University of Illinois at Urbana-Champaign

ACRC

Air Conditioning and Refrigeration Center   A National Science Foundation/University Cooperative Research Center

Dynamic Modeling, Control, and Fault Detection in Vapor Compression Systems

M. C. Keir and A. G. Alleyne

ACRC TR-247   August 2006

For additional information:

Air Conditioning and Refrigeration Center
University of Illinois
Department of Mechanical Science & Engineering
1206 West Green Street
Urbana, IL 61801

(217) 333-3115

Prepared as part of ACRC Project #170
Advanced Fault Detection for Vapor Compression Systems
A. G. Alleyne, Principal Investigator

The Air Conditioning and Refrigeration Center was founded in 1988 with a grant from the estate of Richard W. Kritzer, the founder of Peerless of America Inc. A State of Illinois Technology Challenge Grant helped build the laboratory facilities. The ACRC receives continuing support from the Richard W. Kritzer Endowment and the National Science Foundation. The following organizations have also become sponsors of the Center.

Arçelik A. S.
Behr GmbH and Co.
Carrier Corporation
Cerro Flow Products, Inc.
Daikin Industries, Ltd.
Danfoss A/S
Delphi Thermal and Interior
Embraco S. A.
Emerson Climate Technologies, Inc.
General Motors Corporation
Hill PHOENIX
Honeywell, Inc.
Hydro Aluminum Precision Tubing
Ingersoll-Rand/Climate Control
Lennox International, Inc.
LG Electronics, Inc.
Manitowoc Ice, Inc.
Matsushita Electric Industrial Co., Ltd.
Modine Manufacturing Co.
Novelis Global Technology Centre
Parker Hannifin Corporation
Peerless of America, Inc.
Samsung Electronics Co., Ltd.
Sanden Corporation
Sanyo Electric Co., Ltd.
Tecumseh Products Company
Trane
Visteon Automotive Systems
Wieland-Werke, AG

For additional information:

Air Conditioning & Refrigeration Center
Mechanical & Industrial Engineering Dept.
University of Illinois
1206 West Green Street
Urbana, IL 61801

217 333 3115

Abstract

According to the US Department of Energy (www.energy.gov), air conditioning and refrigeration (AC&R) applications account for approximately one third of the total electrical use in US homes and commercial buildings. The massive energy consumption of AC&R systems provides both economic and environmental motivation to develop highly efficient systems that maintain operating efficiency through the use of online diagnostic modules. Recent developments in actuator technologies (i.e. variable speed and variable displacement compressors, electronic expansion valves) managed by proper control architectures can capitalize on the efficiency gains obtained through continuous system operation, enabling significant efficiency gains to be realized in vapor compression systems. The addition of fault detection and diagnosis (FDD) algorithms to the overall control architecture provides the capability to maintain a high level of system performance over its lifetime of operation.

This thesis makes contributions to both the effective transient control and diagnostic capability of vapor compression systems. Accurate control-oriented models that balance simplicity and accuracy are leveraged to improve the understanding of system level fault impact. The general control architecture for vapor compression systems is analyzed from a design perspective, and effective methods to achieve improved system performance and capacity control are demonstrated. Additionally, a discussion of the presence of dynamic fault signatures that may enable improved detection and identification is presented for a subset of system faults.

Table of Contents

List of Figures

Page

List of Tables

List of Abbreviations

AC&R – air conditioning and refrigeration
COP – coefficient of performance
EEV – electronic expansion valve
FDD – Fault Detection and Diagnosis
MIMO – multi-input, multi-output
ODE – ordinary differential equation
PDE – partial differential equation
PID – proportional/integral/derivative controller
PRBS – pseudo-random binary sequence
R134a – refrigerant commonly used in air conditioning systems
SISO – single-input, single-output
TEV/TXV – thermostatic expansion valve

List of Symbols

Variable	Explanation
A	Area
A,B,C,D	State Space Matrices
α	Heat Transfer Coefficient; Physical Parameter
C_p	Specific Heat
E	Energy
η	Efficiency
f, g	Continuous Functions
γ	Void Fraction
$\bar{\gamma}$	Mean Void Fraction
H, h	Enthalpy, Specific Enthalpy
I	Identity Matrix
L	Length
λ	Eigenvalue
$\dot{m}$	Mass Flow Rate
P	Pressure
p	Perimeter
Q	Heat
Re	Reynold’s Number
ρ	Density
S	Slip Ratio; Dual Youla Parameter
s	Specific Entropy
σ	Singular Value
T	Temperature
t	Time
U, u	Internal Energy, Specific Internal Energy; Controllable Inputs
UA	Lumped Heat Transfer Coefficient
V	Volume; Voltage
W	Work
ω	Specific Humidity; Compressor Speed; Frequency

x	Quality; Dynamic States
y	Outputs
z	Spatial Coordinate

Subscript	Explanation
1,2,3	1st, 2nd, 3rd Region
a, air	Air
ave	Average
c	Cold; Condenser; Gas Cooler
cs	Cross-Sectional
d	Diameter
f	Liquid; Frost
g	Vapor
h	Hot
hx	Internal Heat Exchanger
i	Inner
ice	Ice
in	In
int	Intermediate
k	Compressor
o	Outer; Steady State
out	Out
r	Refrigerant; Reduced Order
sb	Sublimation
sys	System
total	Total
v	Valve
w	Wall

Chapter 1. Introduction

According to the US Department of Energy (www.energy.gov) the typical US family spends $1300 a year on home energy bills. Of the total energy use, 56% is used to heat and cool the home. In 2000, residential energy use accounted for an estimated 20% of the energy related CO_2 emissions in the US, producing 313.4 million metric tons of CO_2. Systems, such as air conditioners and refrigerators, that use a vapor compression cycle account for a significant portion of the total electricity consumption in US homes. A study released in 2005 by the Energy Information Administration (www.eia.doe.gov) estimated the residential consumption of electricity by end use. This study analyzed residential data collected in 2001, and found that refrigerators, air conditioners, and freezers accounted for 33.2% of the total electrical use in US homes.

Another study released by the Energy Information Administration in 2005 presented estimates of the energy consumption and expenditures of commercial buildings from data collected in 1999. According to the study, US commercial buildings spent over $66.4 billion dollars on electricity, and consumed 3,098 trillion Btu of electrical energy in 1999. Of the 3,098 trillion Btu, 1,059 trillion Btu were used for cooling and refrigeration, accounting for 34.2% of the annual electricity usage. Since the vast majority of cooling and refrigeration applications use vapor compression cycles, it is clear that improved efficiency and proper maintenance of these systems could have a lasting economic and environmental impact.

Improved efficiency in vapor compression systems can be realized not only through refined component design, but also through advanced control techniques that include diagnostic capability and offer significant lifetime system efficiency benefits. With recent developments in actuator technologies (i.e. variable speed and variable displacement compressors, electronic expansion valves), capitalizing on the efficiency improvements of continuous versus cycled system operation is now economically viable. The additional sensors required for continuous system operation also provide the opportunity to add fault detection and diagnosis algorithms to the overall control architecture. Effectively detecting and identifying slow forming system faults will reduce system maintenance and operating costs, significantly lowering the energy consumed by these systems over their lifetime.

This thesis makes contributions to both the effective transient control and diagnostic capability of vapor compression systems. Accurate control-oriented models that balance simplicity and accuracy are leveraged to improve the understanding of system level fault impact. The general control architecture for vapor compression systems is analyzed from a control design perspective, and effective methods to achieve improved system performance and capacity control are demonstrated. Additionally, a discussion of the presence of dynamic fault signatures that may enable improved detection and identification is presented for a subset of system faults.

The remainder of this chapter is organized as follows. Section 1 presents a background discussion of vapor compression cycles. Section 2 summarizes the control oriented modeling framework, and Section 3 discusses approaches that have been used to control vapor compression systems. Section 4 overviews the challenges associated with fault detection and diagnosis in vapor compression systems. Finally, Section 5 presents an outline of the thesis.

