



DEPARTMENT OF ELECTRONIC SYSTEMS

10TH SEMESTER

---

# Robust Control of A Refrigerated Container

---

*Authors:*  
Michael RIMESTAD

*Supervisor:*  
Jakob STOUSTRUP  
Kresten KJÆR SØRENSEN

June 3, 2009



**Title:**

Robust Control of A Refrigerated  
Container

**Theme:**

Intelligent Autonomous Systems

**Project period:**

February 2nd, Spring 2009

**Project group:**

09gr1034

**Members of group:**

Michael Rimestad

**Supervisor:**

Jakob Stoustrup  
Kresten Kjær Sørensen

**Copies:** 4

**Pages:** 65

**Attachment:** One CD

**Printed:** June 2ed, 2009

**Synopsis:**

In this project it is investigated if a model based robust controller can be design with focus on controlling the compressor speed and expansion valve. The aim is to ensure good performance and stability of the evaporator in a set of working conditions for a reefer system. To achieve this a non-linear system model has been used to obtain a set of linear models. From the linear models a parametric uncertainty model has been derived which capture the difference in the linear models. The standard control problem for the robust controller design has been set up and a  $H_\infty$  controller is designed for the system with no uncertainty, to ensure that the problem formulation is sound. This was followed by a  $\mu$  controller design using D-K iterations for the system with uncertainty. The found controller is able to achieve robust stability but not robust performance for the formulated control problem. The controller has been tested on the simulation model and a reefer system. It were found that the controller performed better than the existing as it were able to maintain a more stable temperature and superheat pressure.



# Preface

This report has been written by Michael Rimestad in the 10th semester at the Institute for Electronic Systems, in the period from February 2nd to June 3rd, 2009. This report is handed in for approval as a Master's Thesis at the Department of Intelligent Autonomous Systems.

The author would like to thank his supervisors Jakob Stoustrup from Aalborg University and Kresten Kjær Sørensen from Lodam for their help and guidance throughout the project. In particular a big thanks to Kresten Kjær Sørensen for his help in setting up and running the tests on the reefer container. Also a big thanks to Lodam for making a refrigerated container available for test of the robust controller.

It is expected that the reader has some knowledge about modern control, MatLab<sup>TM</sup> and thermodynamic.

The report is organized in 7 chapters and an appendix. In the following a brief overview of all the chapters will be outlined to give the reader an understanding of the overall structure and contents of the report.

- **Chapter 1, Introduction** Gives an introduction to the project, including setting up the objectives.
- **Chapter 2, Refrigeration System** Description of the Vapor-Compression Cycle upon which the used refrigeration system is based.
- **Chapter 3, System Model** The nonlinear system model is introduced and reduced to fit the problem. The model is trimmed at different operating points, after which the linear time invariant state space models are extracted from the nonlinear system model.
- **Chapter 4, Uncertainty Model** In this chapter the uncertainty model needed for the robust controller design is extracted from the set of linear models.
- **Chapter 5,  $H_\infty$  Controller Design** In this chapter the control problem for the controller design is set up. First a  $H_\infty$  controller synthesis is used to obtain a controller for the nominal plant with no uncertainty to confirm that the control problem is reasonable formulate. This then is followed by a robust controller synthesis using DK-Iterations to obtain a controller for the system with uncertainty.

- 
- **Chapter 6, Test** In this chapter the found robust controller is tested, on the simulated system as well as on the actual reefer system.
  - **Chapter 7, Conclusion** This chapter includes a conclusion on the results in the project.
  - **Chapter 8, Future Perspectives** This chapter includes a discussion on further improvement of the design.

References to source material are indicated with reduced name and the year of publication and when known, the page number. Figures, tables and equations are referred to by the number of the object.

Bibliography and appendix can be found in the last part of the report and the CD can be found on the back page. The appendix contains information to support elements in the report, this includes a Log(pressure)-Enthalpy Diagram of the R134a refrigerant and a diagram of the full simulation model. MatLab<sup>TM</sup> files, source code, test results and electronic available literature, can be found on the CD marked with the relevant path on the CD.

**Project participant:**

---

**Michael Rimestad**

*Aalborg University the 3ed. of June 2009*

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Performance Requirements . . . . .	8
1.2	Robust Control Design . . . . .	9
<b>2</b>	<b>Refrigeration System</b>	<b>11</b>
2.1	The Vapor-Compression Cycle . . . . .	11
<b>3</b>	<b>System Model</b>	<b>15</b>
3.1	Simplify System Model . . . . .	16
3.2	Simulated Inputs . . . . .	17
3.3	Linearization . . . . .	19
3.4	Model Reduction . . . . .	21
<b>4</b>	<b>Uncertainty Model</b>	<b>25</b>
4.1	Parametric Uncertainty Model . . . . .	25
<b>5</b>	<b><math>H_\infty</math> Controller Design</b>	<b>31</b>
5.1	Scaling The Model . . . . .	31
5.2	Robust Controller Design Method . . . . .	33
5.3	Standard Problem Formulation . . . . .	35
5.4	Weights . . . . .	36

5.5	Controller Design . . . . .	39
<b>6</b>	<b>Test</b>	<b>45</b>
6.1	Test Objectives . . . . .	48
6.2	Simulation Test . . . . .	49
6.3	Reefer Test . . . . .	53
6.4	Test Conclusion . . . . .	60
<b>7</b>	<b>Conclusion</b>	<b>63</b>
<b>8</b>	<b>Future Perspectives</b>	<b>65</b>
	<b>Bibliography</b>	<b>67</b>
<b>I</b>	<b>Appendix</b>	<b>69</b>
<b>A</b>	<b>Uncertainty Model - Extrapolation</b>	<b>71</b>
<b>B</b>	<b>R134a Log(pressure)-Enthalpy Diagram</b>	<b>75</b>
<b>C</b>	<b>Full Model Diagram</b>	<b>77</b>

# Introduction

Today refrigeration systems are used in many applications. These applications can vary from the food refrigerators, freezers and air conditioners in homes to bigger systems for industrial buildings such as the freezer display units in the supermarket.

In the field of controlling a refrigeration system the approach used until now has mainly been to use classic controllers. It is believed that by using a model based controller design one might achieve better stability and performance of the system.

If there is a chance that these systems can be controlled in a more efficient and stable way, it could become profitable in the sense of lowering the energy usage of the system. In this project the system used is a Maersk refrigerated container or reefer, used on cargo ships, where the container is equipped with a vapor-compression refrigeration system used to control the temperature inside the container. [Mae08]

In this report it will be examined if it is possible to obtain a more stable and better performing controller by using a model based approach. The model used for designing the model based controller will be made using a modular model presented by Kresten K. Sørensen and Jakob Stoustrup in the paper 'Modular Modelling and Simulation Approach - Applied to Refrigeration Systems'. [SS08]

When controlling a vapor-compression refrigeration system there are primarily four actuators with which to control the system. These are the compressor speed, opening of the expansion valve for the evaporator, the fan speed of the fans on the evaporator unit and the fan speed of the fans on the condenser unit (A description of the vapor-compression refrigeration systems is given in Chapter 2). The difficult actuators to control are the compressor speed and the opening of the expansion valve for the evaporator as the behavior of the evaporator are nonlinear. Therefore it is chosen to focus on controlling the compressor speed and expansion valve to ensure good performance and stability of the evaporator.

A good theory for designing a controller that can guarantee this is robust control theory. Thus it is chosen to use this theory to study if it is possible to give a controller with the desired performance and stability for the system.

The requirements that the controller has to fulfill are outlined in the following.

## 1.1 Performance Requirements

The process of controlling a reefer system can be divided into several tasks. These tasks are

1. Reaching a set temperature at system startup.
2. Reaching a new set temperature due to a change in the reference.
3. Maintaining a temperature. Thus keeping the temperature in the box and reject disturbances.
4. Maintaining the temperature during de-icing of the evaporator to remove ice.

The focus in this project will be on task 3, to maintain a temperature inside the box. Task 1 can be tough to handle as this often requires a great change in the temperature. It will therefore be assumed that the temperature in the box will be around the desired set temperature. It should be able to handle task 2, changes in the reference. In this case changes in the reference will be limited to small changes (less than 5 degrees Celsius). Task 4 will require that the controller can handle large heat loads on the evaporator as the heater in the box are turned on to de-icing the evaporator. This will not be handled in this report.

Maintaining a temperature requires different performance of the system. If the temperature difference between the set temperature in the box and the ambient temperature is small the compressor does not have to work as much as if the difference is large. Thus the amount of work the compressor has to do depends on the temperature difference between the set temperature and ambient temperature. If the work load on the compressor becomes less than a speed of 20Hz the system is defined to go into a Pulse-width modulation(PWM) mode. This is done to protect the compressor against the higher load at the lower speeds and to save energy. But as robust control theory is not designed to handle systems that can go into PWM cycles the working range has to be limited. If one were to handle PWM cycles hybrid control theory would for instance be a better choice.

To avoid this the temperature range in which the controller is applied is therefore limited to when the compressor speed exceeds 20Hz.

System tests have shown that this is the case if the difference between ambient temperature and box temperature is at least 30 degree Celsius. The work range are therefore defined to be :

- Temperature inside the box :  
-5 to -25 degree Celsius
- Temperature outside the box :  
+25 to +55 degree Celsius

In this working range the controller must be able to keep the return air temperature with a precision of  $\pm 0.5$  degree Celsius of the set temperature with changing ambient temperature. The controller also has to keep the superheat in the evaporator to avoid liquids getting into the compressor and keep a good cooling capacity of the evaporator.

Now that the requirements for the controller has been outlined a short review of what robust controller design involves will be given to form the basis for the content of the report.

## 1.2 Robust Control Design

Robust control is a kind of modern control theory in which the uncertainty about the system model together with the disturbances affecting the system are included in the approach to design a controller. Thus trying to ensuring that the controller is able to keep performance and system stability even when the system parameters and disturbances are changing.

To design a robust controller a linear system model which includes the uncertainty of the system is needed. Thus a linear model of the system are needed. This will be derived from the nonlinear system model which as mentioned is provided by Kresten K. Sørensen and Jakob Stoustrup. [SS08]

A description for the system and its dynamics on which the nonlinear system model are build is given in Chapter 2 to describe how the system works. The nonlinear system model will then be introduced in Chapter 3.

Before using the nonlinear model to obtain the linear model it is decided to try to reduce the size of the model without losing important information about the part of the system that are to be controlled. This is done as the algorithms for the later robust controller synthesize increase in algebraic complexity for each extra state and uncertainty perturbation added in the system, it is therefore desirable to use as small a system model as possible. Thus before extracting the linear models the system will be reduced. This is done in Section 3.1.

The approach taken in this project to find the uncertainty model is to linearizing the system model in multiple operating points distribute over the defined work range. From this set of linear models the change in between them can then later be captured in a uncertainty model. To linearize the model the operating point is needed, this and the linearization of the nonlinear models will be done in Section 3.3

As mentioned the algorithms increase in complexity for each extra state and uncertainty perturbation added, and it is therefore examined in Section 3.4 if the linear models can be reduced even further before setting up the uncertainty model.

The obtained reduced linear model is then used in Chapter 4 to set up a uncertainty model of the system which is needed for the robust controller design. The uncertainty model can be set up in two ways, using a parametric uncertainty model or using a unstructured uncertainty model. Only the parametric uncertainty model will be used in this project.

With the uncertainty model setup the control problem defining the controller requirements can be setup and a  $H_\infty$  robust controller can be synthesized. This controller then may or may not

fulfill robust performance and robust stability for the given controller requirements. This is done in Chapter 5.

The last part of the robust controller design is to test the found controller on the simulated system and the real reefer system to verify if the controller performs as expected. This will be done in Chapter 6.

With the basis for the steps taken to design a robust controller for the reefer system and the content of the report outlined a description of the used system can be given. This is done in the following chapter.

## Refrigeration System

As mentioned the system used in this case is a Maersk reefer container [Mae08]. The main purpose of the system is to reach and maintain a given set point temperature within the container which lies below the ambient temperature. The reefer container is able to keep a higher temperature within the container than the outside ambient temperature, but to do this a heater is used instead of the refrigeration system. As this is not part of the project objective, this feature will not be discussed further.

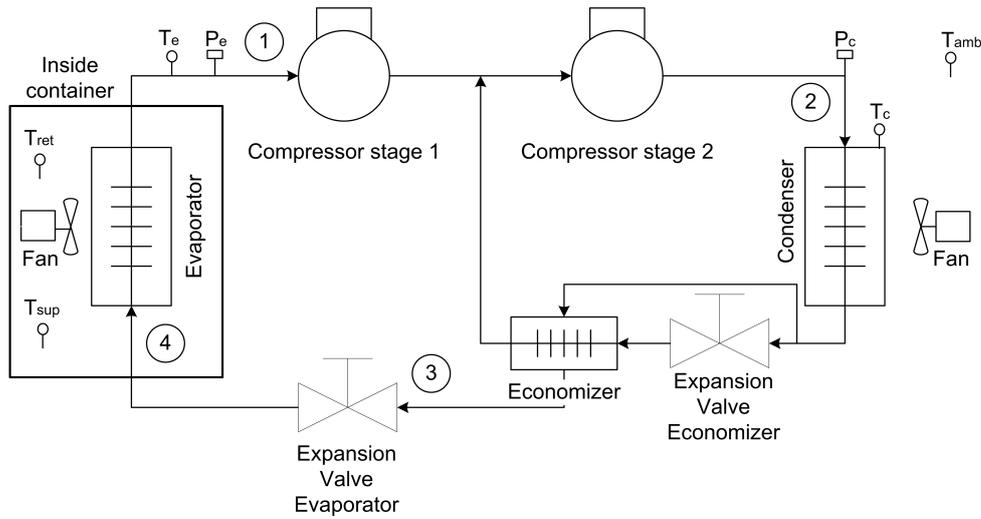
The system is designed mainly to keep a set point temperature in the box, but not cooling the contents inside the box down to the desired temperature. This implies that the cargo inside the container normally already has the desired temperature and therefore does not need to be cooled down, but only kept cold. To get some understanding of how the refrigeration system works, the vapor-compression cycle on which the refrigeration system is based will be described in the following section.

### 2.1 The Vapor-Compression Cycle

The purpose of the vapor-compression cycle is to transfer heat from one space to another. Thus removing heat from the enclosed space inside the box and transfer it to the outside surroundings. This is done by letting a refrigerant circulate between two heat exchangers, an evaporator and a condenser. This is illustrated in Figure 2.1.

In the evaporator the heat is drawn from the air inside the box and in the condenser the heat is rejected to the surrounding air. This process requires that the vapor in the evaporator has a lower temperature ( $T_e$ ) than the temperature in the box ( $T_{box}$ ) and the temperature of the vapor inside the condenser ( $T_c$ ) must have a higher temperature than the ambient temperature ( $T_{amb}$ ). I.e.  $T_e < T_{box}$  and  $T_c > T_a$ .

The refrigerant has the property that the saturation temperature uniquely depends on the pressure. This property is utilized in the vapor-compression cycle by controlling the pressure of the refrigerant, where a low pressure refrigerant has a low saturation temperature and a high



**Figure 2.1:** The layout of the refrigeration system. The circled numbers indicates the stage

pressure refrigerant has a high saturation temperature.

Thus obtaining a low temperature in the evaporator and a high temperature in the condenser can be achieved by controlling the pressure in the evaporator ( $P_e$ ) and in the condenser ( $P_c$ ). This it done with the compressor and expansion valve.

The compressor compresses the low pressure refrigerant from the outlet of the evaporator to a high pressure at the inlet of the condenser. To keep the pressure difference between the condenser and evaporator the expansion valve is installed in between the two to uphold the pressure difference.

The refrigeration systems are also equipped with an economizer which increases the efficiency of the system at high pressure differences between the evaporator and condenser. A further description of this will follow later. [Jon91]

In Figure 2.2 the vapor-compression cycle is sketched in a log(pressure)-Enthalpy diagram. The diagram depends on the used refrigerant which in this case is R134a(A log(pressure)-Enthalpy diagram for this refrigerant are included as Appendix B). The diagram illustrates the steps that the refrigerant goes through and the phase changes that occur. The curve indicates the saturation temperature of the refrigerant. If the refrigerant is in the area on the left of the saturation curve it is in a liquid phase and if it is on the right it is in the gas phase. In the area under the saturation curve the refrigerant is a mixture of liquid and gas. [SBW, p. 435-439].

The normal vapor-compression cycle consists of 4 connected subprocesses which are, stage 1-2 compression, stage 2-3 condensation, stage 3-4 Expansion and stage 4-1 Evaporation. In this case there is an extra stage because of the economizer which is just before stage 3. In the following each stage is described with the economizer last.

### 2.1.1 Compression

In stage 1-2 the refrigerant is in the gas phase. In Figure 2.1 the refrigerant is at the inlet to the compressor where the refrigerant is at low pressure and temperature. When compressed the

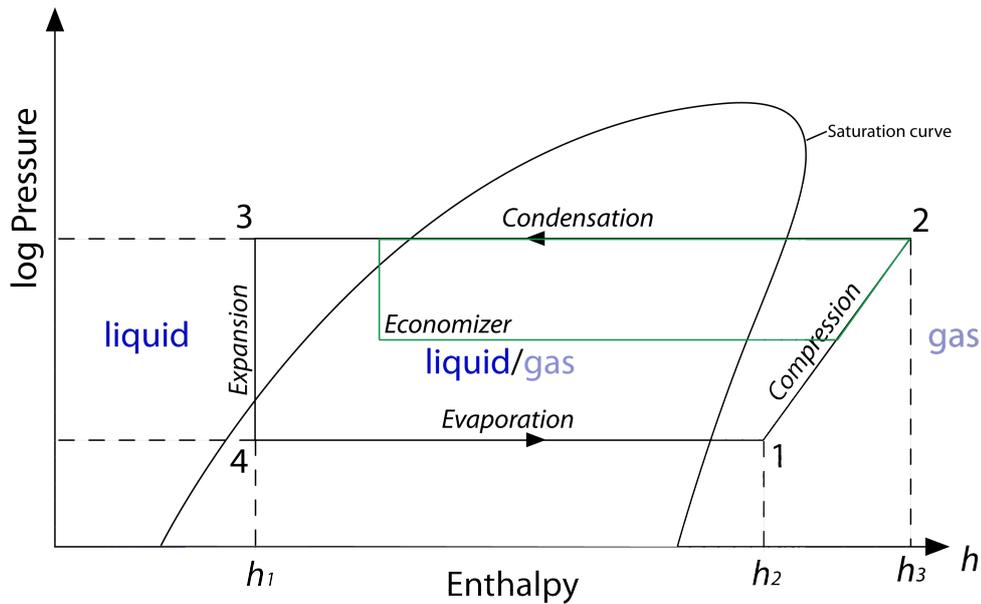


Figure 2.2: The vapor-compression cycle sketched in a Enthalpy  $\log(p)$  diagram

refrigerant's pressure, temperature and enthalpy increases as shown in Figure 2.2. Since some heat is added during the compression stage it is not a isentropic process, and the entropy will therefore increase slightly during the compression.

### 2.1.2 Condensation

From the compressor the refrigerant vapor flows into the condenser. In the condenser the hot gaseous refrigerant transfers heat to the surroundings. The refrigerant starts to condense at constant pressure changing phase from gas into liquid (stage 2-3 in Figure 2.2). To increase the heat transfer a fan is mounted to blow air across the condenser. Through the last part of the condenser the refrigerant temperature should be pulled down below the condensation temperature to ensure that all of the refrigerant is in liquid form before reaching the expansion valve.

### 2.1.3 Expansion

The expansion valve separates the high pressure in the condenser and low pressure in the evaporator. By opening the valve and letting some of the refrigerant through the refrigerant goes through a large pressure drop from  $P_c$  to  $P_e$  (stage 3-4 in Figure 2.2), causing the liquid to start boiling. The enthalpy remains the same as no energy is absorbed or emitted. The partial phase change causes the temperature to drop down to the evaporation temperature, which is determined by the low pressure  $P_e$ . From the valve the refrigerant starts to flow through the evaporator.

### 2.1.4 Evaporation

At the inlet to the evaporator the refrigerant is a mixture of liquid and gas while at the outlet the refrigerant is in gas form. The pressure and temperature remain constant as long as the refrigerant is a mixture of liquid and gas (stage 4-1 in Figure 2.2). The low temperature of the refrigerant allows for it to draw heat from inside the container box. To increase the heat transfer a fan is mounted to blow air across the evaporator and inside the box. At the outlet of the evaporator the temperature of the refrigerant must be higher than the temperature at the inlet to ensure that the refrigerant is in gas form. This temperature difference is called the superheat temperature. The superheat temperature is important, because if it becomes too small liquid could go into the compressor causing the load on the compressor to increase and less refrigerant to be compressed. If the system is allowed to run for too long during which liquid gets into the compressor it can also result in permanent damages on the compressor. But if the superheat temperature is too large the cooling capacity of the evaporator will become smaller which also is undesirable. Thus it is important for good system performance to keep a steady superheat temperature.

### 2.1.5 Economizer

The economizer cycle is like an extra expansion, evaporation and compression stage as shown by the green line in Figure 2.2. Some of the refrigerant from the condenser is diverted to a plate heat exchanger and used to sub-cool the refrigerant going to the evaporator by expanding it and letting it back into the compressor at stage two. The advantage of doing this is that the gas is evaporated at a higher pressure and therefore does not have to be compressed all the way from the lower evaporation pressure. This saves energy and at the same time the evaporator is fed with a higher percentage of refrigerant liquid, which gives a higher cooling capacity. By sub-cooling the refrigerant before letting it into the evaporator a higher cooling capacity can be achieved in the evaporator without increasing the mass flow and the compressor speed at stage one. Thus it saves energy as the compressor does not need to work as much. The higher the pressure difference is between the inlet and outlet on the compressor the more efficient the economizer is.

