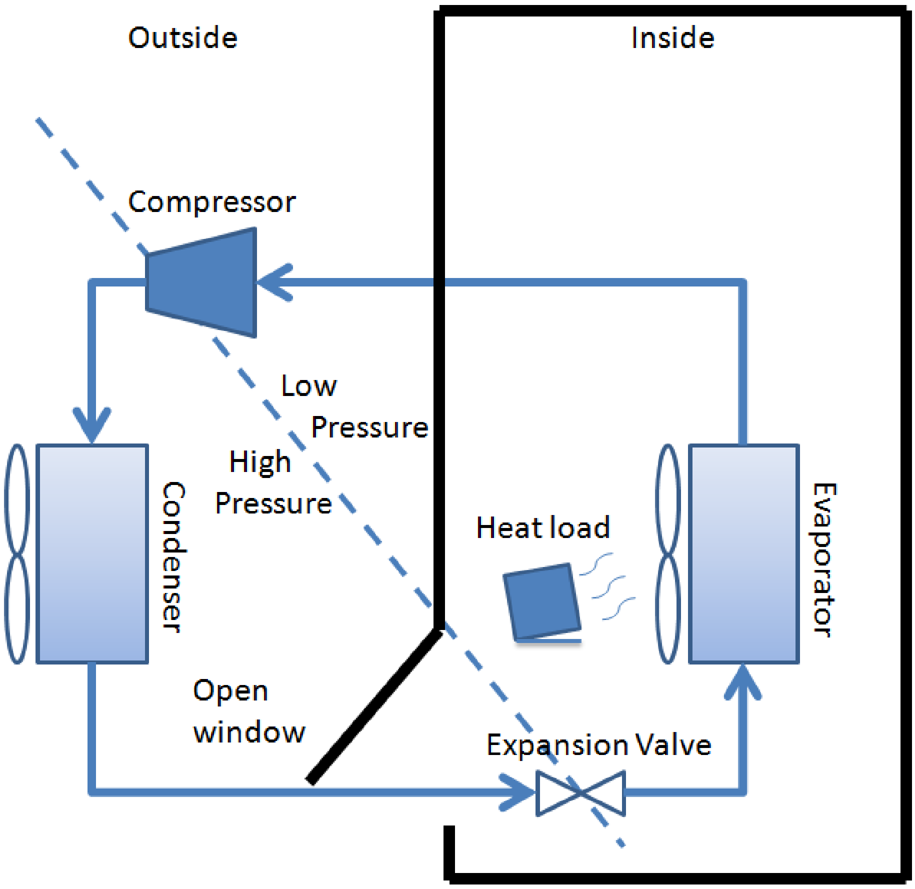


Figure 3.1: Overview of the test setup. The evaporator is located on the insade, with a heater as heat load and an open window to the outside, where the compressor and condenser is located.

3.2 ODSweep (Live) - Obtain Variance

This section will cover the results gathered from the first phase of the algorithm, the ODSweep. The goal here is to find the variance trigger values, VarianceLow and VarianceHigh, and a suitable S2 Reference.