1.1 Vapor Compression Systems

Vapor compression cycles are widely used for heating and cooling in industrial, residential, and automotive applications. A basic vapor compression cycle is composed of four primary components; an evaporator, a compressor, a condenser, and an expansion device. Figures 1.1 and 1.2 present the component layout and P-h diagram of an ideal subcritical vapor compression cycle. Beginning at the condenser inlet, the high-pressure two-phase fluid flows through the condenser rejecting heat. From the condenser the refrigerant flows to an expansion device and transitions from a liquid to a two-phase mixture at a lower pressure. Next, the refrigerant enters the evaporator, where heat is absorbed as the fluid evaporates. The exiting vapor from the evaporator is compressed to a higher pressure, and continues cycling through the system.

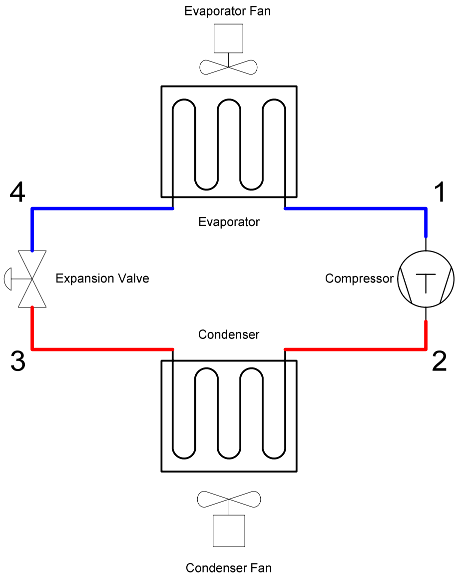


Figure 1.1 - System Diagram - Ideal Subcritical Vapor Compression Cycle

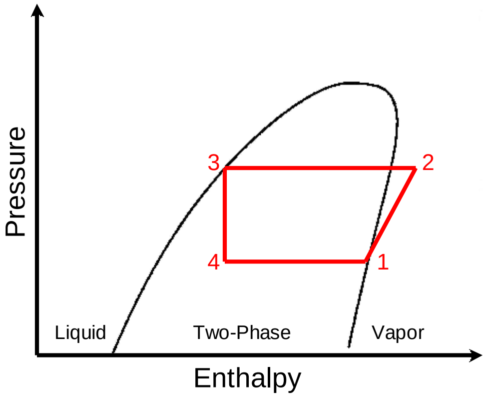


Figure 1.2 - P-h Diagram - Ideal Subcritical Vapor Compression Cycle

There are a number of variations that can be made to the standard vapor compression cycle. A receiver is frequently placed at the condenser outlet, and is used as storage for excess refrigerant in the system. The receiver forces the condition of the condenser outlet to be near a saturated liquid. Accumulators are also occasionally placed at the outlet of the evaporator to prevent liquid from entering the compressor. Internal heat exchangers are used in variations of the basic vapor compression system. Although there is a great deal of variety in specific system configurations, the basic operation of the four major components forces these systems to behave similarly, while the extra components only slightly modify the vapor compression cycles behavior on the P-h diagram.

1.2 Control-Oriented Modeling Framework

To create low order dynamic models of vapor compression systems that are useful for control design, a few standard assumptions are required. First, the compression of the fluid is assumed to be adiabatic with an isentropic efficiency. Second, isobaric conditions in the condenser and evaporator are assumed. Third, expansion through the valve is assumed to be isenthalpic. The expansion valve and the compressor are modeled with static, semi-empirical relationships, while the dynamic heat exchanger models are derived using a lumped parameter moving boundary approach. The nonlinear models have been validated using data taken from experimental systems at the University of Illinois at Urbana-Champaign [22,63].

These models effectively capture the salient transient behavior of vapor compression systems, and are useful for predicting system level behavior. The nonlinear models can be linearized and interconnected, enabling classical linear control design techniques to be applied. The models have been compiled in the Thermosys Toolbox for Matlab, a suite of simulation tools specifically designed for real time simulation of vapor compression systems.

1.3 Control of Vapor Compression Systems

In the United States, vapor compression systems are a primary consumer of electrical power. Due to the massive energy consumption of these devices, it is imperative that system manufacturers design their systems to meet their objectives as efficiently as possible. The goal of any air conditioning or refrigeration application is to move energy from one location to another. In the case of a home on a hot day, this means that the system will use energy to move heat from within the home to the outside air. The heat load within the home varies significantly, depending on the ambient conditions and the level of internal heat generation. Therefore the system must be designed to meet variable cooling loads. Capacity control methods include such strategies as simple on/off control, compressor cylinder unloading, and variable speed compressor control. A summary of theoretical and experimental studies to determine the best capacity control method was conducted by Qureshi and Tassou in 1996 [62]. They found that both theoretical and experimental analyses demonstrated that variable speed compressor control provided the greatest flexibility to match heat loads, resulting in the best overall system efficiency. In many of the papers they summarized, the variable speed control strategies resulted in 20% to 40% reductions in seasonal power consumption. In order to effectively manage variable speed compressor systems it is critical that the control architecture is properly designed.

In general, vapor compression systems are controlled to maximize the energy efficiency (COP) of the system while ensuring the fluid entering the compressor is in the vapor phase. If liquid enters the compressor it can cause a variety of system problems, including decreased efficiency, disruptions to oil circulation, and physical damage to compressor components. To prevent liquid from entering the compressor, vapor compression systems are designed to operate with a certain degree of superheat, defined as the temperature the exiting refrigerant is above the saturation temperature at the evaporator outlet.

To generate the largest cooling capacity for a given power input, the system should operate with the best heat transfer characteristics in the heat exchangers. In the evaporator, the heat transfer coefficient between the refrigerant and the evaporator wall is significantly greater in the two phase region of the heat exchanger. Thus, the best energy transfer is obtained by maximizing the length of the two phase region within the evaporator. This means the ideal evaporator outlet condition is saturated vapor, since this condition maximizes the heat transfer properties while preventing the potential problems associated with liquid entering the compressor. If the vapor compression system is designed to operate with a saturated outlet condition, any system transients will force temporary deviations from this set point, and liquid will enter the compressor. In practice, a generally accepted compromise is to regulate the outlet of the evaporator to the minimum value that retains vapor at the evaporator outlet throughout any anticipated system transients.

The combination of the desire to match external heat loads (capacity control) with the internal refrigerant phase limitations (superheat regulation) necessary for system efficiency/safety requires that the control system simultaneously meet multiple control objectives. Since the dynamics of the system are inherently coupled, it is critical to find the appropriate system signals that can be used in a multivariable controller to effectively manage the system. This thesis provides a detailed discussion of standard and alternative controller configurations in order to develop a simple control framework that achieves high performance while retaining a degree of simplicity that makes the method practical in many industrial applications.

1.4 Fault Detection and Diagnosis in AC&R Systems

Performance degradation resulting from the development of faults within vapor compression systems can result in significant increases in energy consumption [55]. In many cases, slow forming system faults do not manifest themselves in such a way that their existence can be discerned by a reduction in comfort. In fact, a test of more than 4000 residential cooling systems in California revealed that 34% of the systems were undercharged, 28% were overcharged, and only 38% had the correct charge level [61]. A previous study of 27 residential air conditioning systems in Arizona found that 78% of the systems were undercharged [60]. Since cooling and refrigeration compromise over a third of the electrical energy consumption in residential and commercial buildings, the development of diagnostic modules that can effectively detect incipient faults could result in significant cost and energy savings that would have a dramatic economic and environmental impact.

In general, current FDD algorithms for vapor compression cycles fall into two categories, steady-state model-based algorithms and neural network/fuzzy model approaches [34]. Typically the steady-state model approaches use a model identification procedure to develop an accurate model of the steady state vapor compression system behavior. A steady state detector is used to prevent the FDD algorithm from operating during system transients. Neural network/fuzzy model approaches require extensive sets of training data to develop a system model. In general, neural network approaches are limited to steady or pseudo steady state system operation because only steady state data is used to train the model, and therefore a steady state detector is used to prevent false diagnosis during system transients.