The refrigerant is now back at stage 1 which completes the vapor-compression cycle.

## System Model

The nonlinear model used for simulating and linearization is as mentioned supplied by Kresten K. Sørensen and Jakob Stoustrup [SS08]. The model is built upon a modular approach to increase the speed and flexibility of the developed simulation environment. The model is divided into components, each described by an XML file. These can then be connected into a composite system model using the tools developed for MATLAB<sup>TM</sup>. Their model is based on the full system described in Figure 2.1 on page 12.

This system is built up by 10 components. These components and their interconnection are shown in the Figure included as Appendix C. Each component has an associated MATLAB<sup>TM</sup> file referred to through the XML file, which calculates the output of the component given the input and the previous state vector. The size of the state vector depends on the complexity of the component. The full system needs a total of 46 states to describe all states in the system. A component called *controller* is included in the model to control all actuators in the system.

As mentioned in Chapter 1 the goal is only to control the compressor speed and expansion valve opening on the model. To obtain a controller for the compressor speed and valve opening the only interesting components are the box, the evaporator and the first stage of the compressor. The rest of the model dynamics are not of interest when trying to control the compressor speed and valve opening. When extracting the linear model the states of the system will be used to form the state space system. If the full simulation model is used, this will result in a linear model with 46 states which is a too large a model size for the later robust controller design. Thus the first step will be to remove the other components and thereby obtain a more simple model, and remove states and inputs/outputs that are of no interest. This will be done in Section 3.1.

To be able to remove components the disconnected inputs for the remaining components must be simulated to get the model to work correctly. This will be done in Section 3.2. When the reduced simulation model is setup it needs to be linearized in a set of different operating points to get the needed linear model for the uncertainty model. This will be done in Section 3.3.

Before setting up the uncertainty model it is investigated if the linear model can be reduced further. This is done to get as small a model as possible for the later robust controller synthesis as it increase in algebraic complexity for each extra state and uncertainty perturbation in the

uncertainty model. Thus if the model has a too big uncertainty perturbation, which for the parametric uncertainty model will depend on the model size, the algorithm might not be able to synthesize a controller. This will be done in Section 3.4.

This all makes way for the uncertainty modeling for the robust controller design which will be setup in the following chapter.

But as mentioned the first step is to simplify the simulation model which will be done in the following section.

### 3.1 Simplify System Model

To obtain a controller for the compressor speed and valve opening the interesting components are as mentioned the box, the evaporator and the first stage of the compressor. Thus the other components are removed and a new simplified system model is built based on these three components. To be able to remove the other components the block called *controller* has to simulate the otherwise expected outputs from the removed components. The dynamic influence of the outputs have through tests been found to be slow and depend on the operating point. They will therefore not have a big influence when finding the needed control inputs for the compressor speed and valve opening or on the dynamics in the model during linearization. Thus by replacing them with constants corresponding to the operating point is assumed to be acceptable.

It is therefore decided that the simulated inputs can be discrete functions and not continuous. The simplified system model and its interconnections is shown in Figure 3.1.

The model shown in Figure 3.1 is reduced to 18 states instead of the full model's 46 states. The simulated inputs are shown with red in the figure. The three simulated inputs are the enthalpy of the refrigerant into the evaporator, the pressure of the refrigerant at the inlet to the evaporator and the pressure at the output of stage one of the compressor. These will be described in the following.

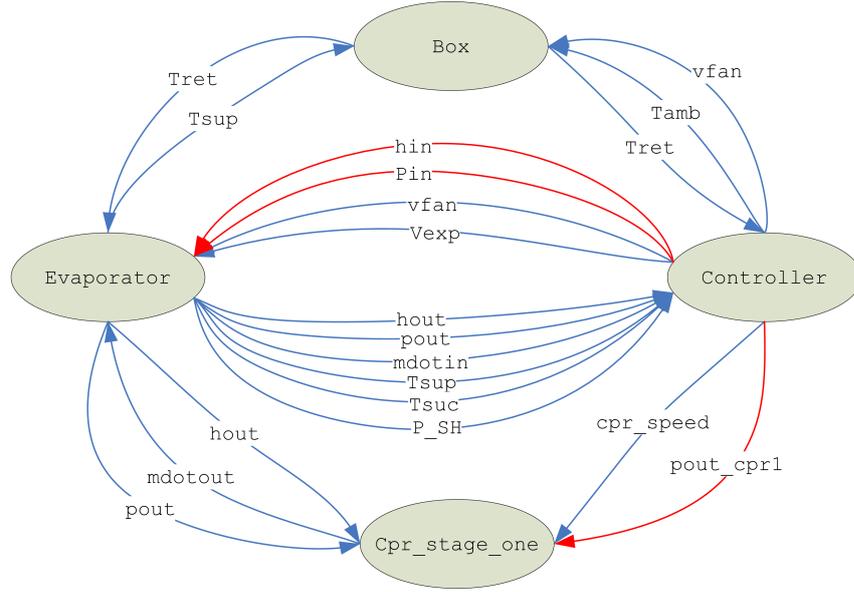


Figure 3.1: Reduced model. The red connections are the simulated inputs.

## 3.2 Simulated Inputs

The pressure output from the economizer into the evaporator can be calculated using the ambient temperature as this limits the condenser in how low it can cool the refrigerant. The refrigerant is normally cooled down to about 11 degree Celsius above the ambient temperature as this is the most energy efficient set point for this system. This value has been supplied by Kresten K. Sørensen. The value has been found by testing the actual system to find the most energy efficient reference for the condenser temperature for the system and this is also the reference used during normal system operating. The pressure can therefore be calculated using the following equation.

$$p_{in} = PDewT(T_{amb} + 11) \quad (3.1)$$

Where  $PDewT$  in Equation 3.1 returns the dew point pressure for the refrigerant.

The enthalpy into the evaporator can be calculated from the bubble temperature of the refrigerant at the pressure  $p_{in}$ . But as the refrigerant is subcooled by the economizer this temperature will be cooler. The extra cooling from the economizer is roughly 10 degrees Celsius. Thus the enthalpy can be calculated using the following equation.

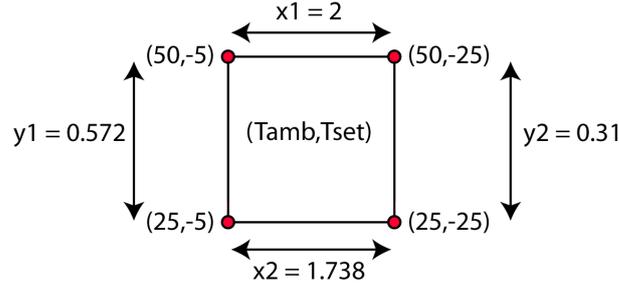
$$T_{Hin} = TBubP(p_{in}) - 10 \quad (3.2)$$

$$Hin = getRefrigHTP(T_{Hin}, p_{in}) \quad (3.3)$$

Where  $TBubP$  returns the bubble temperature for the given pressure. The bubble temperature are subtracted 10 degrees Celsius corresponding to the sub-cooling done by the economizer, after which the enthalpy can be calculated from the pressure and the found temperature using the function  $getRefrigHTP$ .

$T_{set}$	$T_{amb}$	Mean pressure
$-5^{\circ}C$	$25^{\circ}C$	3.74 Bar
$-5^{\circ}C$	$50^{\circ}C$	4.31 Bar
$-25^{\circ}C$	$25^{\circ}C$	2.00 Bar
$-25^{\circ}C$	$50^{\circ}C$	2.31 Bar

**Table 3.1:** Table of set point temperatures and corresponding mean pressure at the compressor stage one output



**Figure 3.2:** Illustration of the difference between the four pressure measurements.

The pressure at the output of stage one of the compressor is however not as straightforward to calculate as it depends on several components. So to find a good estimate for this value four tests were conducted on the full model to get four values to setup a function that can estimate the pressure. The four tests were executed for 50.000 seconds each to ensure that the pressure at the output of compressor stage one had settled. The mean value of the last 1000 pressure samples was then used as estimates for the given set point. The set points and the mean values are shown in Table 3.1.

From the four values the rest will be interpolated with the assumption that the pressure changes linearly. The equation is dependent on  $T_{set}$  and  $T_{amb}$  and it is therefore chosen to setup the equation as  $pout_{cpr1} = a + T_{amb} \cdot b + T_{set} \cdot c$ . The value for b and c are found by taking the mean difference between the two  $T_{set}$  and two  $T_{amb}$  measurements respectively. The differences are illustrated as x1, x2, y1 and y2 in Figure 3.2.

The value for a, b and c can then be calculated as shown in the following equation.

$$b = \frac{y1 + y2}{2} / 25 = \frac{0.572 + 0.31}{2} / 25 = 0.01764 \quad (3.4)$$

$$c = \frac{x1 + x2}{2} / 20 = \frac{2 + 1.738}{2} / 20 = 0.09345 \quad (3.5)$$

Find a at  $T_{amb} = 25$  and  $T_{set} = -5$

$$3.74 = a + 25 \cdot b - 5 \cdot c$$

Inserting b and c to find a

$$a = 3.766 \quad (3.6)$$

From the found a, b and c values found in the equations above the following equation can be

constructed for the pressure at the compressor output.

$$p_{out_{cpr1}} = 3.766 + T_{amb} \cdot 0.01764 + T_{set} \cdot 0.09345 \quad (3.7)$$

To ensure that the estimate is acceptably close to the actual value an extra mean value for the pressure is extracted from the full model at  $T_{amb} = 40$  and  $T_{set} = -15$ . The mean value for the pressure at the compressor output in this point was found to be 3.1648 Bar, and if calculated with Equation 3.7 gives 3.07 Bar, which is an error of 0.0948 Bar which is acceptable.

This gives a good estimate of the pressure on the output of the compressor. By adding the three simulated inputs to the controller component and connecting them to the inputs the other components can be removed without losing any important information in the system. A smaller system model has now been constructed. This can be used for the further linearization which will be done in the following section.

### 3.3 Linearization

To get information on how the system changes accordingly to changes in the set point ( $T_{set}$ ) and ambient temperature ( $T_{amb}$ ) in order to be able to setup the uncertainty model a set of linear models must be extracted. To get the linear models an operating point must first be obtained for the system, where it is in steady state for a given set point temperature and ambient temperature.

To obtain the operating point for the system it is simulated with two simple proportional-integral (PI) controller, one for the expansion valve and one for the compressor. The fan on the evaporator which also is included in the simplified model will be set to medium speed as this is the common speed in this operating range. While running the simulations the limits on the actuators are removed to ensure that the system can settle into a steady state.

The controller for the compressor is set up to use the error in the return air temperature ( $T_{ret}$ ) to control the compressor capacity. Thus if  $T_{ret}$  is larger than  $T_{set}$  the compressor speed is increased to increase the compressor capacity.

The expansion valve is controlled by the pressure in the evaporator, where it should hold the pressure at 0.25 Bar above the pressure for the dew point of the refrigerant. Thus the control error is given as  $error = (evapp_t - p_{out} - 0.25)$  where  $evapp_t = PDewT(T_{suc})$  which is the dew point pressure of the refrigerant at the temperature at the outlet of the evaporator ( $T_{suc}$ ). This will be referred to as the superheat pressure in the rest of this report.

The superheat pressure is used instead of the superheat temperature, as the superheat pressure is nonlinear compared to the superheat temperature. This implies that when using the superheat pressure at a high set point temperatures the superheat temperature is going to be smaller than at lower set point temperatures. The controllers are illustrated in Figure 3.3.

The PI controllers may need to be tuned for it to be able to get the system into a steady state. To ensure that the system finds the operating point for a given set point temperature and ambient temperature, the derivative of the state vector is used. If the derivative is very small the change in the system is small and therefore closer to steady state. The operating points are found for

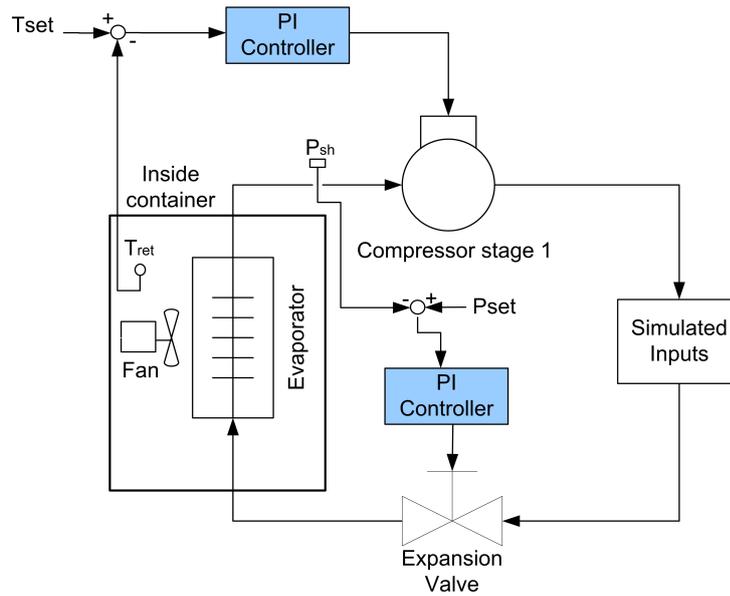


Figure 3.3: Illustration of the PI controller configuration.

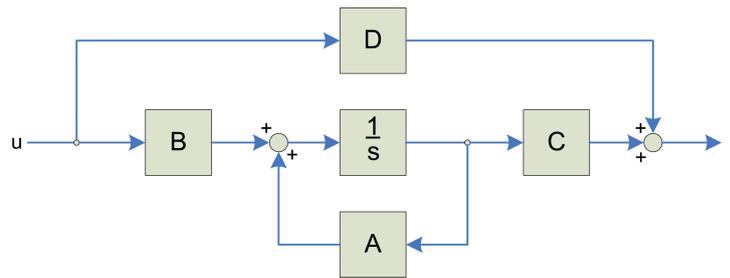


Figure 3.4: State space model

ambient temperature between 25 to 50 degrees Celsius with a step size of 5 degrees Celsius and set point temperatures from -5 to -25 also with a step size of 5 degrees Celsius. This results in a total of 30 operating points.

When the operating points are found, the simulation model is used to linearize the system in each operating point. To linearize the simulation model the controller component is removed and the state space model, as shown in Figure 3.4, is found using the found operating points and the inputs and outputs from the controller component. This is done in steps where each A, B, C and D matrices are found one by one. The method is based on the **linmod** function from MatLab<sup>TM</sup> [TM]. The method is to make a small perturbation around the operating point on all the states and inputs one by one. From the resulting change the A, B, C and D matrices can be set up, where the A matrix is the changes in states to states, the B matrix is the change from input to states, the C matrix is the change from the states to the outputs and the D matrix is the change in input direct to output.

This results in a 18 state model with 7 inputs and 7 outputs. Most of the inputs and outputs can be removed as these are not part of the desired model.

The inputs can be reduced to the opening of the expansion valve (*exp*), the compressor speed

( $cpr_{speed}$ ) and the ambient temperature ( $T_{amb}$ ). The other inputs for the model are  $pout_{cpr1}$ ,  $pin$ ,  $Hin$  and  $Vfan$ . The first three were added to simulate inputs which were removed to simplify the model. They are not controllable and are therefore removed. The last one is the  $Vfan$  which is always set to medium and will not be controlled. This is therefore also removed. These are removed from the model by removing them from the B and D matrices.

The chosen outputs depend upon what errors the controller should control. The calculated superheat pressure ( $p_{SH}$ ) is naturally selected as this is important for the evaporator performance. In the chosen temperature range the system is normally also controlled by the error on the return air temperature compared to the set temperature as done in the location of the operating points. But the return air temperature has been found to have a slow response as this is affected by the cargo and box temperature. It is therefore arguable that it might be easier to get a better performing controller if the supply air temperature is used instead. The error on the return air temperature can then be used to control the reference for the supply air temperature to achieve the right return air temperature. It is therefore chosen to try to design two controllers, one using the return air temperature and another one using the supply air temperature. Thus the two other needed outputs are the return air temperature ( $T_{ret}$ ) and the supply air temperature ( $T_{sup}$ ).

The current system has 18 states and there is a chance that some of these only contribute with little or no extra signal to the system. Thus the system model might be reduced without losing system information. A common way to finding states with little "energy" is by looking at Hankel singular values. The computation of and reduction by these values will be shown in the following.

### 3.4 Model Reduction

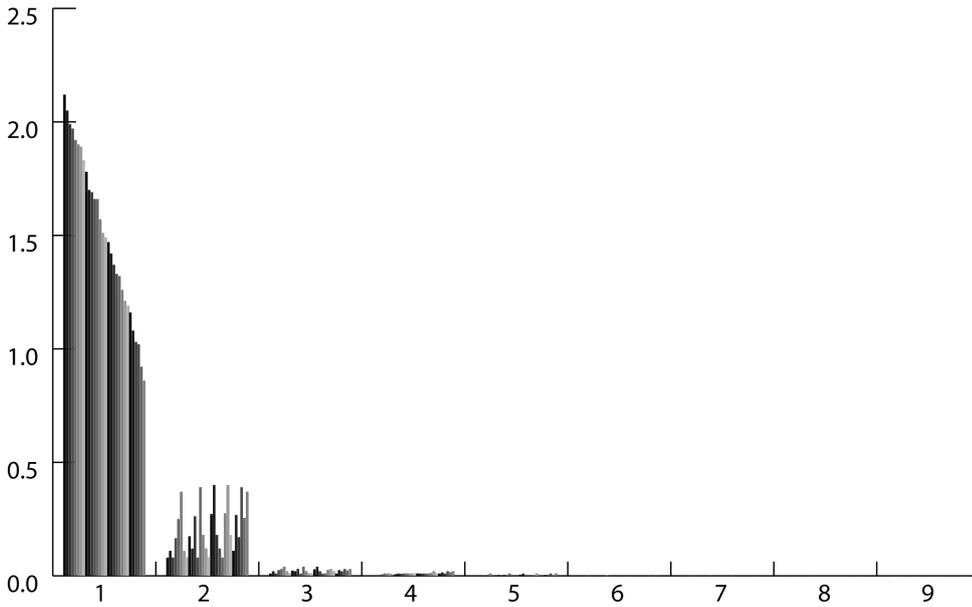
There are several methods for model reduction. Some methods are based on identifying the fast states and then eliminate these since they represent the least controllable and observable states in the system. But these methods do not guarantee that the reduced model fit across the whole frequency and they are therefore not considered in this case. The methods used will be the following two.

1. Balanced model truncation via square root method (balancmr)
2. Hankel minimum degree approximation (hankelmr)

The two methods chosen give a guarantee on the error bound of the infinity norm of the additive error  $\|G - G_{red}\|_{\infty}$  where  $G_{red}$  is the reduced model and  $G$  is the full state model, as a function of the chosen Hankle singular values ( $\sigma_i$ ). The additive error bound is given by Equation 3.8.

$$\|G - G_{red}\|_{\infty} \leq 2 \cdot \sum_{k=1}^n (\sigma_k) \quad (3.8)$$

The basis for these two methods are that they use the Hankel singular values as a measure for which degree the model can be reduced. Hankle singular values represent the "energy" of each



*Figure 3.5: Hankel singular values from the set of linear models for the 9 larger contributing modes*

state in the system. By "energy" is meant the relative size of the influence that the state has on the input-output behavior. Thus by keeping the states with large energy a system will preserve most of its characteristics. This includes the stability, frequency and time responses. [SP05, p. 458-459]

As the two methods give a guarantee for how well the reduced model will fit based on the selected Hankel singular value order, the Hankel singular values for all the linear models must to be found to be able to decide on how much the systems can be reduced without losing too much information. This is done by using the MatLab command `hsvd` on each model. The value for the first nine  $\sigma_i$  value is shown in Figure 3.5.

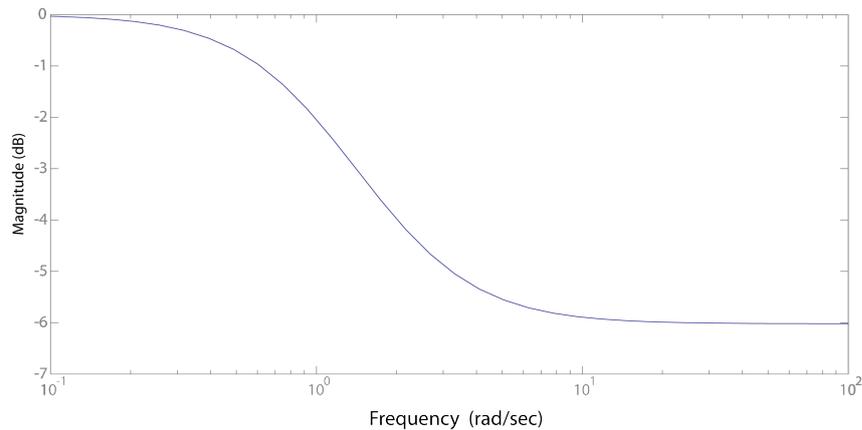
Notice that the values in Figure 3.5 quickly drop down to around zero. This indicates that there are modes that are not controllable or observable. Around  $\sigma_6$  the values are small enough to reduce the rest away. The additive error bound for the infinity norm can be calculated using Equation 3.8 to be less than  $10^{-3}$ . Thus the error is small enough to be neglectable.