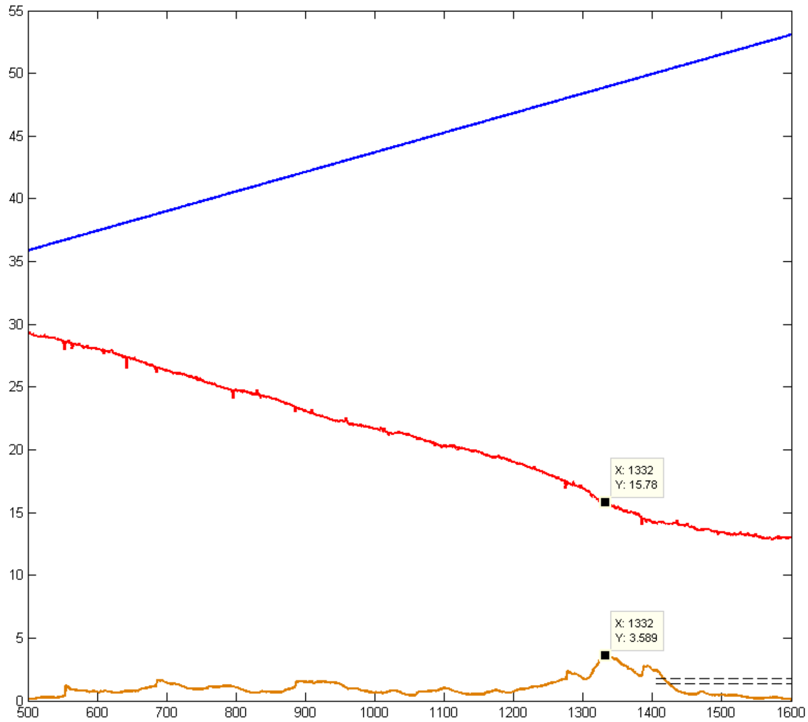


Figure 3.2: ODSweep test result, showing OD(Blue), S2(Red), the VarianceHigh and VarianceLow triggers (Dashed Black) and finally the S2 Reference marked according the the peak variance value. The variance in this case is multiplied by a factor of 10.

Figure 3.2 shows the result of a continuous OD increment during the ODSweep phase of the algorithm. As seen on the plot, the peak in variance occurs at sample 1332, at this time the S2 is at 15.78, which is then used as the initial S2 reference value. Furthermore the maximum variance is 0.358, making VarianceHigh = 0.18125 and VarianceLow = 0.13594.

The result show an obvious increase in the gain relation between OD and S2 at the point where SH is getting close to 0, which is reflected in the variance peak occuring at this time. The maximum variance is collected accuratly by the algorithm, and the triggers and the S2 reference is set, making the ODSweep phase a success.

3.3 ODStep (Live) - Obtain Controller Parameters

This section will cover the results gathered from the second phase of the algorithm, the ODStep. This phase gathers suitable controller parameters to control the refrigerant flow through the evaporator.

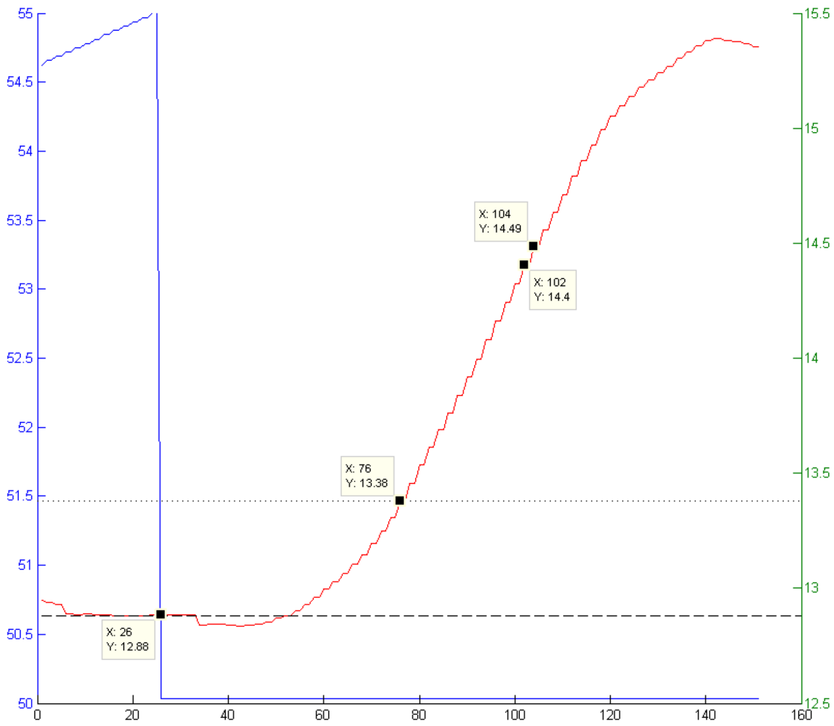


Figure 3.3: ODStep test result, showing OD(Blue), S2(Red), the S2AtOverflow (Dashed Black) with dotted black being the marker to cross to get Lag time, the 2 upper marks indicate the maximum Rate of Change point and the other 2 marks indicate start sample, and the sample where S2 crosses the Lag marker.