A typical vapor compression system achieves a steady or nearly steady state operating condition on the order of minutes or hours (depending on system size). Due to the lengthy settling time, many vapor compression systems perpetually operate under transient conditions. For example, automotive air conditioning systems experience persistent excitation due to the coupling between the engine speed and the compressor. There is also a wealth of system information contained within the system transients that could potentially reduce the number of sensors required to detect a set of system faults. This thesis explores the transient impact of a subset faults on vapor compression systems and highlights some of the advantages of a dynamic model based approach.

1.5 Organization of Thesis

The remainder of this thesis is organized as follows. Chapter 2 provides a detailed discussion of FDD methods, faults in vapor compression systems, and FDD algorithms that have been applied to vapor compression systems. Chapter 3 presents the 1st principles modeling framework used to model the transient behavior of vapor compression cycles. The inclusion of evaporator frosting, refrigerant leaks, and valve actuation faults into the modeling framework is discussed. Chapter 4 discusses real-time simulation of vapor compression systems, and overviews the Thermosys Toolbox for Matlab. In Chapter 5, a detailed description of the experimental system used for model validation, control design, and FDD algorithm development is presented. Chapter 6 discusses the relevance of parameter sensitivity with regard to the control oriented modeling framework. Trajectory sensitivity functions are shown to be useful for tuning immeasurable parameters and identifying signals sensitive to specific faults within vapor compression systems. Chapter 7 provides an in depth discussion of the appropriate methodology to control vapor compression systems. Multivariable control techniques are used to obtain increased control of system capacity while appropriate input-output combinations are shown to significantly reduce the challenges associated with controller design. Chapter 8 presents a discussion of the presence of dynamic fault signatures that may enable improved detection and identification for a subset of system faults. Conclusions and recommendations for future research are provided in Chapter 9.

Chapter 2. FDD in Vapor Compression Systems

Although the exploration of fault detection and diagnosis (FDD) algorithms for vapor compression systems has primarily occurred within the past decade, the general field of fault detection and diagnosis is far more developed [43]. Fault detection and diagnosis algorithms primarily grew out of the need for online system assessment in safety-critical and cost-prohibitive applications. Safety-critical applications are frequently found in areas where the process control and safety requirements dictate the need for fast response to failures, and are commonly found in the aircraft and nuclear power industries. Cost-prohibitive applications include manufacturing processes where inefficient production or equipment failure can significantly impact the product cost. Cost-prohibitive manufacturing applications can be found in many chemical process plants [10], and in fact it is estimated that the US petrochemical industry incurs approximately 20 billion dollars in annual losses due to poor process fault management [57]. In this chapter a brief review of fault detection is provided, summaries of the faults common to vapor compression systems are presented, and a literature review of relevant FDD work in vapor compression systems is included.

2.1 Fault Detection and Diagnosis

Fault detection and diagnosis is typically described as consisting of three primary tasks; fault detection, fault isolation, and fault identification [30]. Fault detection is the process of detecting the presence of a fault within a system. Fault isolation includes the decision scheme that locates the fault within the system. Fault identification then assesses the magnitude of the fault, thus providing the control architecture with sufficient information to choose an appropriate course of action. Figure 2.1 provides a description of the incorporation of a FDD algorithm into the control architecture of a vapor compression system; similar descriptions can be found in [10] and [41].

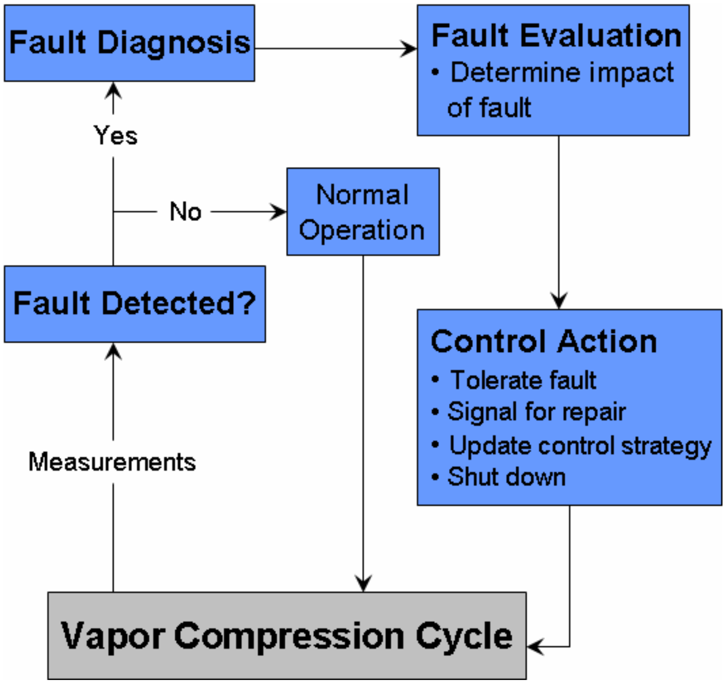


Figure 2.1 Generic FDD framework applied to a vapor compression system.

2.1.1 Fundamental FDD Research

As research in the area of fault detection and diagnosis has progressed, a wide range of approaches have been applied to a variety of systems and processes. The first major survey paper on the area of fault detection was written by Willsky [89], and focused on the use of observers to detect changes in linear system behavior. The next major survey paper was written by Isermann [41], and provided the general framework for model based fault detection and diagnosis. In 1988, Gertler published a survey paper that included a discussion of both model based and statistical detection methods, and commented on the robustness of various fault detection schemes [30]. Frank published a survey paper that expanded on the robustness considerations for FDD algorithms in 1990 [27]. Isermann and Balle conducted a survey of application oriented developments in FDD, gathering interesting statistics on the usage of FDD in various processes [42]. In 1997, Frank summarized the usage of neural networks, fuzzy logic, and qualitative simulation in FDD algorithms [28]. Most recently, Venkatasubramanian et. al. published a three part series surveying the use of quantitative model based methods [84], qualitative models and search strategies [82], and process history based methods [83] in FDD applications.

Over this time period there have also been a variety of books published on various subjects relating to FDD. The first book on model based methods for FDD in chemical processes was published by Himmelblau [37]. In 1998, Gertler published a book with a detailed discussion of the development and implementation of model based FDD algorithms [29]. Chen and Patton published a text on robust model based FDD in 1999 [17]. Patton, Simani, and Fantuzzi published a book on FDD using system identification techniques in 2003 [58]. A useful book, though not directly on FDD algorithms, was published by Gustafsson in 2000 [33]. Gustafsson's book summarizes a large number of the statistical and adaptive filtering techniques that are commonly used for FDD.

2.1.2 Fundamental FDD Methods

The methods used to detect and diagnose faults in processes form the foundation of all FDD systems [43]. There are three fundamental methods that encompass almost all FDD systems; quantitative model based methods, qualitative model based methods, and process history based methods [84]. This section provides an outline of these three methods and discusses some of the merits of each approach as they apply to FDD.

2.1.2.1 Quantitative Model Based Methods

Quantitative model based FDD methods utilize an explicit mathematical model of the monitored plant to achieve analytical redundancy within the system [29]. The analytical redundancy allows for the generation of process residuals based on deviations between the output from the model and the physical plant. The models are generally derived from a detailed knowledge of the physical principles that govern the behavior of the system [43]. Most of the quantitative model based FDD methods available in the literature use discrete linear state space models [84], and are well suited to capitalize on the robust analysis techniques that have been developed for this class of models.

Since quantitative model based FDD methods are based on a first principles physical understanding of the system behavior, they are highly useful for characterizing the physical impact of various faults on the system. The challenge with quantitative models is balancing the complexity and accuracy of the model. Frequently, highly accurate quantitative models use finite volume or discretized approaches that utilize hundreds or thousands of dynamic states to capture system behavior. These computationally complex models are typically too cumbersome to be used in real time system monitoring; therefore some model accuracy must be sacrificed to obtain a feasible FDD algorithm. Although lower order simplified models are frequently available, these models must be sufficiently developed to the point where the loss in accuracy does not significantly obstruct accurate fault detection and identification.