The two methods support the option to weigh the inputs and outputs to increase accuracy in one frequency region and decrease it in another. This is useful since the system has a slow dynamic, which makes it acceptable to have less accuracy at high frequency. The transfer function in Equation 3.9 is used to weight the inputs and outputs when reducing a model.

$$R_{weight} = \frac{s \cdot 0.5 + 1}{s + 1} \quad (3.9)$$

A bode magnitude plot of the transfer function in Equation 3.9 is shown in Figure 3.6.

All the models were reduced using the two mentioned methods both with and without weights applied. The models that gave the best result was the Balanced model truncation via square root method not using weights and Hankel minimum degree approximation using weights.



**Figure 3.6:** Bode magnitude plot of the weight function used in model reduction

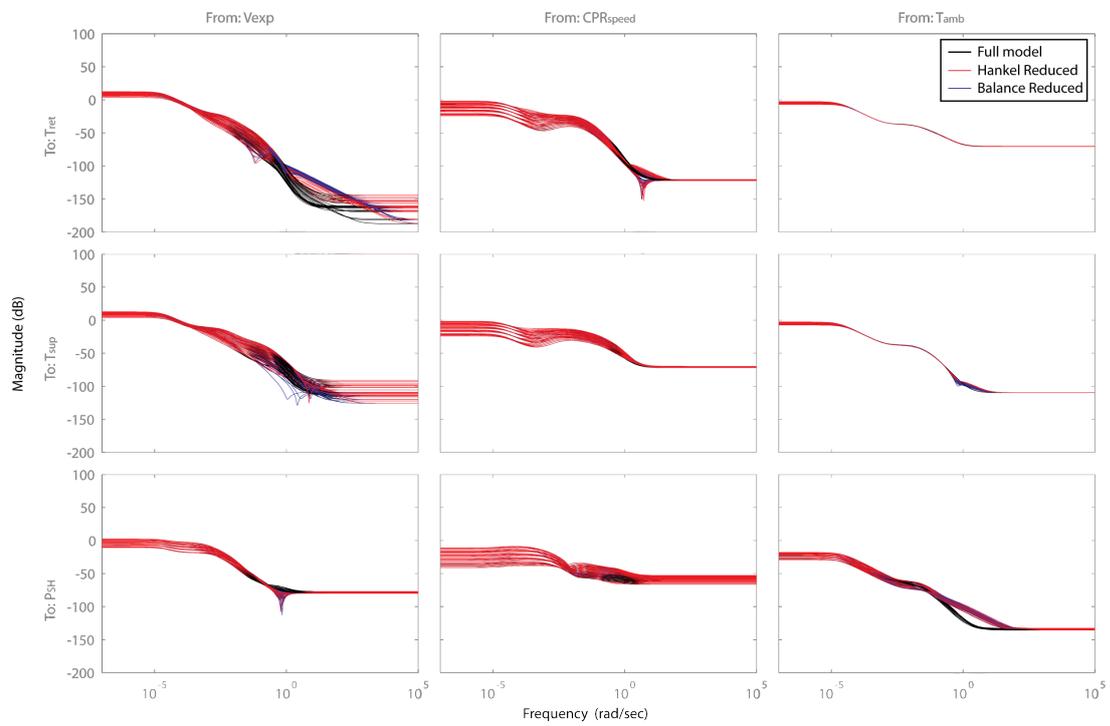
The two mentioned reduced models and the full model are compared in Figure 3.7.

As it can be noticed in the bode plot from  $v_{exp}$  to the output  $T_{ret}$  differs quite a lot. But this can be neglected for two reasons, first of all the dynamics of  $T_{ret}$  are slow, the time constant for  $T_{ret}$  has been found to be 67970 seconds, and secondly the gain might look like it differs a lot but the additive error is still less than  $10^{-3}$ . Another observation is that the other outputs from  $v_{exp}$  and  $cpr_{speed}$  fits well which is important as these are the inputs that are to be controlled.

This would appear to look acceptable, but it was found that these methods result in some unwanted effects in the later construction of the parametric uncertainty model as the reduced system matrices are transformed using different transformation matrices. The transformation matrix is for both methods constructed from the controllability and observability grammians to weight the states with large "energy", and these change for each linear model resulting in different transformations. Thus the new system matrices are not comparable from model to model, and this is needed in the construction of the parametric uncertainty model. Therefore a more controlled approach to the model reduction is needed.

The new approach will be to locate the states that has no or very little effect on the system, and remove these. This is done through a trail and error method, in which one state at a time is removed from the A, B and C matrix and the resulting model are compared to the original system through a bode plot. It was found that four of the states in the system could be removed directly without any effect on the system. The other states all have some influence on the system behavior. Thus the model is only reduced with 4 states resulting in a 14 state model which was decided to be acceptable for the further setup of the uncertainty model and robust controller design.

The reduced models can now be used to setup a uncertainty model for the system which then can be used for robust controller synthesis. This will be done in the following chapter.



*Figure 3.7: Bode plot of the 18 state linear model and the two reduced 6 state models*

# Uncertainty Model

In this chapter the uncertainty model of the system will be set up. The uncertainty model is needed for the controller design to provide information about how the model changes with the changes in operating points. The uncertainty model can be setup in two ways, either using a parametric uncertainty model or a unstructured uncertainty model.

The parametric uncertainty model approach uses real uncertainty parameters to describe how the system parameters changes between the models, while the unstructured uncertainty model uses an uncertain linear time-invariant object to describe the change between the models. The parametric uncertainty model will work well if the system changes in a structured way, while if the system changes in a more unstructured manner when the dynamics in the system change the unstructured uncertainty model will work better. The parametric uncertainty model will, if the system changes in a structured way result in a better fitting uncertainty model, thus also resulting in a less conservative controller design. [SP05, 260]

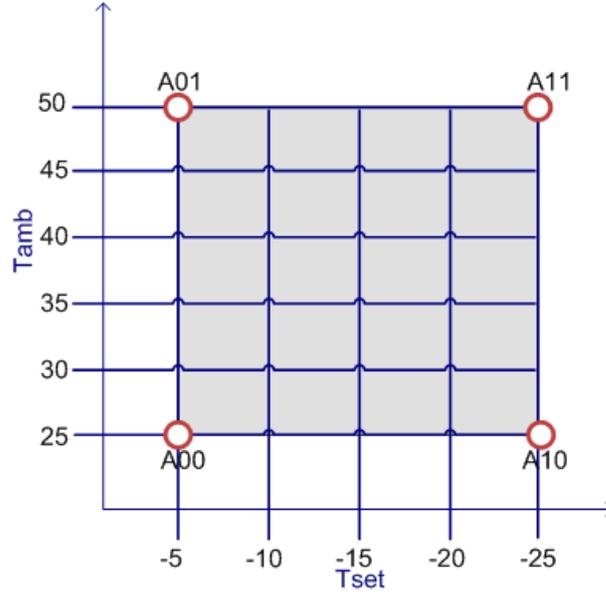
Therefore the approach will be to try to setup a parametric uncertainty model at first. This will be done in the following.

## 4.1 Parametric Uncertainty Model

There are different ways to try to set up a parametric uncertainty model. The approach used here will be based on interpolation using the extracted linear plant models at different operating points to try to obtain one model which should describe the whole model set.

The models are changed depending on the two parameters  $T_{set}$  and  $T_{amb}$ , thus the linear models can be illustrated in a 2D grid where the x axis is the set point temperature and the y axis is the ambient temperature. This is illustrated in Figure 4.1.

As mentioned the parametric uncertainty works best if the model changes in a structured manner, otherwise the parametric uncertainty models have to be more complex to capture the changes in the model. To verify that  $T_{set}$  changes the models in a structured manner a bode plots of the



**Figure 4.1:** Illustration of the model 2D grid

systems where the  $T_{amb}$  value is fixed and the  $T_{set}$  value changes is plotted. This is illustrated in Figure 4.2 where the linear models with  $T_{amb}$  set to 25 degrees Celsius and different  $T_{set}$  operating points are shown.

As it can be noticed from Figure 4.2 the system changes in a structured manner. The same is observed when  $T_{set}$  is kept constant and  $T_{amb}$  is variable. This suggest that an affine parameter combination might be able to capture the changes in the system. This will be attempted in the following using interpolation.

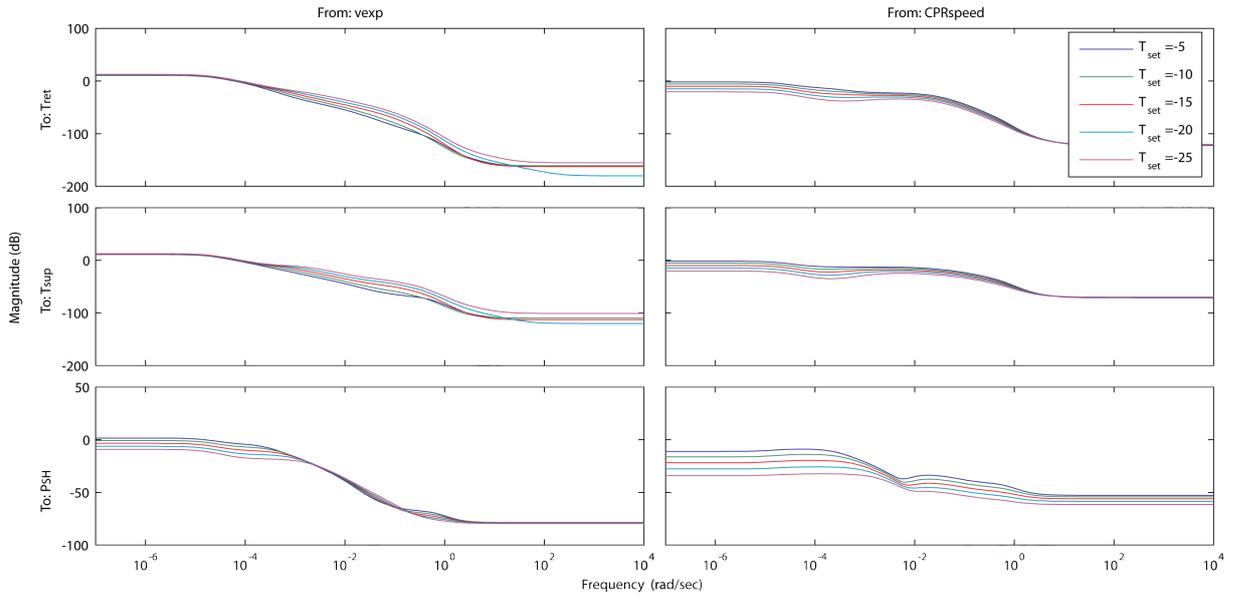
#### 4.1.1 Interpolation

Interpolation uses the corner models from the model set to interpolate the models inbetween. The interpolated model is going to depend on two parameters to express the whole model set as it has to capture both the changes from  $T_{set}$  and  $T_{amb}$ . It has been verified that the model is changing in a structured manner, thus the four corner models, A00, A10, A01 and A11 from Figure 4.1 should be able to represent all the models.

The first attempt will be to interpolate a parametric model using the following affine combination

$$\begin{bmatrix} A(P_1, P_2) & B(P_1, P_2) \\ C(P_1, P_2) & D(P_1, P_2) \end{bmatrix} = \underbrace{\begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix}}_{G_0} + P_1 \underbrace{\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}}_{G_1} + P_2 \underbrace{\begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}}_{G_2} \quad (4.1)$$

Where  $P_1$  and  $P_2$  are between zero and one.  $G_0$  is then a selected nominal model, which in this case could be any of the four corner models and  $G_1$  and  $G_2$  are the difference between the selected nominal model and the corner models along the x-axis and y-axis respectively. If the

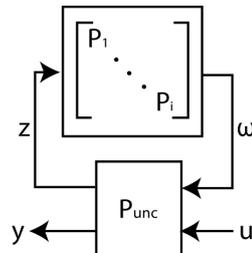


**Figure 4.2:** The linear models for with  $T_{amb} = 25$  and changing  $T_{set}$

nominal model is selected as A00 the two models selected to describe the difference will be A01 and A10. The result is then a model where if  $P_1$  is one and  $P_2$  is zero the model is equal to that of A01, and the same the other way around for A10. If both  $P_1$  and  $P_2$  is one then the model should hopefully be close to the model A11. This parametric uncertainty model can then be described by the following state space model.

$$\begin{aligned}\dot{x} &= (A_0 + A_L P A_R)x + (B_0 + B_L P B_R)u \\ y &= (C_0 + C_L P C_R)x + (D_0 + D_L P D_R)u \\ P &= \begin{bmatrix} P_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & P_i \end{bmatrix}\end{aligned}$$

This can then be rewritten by lumping the perturbation P together as show in Figure 4.3.



**Figure 4.3:** Parametric uncertainty model

This results in the following Equation.

$$\begin{aligned}\dot{x} &= A_0x + B_0u + [A_L \ B_L \ 00]\omega \\ y &= C_0x + D_0u + [0 \ 0 \ C_L \ D_L]\omega \\ z &= [A_R \ 0 \ C_R \ 0]x + [0 \ B_R \ 0 \ D_R]u\end{aligned}$$

This state space model is straightforward to construct using the uncertainty elements from the Robust Control Toolbox in MatLab<sup>TM</sup> [TM]. In this case the uncertain real parameter (**ureal**) is used which can represent a real number whose value is uncertain. In this case that uncertainty has to be between 0 and 1 which can be defined with the 'range' option when creating the uncertain element. This uncertainty element can then be used in the construction of the  $A$ ,  $B$ ,  $C$  and  $D$  matrices for the parametric uncertainty model. The following MatLab<sup>TM</sup> code constructs the two uncertainty elements and the  $A$  matrix for the interpolated parametric uncertainty model.

```

1 P_1 = ureal('P_1',0.5,'range',[0 1]);
  P_2 = ureal('P_2',0.5,'range',[0 1]);
3
  A_diff_10 = A10.a - A00.a;
5  A_diff_01 = A01.a - A00.a;
7
  A = A00.a + A_diff_10*P_1 + A_diff_01*P_2;
```

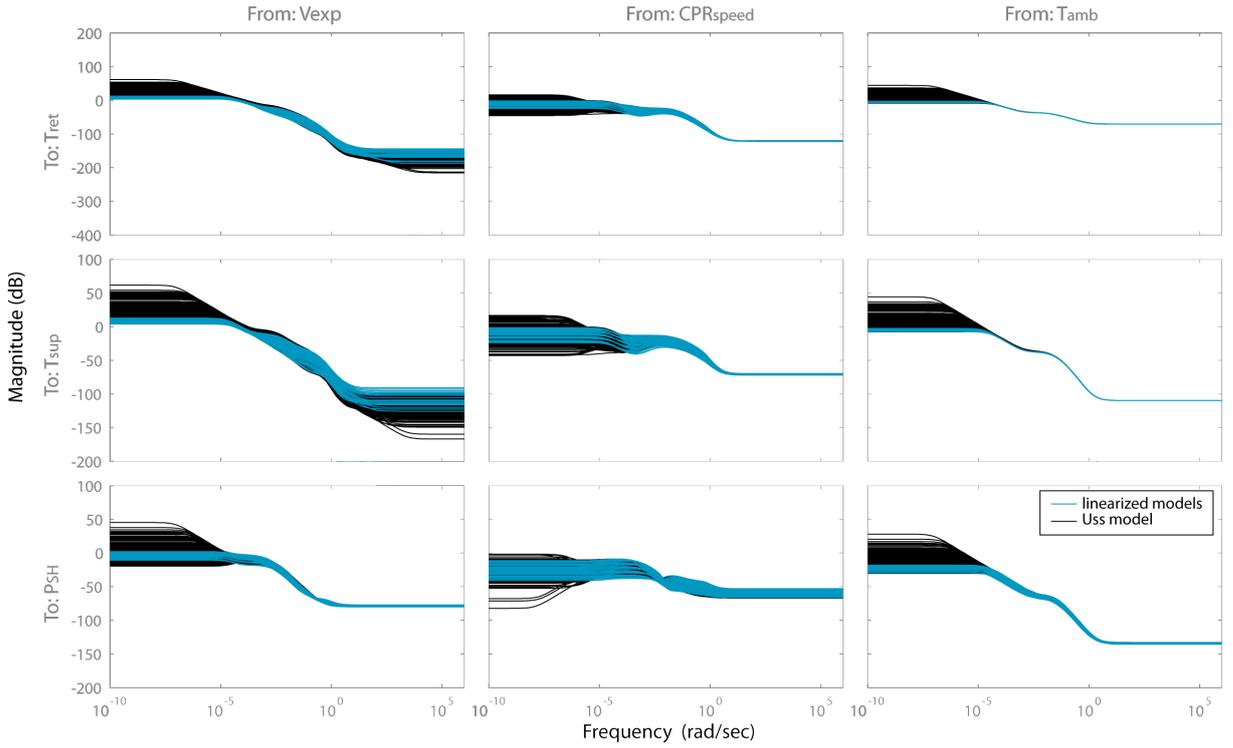
This has to be done for the  $B$ ,  $C$  and  $D$  matrices as well and then join them into a uncertainty state space model. The  $P_1$  and  $P_2$  parameters can then be varied to change the state space model to represent the system model interpolated by A00, A01 and A10 model. To test how well the parametric uncertainty model represents the real model set the parametric uncertainty model is sampled with changing  $P_1$  and  $P_2$  parameters. This is done with a step size of 0.05 on each parameter through two for loops. Thus first changing  $P_1$  0.05 once and then changing  $P_2$  through the whole range. This results in  $\frac{1}{0.05} \cdot 2 = 400$  models. The sampled models and the linearized models are plotted together in a bode magnitude plot in Figure 4.4.

As it can be seen in Figure 4.4 the parametric uncertainty model differs a great deal from the original model, in particular at low frequencies.

It is believed that a better parametric uncertainty model can be found. From studying the parametric model it is observed that the parametric model especially differs a lot from the A11 model which is the point where  $P_1$  and  $P_2$  are one. To compensate for this the parametric model is extended to the following form :

$$\begin{bmatrix} A(P_1, P_2) & B(P_1, P_2) \\ C(P_1, P_2) & D(P_1, P_2) \end{bmatrix} = \underbrace{\begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix}}_{G_0} + P_1 \underbrace{\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}}_{G_1} + P_2 \underbrace{\begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}}_{G_2} + P_1 P_2 \underbrace{\begin{bmatrix} A_3 & B_3 \\ C_3 & D_3 \end{bmatrix}}_{G_3} \quad (4.2)$$

Thereby include the corner model A11 from Figure 4.1 when setting up the parametric uncertainty model. This is setup by adding the following MatLab<sup>TM</sup> code for the construction of the A Matrix.



**Figure 4.4:** Bode magnitude plot of the interpolated Parametric uncertainty model generated from  $A00$  to  $A01$  and  $A10$  together with the linearized models

```

1 A_diff_11 = (A11.a - A00.a - A_diff_10 - A_diff_01);
3 A = A00.a + A_diff_10*P_1 + A_diff_01*P_2 + A_diff_11*P_1*P_2;

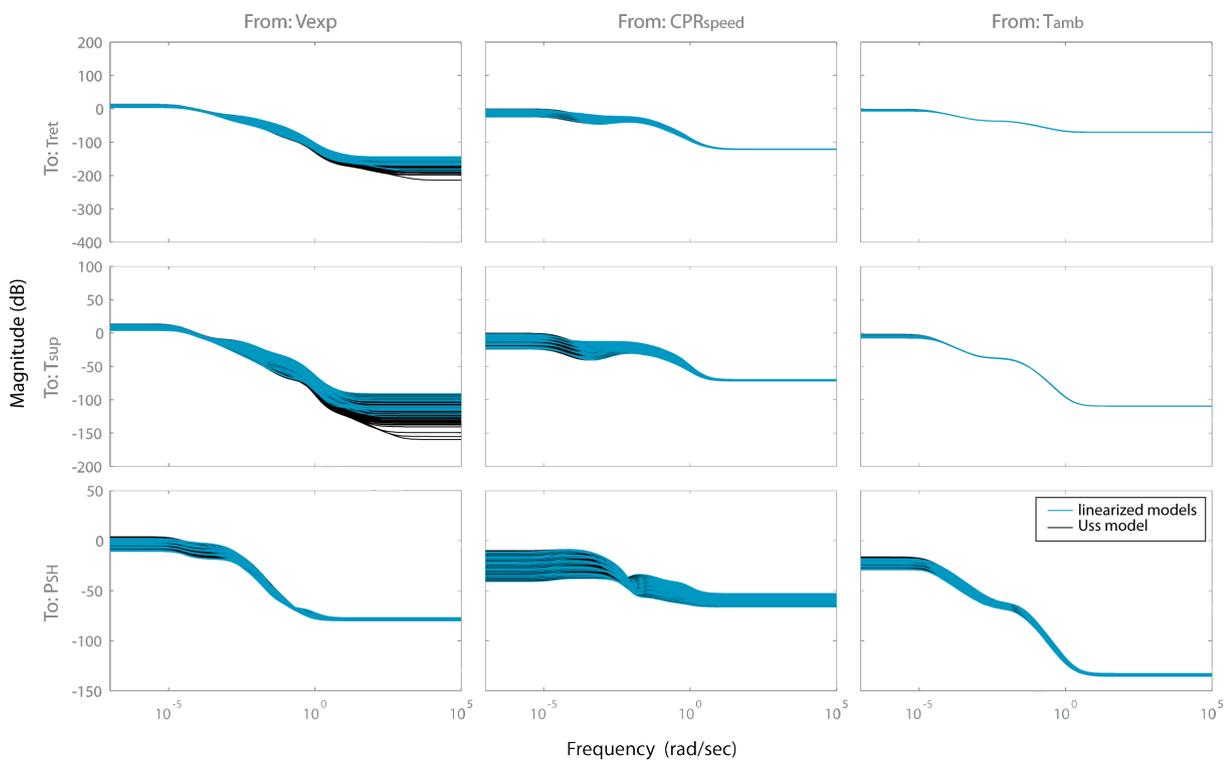
```

This again also has to be done for the  $B$ ,  $C$  and  $D$  matrices. Again it is plotted with changing  $P_1$  and  $P_2$  parameters to observe how the model changes. This is shown in Figure 4.5.