Figure 3.3 shows the result of an OD step decrease of -5 in the ODStep phase of the algorithm. From the markers it's observed that Lag is $= 40$ samples, and the maximum RoC is 0.09. Using these two values the algorithm sets the controller parameters, kp and Ti.

The step input to the OD of -5 provides a response of ≈ 2.5 degrees of temperature change in the S2 value, around the operation point, which is a provides a good basis for estimation of controller parameters. Multiple tests placed the parameters of kp and Ti at ≈ 0.3 and ≈ 160 respectively.

3.4 FinalStage (Live) - Controller running

This section will cover the 3rd and final phase of the algorithm, the FinalStage. In this phase the controller is running and the S2 reference is updated according to the behaviour of the variance. However, this change in reference is not suppose to start until the variance shows a calm system, which never happend during the tests on the live system. As seen on figure 3.4 the controller isn't working as intended. Normally the algorithm has S2 reference set as the S2 at maximum variance in "ODSweep", but in this case the reference is set even higher (+2), and the controller still isn't able to control the system. Furthermore the reference tries to smooth out the steep decreases of the S2 value, by having a maximum difference between reference and actual S2 (causing the small bumps in the reference), yet still it fails.

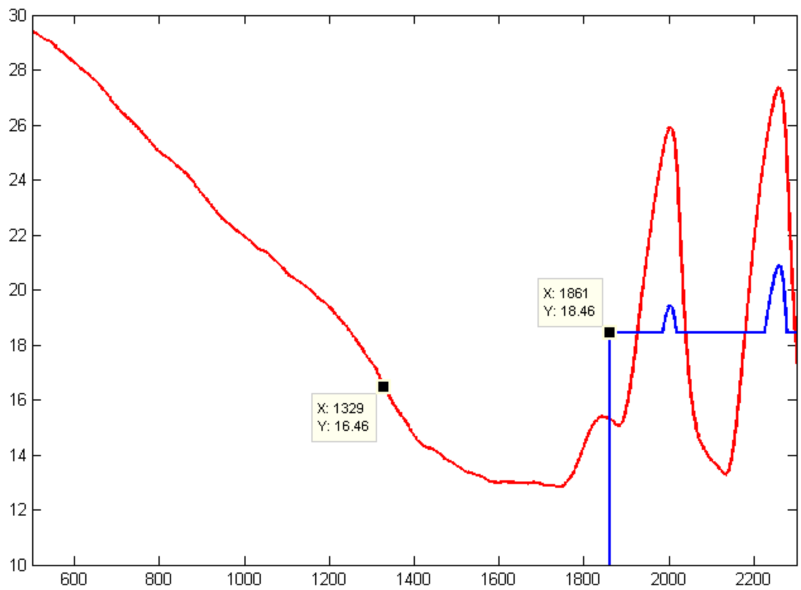


Figure 3.4: This plot shows the unstable controller from running the main algorithm. S2(Red), S2 reference(Blue).

Since the controller wasn't working on the live system, in the main algorithm, the controller is tested in a standalone piece of code, to evaluate if the estimated controller parameters were usefull or not. Figure 3.5 shows this test, from the plot it's seen that the controller does in fact work, it might not be optimal but it works as intended. This was the last test performed on the live Residential Air Condition unit at Danfoss, the test was executed to evaluate if the controller had any chance of working, using the parameters obtained in ODStep.