2.1.2.2 Qualitative Model Based Methods

Qualitative model based FDD methods rely on a base of previous process knowledge to draw conclusions about the state of a system. Qualitative model based FDD methods include both rule-based FDD methods, such as digraphs and fault-trees, and qualitative physics FDD approaches [82]. In general, qualitative model based FDD methods employ a large set of if-then-else rules and an inference engine to identify the process condition from a previously defined set of potential states.

Qualitative model based FDD methods are well suited to data rich environments, and they are simple to develop and apply. The reasoning behind a qualitative FDD algorithm is typically transparent and inference engines exist that can reason under uncertainty. Qualitative FDD methods are specific to a system or process, and when the process is complex, the knowledge base can frequently be insufficient or highly complex. Qualitative FDD methods may not be well suited identifying faults on systems where the disturbances and faults require a rigorous approach to residual generation.

2.1.2.3 Process History Based Methods

The most common approaches to FDD in vapor compression systems are process history based FDD algorithms. Process history based approaches include both grey and black box models, where previous process information is used to predict the future behavior of the system. Grey box model approaches are formulated in such a way that the parameter estimates used in FDD can be traced to actual physical parameters that govern the system [43]. Black box model FDD approaches use parameter estimation to identify faults in the system, although the physical meaning of the parameter deviation is not known. Black box models include both dynamic and static models, as well as neural network approaches to FDD.

The main challenge with process history based FDD methods is collecting the data and identifying the trends that indicate a fault within the system. Process history based methods are not adaptable to new system configurations; instead the algorithm must be retrained to account for the changes in system behavior. Process history approaches are best suited to applications where theoretical models of system behavior are insufficiently developed or inadequate to explain system observations. In general, process history based methods are easy to develop, although they require significant training data from the system, which can be time consuming to collect.

2.2 Faults in Vapor Compression Systems

An integral step in developing an effective FDD system for vapor compression equipment is the identification of the most important faults that should be considered in the FDD algorithm. Before proceeding with this discussion, the terminology common to vapor compression system faults should be explained. Faults in vapor compression systems are frequently lumped into two general classes; hard faults, and soft faults. Hard faults consist of any total component failures within the system, and primarily involve fan or compressor motor failures and valve failures. Soft faults are the faults that develop slowly over time in a refrigeration system, and include refrigerant loss, heat exchanger fouling, frosting, and internal and external partial blockages in fluid flow.

The most complete study on vapor compression system failure was conducted by Stouppe and Lau [73] who summarized the cause of 15,716 failures that led to insurance claims in HVAC&R equipment over an eight year period from 1980 through 1987. They found that 76.6% of the failures in hermetic air conditioning and refrigeration units were the result of electrical failures, with 65.8% of the total failures occurring in the motor windings. Failures in the motor windings are generally attributed to a deterioration of the insulation in the motor, resulting in an increase in winding temperature that significantly reduces the life of the motor.

In 1998, Breuker and Braun [12] analyzed a database from a company that primarily repairs rooftop air conditioners for commercial buildings. The database contained over 6000 separate fault cases from 1989 to 1995. Of the 6000 faults, 60% were the result of electrical failures; with 21% stemming from controller failures, 20% from electrical problems, and 19% from motor failures. Breuker and Braun also analyzed the relative repair cost as a result of the various failures, and found that compressor failures accounted for 24% of the total service cost in the database. It was noted that these compressor failures were frequently caused by other faults within the system that resulted in either liquid refrigerant entering the compressor, resulting in a mechanical failure, or high compressor temperatures that cause the motor winding to burnout.

Comstock et. al. [18] performed a similar survey on common faults in chillers. This survey collected data from four chiller manufacturers for centrifugal, water-cooled screw, and air-cooled screw chillers. Similar to the results reported by Breuker and Braun [12], service costs associated with compressor failures accounted for 35% of the total repair cost. Refrigerant leakage was the second most costly fault, accounting for 20% of the total service cost. Comstock et. al. noted that soft faults accounted for 42% of the service calls made for the chillers studied. These soft faults also frequently lead to premature failure of components, a loss in comfort, or a reduction in efficiency.

Table 2.1 summarizes the faults that typically occur in vapor compression systems and provides a brief explanation of the cause and type of fault. Since the surveys presented in this section only provide a cursory understanding of the frequency and impact of faults on vapor compression systems, the following sections will explore some of the important soft faults in greater detail.

Table 2.1 - A summary of faults common to vapor compression systems.

Compressor Failures (Stouppe and Lau, 1989)		
Fault	**Cause**	**Type**
Motor Winding Failure	Most failures involve a deterioration in the insulation. Generally insulation deterioration is attributed to an increase in winding temperature. A 10 C increase will result in a 50% decrease in the life of the windings.	Hard
Unbalanced Voltage	In 3-phase motors a 3% voltage imbalance results in approximately a 25% increase in winding temperature.	Soft
Short Cycling	Rapid starting and stopping of the motor increases temperature of the windings	Soft
Refrigerant Contamination	Some refrigerants will react with water and oil to produce an acid which can corrode the metal in the compressor	Soft
Compressor Valve Failtures	Metal fatigue from cycling can result in failures of the suction and discharge valves	Hard
Liquid Slugging	If liquid enters the compressor, hydraulic forces act on the compressor valves, valve plates, pistons and connecting rods which break	Hard
Bearings	Loss of lubrication can result in damage to the bearings of the compressor	Hard

EEV Failures (Sporlan Documentation)		
Fault	**Cause**	**Type**
Stepper Motor Failure	Winding burns out, opening one or more of the motor phases.	Hard
Electronic Controller Failure	Multiple causes, such as faulty power supply.	Hard

Heat Exchanger Faults (Braun, 2002 and 2003)		
Fault	**Cause**	**Type**
Fan Motor Burnout	Motor windings in fan burn out.	Hard
Air Flow Blockage	An impediment to air flow covers or blocks the air flow across the heat exchanger.	Soft
Fouling	Sediment deposits on internal or external surfaces of heat exchangers	Soft
Frosting	A layer of frost develops as water is condensed out of the air in the evaporator.	Soft

System Faults (Braun, 2002 and 2003)		
Fault	**Cause**	**Type**
Refrigerant Leak	Improper seal allows refrigerant to leak to the environment.	Soft
Liquid Line Rescriction	An impediment to flow develops in the liquid line after the condenser which introduces an additional pressure drop before the valve.	Soft

2.2.1 Refrigerant Leaks

Refrigerant leaks occur when a compromised seal or joint within the refrigeration system allows refrigerant to leak into the surrounding environment. The following two subsections review previous work on detecting refrigerant leaks and present a first principles based explanation of the impact of a refrigerant leak on a system.

2.2.1.1 Previous Studies on Refrigerant Leaks

One of the first significant studies to explore multiple refrigerant undercharge or overcharge conditions was published by Farzad and O’Neal in 1991 [24]. In this work they used the DOE/ARI test procedure for residential sized air conditioners to determine the performance of the system from a condition of 20% undercharge to 20% overcharge in 5% increments. Data was collected for the tests at 15 s intervals. It was found that the capacity of the system peaked at the full charge condition and decreased for both positive and negative deviations from this point. The capacity drop off was steeper for refrigerant undercharge conditions. It was also noted that the capacity of the system for undercharge conditions increased when the condenser air inlet temperature increased.

In 1992, Inatsu et. al. [39] presented a paper detailing the development of a refrigerant monitoring system for automotive air-conditioners. The proposed method was to use a sensor to detect bubbles in the liquid line after the receiver. It was shown that this sensor could detect a refrigerant leak by the time the system reached a 60% charge level.