As it can be noticed from Figure 4.5 the parametric model differs very little from the linearized models. The only areas where it differs are at the higher frequencies where the controller can not actuate the system since it is limited to a maximum sampling frequency of 1Hz. The gain of the signal is also already below -100 dB thus the influence will be minimal.

This parametric uncertainty model describes the system very well and there are therefore no reason to further investigate if a better model can be obtained. This was however not the case at the beginning of the project when another linear model was used. Therefore it was also examined if a better parametric uncertainty model might be obtained from extrapolation the models. This method was however no longer necessary as the interpolated model covers the model set well but a description of the methods and the results from extrapolation the model can be found as Appendix A.

As it has been shown that the parametric uncertainty model covers the model uncertainty very well there is no need to try to setup a unstructured uncertainty model. Thus the next step is to set up the robust controller design problem. This will be done in the following chapter.



**Figure 4.5:** Bode magnitude plot of the interpolated Parametric uncertainty model generated from A00 to A01, A10 and A11 together with the linearized models

## $H_\infty$ Controller Design

Now that an uncertainty model have been setup the robust control problem can be setup. The control problem includes the requirements to how the controller should perform and what disturbances affect the system. In this case the only disturbance will be the ambient temperature.

The normal procedure for designing a robust controller is consist of four steps. First the model should be scaled to simplify the controller design and analysis. Secondly the weights for the  $H_\infty$  controller design are setup, which includes weights for the performance, disturbances and limitations on the input for the system. Third step is to design a controller for the nominal model with no uncertainty to verify that the control problem with the given weights is reasonable. The reason to first synthesize a controller for the system with no uncertainty is that the problem will be less restrictive, thus if it is impossible to compute a controller for the system with no uncertainty it will for sure not be able to synthesize one for the system with uncertainty. Fourth step is then to synthesize a controller for the system with uncertainty.

### 5.1 Scaling The Model

Scaling of the model is important as it simplifies model analysis and controller design. The approach taken for scaling the model is to make the inputs, outputs and disturbances less than 1 in magnitude by scaling each with the maximum allowable or expected change. The disturbance and inputs can then be defined as the following. [SP05, p.5-6]

$$d = \frac{\hat{d}}{\hat{d}_{max}}, \quad u = \frac{\hat{u}}{\hat{u}_{max}} \quad (5.1)$$

Where  $\hat{d}$  is the unscaled disturbance and  $\hat{d}_{max}$  is the largest expected disturbance.  $\hat{u}$  is the unscaled input and  $\hat{u}_{max}$  it the maximum allowed input. The output must be scaled by the maximum allowed control error,  $\hat{e}_{max}$ . That results in the following scaling of the output, reference and error.

$$y = \frac{\hat{y}}{\hat{e}_{max}}, \quad r = \frac{\hat{r}}{\hat{e}_{max}}, \quad e = \frac{\hat{e}}{\hat{e}_{max}} \quad (5.2)$$

The scaling factors can more formal be setup by using the following.

$$D_e = \hat{e}_{max}, \quad D_u = \hat{u}_{max}, \quad D_d = \hat{d}_{max}, \quad D_r = \hat{r}_{max} \quad (5.3)$$

As this is a MIMO system each variable in the vectors  $\hat{d}$ ,  $\hat{r}$ ,  $\hat{u}$  and  $\hat{e}$  may have different maximum values in which case the D values are diagonal scaling matrices. This is done to ensure that the outputs are weighted equal in terms of their magnitude. The transformation from the unscaled model to the scaled model can then be setup in the following way.

$$G = D_e^{-1} \hat{G} D_u, \quad G_d = D_e^{-1} \hat{G}_d D_d \quad (5.4)$$

The scaling factors for  $D_e$  should be the maximum allowable error, this is therefore defined as :

$$D_e = \begin{bmatrix} T_{ret} & 0 \\ 0 & p_{SH} \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.25 \end{bmatrix} \quad (5.5)$$

There are only one disturbance for  $D_d$  which is  $T_{amb}$ , thus this will only be a scalar. The maximum disturbance is the maximum change in the ambient temperature which is 25.  $D_d$  is therefore defined as  $D_d = T_{amb \ max} = 25$ .

The scaling factor for  $D_u$  includes the control inputs. These have to be scaled by the maximum allowable input change to make the inputs less than 1 in magnitude. The maximum allowed input for  $Vexp$  is 100% and for the  $CPR_{speed}$  it is 110Hz. The maximum values found for these two control inputs during the linearization were 81Hz for the  $CPR_{speed}$  and 12 for the  $Vexp$ . Thus in worst case the maximum allowable input change are approximately 30 and 90 respectively if only positive changes are considered. If negative changes are considered the limits would be much smaller, but as the system should be able to reach a temperature rather quick the inputs should not be limited too much. Therefore only the positive changes are considered here. The scaling should therefore be 90 and 30, but it was found that this skew weighting of the inputs resulted in much less utilization of the compressor and it was therefore decided to weight the two inputs equally. They are therefore both set to 30. The scaling matrix  $D_u$  is therefore defined as the following.

$$D_u = \begin{bmatrix} Vexp_{max} & 0 \\ 0 & CPR_{max} \end{bmatrix} = \begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix} \quad (5.6)$$

Thus now the scaled model can be constructed with  $G = D_e^{-1} \hat{G} D_u$  and  $G_d = D_e^{-1} \hat{G}_d D_d$ . The scaled system can then be used for the further controller design and analysis.

Before the actual controller design is described the principles of robust controller design will be outlined.

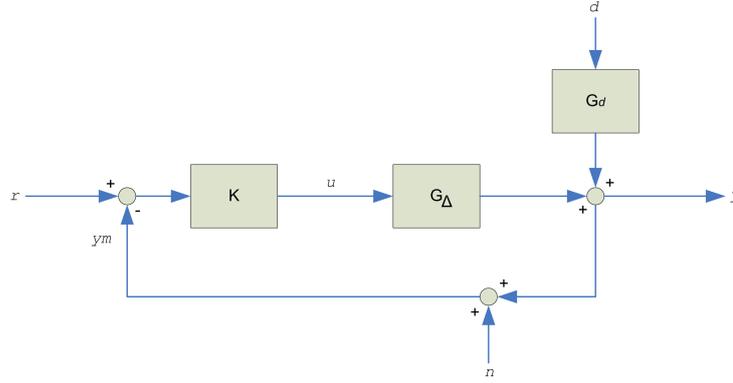


Figure 5.1: Block diagram of one degree-of-freedom feedback control system

## 5.2 Robust Controller Design Method

The controller structure chosen is a one degree-of-freedom controller as shown in Figure 5.1.  $G_\Delta$  is the plant model including the uncertainty,  $G_d$  is the disturbance model and  $K$  is the controller. The input  $n$  is added sensor noise. From the Figure the input for the plant can be described as  $u = K(s)(r - y - n)$ . The objective for the controller is then to manipulate  $u$  such that the control error remains small in spite of disturbances  $d$ . [SP05, p. 21]

The output  $y$  in Figure 5.1 can then be written as  $y = G_\Delta(s)u + G_d(s)d$ . By substituting the above expression for  $u$  into  $y$  yields the following.

$$y = G_\Delta K(r - y - n) + G_d d \quad (5.7)$$

or

$$(I + G_\Delta K)y = G_\Delta K r + G_d d - G_\Delta K n \quad (5.8)$$

This results in the following closed-loop response for the system.

$$y = \underbrace{(I + G_\Delta K)^{-1} G_\Delta K}_T r + \underbrace{(I + G_\Delta K)^{-1} G_d}_S d - \underbrace{(I + G_\Delta K)^{-1} G_\Delta K}_T n \quad (5.9)$$

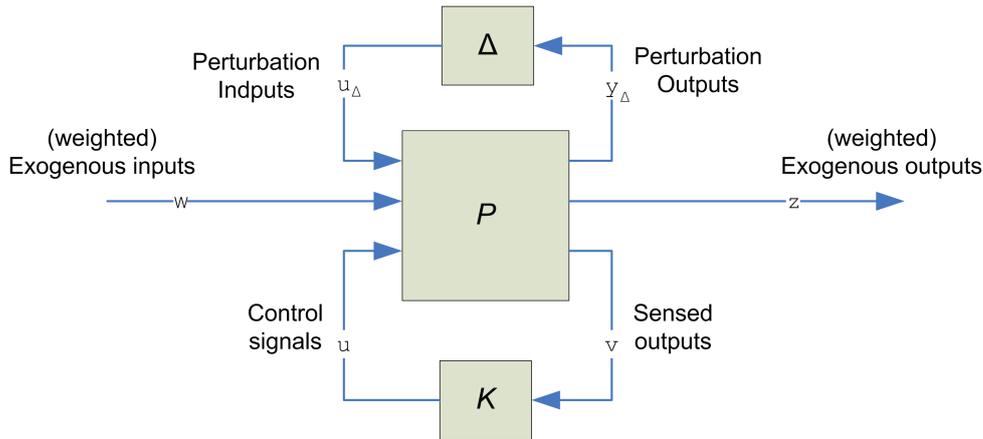
From the above the following three terminologies can be formed.

$$L = G_\Delta K \quad \text{Loop transfer function} \quad (5.10)$$

$$S = (I + G_\Delta K)^{-1} = (I + L)^{-1} \quad \text{Sensitivity function} \quad (5.11)$$

$$T = (I + G_\Delta K)^{-1} G_\Delta K = (I + L)^{-1} L \quad \text{Complementary sensitivity function} \quad (5.12)$$

The sensitivity function  $S$  is a very good indicator of closed-loop performance. The main advantage of considering  $S$  is that it ideally should be kept small, and it is sufficient to consider its magnitude and not worry about its phase. This enable us to setup specifications in terms of



**Figure 5.2:** General control problem formulation for the case with model uncertainty

$S$  such as maximum tracking error at selected frequencies, maximum steady-state tracking error or maximum peak magnitude, as  $\|S(j\omega)\| \leq M$ . [SP05, p.60]

These specifications may be defined as an upper bound,  $\frac{1}{\|W_p(s)\|}$ , on the magnitude of  $S$  where  $W_p(s)$  can be selected to match the requirements for the system. The performance specification can then be formulated as

$$\|S(j\omega)\| < \frac{1}{\|W_p(j\omega)\|}, \quad \text{for all } \omega \quad (5.13)$$

$$\Leftrightarrow \|W_p S\| < 1, \quad \text{for all } \omega \quad (5.14)$$

$$\Leftrightarrow \|W_p S\|_\infty < 1 \quad (5.15)$$

The  $H_\infty$  norm of  $W_p S$  is the peak value or supremum of  $\|W_p(j\omega)S(j\omega)\|$  as a function of the frequency.

Further, from the identity that  $S+T=I$  and Equation 5.9 it can be observed that the controller design must trade off between the effect of noise against the rejection of disturbances and the tracking of the reference signal over frequency.

The  $H_\infty$  design synthesis which is used to obtain a controller, then seeks to minimize the largest closed-loop gain across all frequencies. Thus the requirements have to be set up as constraints on the closed-loop gains by setting up the performance requirements as weight functions to normalize the specifications across all the frequencies and to weight each output according to the requirements. [SP05, 104-106]

The general control problem for a model with uncertainty can then be setup as shown in Figure 5.2.

Where  $P$  is the generalized plant,  $K$  is the controller and  $\Delta$  is the uncertainty perturbation. The generalized plant  $P$  contains the nominal plant model, performance and uncertainty weighting functions. The input  $w$  contains all external inputs, including disturbances, sensor noise, and reference. The output  $z$  is an error signal which has to be minimized to meet the control objectives.

The overall control objective is then to minimize the  $H_\infty$  norm of the transfer function from  $w$  to  $z$ . Thus find a controller  $K$ , which, based on the information in  $v$ , generates a control signal  $u$ , which counteracts the influence of  $w$  on  $z$ , thereby minimizing the closed-loop norm of the transfer function from  $w$  to  $z$ . [SP05, 107]

The design method for the model with uncertainty and the model without uncertainty differs from this point, and they will therefore each be described in the sections below when the standard problem has been setup.

### 5.3 Standard Problem Formulation

The system description for the generalized plant  $P$  from Figure 5.2 without the uncertainty pulled out is shown in Figure 5.3 where the weights for the performance and inputs are added. In Figure 5.3  $T_{mes\ error}$  is the chosen temperature reference which is either the return air temperature or the supply air temperature. This is as mentioned in Section 3.3 to examine if a better controller might be obtained by using the supply air temperature instead of the return air temperature as the latter has a slow dynamic response which might make it difficult to get a good performing controller. The weights and problem formulation are the same regardless of the temperature reference used.

Figure 5.3 shows the connections from control inputs  $u$  and exogenous inputs  $w$ , to the measured outputs  $v$  and Exogenous outputs  $z$  together with the weight functions. The weight block added will be outlined in the following.

- $W_{T_{amb}}$  - Weight that shapes the ambient temperature disturbance to match that dynamics of the ambient temperature.
- $W_{p1}$  and  $W_{p2}$  - Weight functions defining maximum tracking error on the return temperature or supply air temperature and the superheat pressure respectively.
- $W_{u1}$  and  $W_{u2}$  - Weight functions defining maximum allowable control input for the compressor speed and expansion valve respectively.
- $W_{T_{ref}}$  - Weight for how the controller should track changes in the reference for the return air temperature or supply air temperature.
- $W_{noise}$  - Weight for the actuator noise on the input channel and sensor noise on the output channel.

These weights need to be defined according to the requirements of the system. This will be done in the following.

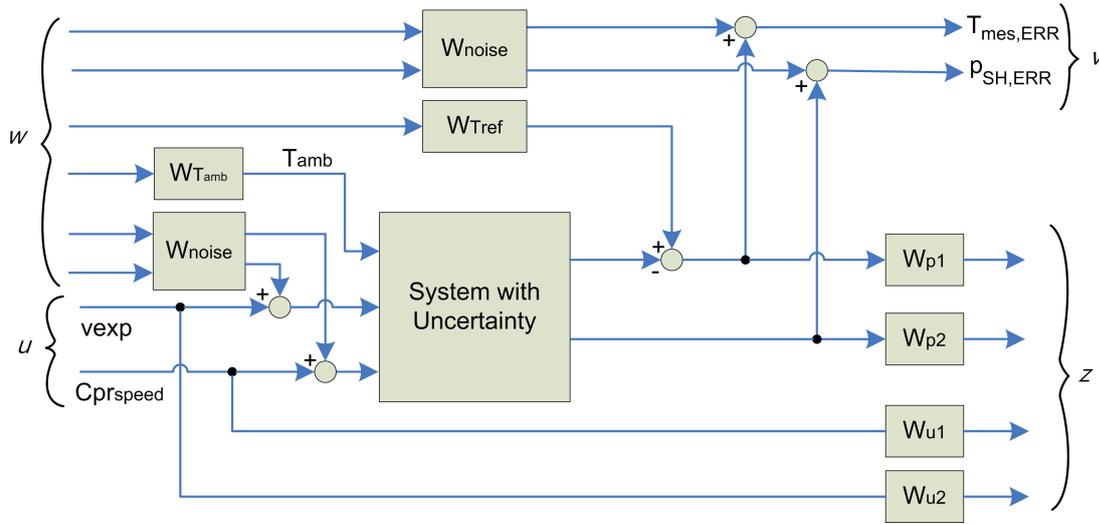


Figure 5.3: System setup

## 5.4 Weights

There are as mentioned a set of weight functions that must be defined to setup a bound on the performance, input and disturbances of the system. These should be defined based on the requirements for the system.

### 5.4.1 Performance Weights

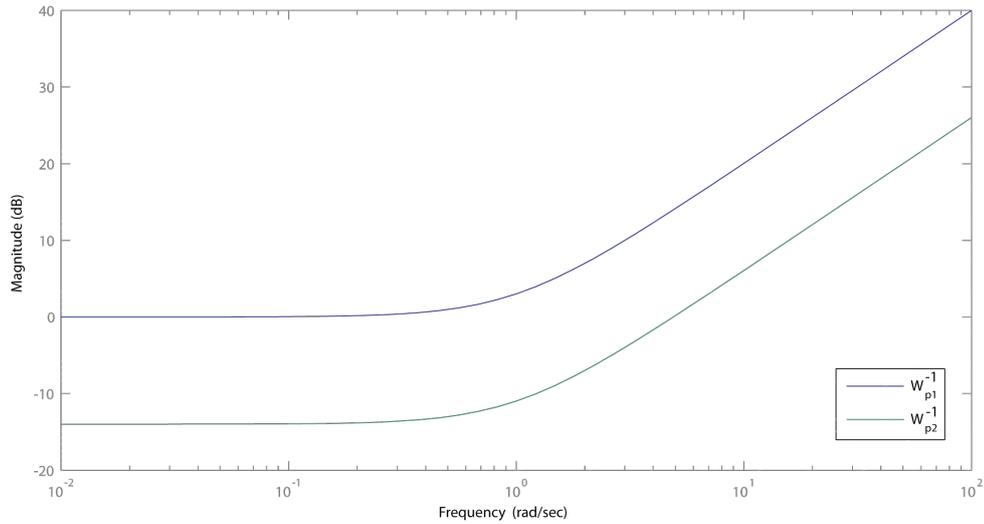
The performance weights for each output channel of  $S$  are chosen to be first order systems. As a fast response is desired on both the superheat pressure and temperature, it is decided to require good tracking of both up to 1 Hz after which a larger error can be allowed. Thus giving the weights a cutoff frequency at 1 Hz.

The magnitude of the two performance weights will sets the requirement to how tight the controller should track the error on the channel. For the return air temperature the requirement is that it should be kept within  $\pm 0.5$  degrees Celsius, thus the magnitude of the output should be kept below 1 as the channel is multiplied by  $0.5^{-1}$ . Thus allowing for a maximum error of 0.5 degree Celsius. For the superheat pressure the control should be more tight to ensure a good cooling capacity and that no liquids gets into the compressor. For this controller design it is decided to set it to a maximum error of  $\pm 0.05$ . Thus with the scaling of the channel (which is  $0.25^{-1}$ ) the magnitude of the weight should be 5, allowing for a maximum error of  $\frac{1}{5} \cdot 0.25^{-1} = 0.05$ . The two weights are therefore chosen as the following two first order transfer functions.

$$W_{p1} = \frac{1}{s + 1} \quad (5.16)$$

$$W_{p2} = \frac{5}{s + 1} \quad (5.17)$$

Figure 5.4 depicts the magnitude of  $W_{p1}^{-1}$  and  $W_{p2}^{-1}$  which is the upper magnitude bound for output channels of  $S$ .



**Figure 5.4:** Upper magnitude bound for output channels of  $S$

### 5.4.2 Input Weights

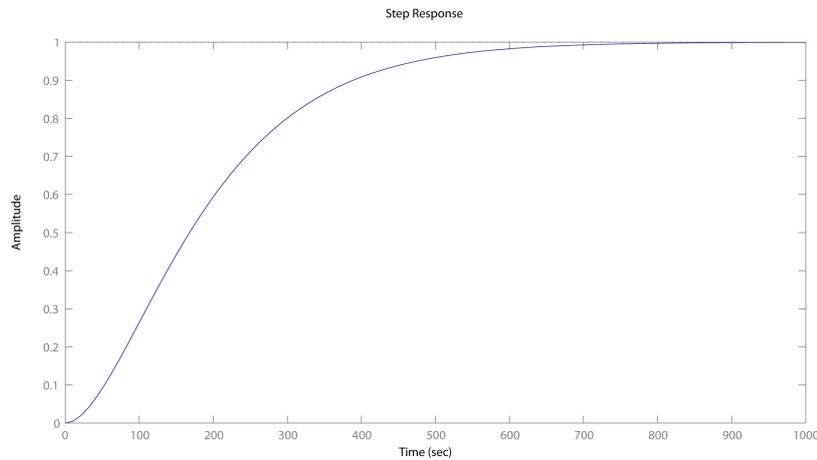
The input weights have to bound the maximum allowable control input for the system. Given the scaling of the two control inputs the weight are defined as  $W_{u1} = 1$  and  $W_{u2} = 1$  where  $W_{u1}$  is for the compressor speed and  $W_{u2}$  is for the expansion valve. From the scaling the compressor speed was limited to the worst case maximum allowable change of 30Hz in the control input, while the expansion valve could handle a larger change. But it is decided to limit the expansion valve to the same as the compressor to weigh them equally. This will allow the two inputs to have a maximum peak of 30 across the whole frequency which should be enough to suppress errors and disturbances.