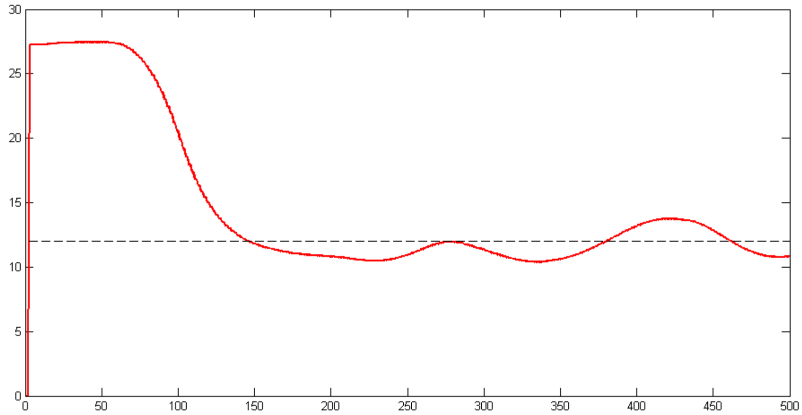


Figure 3.5: This shows the controller being tested in a standalone code that only contains the controller with fixed values and a fixed reference point. S2(Red) and S2 reference(Dashed black).

Chapter 4

Results Simulink Model

4.1 The simlink test system

The simulation setup is made in simulink with focus on a varying gain, increasing as SH gets closer to 0, but with a minimum SH of 0. Standard white noise is also added, comparing the test results in simulation to the real system result, the simulink model provides a good subject for tests of the method. The simulink setup is shown in figure 4.1.

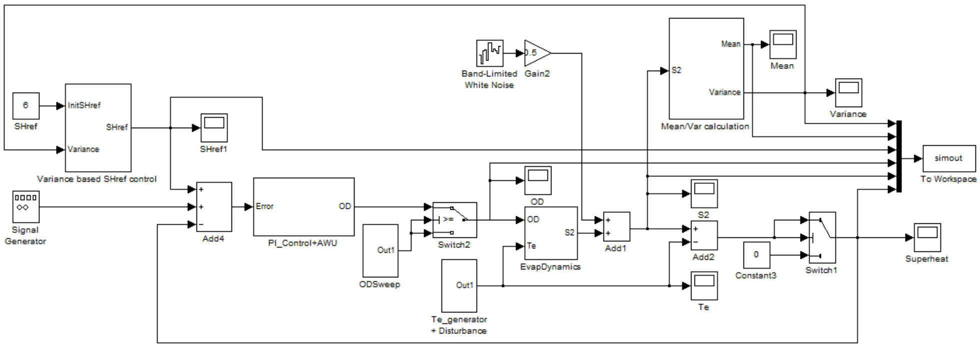


Figure 4.1: Overview of the simulink setup.

4.2 ODSweep (Simulation) - Obtain Variance

This section will cover the results gathered from the first phase of the algorithm, the ODSweep.

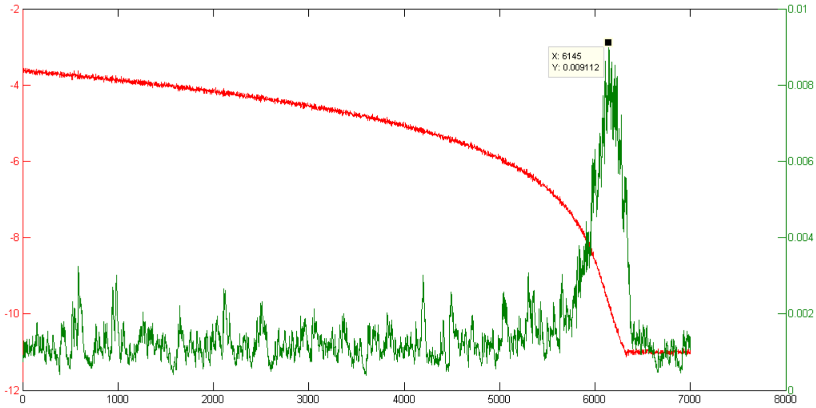


Figure 4.2: ODSweep test result, showing S2(Red), variance(Green) and a marker at the max variance.

Figure 4.2 shows the result of a continuous OD increment during the ODSweep phase of the algorithm. As seen on the plot, the peak in variance occurs at the steepest point of the S2 reading with a value of 0.009112, making VarianceHigh ≈ 0.0045.