Farzad and O'Neal [25] published a paper in 1993 that compared the performance of a thermostatic expansion valve (TEV) to an orifice tube on a residential sized air conditioner under a range of charge conditions. The test procedure was the same as that used in [24], and it was shown that the inclusion of a TEV resulted in nearly constant system capacity from a 15% undercharge condition to a 5% overcharge condition. As a comparison, for a 20% undercharge condition, the SEER of the system regulated by an orifice tube decreased by 27%, whereas the SEER of the system regulated with a TEV only dropped 3.5%. Therefore the system is far more robust to changes in refrigerant charge when an active expansion device is used.

In 1998, Bailey [6] presented results for the impact of refrigerant undercharge and overcharge spanning the range of -60% to +15% in 5% charge increments. The tests were conducted on a helical rotary screw air-cooled chiller with electronic expansion valves regulating the chiller circuits. Bailey also considered the impact of refrigerant leaks when the system was operating at partial load conditions. Relationships between refrigerant charge level and the chiller capacity, subcooling temperature, compressor discharge pressure, superheat temperature, and suction pressure were identified. This work was later used to create a data fault library for a neural network based FDD scheme presented by Bailey in [7].

Braun has produced a number of articles on fault detection and diagnosis methods applied to rooftop air conditioning units. The FDD schemes typically include refrigerant leak as a fault. In 2000, he published the impact of refrigerant charge leakage on the capacity, COP, superheat temperature, and compressor discharge temperature on a fixed orifice tube rooftop air conditioning unit [11].

Grace et. al. [31] explored the impact of refrigerant leakage on chillers with liquid to liquid interactions in both the condenser and evaporator. The chiller system was regulated using a TEV and did not contain a receiver. Grace et. al. found that the cooling capacity of the system was fairly constant from a 25% undercharge to a 25% overcharge condition. At conditions lower than 25% undercharge the cooling capacity drops off rapidly, and at conditions above the 25% overcharge condition the capacity drops off slowly. A detection method is proposed that utilizes measurements of evaporator superheat and condenser subcooling to detect undercharge and overcharge conditions, although it is noted that refrigerant leaks are not the only fault that can impact these system outputs.

2.2.1.2 Thermodynamic Impact of Refrigerant Leaks

The impact a refrigerant leak or overcharge will have on a vapor compression system is entirely dependent on the system configuration. Important components that significantly impact the sensitivity of the vapor compression system to refrigerant leaks include receivers and the expansion device. The ability of the expansion device to adapt to off design operation directly impacts the sensitivity of the system to refrigerant leaks, as shown in [25]. A system that uses an orifice tube expansion device has no means to regulate the system, and relatively small deviations from the design condition will impact the performance of the system. Active valve control provided by a TEV or an EEV allows the system to compensate for variations in charge level while maintaining a specified superheat condition at the evaporator outlet.

A liquid receiver is frequently included in vapor compression systems, and its presence can mask the impact of a refrigerant leak. The change in performance associated with refrigerant leaks is commonly associated with gas entering the expansion device. As a larger percentage of the refrigerant entering the expansion device is vapor, the valve will eventually reach a point where the fluid flow becomes choked. This limits the mass flow through the valve and drives the system away from the design condition. The liquid receiver increases the level of refrigerant leak required for vapor to enter the expansion device, since it stores extra refrigerant in the loop, and must be emptied before vapor can enter the expansion device.

2.2.2 Heat Exchanger Fouling

Heat exchanger fouling is defined as the build up of a thermally insulating material on a heat transfer surface. The following two subsections review previous work on heat exchanger fouling as applied to HVAC systems, and present a first principle explanation of the impact of heat exchanger fouling on vapor compression systems.

2.2.2.1 Previous Work on Heat Exchanger Fouling

A general discussion of fouling modes in heat exchangers is presented by Epstein [23]. In this paper fouling is separated into six categories according to the method which the deposit forms on the heat exchanger surface. The fouling categories are scaling, particulate fouling, chemical reaction fouling, corrosion fouling, biofouling, and freezing fouling.

A study of fouling specific to HVAC systems was presented by Siegel and Nararoff [70]. In this paper they state that fouling typically occurs on the external side of the heat exchanger. When the external fluid is air they identified particulate fouling as the primary mode for fouling in HVAC systems. A particulate model for particle deposition on the heat exchanger surface is provided in this paper.

A variety of FDD schemes have included external fouling in their FDD algorithms. There are slight differences in the way this fault is simulated on experimental systems, but the general consensus is that the behavior can be effectively introduced into a system by reducing the rate of air flow over the heat exchanger. Methods for reducing air flow include placing an obstruction on the air side of the coils or reducing the fan speed associated with the heat exchanger [7,16,52,66].

2.2.2.2 Thermodynamic Impact of Heat Exchanger Fouling

External heat exchanger fouling would have two effects on a vapor compression system. Since the particulate matter collecting on the surface of the heat exchanger has a lower conductivity that the metal of the heat exchanger, the thermal resistance between the refrigerant and the external fluid increases. As the layer of particles collects on the external surface of the heat exchanger, the external fluid flow can become partially blocked. The reduction in mass flow rate of the external fluid can significantly impact the performance of the vapor compression system.

For example, as particulate matter collects on the condenser the thermal resistance between the refrigerant and the external fluid will increase and the mass flow rate of the external fluid will decrease. The decrease in external fluid flow will increase the temperature difference between the inlet and the outlet of the condenser. The heat transfer from the refrigerant to the external fluid is also impeded. Therefore for the system to reject the same amount of energy a higher temperature differential will be required. This will result in higher condenser pressures, forcing the compressor to work over a higher pressure differential.

2.2.3 Evaporator Frosting

When the evaporation temperature drops below freezing, water from the air may collect in a layer of frost on the coil. The following two subsections review previous work on evaporator frosting in HVAC systems, and present a first principle explanation of the impact that evaporator frosting will have on a vapor compression system.

2.2.3.1 Previous Work on Evaporator Frosting

In 1990, Rite [65] detailed the effect of frosting of domestic refrigerator-freezer finned tube evaporator coils. This work focused on detecting changes in heat transfer coefficients that occurred as frost accumulated on the evaporator. As a part of this work, Rite demonstrated that the velocity of air flowing over the evaporator coil determined the rate a layer of frost will form. Experimental results demonstrated that for a particular flow rate of air, the rate of frost growth is constant with respect to time, as shown in Table 2.2. The baseline condition for the experimental results provided in Table 2.2 consist of: a refrigerant inlet temperature of -23° C, an evaporator inlet quality of 0.11, and evaporator outlet superheat of 0° C, an inlet air temperature of -12° C, an inlet air relative humidity of 52%, and an air flow rate of 18 L/s.

Table 2.2 - Average frosting rate flux found by Rite [65]

Test Case	**Frost Flux (kg/hr-m^2)**	
	First Five Hours	**Second Five Hours**
Baseline	0.008	0.008
72% RH	0.020	0.022
36 L/s	0.009	0.008
11 L/s	0.006	0.006
$T_{air,in}$ = -7° C	0.024	0.025
$T_{air,in}$ = -29° C	0.017	0.018

Kondepudi and O’Neal [48,49] provided a finite volume quasi-steady state model of frost formation on finned tube heat exchangers. The model accounts for both the increase in thickness of the porous frost layer and the increase in the density of the frost as the growth progresses. The model also allowed the free flow area of the heat exchanger to vary, and calculated the increase in pressure drop across the heat exchanger that would occur as the frost layer grew. The model was compared to experimental results using a glycol-water mixture as the refrigerant. It should be noted that the energy transfer coefficient (W/m^2-K) of the heat exchanger decreased less than 10%, while the pressure drop across the heat exchanger increased 240% from the nominal measured value.

In 1994, Bejan and Vargas [8] confirmed the frost build up model presented by Rite, and demonstrated that an optimal on-off cycle exists for household refrigerators. This on-off cycle maintained the prescribed cooling capacity while minimizing average power consumption and intermittently removed the ice layer that accumulates on the evaporator surface.