### 5.4.3 Disturbance Weight

The disturbance to the system is the ambient temperature. The ambient temperature has a slow dynamic and the change of the ambient temperature is included in form of change in operating point. It can therefore be argued that it only has to reject the disturbance around the operating point. But to ensure good disturbance rejection of the ambient temperature it is decided to allow the maximum change of the ambient temperature to be 25 degree Celsius. The ambient temperature is scaled with 25 and the magnitude therefore have to be 1 at maximum gain. The following second order transfer function is used to shape the disturbance.

$$W_{T_{amb}} = \frac{1}{10000s^2 + 200s + 1} \quad (5.18)$$

A step response of the transfer function is shown in Figure 5.5. From the figure it is noticed that the rise time, which is the time it takes to go from 0.1(10%) to 0.9(90%) is roughly 400 seconds which is approximately 7 minuts for a change of 25 degree Celsius in the ambient temperature. This might seem as a rather fast and large change, but this is also if worse case scenario are



**Figure 5.5:** Transfer function for shaping the disturbance

considered. If the controller is able to handle this it should also be able to handle slower temperature changes.

#### 5.4.4 Reference Weight

The reference weight sets the requirement for how the system should track a change in the temperature reference. The magnitude represents the maximum expected change in the reference, and the response should reflect how the system should track the reference. It is decided to shape the reference as the following first order system.

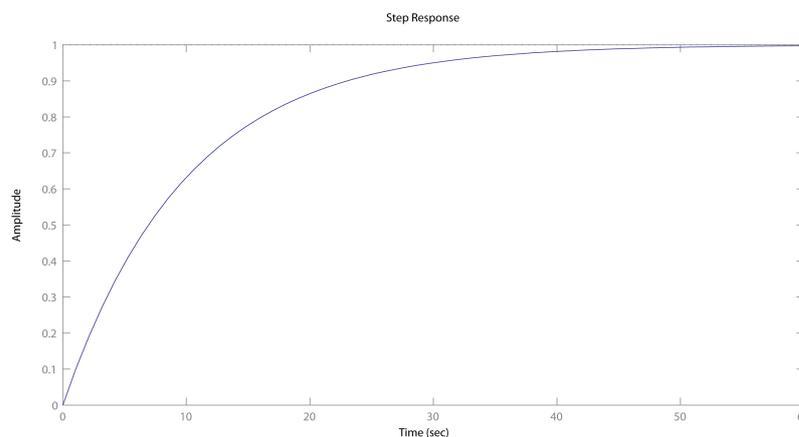
$$W_{Tref} = \frac{0.1}{1s + 0.1}; \quad (5.19)$$

The gain of the weight is 1 which corresponds to 2 degrees Celsius with scaling included. The transfer function for the reference is shown in Figure 5.6.

#### 5.4.5 Noise Weight

Normally weighted noise is added to account for actual sensor and actuator noise. In this case it has been implied that there are no sensor noise nor actuator noise. But the noise is added to aid the controller design in obtaining a better controller as it has been found that by adding a very small amount of noise to the model the design is able to find a better performing controller.

The noise on the input and outputs are weighted as follows



**Figure 5.6:** Transfer function for shaping the reference

$$noise_{input} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} \quad (5.20)$$

$$noise_{output} = \begin{bmatrix} 0.002 & 0 \\ 0 & 0.002 \end{bmatrix} \quad (5.21)$$

For the input the added noise equals roughly 0.3% of the signal and for the output it is less than 0.1% thus the noise should not have a big influence but it has shown to aid the controller design in obtaining a controller with a better performance.

Now that the weight for the system has been setup the controller for the system can be designed.

## 5.5 Controller Design

First a controller for the nominal system will be synthesized as mentioned in the beginning of the chapter, to ensure that the problem formulation is sound. Afterward a controller for the system with uncertainty is synthesized using  $\mu$ -synthesis D-K iteration, to account for the uncertainty in the system.

### 5.5.1 $H_\infty$ Controller For The Nominal Plant

The  $H_\infty$  controller design problem for a system with no uncertainty is to find a linear controller,  $K$ , such that the closed-loop system  $z = F_L(P, K)w$  is stable and the  $\|F_L(P, K)\|_\infty$  norm is less than  $\gamma$ . Where  $F_L$  is a linear fractional transformation (LFT) which connects  $P$  and  $K$  on the form as shown in Figure 5.2 but without the uncertainty. [SP05, 357]

To synthesize the controller the *hinfsyn* function from MatLab<sup>TM</sup> is used. The function iteratively lowers the  $\gamma$  value and solves two Riccati equations to obtain a  $H_\infty$  optimal controller.

This is done for the system setup in 5.3 for the nominal model without uncertainty where both the return air temperature and the supply air temperature are used. The result is two controllers that achieve the following  $\gamma$  value :

- Using return air temperature :  
 $\gamma = 0.9536$
- Using supply air temperature :  
 $\gamma = 0.6953$

Thus the design algorithm is able to get a controller in both cases, but with the given weights the controller using the supply air temperature is better at fulfilling the requirements as it has a lower  $\gamma$  value. This was also assumed, as the supply air temperature has a faster response. Since the problem formulation now seems sound the uncertainty can be added into the design process.

### 5.5.2 $H_\infty$ Robust Controller via $\mu$ -synthesis D-K iteration

When adding the uncertainty to the control problem the *hinfsyn* function will not work as it can not handle the uncertainty elements used to setup the uncertainty model. The function *dksyn* will therefore be used instead which is also available in MatLab<sup>TM</sup> and that can handle the uncertainty elements. The function *dksyn* combines the  $H_\infty$  synthesis and  $\mu$ -analysis to derive a controller.

The design objective is to find a stabilizing controller  $K$ , such that for all  $\Delta$ , the closed-loop system is stable and satisfies

$$\|F_L[\underbrace{F_U(P, \Delta)}_{\text{perturbed plant}}, K]\|_\infty \leq 1 \quad (5.22)$$

Where  $F_U$  is a LFT which connects  $P$  and  $\Delta$  on the form shown in Figure 5.2. As the  $\Delta$  structure can change in value another approach is needed to represent the  $\Delta$  perturbation. To achieve this the perturbation is replaced with a scaling matrix ( $D$ ) which are calculated to match the upper bound of the structured singular value ( $\mu$ ). The  $\mu$  value is used as it relates to the corresponding robust performance problem associated with the uncertain system. The  $D$  matrix has the property that  $D\Delta = \Delta D$ .

For this to be true the  $\Delta$  block must have a diagonal structure as shown in the following.

$$\Delta = \text{diag}[\Delta_1, \Delta_2, \dots, \Delta_N] \quad (5.23)$$

Where the weighted exogenous performance requirement can be included in the  $\Delta$  structure as an additional fictitious block on the form.

$$\Delta_{full} = \text{diag}[\Delta_1, \Delta_2, \dots, \Delta_N, \Delta_F] \quad (5.24)$$

The scaling matrix ( $D$ ) can therefore be defined as

$$D = \text{diag}[d_1I, d_2I, \dots, d_NI, I], \quad \text{where } d_i > 0 \quad (5.25)$$

By replacing the  $\mu$  upper bound value with the scaling set  $D$  we are able to calculate a minimizing controller for the system. For this the *hinfsyn* is used where the minimization problem is as follows [SP05, p. 305]

$$\min_{\substack{K \\ \text{stabilizing}}} \|DF_L(P, K)D^{-1}\|_{\infty} \quad (5.26)$$

With the calculated stabilizing controller  $K$  the scaling set  $D$  is calculated again from the upper bound of  $\mu$  after which a new controller can be found. Thus it iterates between

- Holding the  $D$  variable associated with the scaled  $\mu$  upper bound fixed and find the controller variable  $K$
- Holding the controller  $K$  variable fixed and find the scaling set  $D$

The D-K iteration procedure is however not guaranteed to converge to a global or even local minimum  $\mu$  value, but the method has shown to work well in practice for many problems [SP05, p. 328].

The D-K iteration are performed as mentioned by using the MatLab<sup>TM</sup> function *dksyn*. The resulting controller when using the supply air temperature were found to have a upper bound value of 1.3105. This indicates that it do not fulfill robust performance, but it still might fulfill robust stability.

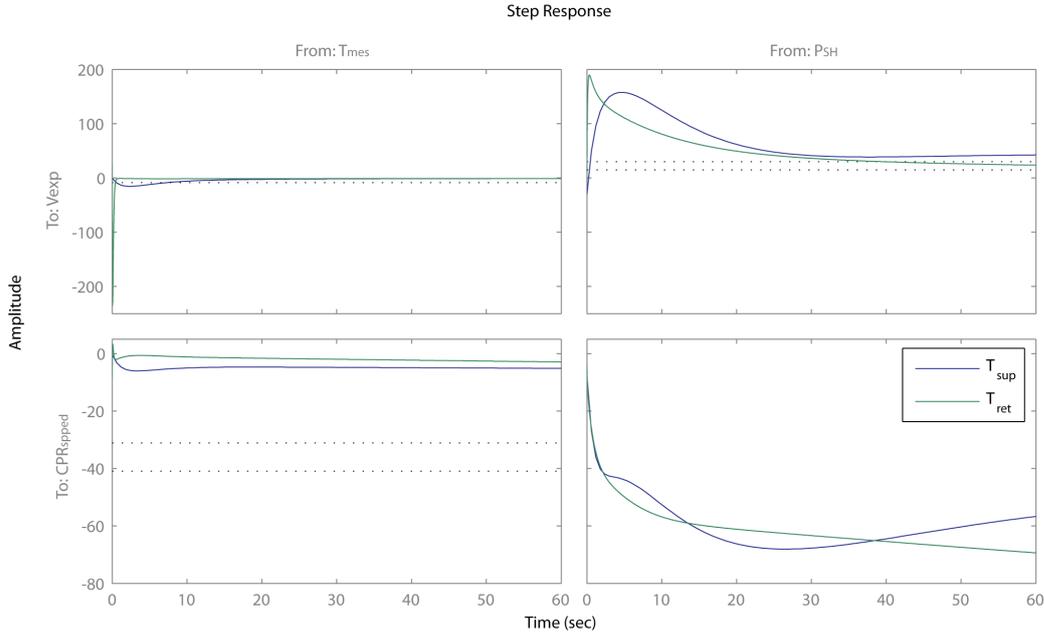
When using the return air temperature the resulting controller had a worse upper bound value of 1.5154. It was found by doing step response analysis on the controller that it has a slow response to errors in the return air temperature. This is shown in Figure 5.7 where the two found controllers are compared.

In Figure 5.7 it can be noticed that the controller found using the return air temperature acts on errors in the superheat pressure, but not as much on the return air temperature where it has a much slower response. It also has a large spike on the input for the expansion valve on a error on the return temperature. This implies that the *dksyn* algorithm can not obtain a controller that can fulfill the given requirements. The controller has been tested on the simulation model, and it was found that the controller performed badly while the controller found for the supply air temperature performed much better. It is therefore decided to continue only with the controller developed to control based on the supply air temperature.

The controller found for the supply air temperature is a state space model with 2 outputs, 2 inputs, and 79 states. This is a large number of states, but most of them can be reduced away using the model reduction methods discussed in Section 3.4. It was found that the controller can be reduced to 16 states with a maximum error of 0.35%. The reduced controller will be used when testing the system.

Before testing the controller it it investigated how well the controller fulfills the given requirements for robust performance and robust stability and for this the full controller model will be used. To do this the MatLab<sup>TM</sup> functions *robustperf* and *robuststab* are used that calculates the robust performance and robust stability margin.

Using the functions *robustperf* it was found that the controller achieves a robust performance margin of 0.7915. Thus there exists a model which excites the performance margin by  $\frac{1}{0.7915} =$



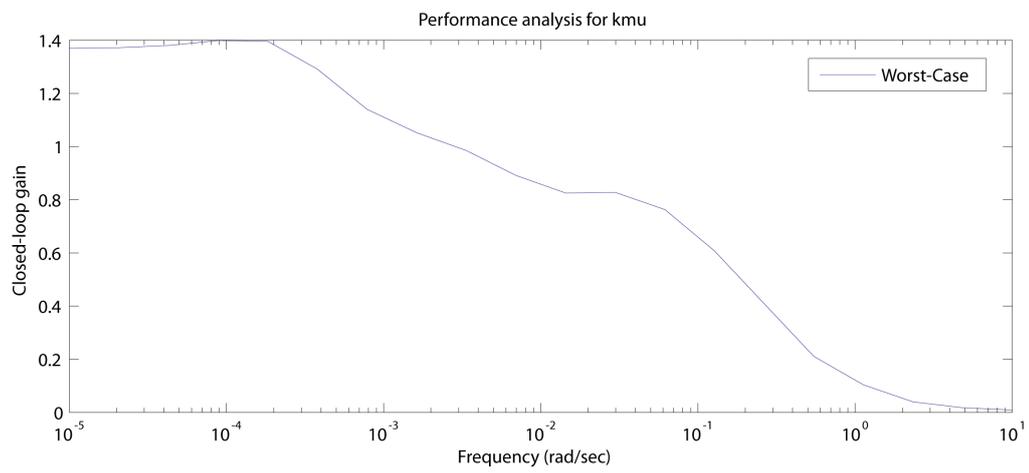
**Figure 5.7:** Step response of the two found controllers using the return air temperature and supply air temperature. The step is equal to a error of 1 degree Celsius on  $T_{mes}$  and 1 Bar on  $P_{SH}$  as the controller are normalized

1.2634. From examining the singular values it was found that the performance requirement that is not always fulfill is the requirement to the supply air temperature. Thus in worse case with the given weights and uncertainty the temperature might differ with  $0.5 \cdot 1.2634 = 0.6317$  degree Celsius from the reference. This is acceptable for the system, but this entail that the controller are not able to achieve robust performance.

To analyze in which frequency region the controller are not able to achieve robust performance the system can be analyzed using the MatLab<sup>TM</sup> function *wcgain* which computes the worst-case gain as a function of frequency. The worse case gain are shown in Figure 5.8. It can be noticed at that the controller have problem with keeping the performance margin in the low frequencies. It has through analysis been found that the weight that has the impact on the performance margin is the large change in the ambient temperature. If this is lowered the controller design are actual able to achieve robust performance. It is how ever decided to keep the weights as defined to ensure the best possible disturbance rejection.

The robust stability is the more impotent of the two as this will indicate if the controller is robustly stable to modeled uncertainty. This is tested by using the *robuststab* function. The function computes the controller to be able to tolerate up to 152% of the modeled uncertainty. Thus the controller is robustly stable.

The fact that the controller are not able to get robust performance are accepted at the margin is close to the requirements. It is therefore decided to test how well the controller performs. The test setup will be defined and performed in the following chapter.



**Figure 5.8:** Worst case gain for the  $\mu$  controller found using the supply air temperature



# Chapter 6

## Test

Now that the controller has been obtained it can be tested on both the full simulation model and the actual Reefer system. But to be able to test the controller it must be setup to work with the systems.

First of all the controller must be scaled back to match the inputs and outputs for the systems. The controller is a two input two output controller. The two outputs from the controller are the control input for the expansion valve and the compressor speed. The two inputs for the controller are the supply air temperature and the superheat pressure. These were all scaled during the controller design, and therefore have to be scaled back to match this. These inputs and outputs can be unscaled using the scaling matrix found in Section 5.1 as follows.

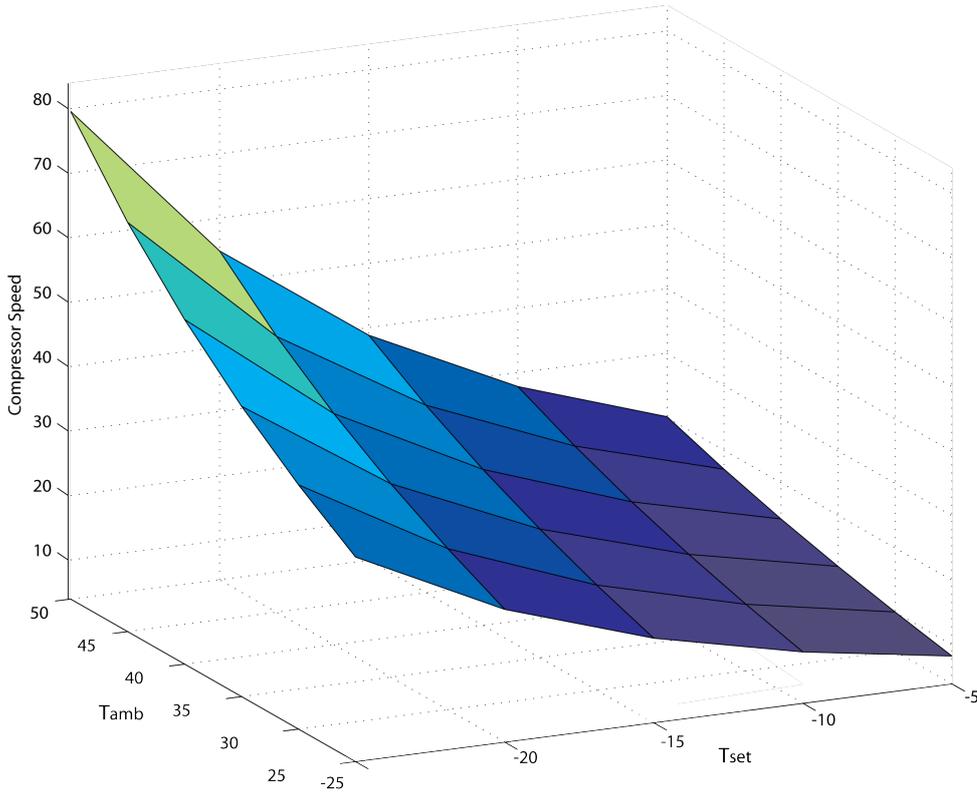
$$\hat{K}\mu = D_u K \mu D_e^{-1} \quad \text{where } D_u = \begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix} \quad \text{and } D_e = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.25 \end{bmatrix} \quad (6.1)$$

Secondly the operating point for the expansion valve and compressor speed in which the linear models have been linearized have to be included. These are important as they represent the steady state offset for the system depending on the set temperature and ambient temperature. As these were found with 5 degrees interval they have to be interpolated to be able to control the systems between the operating point. It was found that the operating points changed in a nonlinear way, this is shown for the compressor speed in Figure 6.1.

Thus to get a good estimation of the operating point it was chosen to interpolate the operating point using a second order polynomial of the following form.

$$\text{operating point} = A_0 + B_x T_{set} + C_x T_{set}^2 + B_y T_{amb} + C_y T_{amb}^2 + D T_{set} T_{amb} \quad (6.2)$$

The second order polynomial was chosen as a affine combination resulted in a large error in the center region of the operating points. The values for  $B_x$ ,  $C_x$ ,  $B_y$  and  $C_y$  in the function are found using the MatLab<sup>TM</sup> function *polyfit* which fits a polynomial of a desired order to a data set.



**Figure 6.1:** 3d plot of the compressor speed operating point as a function of ambient and set temperature

The values for the operating point was found to be the following :

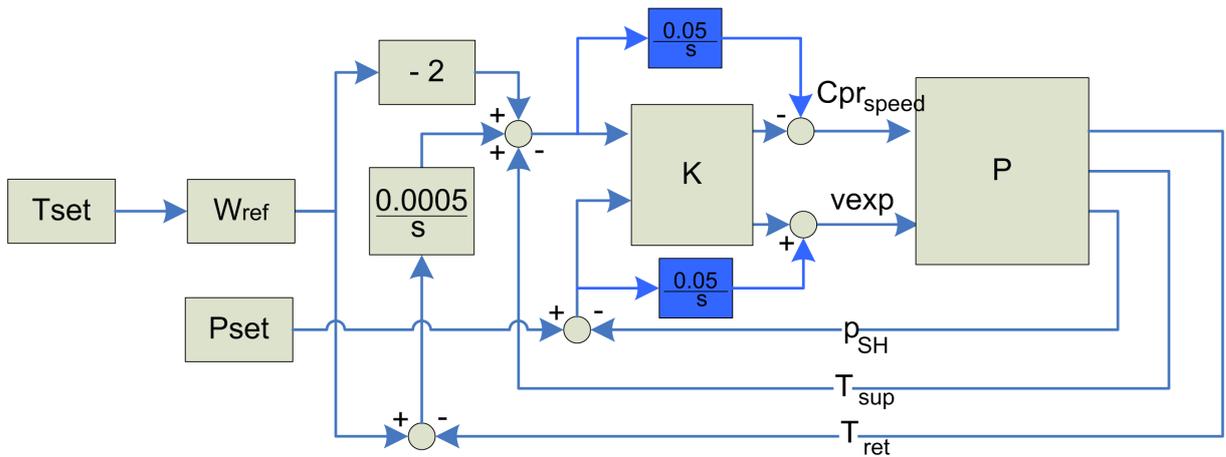
$$\begin{bmatrix} vexp_{op} \\ fcpr_{op} \end{bmatrix} = \begin{bmatrix} 5.55 \\ 8.24 \end{bmatrix} + \begin{bmatrix} 0.26 \\ 19.77 \end{bmatrix} T_{set} + \begin{bmatrix} 2.85 \\ 7.21 \end{bmatrix} T_{set}^2 + \begin{bmatrix} 0.65 \\ 4.12 \end{bmatrix} T_{amb} + \begin{bmatrix} 2.75 \\ 7.74 \end{bmatrix} T_{amb}^2 + \begin{bmatrix} 0.70 \\ 32.27 \end{bmatrix} T_{set} T_{amb} \quad (6.3)$$

Where  $\hat{T}_{set}$  and  $\hat{T}_{amb}$  are scaled to go from 0 to 1 with the following equation.

$$\hat{T}_{set} = \frac{(T_{set} + 5)}{-20} \quad \text{and} \quad \hat{T}_{amb} = \frac{(T_{amb} - 25)}{25}$$

As the calculated operating points might be a little away from the actual operating points for the simulation model, and for sure will differ from the real reefer systems operating points, two integral parts are added to correct the steady state error. This is shown in Figure 6.2 as the two blue integrators. The integrators has to integrate slowly enough to not interfere with the controller dynamics, but they should also be fast enough to correct errors. The delay is chosen to be  $\frac{0.05}{s}$  for both which should be slow enough to not interfere with the controller.