Just like the real test setup, the algorithm collects the variance triggers and the S2 reference at a suitable time, making the ODSweep phase a sucess in simulation aswell.

4.3 FinalStage (Simulation) - Controller running

This section will cover the 3rd and final phase of the algorithm, the FinalStage. In this phase the controller is running and the SH reference is updated according to the behaviour of the S2 variance.

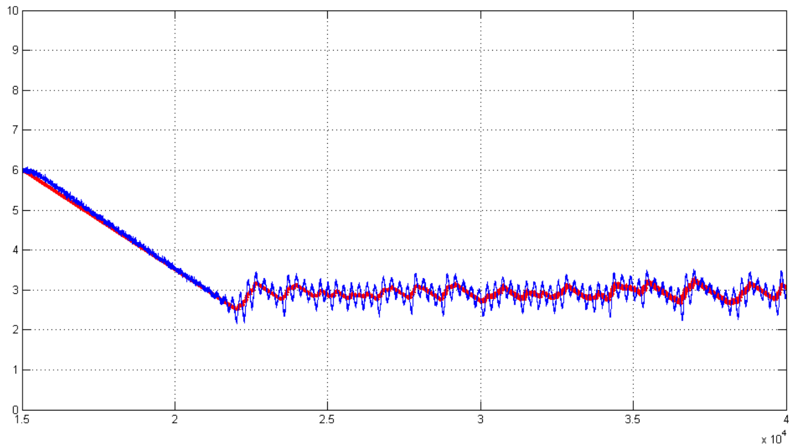


Figure 4.3: Shows SH(Blue) and SHref(red), SHref is changed according to the S2 variance.

Figure 4.3 shows a plot of SH and the SH reference. The SH reference is changed according to the behaviour of S2 using the variance to descide if the reference should be increased or decreased. As seen on the plot, the reference is around 3 degrees of SH.

Chapter 5

Discussion

5.1 Discussion

5.1.1 Utilizing knowledge of system behaviour

The gain relation between the opening degree (OD) of the valve, and the corresponding supearheat and S2 temperatures are nonlinear, meaning as the evaporator fills up the gain gets increasingly larger. This knowledge was used in the "ODSweep" phase, as a slowly increasing OD will result in an increasingly faster decrease of the S2 value, this phenomena is show in section 2.1.

The gain will increase as SH gets closer to 0, and since the minimum value of the SH is 0 degrees, the maximum gain will occur shortly before SH reaches 0. Perfoming a slow, continuous increase of the OD will result in a slowly decreasing SH and S2, and at some point these will decrease faster, and then flat out at SH ≈ 0 and S2 at the corresponding value. The fast decrease will provide a peak in a variance calculation of S2, and the flat reading indicates the flooded evaporator's S2 temperature.

5.1.2 Obtain suitable Controller Parameters

The controller is intended to work around the maximum capacity of the evaporator, so ideally the controller parameters should be set when the S2 temperature indicates a low SH. The "ODStep" starts once S2 has flattend out after the variance peak, making the step response happen just in the middle of the operation point, to provide optimal controller parameters.

5.1.3 Variance based reference update

In the final phase of the algorithm, the controller is running on fixed values with a varying reference. The reference is changed based on variance of the S2 reading. When S2 gets low, variance will increase as a result of the larger gain, this triggers an increase in the reference. Once the reference gets suitably high for the controller to handle, the variance will decrease and trigger a decrease in the reference value once again. Constantly slowly forcing a decrease in the reference value, resulted in a controller that is always pushed to it's limits, and an optimized refrigerant flow through the evaporator.

Chapter 6

Conclusion

6.1 Conclusion

The objective of the assigned project, was to develop a way to optimize refrigerant flow, through the evaporator of a refrigerant system, using only one sensor. Normally two sensors are utilized for this purpose, the more expensive pressure sensor, located at the inlet inside the evaporator, and the less expensive temperature sensor located on the outlet tube of the evaporator. During the project an algorithm was developed to do so, using knowledge of system behaviour during different levels of superheat, and the corresponding outlet temperature (S2).