Thybo et. al. [78] showed that the dominant effect on supermarket refrigerated display cabinets as a result of frosting of the evaporator coil was a reduction in the air mass flow rate. In experimental frosting trials they found that the temperature drop across the evaporator increased as frost formed, indicating that the reduction in air flow dominated the effect of the reduction in heat transfer coefficient. The increase in air temperature drop across the evaporator was used as the residual signal for FDD, and a CUSUM filter was applied to reduce false alarms. They also detailed the impact of the frost formation on the temperature control and cycle behavior of the cabinet, as detailed in Figure 2.2.

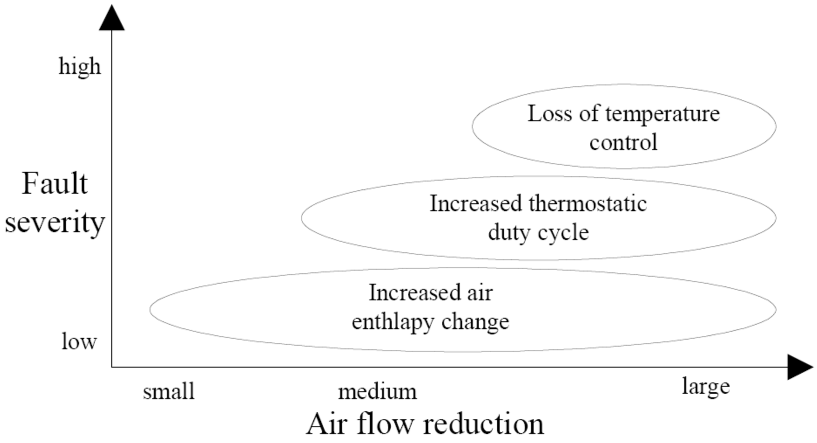


Figure 2.2 - Impact of reduced air flow caused by frost formation in supermarket refrigerated display cabinets [78].

In 2004, Seker et. al. [68] applied the pressure drop correlation presented by Kays and London [45] to the frost model presented by Kondepudi and O’Neal [48]. The model is again a finite volume model and applies a variety of correlations related to the heat exchanger geometry to capture the heat transfer characteristics. Only simulation results are provided.

Yao et. al. [92] presented a finite volume heat exchanger model with a frosting sub-model. The frosting model was similar to that presented by Kondepudi and O’Neal [48], although it treated the water vapor on a frost surface as an ideal gas to develop an advanced equation for the rate the density increased as frost formed. The air-side calculations for the heat exchanger model incorporated the j-factor method to develop the external flow and heat transfer characteristics. The air side pressure drop was evaluated using the pressure drop factor presented by Turaga et. al. [81]. The paper included a brief experimental validation and a number of simulation results. Table 2.3 details the most interesting result of the simulation studies.

Table 2.3 - Simulation results after 30 minutes with an air inlet temperature of 1.5° C.

Relative Humidity	**65%**	**75%**	**85%**
% Loss in heat transfer to air	1.20%	1.50%	2.00%
% Loss in volumetric flow rate of air	20%	30%	40%

From Table 2.3 it is clear that the reduction in volumetric flow rate across the heat exchanger dominates the impact of the fault on the system. As the area between the heat exchanger fins decreases, the flow rate will decrease, but the velocity of the air in the channels will increase. The net result is that the temperature drop of the air across the heat exchanger will deviate significantly from the nominal operating condition, which is in agreement with the experimental results presented by Thybo et. al. [78].

In 2006, Tso et. al. [79] expanded on the original finite volume frost model presented by Kondepudi and O'Neal [48]. The improved model allowed for the variation in frost thickness along the length of the evaporator fins as well as the tube. Using this additional dimension the model was able to more accurately match the experimental results presented in [49].

2.2.3.2 Thermodynamic Impact of Evaporator Frosting

The formation of a layer of frost on the external surface of a compact heat exchanger has two distinct effects. First, the layer of frost provides a layer of thermal insulation between the refrigerant and the external fluid, increasing the thermal resistance and reducing the effective heat transfer coefficient of the evaporator. Second, the layer of frost will impede the flow of air over the coil by reducing the free flow area between adjacent heat exchanger fins. The reduction in free flow area increases the pressure drop of the air across the heat exchanger and increases the velocity of the air within the heat exchanger channels. The increase in velocity combined with the increase in surface roughness from the frost layer results in an increase in the convective heat transfer coefficient, which would in turn reduce the thermal resistance between the refrigerant and the air. The net effect is that the temperature of the air at the outlet of the evaporator will decrease appreciably since the flow rate of air has decreased significantly but the total energy transfer has only experienced a minor reduction.

2.3 FDD Algorithms for Vapor Compression Systems

In the past 15 years a number of FDD algorithms have been developed for various configurations of vapor compression systems. In the past four years there have been three literature reviews that detail the current state of research into FDD for HVAC systems. The first review was written by Halm-Owoo and Suen in 2002 [34]. Their review provides a basic outline of neural network, and rule-based (expert system) approaches to detecting faults in refrigeration and air conditioning systems. In 2003, Braun summarized the relevance of FDD to vapor compression systems and presented a discussion on statistical black box approaches to FDD in air conditioners [10]. The most complete discussion of HVAC FDD to date, was provided in two parts by Katipamula and Brambley in 2005 [43,44]. In these two papers Katipamula and Brambley summarize the various approaches to FDD and provide a discussion of their application to HVAC equipment. This literature review extends beyond the scope of vapor compression equipment and includes extensive information on the detection of faults in air handling units (AHUs), absorption chillers, and building level systems.

This section summarizes the major results of 23 publications that provided algorithms and insights for FDD in vapor compression systems. Of the 23 publications, 11 presented algorithms for chillers (liquid to liquid heat exchange in the evaporator), 5 were on rooftop cooling equipment, two on household refrigerators, three on grocery display cases, one on a residential central air conditioning system, and one on a packaged air conditioner. Since certain authors presented multiple approaches to FDD on some of the systems it is also interesting to note the number of individual research groups that worked on FDD algorithms for each system type. There were 10 different groups of authors that presented FDD work on chillers, two groups worked on rooftop air conditioning units, and one author group for each of the other system configurations. These results are summarized in Figure 2.3.

Valves, compressors, and receivers have a significant impact on the sensitivity of the vapor compression system to specific faults. A summary of the types of actuators included in the systems in the 23 papers is presented in Figure 2.4. It is interesting to note that in only 3 publications was a system described with the ability to alter the speed or mass flow in the compressor. It should also be mentioned that the only vapor compression systems with a refrigerant to air heat exchange in the evaporator and active valve control were the grocery display case (EEV) and the packaged air conditioner (TEV).

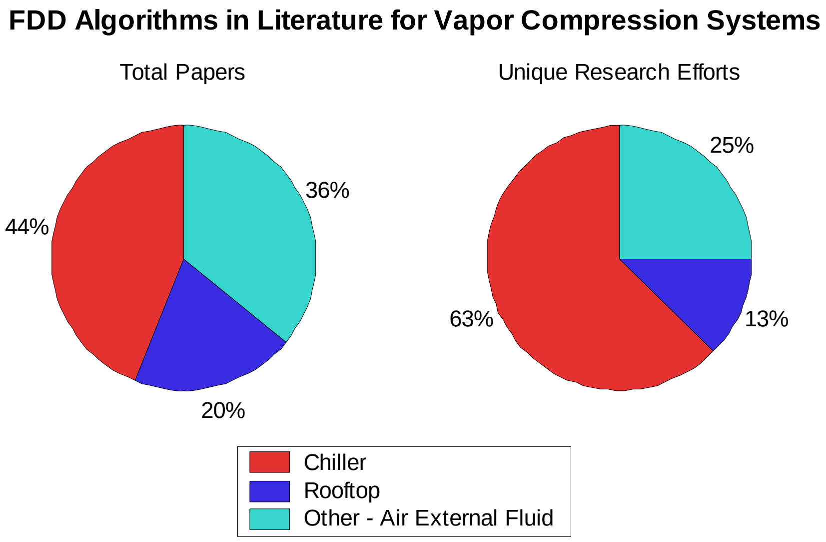


Figure 2.3 - Breakdown of system configurations and research focus in the literature