Thirdly to be able to control the whole system a extra part has to be added to the controller file to control the condenser fan and the opening valve for the economizer. The economizer is set to be  $\frac{2}{3}$  of the control input for the expansion valve opening. This value was found by Kresten K. Sørensen to give a reasonable result when running with the system. The condenser fan is controlled by the error in the refrigerant temperature in the condenser compared to the ambient temperature minus 11 degrees Celsius. The condenser fan speed has three steps, full speed, medium speed or turned off which it changes between compared to the magnitude of the error.



**Figure 6.2:** Control configuration for supply air temperature, the blue integrators are the ones added to correct steady state errors

The MatLab<sup>TM</sup> code needed to control the condenser fan are supplied by Kresten K. Sørensen. The speed of the evaporator fan is set to medium speed as it was during location of the operating point and linearization.

Fourthly the control signals from the controller to the system has to be rounded to nearest whole integer and limited to be within the actuator control range with is 0 to 100 for the expansion valve and 0 to 110Hz for the compressor speed.

Lastly to be able to use the supply air temperature as measurement instead of the return air temperature, a coupling between the return air temperature and the reference for the supply air temperature is needed. This is done as it still is the return air temperature that is to be controlled to match the set temperature. This will be setup on the form shown in Figure 6.2.

As it can be seen in Figure 6.2 the set temperature are shaped with the reference weight. This will at startup be set to the set temperature, but when the temperature is changed during a run the change in the reference will follow the defined weight function. This is done to match the requirements for the controller design. The reference for the supply air temperature are offset with -2 degrees Celsius as it, from the found operating points, was observed that the supply air temperature on average is -2 degrees Celsius lower than the return air temperature.

But to ensure that it is the return air temperature that is kept at the set temperature the extra integral part on the error from the return air temperature are added. This integrator are chosen sufficiently slow to not interfere with the dynamics of the controller or the integrator to correct the compressor speed steady state error. To ensure anti windup of the integrator it is limited to  $\pm 2$  degrees Celsius. This should be sufficient to correct the supply air to give the desired return air temperature.

This controller structure are implemented in the controller file for the simulation model and reefer system.

With all of this setup the controller is ready to be tested on the simulated system and the reefer system. Before performing the actual test the test objectives will be outlined, this will be done in the following.

## 6.1 Test Objectives

To verify that the controller fulfills the given requirements three things has to be tested. First the ability for the controller to reach and maintain a given temperature. Secondly be able to suppress disturbances from changes in the ambient temperature. And thirdly it should be tested how the controller handles a change in the reference. In all three cases the set temperature inside the box should be tested at all set points from -5 to -25. It is assumed that it is sufficient to do this with 5 degrees interval for the set temperature.

During the tests the controller should at all times maintain a super heat pressure between  $0.25 \pm 0.05$ , to ensure that the refrigerant are on gas form before reaching the compressor and to ensure a good cooling efficiency of the evaporator. The other requirement is that the return air temperature has to be maintained between  $\pm 0.5$ degrees from the set temperature.

To test all of these requirements three test are to be performed. The first test will be performed as follows :

- Initialize the system at +2 degree Celsius compared to the desired set point temperature and set the ambient temperature to 30 degrees Celsius.
- Start the simulation with the controller and let the system stabilize.
- Validate that the system were able to reach and maintain the set temperature within reasonable time and that the superheat pressure are kept within the allowed boundaries.

The second test will be performed as follows :

- Initialize the system to the desired set point temperature and start the ambient temperature at 50 degrees Celsius.
- Start the simulation with the controller and while it is running change the ambient temperature with the following function for the simulation model :  $T_{amb} = 37.5 + 12.5 \cdot \cos(\frac{t}{86400} \cdot 2\pi)$ .
- Let the simulation run for 86400 seconds(24 hour) thus letting the ambient temperature change from 50 degrees Celsius, down to 25 degrees Celsius and back up to 50 degrees Celsius following the cosine function.
- Validate that the system were able maintain the set temperature and the superheat pressure during the whole run.

The third test are performed to study how the system handle a change in the reference. This test will be performed as follows :

- Initialize the system at the decried set point temperature and set the ambient temperature to 30 degrees Celsius.
- Start the simulation with the controller and let the system stabilize.

- When the system is stable change the reference with -2 degrees Celsius.
- Let the system stabilize at the new temperature.
- Validate that the system were able to reach and maintain the set temperature within reasonable time and that the superheat pressure are kept within the allowed boundaries.

Now that the test have been outlined they can be performed on the simulated model and the reefer system.

## 6.2 Simulation Test

To test if the controller behaves as expected the three mentioned tests are performed on the full simulation model. The full simulation model is chosen as it is more representative for the actual system than the reduced simulation model. Thus it is expected that if it performs well on the full simulation model it will also performs well on the actual reefer system. It is decided to run the simulation model with a small cargo load of 20Kg as this better will represent the actual reefer system which has the same small load. This will have influence in the sens that the system will respond to changes in the temperature as the smaller cargo load will change temperature faster.

### 6.2.1 Test 1 - Reach and Maintain Temperature

The objective of test 1 is to observe how fast the controller reaches and maintains a set temperature when the system is initialized at +2 degrees Celsius away from the set temperate. It should also be observed how the controller is able to keep the superheat pressure during the test. The test is as mentioned done for the five set temperatures, -5, -10, -15,-20 and -25 degrees Celsius where the ambient temperature is set to 30 degree Celsius. The simulation systems states are initialize with the set temperature +2 degrees Celsius for the box temperatures. The simulation is run for 10000 seconds at each set temperature. It is expected that the system will reach and stabilize at the set temperature within 60 minutes. Further it is expected that the superheat pressure might be off at startup, but is should quickly reach a superheat pressure of 0.25 and remain stable within  $\pm 0.05$ . The superheat pressure during the five tests is shown in Figure 6.3 and the box temperatures is displayed in Figure 6.4.

As it can be seen from the two figures, it takes the controller around a hour to decrease the temperature the two degree Celsius to the set temperature and it does a small overshoot on the return air temperature. The overshoot was expected as the supply air error also is driven by the integrator on the return air temperature error. In Figure 6.4 the reference for the supply air is also shown, and from this it can be noticed that the controller is unable to track the reference perfectly. Notice that the requirements allow the temperature to vary with up to  $\pm 0.5$  degrees Celsius. To examine if the controller is able to this the maximum and mean difference between the reference for the supply air temperature are calculated and shown in Table 6.1. The same table also shows the minimum return air temperature where the overshoot for each test can be noticed.

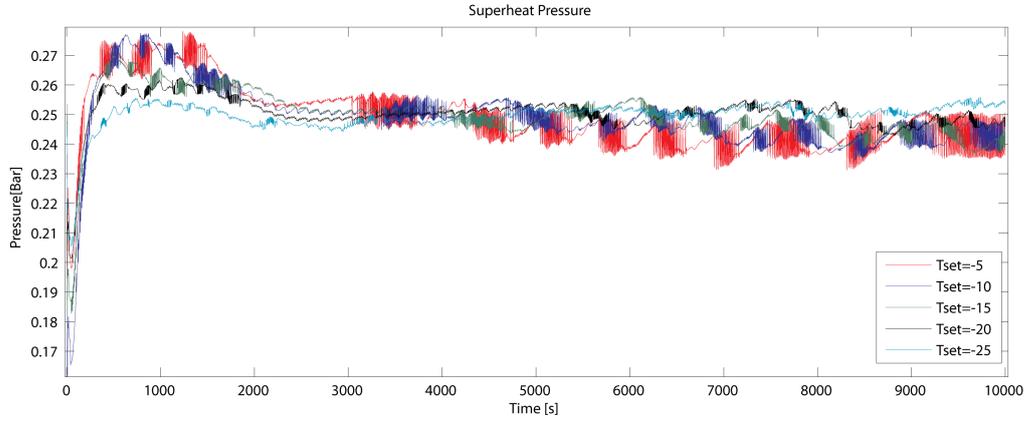


Figure 6.3: The superheat pressure during the five simulations

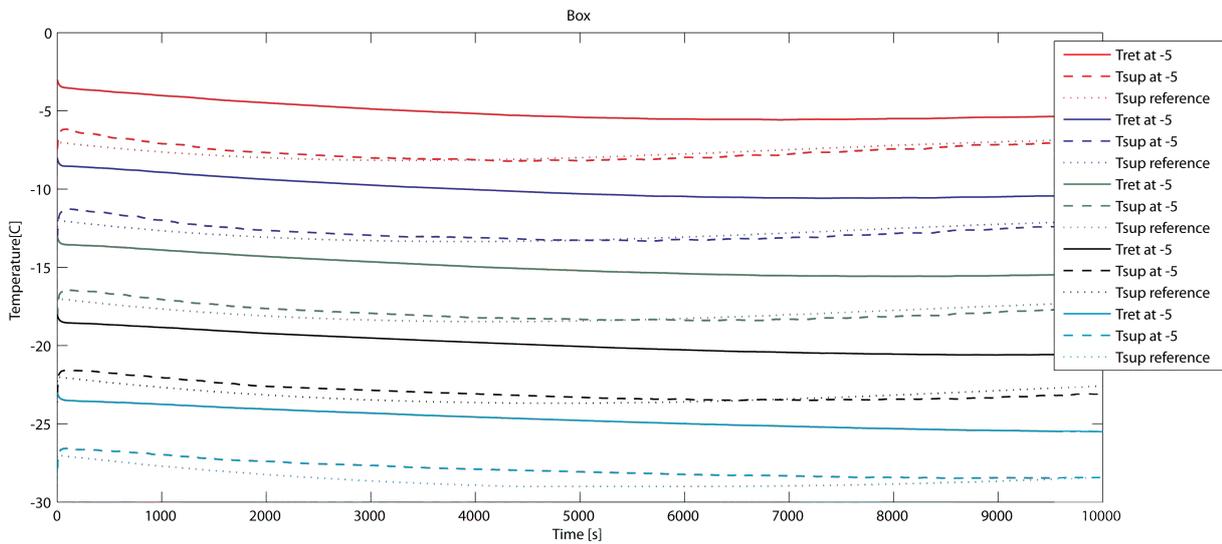
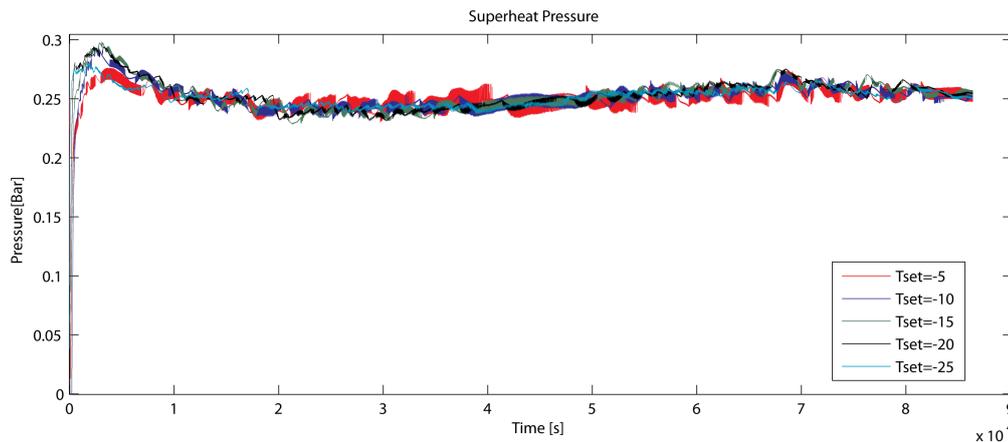


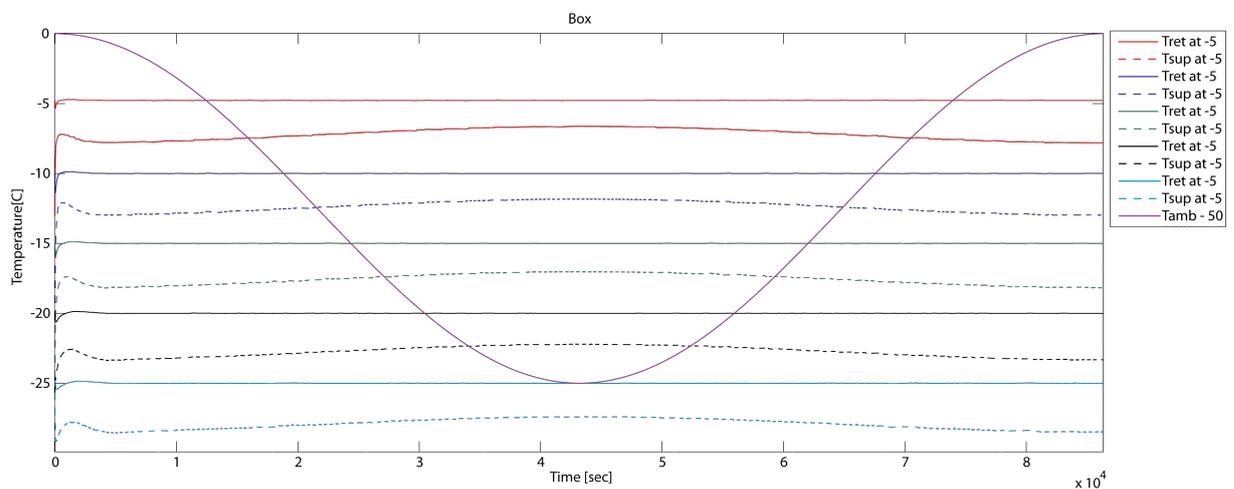
Figure 6.4: The temperatures in the box during the five simulations together with the reference for the supply air

Set temperature	Max variation	Mean variation	Min <i>Tret</i> temperature
$-5^{\circ}C$	$0.40^{\circ}C$	$-0.02^{\circ}C$	$-5.57^{\circ}C$
$-10^{\circ}C$	$1.06^{\circ}C$	$-0.08^{\circ}C$	$-10.59^{\circ}C$
$-15^{\circ}C$	$1.16^{\circ}C$	$-0.08^{\circ}C$	$-15.57^{\circ}C$
$-20^{\circ}C$	$1.06^{\circ}C$	$-0.21^{\circ}C$	$-20.60^{\circ}C$
$-24^{\circ}C$	$0.86^{\circ}C$	$-0.68^{\circ}C$	$-25.50^{\circ}C$

Table 6.1: Maximum and mean difference between the supply air temperature and the reference together with the minimum return air temperature



**Figure 6.5:** The superheat pressure during Test 2



**Figure 6.6:** The return and supply air temperature together with the change in the ambient temperature during Test 2. The ambient temperature is subtracted by 50 degree Celsius to better compare the results

The superheat during all five sub test is maintained well within  $\pm 0.05$ . Thus during this test on the simulation model the controller is able to keep the superheat well within the requirement.

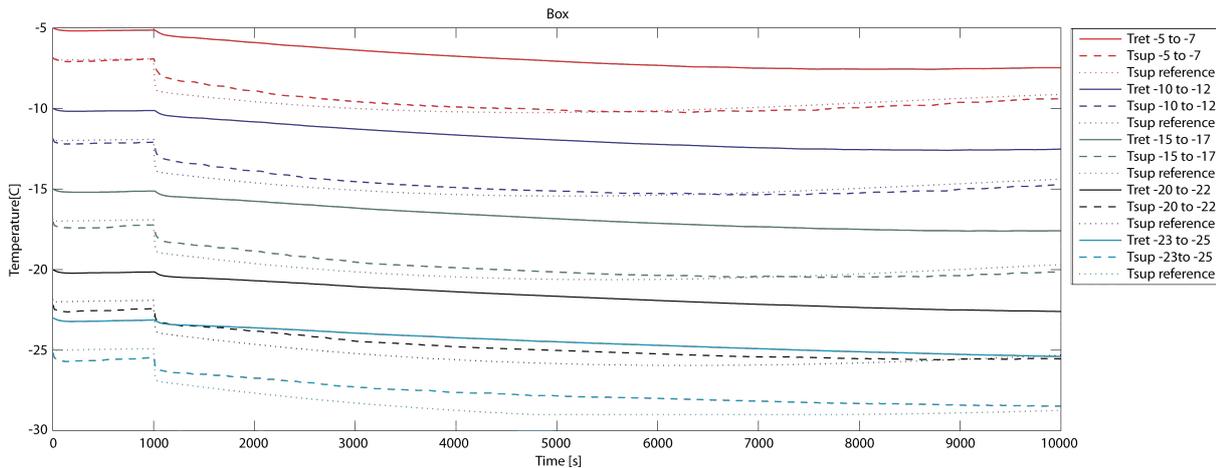
### 6.2.2 Test 2 - Reject disturbance

This test consist as test one of five sub tests with five different set temperatures but with the second test objective listed in Section 6.1. The results from all five sub test are shown in Figure 6.5 for the superheat pressure and in Figure 6.6 for the temperatures. It is noticed that the system at startup has to settle at the set temperate and once the system has stabilized the controller is doing a good job at keeping the superheat pressure and return air temperature stable. The superheat pressure which is shown in Figure 6.5 is kept well within  $\pm 0.05$  during the entire test.

From the temperatures shown in Figure 6.6 it can be noticed how the supply air is lowered when the ambient air is low as the system does not need to supply as much cold air to keep the return

Set temperature	Min $T_{ret}$ temperature	Max $T_{ret}$ temperature
$-5^{\circ}C$	$-4.93^{\circ}C$	$-5.02^{\circ}C$
$-10^{\circ}C$	$-9.87^{\circ}C$	$-10.02^{\circ}C$
$-15^{\circ}C$	$-14.88^{\circ}C$	$-15.09^{\circ}C$
$-20^{\circ}C$	$-19.87^{\circ}C$	$-20.18^{\circ}C$
$-24^{\circ}C$	$-24.86^{\circ}C$	$-25.21^{\circ}C$

**Table 6.2:** Minimum and Maximum return air temperature during test 2 after the first 500 seconds



**Figure 6.7:** The temperatures during the five simulations together with the reference for the supply air

air temperature. The change in the supply air temperature was measured to be approximately 1.2 degree Celsius in all five cases. The return air temperature is at startup changing a little, but after 500 seconds it is settled at the set temperature and vary very little. The minimum and maximum return air temperatures after the first 500 seconds are shown in Table 6.2. From the table it can be noticed that the temperature maximum vary with 0.21 degree Celsius after the first 500 seconds, which is well within the requirements.

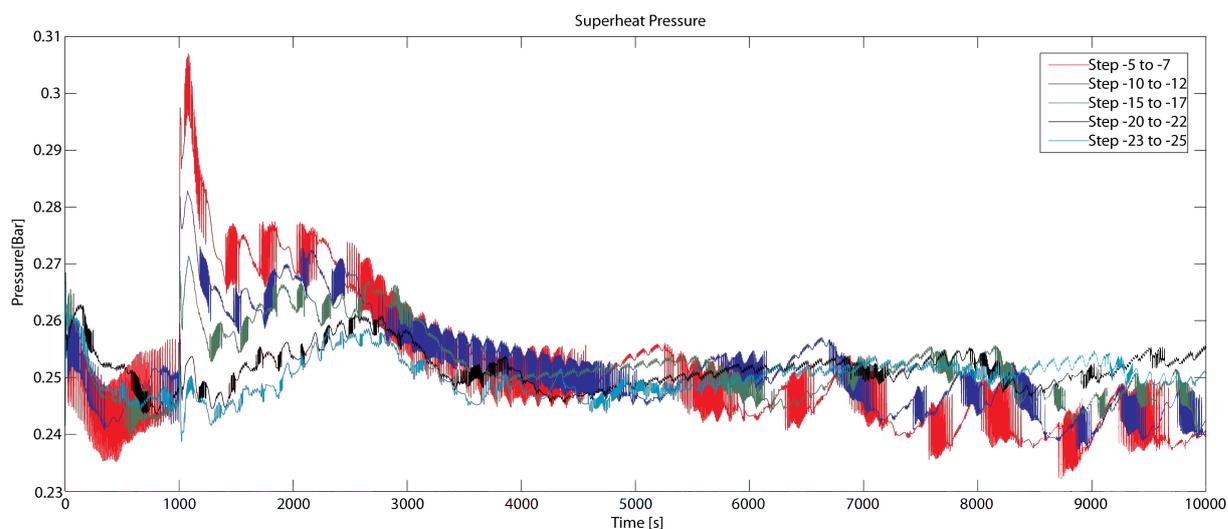
### 6.2.3 Test 3 - Change Reference

The last test is as mentioned there to observe how well the system handles a change in the reference temperature. This will again be tested for five different temperatures, but to not exceed -25 degrees Celsius the last step will be done from -23 to -25 degrees Celsius. The tests are performed as defined in Section 6.1, with an ambient temperature of -30 degree Celsius. The resulting temperatures and their references for the supply air temperature are shown in Figure 6.7 and the superheat pressure shown in Figure 6.8.