The first two phases were tested with success on the live system at Danfoss, whereas the last phase has only been working in the simulation afterwards. The first phase, ODSweep, provided an initial S2 reference value, and a variance trigger level based on a slow sweep of the OD starting at 30%. The second phase, ODStep, resulted in controller parameters, that worked fine in a standalone code on the same system.

The last phase included some bug, and didn't work in the main algorithm, however, as stated, the parameters obtained for the controller worked acceptably in a standalone code. As this was the last test performed at Danfoss, the remaining tests to evaluate a changing reference based on S2 variance, has only been tested in simulation.

The changing reference secures an optimal SH for the system and the corresponding controller. As seen in the simulation results, the SH is taken to ≈ 3 degrees, and judging by the variation in the reading it appears to be pushed to it's limits.

Overall the method is approved as viable, with reference and variance trigger values obtained initially, and controller parameters successfully obtained around the operation point. Furthermore the idea of using variance as indication of low SH has been proved to work in simulation, but needs further work to work on the live system.

Chapter 7

Perspective

7.1 Perspective

7.2 Alternative use

As briefly mentioned in the introduction, this method can reduce production costs since it doesn't require the pressure sensor to function. Furthermore the alternative use of this method is to use it as backup on a system that has both sensors installed already, able to take over in case the pressure sensor fails.

Consider a standard refrigeration system running SuperHeat control using both sensors, then suddenly the pressure sensor breaks. Normally this would trigger some sort of static safe mode settig at very poor, but safe, performance. Using the method covered in this project, the system could keep running using only one sensor:

1. System is controlled normally, while a mean value of the valve opening degree is updated

2. Pressure sensor fails

3. Set opening degree equal to mean value minus 10

4. Run the 2 initial phases "ODSweep" and "ODStep"

5. Let "FinalStage" control the system

7.3 Future work

Obviously some work is still needed to make the controller work on the live system. It appeared that some other control within the refrigerant system, was acting against the sudden changes of the controller in the algorithm. To counter this, the "soft descent" as shown in 3.4 needs better fine tuning. As seen on figure 7.1 a properly executed slow descent can take the SuperHeat temperature low without any odd oscilations occuring. The plot shown is from a test running on similar controller parameters as the one used in the standalone test of the controller.

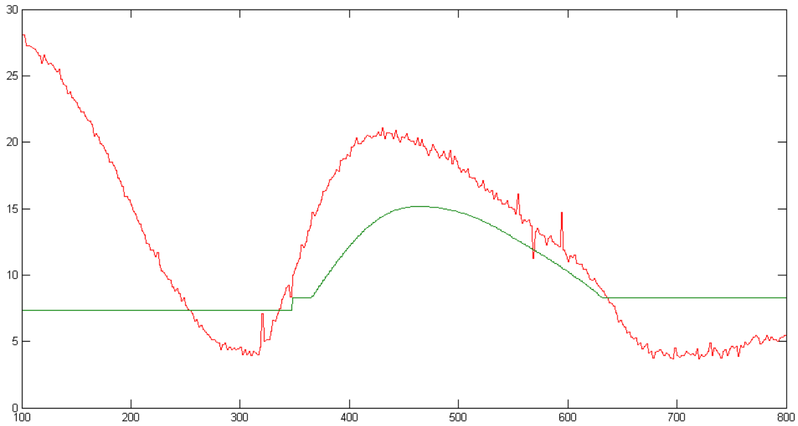


Figure 7.1: Shows SH(Red) and the "slow descent" reference(Green).