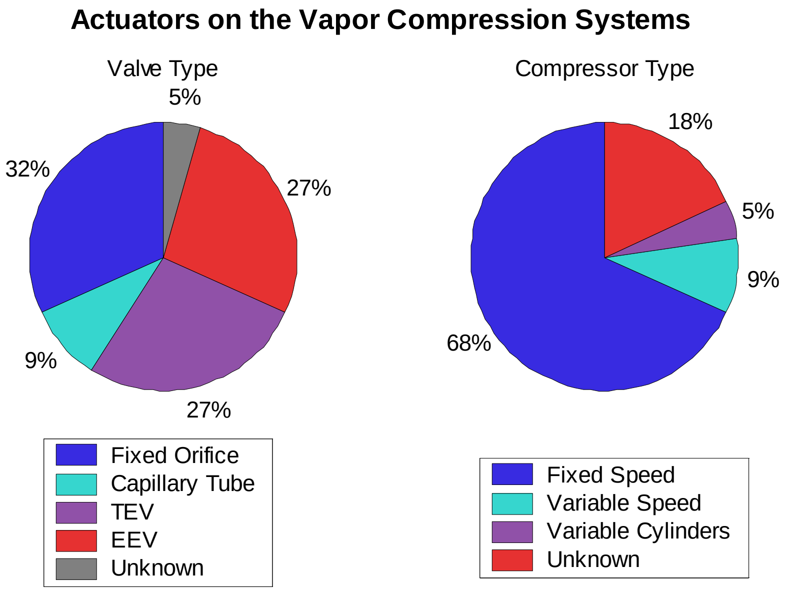


Figure 2.4 - Comparison of actuator technologies used in the systems for FDD development

2.3.1 FDD Algorithms for Non-Chiller Systems

In 1992, Wagner and Shoureshi published two papers on fault detection and diagnosis in household refrigerators [85,86]. The household refrigerator studied had a fixed speed compressor, a capillary tube expansion device, and did not contain a receiver in the refrigerant circuit. In these papers two methods were presented for detecting faults within the refrigerator. In the first method, measured signals from the system were used in a simple limit/trend checking scheme to detect condenser fan motor failures, evaporator fan motor failures, liquid line restrictions, compressor piston leakage, and refrigerant leaks. In the second method a nonlinear model of the compressor and condenser dynamics was developed and simplified using quasi-equilibrium assumptions. Once the dynamic model of the compressor and condenser was complete, a variable structure system observer (sliding observer) was used to track the system states. Residuals were calculated based off the deviation between the predicted model output and the measured system output, and thresholds were used to reduce false alarms. The observer based scheme was able to detect condenser fan failures and liquid line restrictions. The limit/trend checking scheme could detect refrigerant leaks, compressor piston leakage, liquid line restrictions, and evaporator fan failures.

In 1997, Rossi and Braun [66] presented a statistical, rule-based FDD algorithm for rooftop air conditioning units. The rooftop air conditioners they studied used a fixed orifice expansion device, a fixed speed compressor, and did not include a receiver in the refrigerant circuit. A steady-state detector was used to restrict the algorithm to steady-state system conditions, and a directional residual structure with a proximity threshold was used to isolate faults. The algorithm used 7 sensors to detect and distinguish refrigerant leaks, compressor valve leakages, liquid-line restrictions, and condenser and evaporator air flow faults. In simulations, the algorithm was able to detect a refrigerant leak of 2%, a 5% leak in the compressor valve, a liquid line restriction resulting in 80% of the total pressure drop across the system, a 20% decrease in condenser air mass flow rate, and a 40% decrease in evaporator air mass flow rate. The only experimental result presented was that the algorithm was able to effectively detect a 5% reduction in refrigerant charge.

In 1998, Breuker and Braun published two papers on detecting faults in rooftop air conditioning units. In [12], an industry survey is presented that summarizes the common faults in rooftop cooling equipment, their frequency, and the resulting repair cost associated with each fault type. The paper also included plots of variations in system start-up transients when the amount of refrigerant charge in the system is reduced. The transient plots indicate that the steady state gain of the measured signals exhibit consistent deviations as a result of the refrigerant leak. In [13], a steady state lookup model is generated using a least squares fit to experimental data. The system was the same as that used by Rossi and Braun [66], and 7 sensors were used to detect refrigerant leaks, liquid line restrictions, compressor valve leakages, and condenser and evaporator air flow faults. The paper compared the simulated FDD performance of a variety of input-output model configurations with varying order, and identified the best low cost configuration and high performance configuration to detect the fault set.

In 2001, Chen and Braun [16] published a rule-based FDD algorithm for rooftop air conditioning units with a thermostatic expansion valve. As with their previous studies, the system had a fixed speed compressor and the refrigerant circuit did not contain a receiver. The FDD algorithm was designed to detect 7 faults within the system; evaporator air flow faults, condenser air flow faults, liquid line restrictions, compressor valve leakage, refrigerant leaks and overcharge, and non-condensable gas mixed with the refrigerant. Two methods for fault isolation were presented. The “sensitivity ratio” method created temperature residuals with an insensitive signal in the numerator and a sensitive signal in the denominator. The algorithm required 6 temperature sensors and one humidity sensor, and compared measured steady state system output to the output of an identified model. The second method used residuals that were designed to be insensitive to variations in operating conditions but sensitive to specific faults and did not require the identification of a system model. The humidity sensor was not required for the second method. Neither method was capable of distinguishing between a refrigerant overcharge and a non-condensable gas mixing with the refrigerant.

In 2002, Thybo et. al. published two papers on detecting air flow faults in refrigerated display cabinets. In [78], the FDD residual was the difference between the measured energy transfer on the refrigerant side of the evaporator and the energy transferred on the air side of the evaporator. When an air flow fault occurred they showed that the temperature drop of the air over the evaporator coil will increase resulting in a higher predicted energy transfer on the air side, since constant air mass flow rate is assumed in the residual calculation. A CUSUM filter was used to filter the signal noise and the algorithm was shown to effectively detect frost formation on the evaporator. In [77], details of the evaporator model used in the FDD algorithm are presented and an observer based control scheme is presented for the refrigerated display cabinet. It was shown that the FDD algorithm could effectively detect fan failures within the system.

Also in 2002, Siegel and Wray [71] presented a comparison of superheat based refrigerant charge detection methods used by residential AC technicians. The study used four 3-4 ton central air conditioning systems with fixed orifice expansion valves, fixed speed compressors. It was unstated if the systems contained receivers in the refrigerant circuit. They demonstrated the ability to use superheat in a fixed orifice system to detect refrigerant leaks and detailed the accuracy of three commercially available superheat diagnostic methods.

In 2003, Li and Braun [52] published a paper discussing improvements to the original statistical ruled-based FDD algorithm for rooftop air conditioning units presented by Rossi and Braun [66]. The diagonal covariance matrix assumption used by Rossi and Braun was first evaluated and compared to Monte-Carlo simulation (MCS). It was determined that the original method was not sufficiently robust, and the MCS method was robust but not useful due to the lengthy computation time required. The improved method used a combination of the variance threshold and moving window slope methods to filter out any transient data or large noise signals (oscillations) that might appear in the system. After this filtering had been applied the FDD method would only be applicable during steady state operation. A simple FDD method was then used that did not require a covariance matrix, but instead used a normalized distance method that takes advantage of statistical methods to minimize false alarms. It was determined through experimental results that this FDD method was far less sensitive to changes in the operating condition of the system.