The temperatures shown in Figure 6.7 indicates that the system uses a longer time to reach the set temperature after the reference change. The time vary from 1 hour to 1 hour and 45 minutes, before the return air temperature is equal to the new set temperature. The lower the temperature is the longer it takes for the controller to adjust to the new set temperature. The supply air temperature looks like it vary a lot from the reference, but it is in fact at most 0.73

Set temperature	Max variation	Mean variation	Min $T_{ret}$ temperature
$-5^{\circ}C$ to $-7^{\circ}C$	$0.36^{\circ}C$	$-0.09^{\circ}C$	$-7.58^{\circ}C$
$-10^{\circ}C$ to $-12^{\circ}C$	$0.41^{\circ}C$	$-0.13^{\circ}C$	$-12.60^{\circ}C$
$-15^{\circ}C$ to $-17^{\circ}C$	$0.46^{\circ}C$	$-0.19^{\circ}C$	$-17.61^{\circ}C$
$-20^{\circ}C$ to $-22^{\circ}C$	$0.64^{\circ}C$	$-0.44^{\circ}C$	$-22.60^{\circ}C$
$-23^{\circ}C$ to $-25^{\circ}C$	$0.73^{\circ}C$	$-0.71^{\circ}C$	$-25.39^{\circ}C$

**Table 6.3:** Maximum and mean difference between the supply air temperature and the reference together with the minimum return air temperature



**Figure 6.8:** The superheat pressure during the five simulations. The  $-2$  degrees Celsius reference change is applied after 1000 seconds

degree Celsius away from the reference. This is shown in Table 6.3 where the maximum and mean difference between the supply air temperature and the reference are shown. The minimum return air temperature is also shown in the table from which it can be noticed that the controller again overshoot the set temperature with roughly 0.7 degrees Celsius.

From the superheat pressure shown in Figure 6.8 it can be noticed that when the step is applied the superheat pressure increase. It can also be noticed that it increases more at high temperatures than at low temperatures. The superheat pressure at the step from  $-5$  to  $-7$  degrees Celsius actually differ with more than the allowed  $\pm 0.05$  Bar, while at the step from  $-23$  to  $-25$  it differ with less than  $\pm 0.01$ .

## 6.3 Reefer Test

To test the controller on the actual reefer an interface developed by Lodam called MaSCHIL is used. The interface is a MatLab<sup>TM</sup> based tool which can control the reefer through a tool connected to the computer. The tool enables the user to display the measured data while running the system and load different controllers at run-time. The controllers can be written

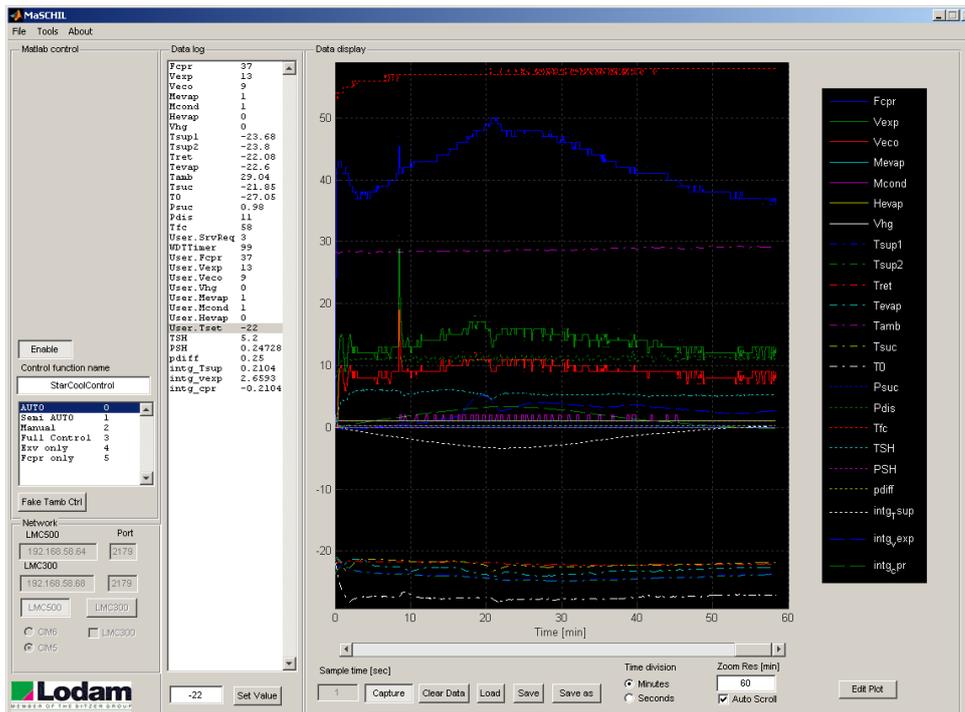


Figure 6.9: Screen shot of the interface for the reefer test system

in MatLab<sup>TM</sup> code and executed on the host PC given the measurements received and then change the control inputs on the reefer based on the outputs from the controller. This makes way for easy development and testing of new controllers for the reefer system. A screen shot of the software is shown in Figure 6.9.

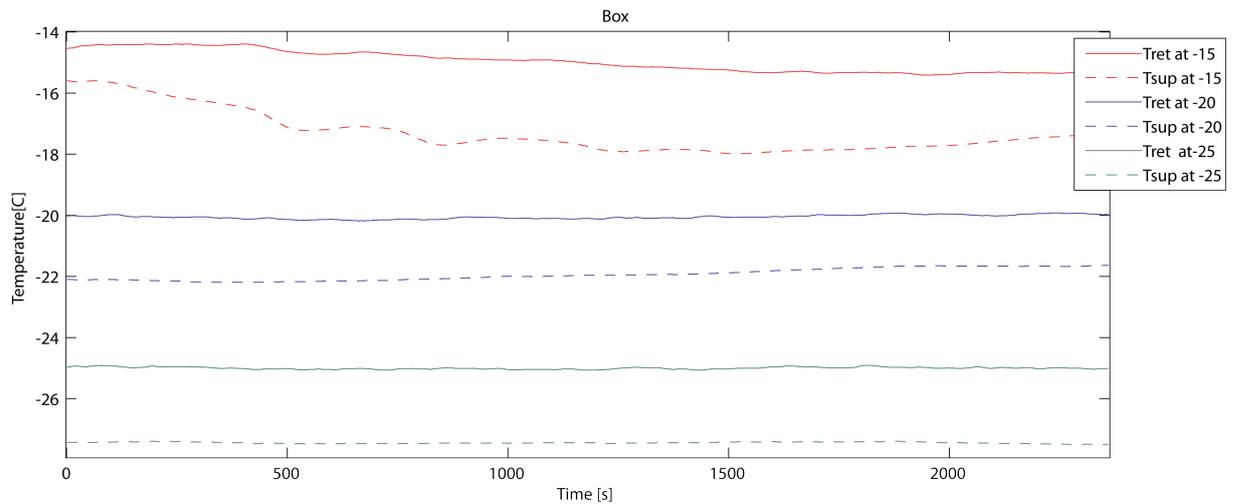
The reefer container on which the tests is performed is located at Lodam in Sønderbog where a server is connected to the system.

It was decided for the test on the reefer system to only use a set temperature between -15 to -25 as it was found that with a lower set temperature the system comes to close too a compressor speed of 20Hz where the system goes into PWM mode.

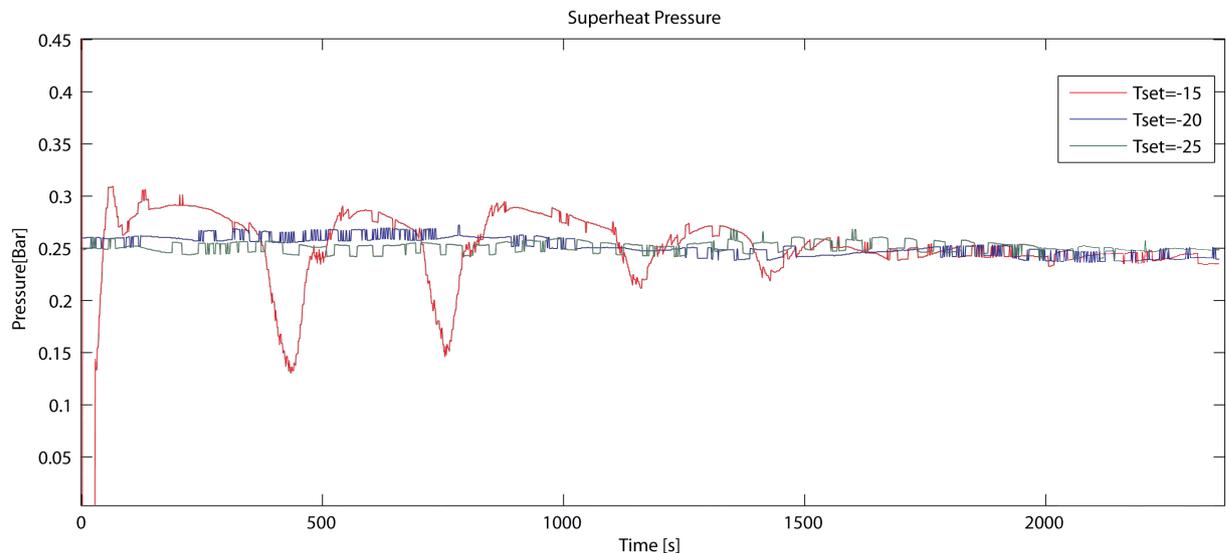
The reefer system is difficult to start with different conditions and therefore some of the test conditions from Section 6.1 were changed. If a test is modified how and what is changed is outlined in the test description.

### 6.3.1 Test 1 - Reach and Maintain Temperature

Test 1 consists of setting up and starting the system 2 degree Celsius off the set temperature and observe how the controller manage to adjust the system temperature. As this is difficult to setup, it was decided to use the controller developed by Lodam to get the wanted set temperature and then switch to the designed controller for the test. The objective is then for the controller to reach the set temperature and stabilize from wherever it is initiated. This was performed with a set temperature at -15, -20 and -25 degree Celsius. During the tests the ambient temperature around the reefer was between 27 to 28 degrees Celsius.



**Figure 6.10:** Measured supply air temperatures and return air temperatures during the three tests

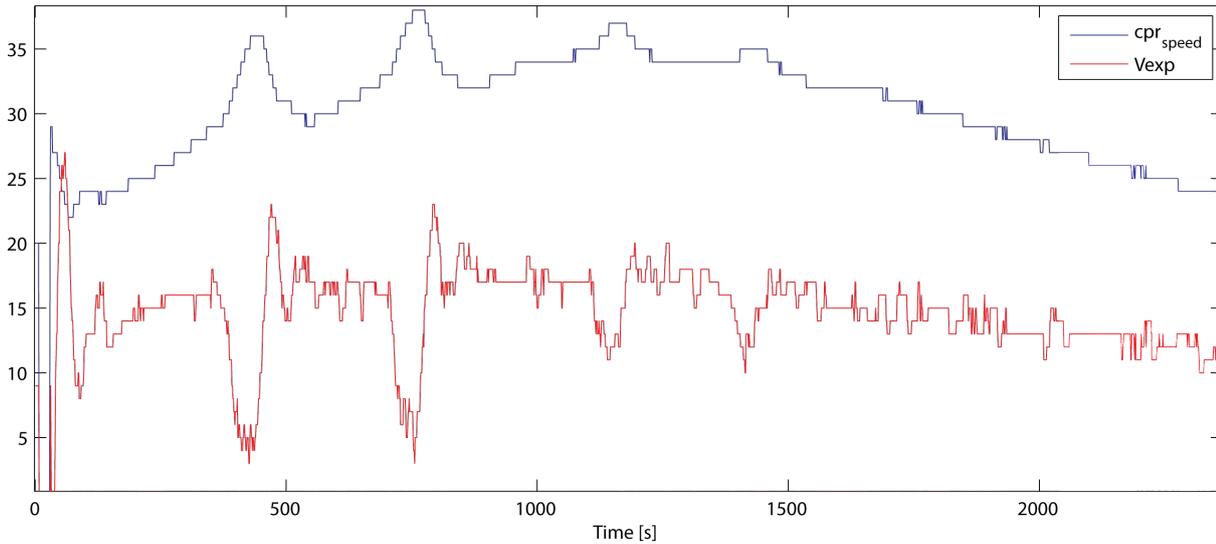


**Figure 6.11:** Measured superheat pressure during the three tests

As the measurements last over a long time period only parts of it will be included to better compare the results. For this test approximately 40 minutes of the measurements are shown. The measured temperatures for the three tests are shown in Figure 6.10 and the measured superheat in Figure 6.11.

The controller clearly changes the supply air temperature to maintain the return air temperature at the set level and once the controller has reached the set temperature it also stabilizes. It has however been found that when the controller has to change the temperature for some set temperatures like at  $T_{set} = -15$ , the controller cannot always maintain the superheat pressure. This can be seen in Figure 6.11 where the superheat pressure, for the test with  $T_{set} = -15$ , drops when the system has to deliver colder supply air to lower the return air temperature.

From Figure 6.11 it is also noticed that the controller is quick to counteract the drop in the superheat pressure. This is because the controller is able to utilize both actuators to work



*Figure 6.12: Control inputs during test 1 with set temperature -15*

against this effect. This can be observed by looking at the control inputs during this time period for the test with  $T_{set} = -15$ . These are shown in Figure 6.12.

The controller performed well once the set temperature are reached which can be seen from the other measured temperatures and superheat pressures.

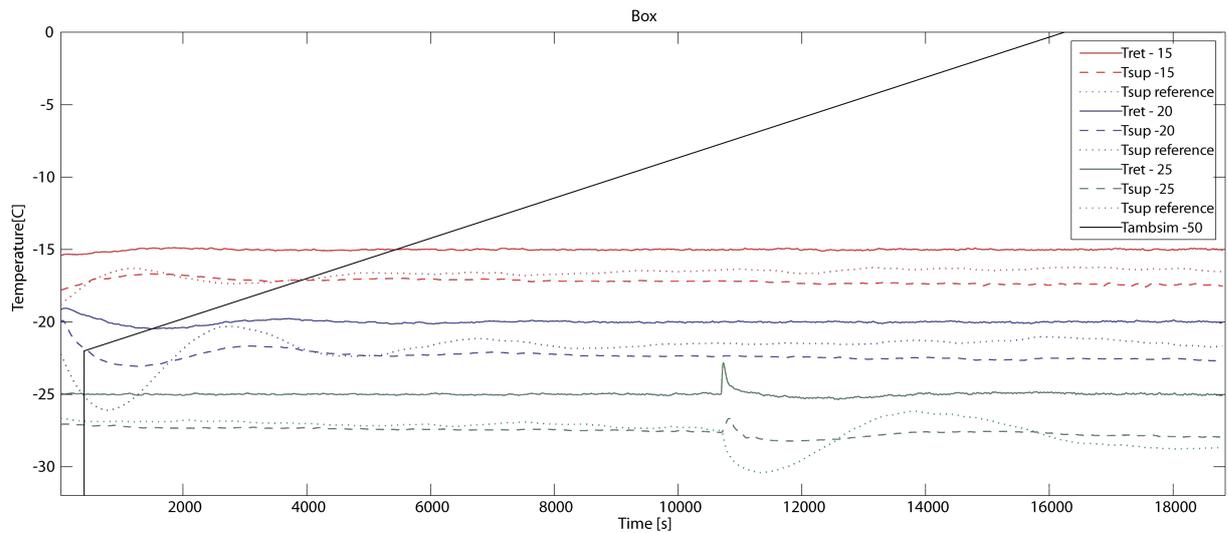
### 6.3.2 Test 2 - Reject Disturbance

To test how well the controller rejects disturbances on the reefer system a simulated ambient temperature was needed. This was done by installing a heating fan inside the reefer, to simulate the disturbance in the return air. The controller for the condenser was also modified to let the temperature of the refrigerant in the condenser rise to match simulated ambient reference. With these two things added the system was able to simulate a change in the ambient temperature.

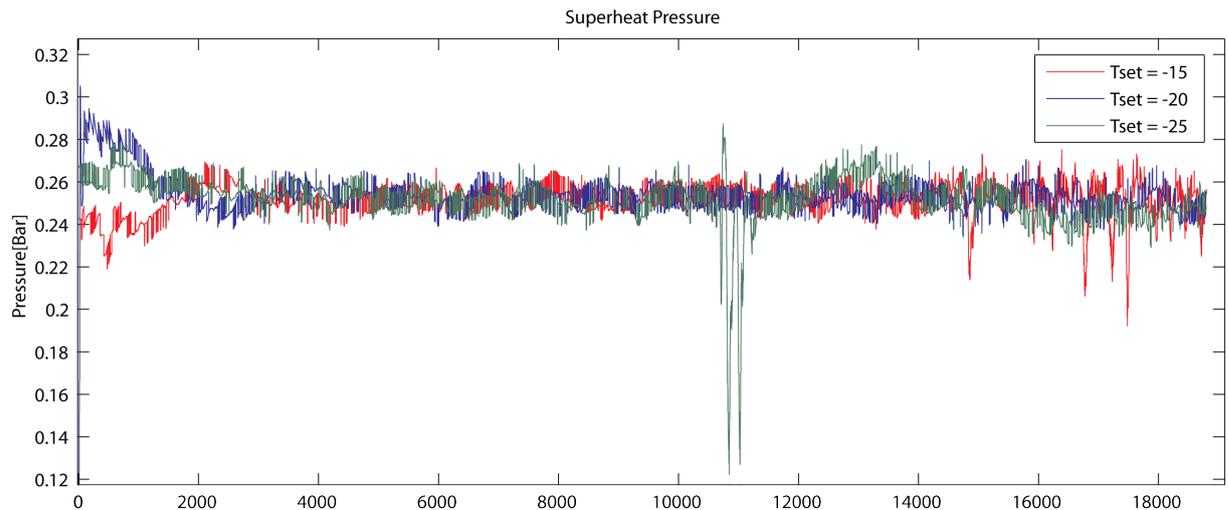
It was decided that the second test would take too long to run on the reefer system. Therefore the test was modified to change the ambient temperature with a ramp function instead and change the ambient temperature with 5 degrees Celsius per hour. This reduced the test to take 5 hour instead of 24. The ambient temperature was therefore set to follow the function :  $T_{ambsim} = 28 + \frac{5}{3600} \cdot t$ . The simulated ambient temperature was set to start at 28 degrees Celsius as the real ambient temperature when the test was conducted was around 28 degrees Celsius.

The results from the tests in form of the measure temperatures and superheat pressures are shown in Figure 6.13 and Figure 6.14 respectively.

From the measured temperatures it can be observed that the system did not settled in all cases before the test is initiated, but the system still settles at the set temperature after a short while. The temperature for the test at  $T_{set} = -25$  also suddenly drops. This is because it loses the superheat pressure for a short while but manages to stabilize it again shortly after. This can also be observed in Figure 6.14. Another thing that can be noticed from Figure 6.13 is that the



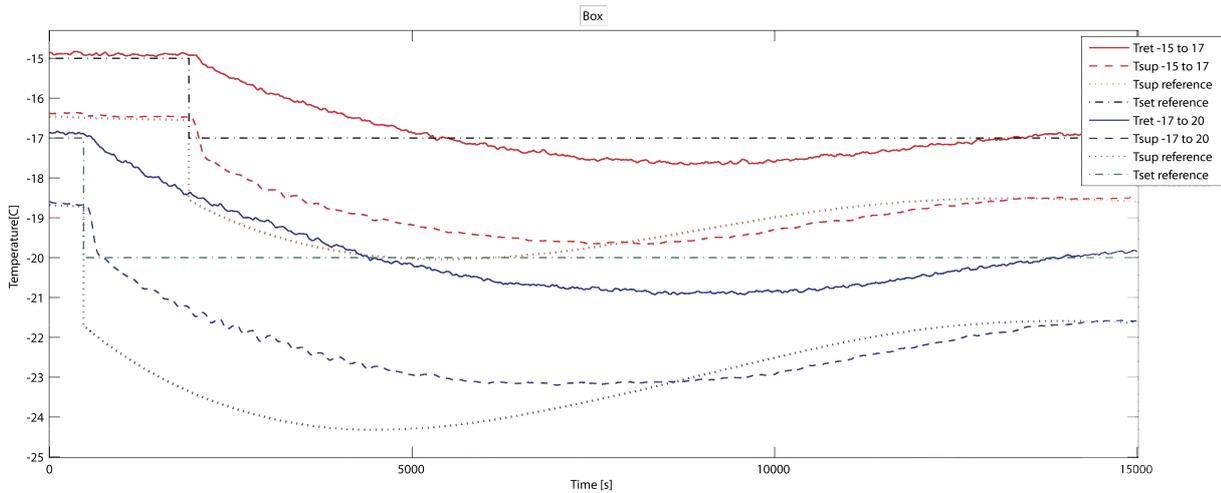
**Figure 6.13:** Measured temperatures, reference and simulated ambient reference during the three tests



**Figure 6.14:** Measured superheat pressure during the three tests

reference for the supply air drifts away, and the supply air does not follow. This was found to be because the controller configuration for this test was not setup correctly as the integrator which should correct a steady state error on the supply air temperature was connected to the error on the return air temperature instead. The steady state error was therefore never suppressed. It has been shown afterwards that by setting the controller configuration up correctly this does not happen any more and the error in this case is therefore accepted.

For Figure 6.14 it can be observed that the controller is able to keep the superheat steady at 0.25 Bar except for the small glitch for the  $T_{set} = -25$  test.