List of Figures

Bibliography

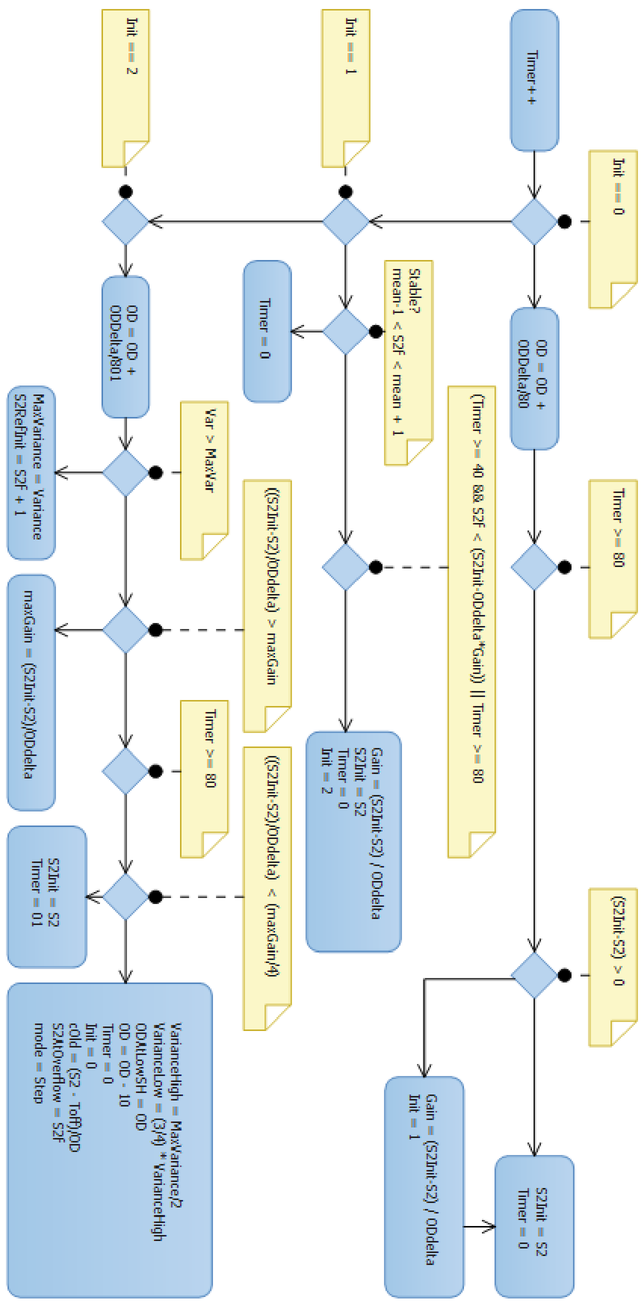
[1] Danfoss. *Thermostatic Expansion Valves*, 2005. See the Reference folder on the CD.

[2] J. G. Ziegler og N. B. Nichols. Optimum Settings for Automatic Controllers. 1942. See the Reference folder on the CD.

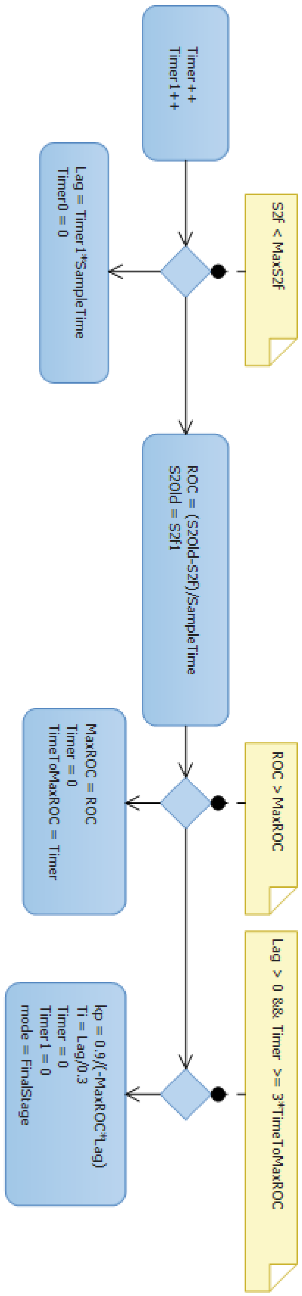
Appendix A

Code models

A.1 ODSweep Model



A.2 ODStep Model



A.3 FinalStage Model