**Figure 6.15:** Measured temperatures, reference and set temperature during the two reference change tests

Set temperature	Max variation	Mean variation	Min $T_{ret}$ temperature
$-15^{\circ}C$ to $-17^{\circ}C$	$2.10^{\circ}C$	$-0.25^{\circ}C$	$-17.67^{\circ}C$
$-17^{\circ}C$ to $-20^{\circ}C$	$3.080^{\circ}C$	$-0.64^{\circ}C$	$-20.93^{\circ}C$

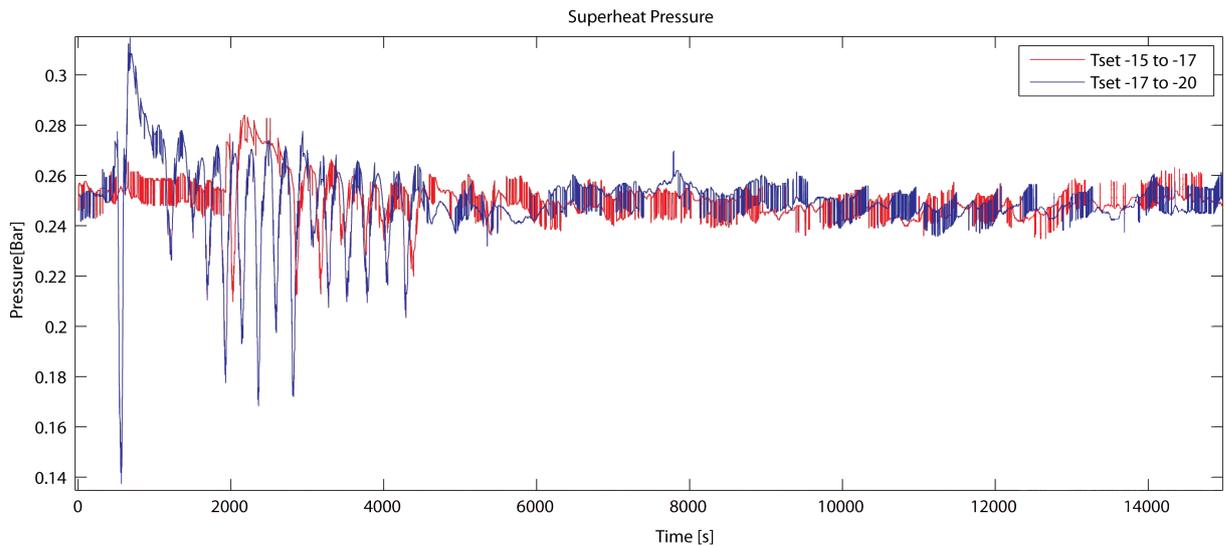
**Table 6.4:** Maximum and mean difference between the supply air temperature and the reference together with the minimum return air temperature

### 6.3.3 Test 3 - Change Reference

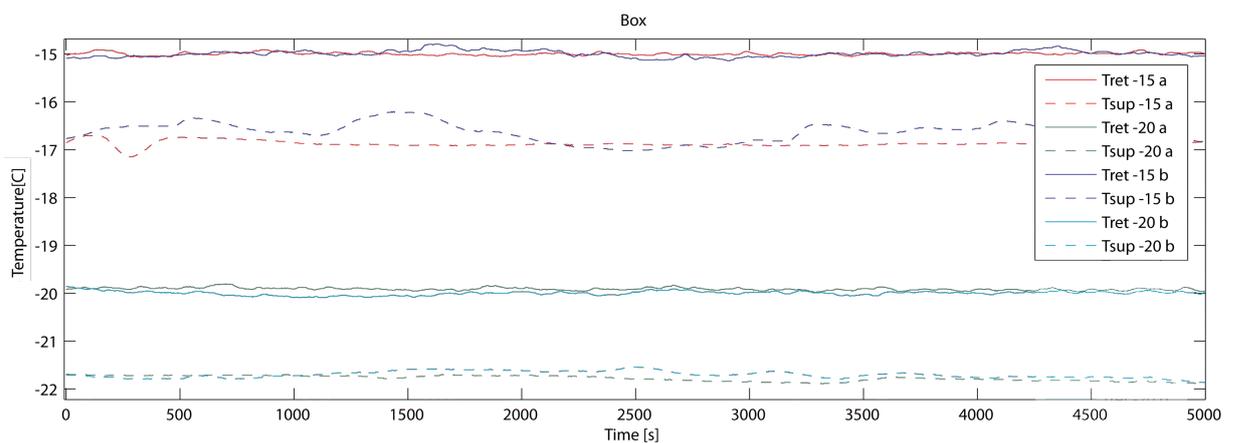
This test was performed more according to the original specifications. The steps are however a bit different. The two steps performed are from  $-15$  to  $-17$  degrees Celsius and from  $-17$  to  $-20$  degrees Celsius. The measured temperatures and references during the two tests are shown in Figure 6.15. The measured superheat are shown in Figure 6.16. The ambient temperature around the reefer during these two tests were between  $27$  to  $29$  degrees Celsius.

In both tests it took the system around a hour to lower the return air temperature to the set temperature. The return air again overshoots the set temperature. The supply air also has problems following the reference as in the simulated tests. The maximum and mean differences between the supply air temperature and the supply air reference are shown in Table 6.4 together with the minimum return air temperature. The table shows that the supply air temperature is at maximum 3 degrees away from the reference. From the table is can also be noticed that with this controller configuration the return air overshoots with almost one degree Celsius with a step of 3 degrees Celsius while the 2 degrees step only overshoots with 0.7.

From Figure 6.16 it can be noticed that the controller has some problems keeping the superheat within the requirements for the 3 degrees step while it manages to keep the superheat within the requirements for the 2 degrees step.



*Figure 6.16: Measured superheat pressure during the two step tests*

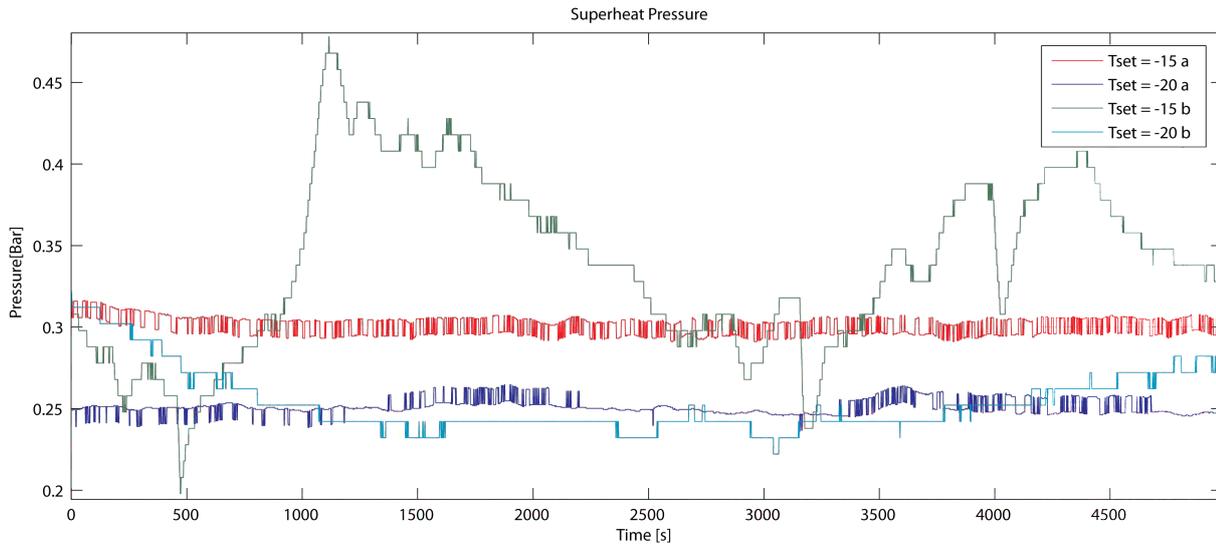


*Figure 6.17: Measured temperatures during the tests*

### 6.3.4 Test 4 - Comparison

An extra test is performed on the reefer to compare how the controller developed by Lodam handles the system in this temperature region compared to the one designed in this project. The comparison will be based on how well the two controllers are at maintaining a set temperature and superheat pressure when stabilized at the set temperature. There are two tests performed, one at a set temperature of -15 degrees Celsius and one at a set temperature of -20 degrees Celsius. To be able to separate the results they will be marked with a 'a' when the controller designed for this project is used and a 'b' when the controller developed by Lodam is used.

The results of the two tests are shown in Figure 6.16 for the superheat pressure and in Figure 6.17 for the temperatures. As it can be noticed from the temperatures, the developed controller is better at keeping a steady return temperature and supply air temperature compared to the Lodam controller. Both controllers are, however, able to keep the return air temperature within the requirements.



**Figure 6.18:** Measured superheat pressure during the tests

As it can be seen in Figure 6.18 the superheat is raised to 0.3 Bar when running with  $T_{set} = -15$ . This was done to get the superheat pressure to settle as it with a reference at 0.25 were not able to maintain a stable superheat pressure. But otherwise it can be noticed that the developed controller is much better at keeping the superheat pressure stable. The Lodam controller fluctuate allot more to maintain the superheat pressure.

## 6.4 Test Conclusion

From the performed tests it can be concluded that the developed controller is really good at keeping a set temperature and a superheat pressure when settled at the set temperature. It has been noticed that the small superheat pressure some times causes the superheat to drop. It has however been found that by increasing the superheat reference slightly the controller is able to keep the superheat pressure stable.

The problems with keeping the superheat pressure occur mainly in the time periods where the evaporator has to deliver a higher cooling capacity. This can be noticed from the reference change test performed on the reefer system where the superheat pressure starts to fluctuate after the step as it has to deliver a higher cooling capacity. It is assumed that by raising the superheat reference in these periods the drop in the superheat pressure could be avoided. If this were implemented it is also assumed that larger steps in the set temperature reference can be performed without the superheat pressure dropping.

The controller is with the current controller setup able to handle changes in the reference, but not with a larger step size than 2-3 degrees. During the reference change test the controller does a good job of controlling the supply air temperature to follow the reference.

In the tests where the disturbance is added the controller does a great job at keeping the return air temperature and superheat pressure stable. During these tests the controller is able to keep the return air temperature and superheat pressure well within the requirements.

The model based controller approach used has shown to be successful for this problem because both actuators are used to take cross-coupling in the system into account and therefore correct an error in the superheat pressure fast and stabilize faster.



## Conclusion

The objective for this project was to investigate if a model based robust controller could be designed with focus on controlling the compressor speed and expansion valve to ensure good performance and stability of the evaporator under a set of working conditions for a reefer system.

To achieve this an existing modular model presented by Kresten K. Sørensen and Jakob Stoustrup in the paper 'Modular Modelling and Simulation Approach - Applied to Refrigeration Systems' was used [SS08]. The model is a full system model which included all parts of the reefer system. It is decided to reduce the model to only include the dynamics directly affecting the evaporator to get a reasonable model size for the later controller design.

To be able to reduce the model the missing inputs from the removed components had to be simulated. It was found that discrete functions depending on only the set temperature and ambient temperature is sufficient as the dynamic influence of the simulated signals is neglectable.

The reduced simulation model was then used to identify the operating points needed to obtain the set of linear models derived from the simulation model. It was then attempted to further reduce the linear model, as the later controller design algorithms cannot handle a model with too many states and uncertainty perturbations. This was first attempted by using the two methods *balanced model truncation via square root* and *Hankel minimum degree approximation*. It was however concluded that these two methods result in an unwanted transformation of the state space realization of the models, which made it difficult in practice to compare the linear models directly afterwards. It was therefore decided to try to remove states directly and examine if the state had any influence on the system dynamics. It was found that 4 of the states had no influence and they were therefore removed.

From the found set of linear models a parametric uncertainty model was set up using interpolation of the found linear models. It was found that a parametric uncertainty model which covers the set of linear models really well could be constructed.

The standard problem formulation for the  $H_\infty$  controller design was then set up together with the needed performance weight after which a  $H_\infty$  controller using D-K iterations was synthesized. It was through analysis found that the controller does not achieve robust performance but

guarantees robust stability for the given uncertainty model.

To examine how well the controller performed a set of tests were conducted on both the full simulation model and a real reefer system. From the tests it was concluded that the controller performs really well. It did in some cases, when the cooling capacity was increased, have problems keeping the superheat pressure. The test comparing the currently used controller and the one developed in this project further demonstrated that the developed controller performs much better than the one currently used.

Thus the project has shown that it is possible to obtain a more stable and better performing controller by using a model based robust controller design. An even better performance might still be achieved. Proposals on how to achieve this will be presented in the following chapter.

## Future Perspectives

As it was noticed from the test the controller had problems keeping the superheat pressure when the cooling capacity was increased. To avoid this problem it is suggested to do gain scheduling of the superheat pressure reference based upon the current cooling capacity of the evaporator. Thus when running with a high cooling capacity the reference should be raised and when running with a low cooling capacity the reference can be lowered. Thus helping the controller to keep a more stable superheat pressure in all conditions.

At project start-up it was decided to limit the working conditions of the controller to primarily handle disturbances in the form of changes in the ambient temperature. But a task that might be obvious to also include is the process of de-icing the evaporator. This has to be done from time to time to ensure good performance of the evaporator, especially if the load in the reefer is very humid. To do this the heater is turned on, which increases the load on the evaporator. During this task the supply air should still be kept low, thus this is a large disturbance that the controller has to oppose. For this task the development of a robust controller would be applicable.

Another thing that should be included in the controller design is the fact that the fan mounted on the evaporator in this case always is set to medium speed, but that it in some cases might be desirable to set it to high speed. This should be included in the robust controller design to ensure that it can also handle these conditions.

When applying robust controller design it could also be interesting to include facts like a loose temperature sensor on the evaporator and attempt to design a robust controller that can handle these cases as this will change the dynamics of the system. Thus ensuring system stability even if the sensor is loose.



# Bibliography

- [Jon91] Sven Jonsson. Performance simulations of twin-screw compressors with economizer. *International Journal of Refrigeration*, 14:345–350, 11 1991. ISSN: 0140-7007.
- [Mae08] Maersk. Maersk box. <http://www.maerskbox.com/>, 2008.
- [SBW] Richard E. Sonntag, Claus Borgnakke, and Gordon J. Van Wylen. *Fundamentals of Thermodynamics*. Sixth edition.
- [SP05] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control*. John Wiley & Sons inc., 2005.
- [SS08] K.K. Sørensen and J. Stoustrup. Modular modelling and simulation approach - applied to refrigeration systems. In *Proceedings of the 2008 IEEE Multi-conference on Systems and Control*, San Antonio, Texas, USA, September 2008. DOI:<http://dx.doi.org/10.1109/CCA.2008.4629691>.
- [TM] Inc. The MathWorks. Matlab & simulink. <http://www.mathworks.com/>.



Part I

Appendix



## Uncertainty Model - Extrapolation

As the first attempts to get a parametric uncertainty model using interpolation did not show as good results as expected the method of extrapolating the model was explored. But as it was later found that the interpolated model was good but that the model reduction method was the the reason that it did not give good results in the first attempt. Thus the method of extrapolation the uncertainty model was no longer needed. It has been included as appendix as the method has been explored and to show that it in this case produce a worse uncertainty model than by interpolating the uncertainty model.

Extrapolation is to calculate data points which lies outside of the known data set. This is illustrated in Figure A.1 where three models picked out are Px, Py and Pz. These model can then be used to setup a function that can describe the whole model set.

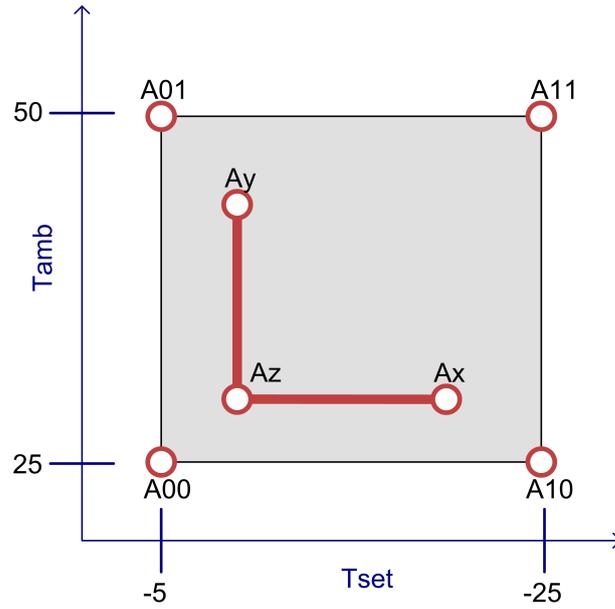
The models Px, Py and Pz have to be selected from the set of linear models. They are chosen to be as follows, Px at  $T_{set} = -20$  and  $T_{amb} = -30$ , Py is at  $T_{set} = -10$  and  $T_{amb} = -45$  and Pz is at  $T_{set} = -10$  and  $T_{amb} = -30$ . These are selected as they look to have a more general dynamic. The parametric uncertainty model can then be setup by solving the following equations for  $A_0$ ,  $A_x$  and  $A_y$ .

$$P_x = A_0 + A_x \cdot \frac{3}{4} + A_y \cdot \frac{1}{5} \tag{A.1}$$

$$P_y = A_0 + A_x \cdot \frac{1}{4} + A_y \cdot \frac{4}{5} \tag{A.2}$$

$$P_z = A_0 + A_x \cdot \frac{1}{4} + A_y \cdot \frac{1}{5} \tag{A.3}$$

This can then be put onto matrix form as shown in the following equation.



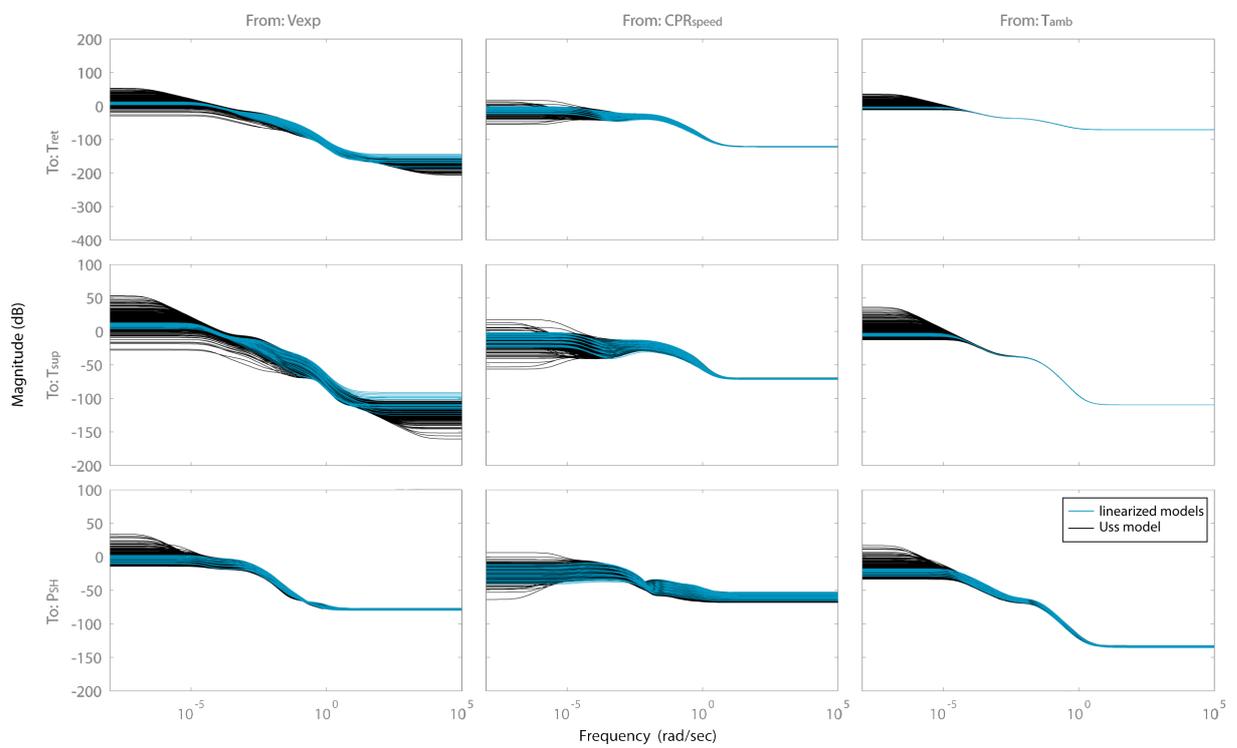
*Figure A.1: illustrate of the extrapolation model selection from the linear model seen as a 2D array*

$$\underbrace{\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}}_B = \underbrace{\begin{bmatrix} 1 & \frac{3}{4} & \frac{1}{5} \\ 1 & \frac{1}{4} & \frac{4}{5} \\ 1 & \frac{1}{4} & \frac{1}{5} \end{bmatrix}}_A \underbrace{\begin{bmatrix} A_0 \\ A_x \\ A_y \end{bmatrix}}_X \quad (\text{A.4})$$

Which then can be solved as  $X = A^{-1}B$ .

This can be setup using the MatLab uncertain atoms like with the interpolated uncertainty model. The resulting model is sampled 400 times and plotted together with the found linear models. This is shown in Figure A.2

As it can be noticed from Figure A.2 the extrapolated parametric uncertainty model has a worse fit than the model produced using interpolation. It is therefore decided not to further investigate this approach.



**Figure A.2:** Bode magnitude plot of the extrapolated Parametric uncertainty model together with the linearized models



Appendix **B**

R134a Log(pressure)-Enthalpy Diagram



Appendix **C**

Full Model Diagram