In 2004, Thybo and Izadi-Zamanabadi [76] presented an improved version of their FDD algorithm for refrigerated display cabinets. The paper compared a neural network and steady state model-based FDD approach, and determined that due to variations in system configuration the model-based approach is more appropriate for refrigerated display cabinets. The model-based approach also provided useful physical insight into the fault impact that was helpful in controller reconfiguration. The algorithm was again shown to rapidly detect evaporator fan failures and was also able to detect the formation of frost on the evaporator.

In 2004, Armstrong [4] used power signature analysis to detect a series of faults in a 3-ton rooftop air conditioning unit. System start-up transients were used to detect liquid ingestion in the compressor, compressor valve leakage, and refrigerant undercharge. During steady-state operation, an identified ARX model was used to detect liquid ingestion in the compressor. The power draw of each fan was also used during steady-state operation to detect air flow faults in either the condenser or the evaporator.

2.3.2 FDD Algorithms for Chillers

In 1995, Grimmelius et. al. [32] presented a steady state FDD algorithm for a chiller. The chiller had a thermostatic expansion valve, a variable cylinder compressor, and did not contain a receiver in the refrigerant circuit. An identified steady state regression model of the system was used to predict system outputs given a set of operating conditions. The variation between the model output and measured signals was used to detect and isolate compressor suction and discharge blockages, reductions in condenser and evaporator water flow rates, liquid line restrictions, and a loose TEV bulb.

In 1996, Peitsman and Bakker [59] presented a paper on the application of black box models in FDD algorithms for HVAC systems. Their study explored the accuracy of auto-regressive (ARX) and artificial neural network (ANN) models trained on a chiller system with a thermostatic expansion valve. The ARX models were enhanced in an effort to capture the nonlinear dynamics of the vapor compression system by including second order inputs to the linear regression. As a result of the inherent ability of ANN models to capture nonlinear effects, it was determined that ANN models were slightly more accurate than ARX models. The models were used to detect non-condensable fluid in the refrigerant circuit. Models were also developed for an air handling unit and were used to detect an air flow fault over the cooling coil.

In 1996, Stylianou and Nikapour [75] presented a variation on a black box model-based FDD scheme that was implemented on a chiller with a thermostatic expansion valve, a fixed speed compressor, and a high side receiver. The FDD scheme included three modules to encompass the various modes of operation of the chiller. The first module was an off-cycle module that would test for sensor failures when the compressor is turned off by comparing the sensor's equilibration response to the response identified for the system. The second module was used during system start-up, and limit checks were used to detect faults in the identified transient system response. The final module was for steady-state system operation, and was the main portion of the FDD algorithm. The steady-state module checked for faults by comparing a set of sensor readings to those predicted from an ARX chiller performance model. Simple rules were used to distinguish four faults; liquid line restrictions, refrigerant leaks, and reductions in water flow rate over the evaporator and condenser.

In 1997, Stylianou [74] published a follow up paper that improved upon the steady state FDD algorithm from [75]. In this paper the rule-based FDD algorithm was replaced with a statistical pattern recognition algorithm (SPRA) to detect and diagnose the same four system faults. The SPRA was designed using assumed statistical properties that were obtained from the multiple linear regression procedure used to identify the ARX system model.

The SPRA allowed the FDD designer to control the sensitivity of the algorithm, providing a statistical design tool to lower the likelihood of false alarms.

In 2000, McIntosh et. al. [55] applied a first principles steady-state model of a chiller system with a variable speed compressor in an FDD algorithm. The static model was verified and tuned to match the actual chiller performance. A set of characteristic quantities were identified that would provide sufficient information to detect reductions in water flow rate over the condenser and evaporator, condenser and evaporator refrigerant side tube fouling, compressor efficiency, and motor failures. The faults were then applied to the steady state model in varying degrees and the sensitivity of the characteristic quantities was identified. Using the sensitivity analysis, unique diagnostic patterns of specific faults were identified, enabling the isolation of the various faults.

In 2001, Zogg et. al. [95] created a black box dynamic model based FDD algorithm for heat pumps. An ARX model was identified for a heat pump with a fixed orifice expansion device and a fixed speed compressor. Using the identified model of the transient system, the FDD algorithm was used each time the heat pump was turned on. From the identified system, the statistical variances in the parameters due to specific faults were identified. These variances were then used in both a fuzzy classification scheme and a neural network to identify reductions in water flow over the subcooler and the evaporator, liquid line restrictions, reductions and increases in water temperature entering the evaporator, and compressor volumetric efficiency faults.

In 2002, Castro [15] used an identified steady state model and a rule based fault classifier in an FDD algorithm for a chiller with a fixed speed compressor, a thermostatic expansion valve, and a high side receiver. A statistical classifier was used to threshold the fault sensitivity of the algorithm, and the algorithm was able to detect condenser air flow blockages, evaporator water flow faults, liquid line restrictions, and refrigerant charge faults. Castro noted that the thermostatic expansion valve made it difficult to detect liquid line restrictions, and the high side receiver masked the effects of faulty charge levels any time it was partially full.

In 2003, Bailey and Kreider [7] used a neural network FDD scheme to detect faults in a two circuit chiller with fixed speed compressors and electronic expansion valves. The neural network was trained on both normal and fault operating data to detect refrigerant charge faults, oil charge faults, and condenser air flow faults. Using 12 sensors, the neural network fault classifier approach had a 3% misclassification rate on the experimental system.

In 2003, Jia and Reddy [93] presented a characteristic parameter approach which allows a baseline fault-free model of a chiller with a fixed orifice expansion device, and a fixed speed compressor to be identified from experimental data. After the baseline steady state model is identified, the authors identify a performance parameter for each component of the system that can be used to track that component's condition. To account for variations in operating conditions, an empirical fudge factor is used to adjust the parameters to the current operating condition. This method does not identify specific faults, but rather assesses the overall health of the chiller and its components.

In 2005, Kim and Kim [47] presented a steady state FDD algorithm for a chiller system with an electronic expansion valve and a variable speed compressor. A rule based classification scheme was used for fault isolation, and simple experimentally identified thresholds were used to detect faults in the compressor volumetric efficiency, reductions in water flow rate over the evaporator and condenser, and refrigerant leaks.

In 2005, Rueda et. al. [67] developed a neural network steady state model of a chiller with a thermostatic expansion valve and a fixed speed compressor. Residuals were calculated as the difference between the model output and measured signals on the experimental system, and were used to detect faults in refrigerant charge level. A steady state detector was employed to ensure the algorithm would only operate when the system was operating in a quasi-steady state. Under normal operation the misclassification (false alarm) rate was 4.1%.

2.4 Summary

In the past 10 years, significant progress has been made in the development of FDD algorithms for vapor compression systems. Yet, there are still a number of areas that should be addressed in future research efforts to improve the effectiveness, reliability, and cost of FDD algorithms in vapor compression systems. One of the key areas for algorithm improvement is the development of reliable first principles model based approaches that eliminate the need for model identification or algorithm training. The use of first principles model based approaches would significantly reduce the time required to develop and implement the algorithms, and would provide greater flexibility for adapting the algorithms to changes in system architecture.

From this literature review there also appears to be a relative void in FDD algorithms developed for variable capacity vapor compression systems. As energy efficiency becomes more of a priority in future systems, many AC&R systems will likely convert to more flexible variable capacity system components. Therefore it may be valuable for research efforts to address the shortage of FDD algorithm development for variable capacity systems in anticipation that more systems will operate with variable capacity control in the future.

There also are likely significant improvements that can be made to the FDD algorithms by including dynamic information in the algorithm. Of the papers reviewed in this Chapter, only Armstrong [4] and Stylianou [75] capitalized on information contained in the system start up transient. They both found valuable fault information in the dynamic system response; therefore it is highly likely that other transient system responses may be beneficial to effectively detecting certain fault conditions. The use of dynamic information may also lead to a reduction in the number of sensors necessary to detect a set of faults by providing the algorithm with more information that can be used to distinguish between each of the fault conditions.

Chapter 3. Dynamic Modeling

Accurate dynamic models of vapor compression systems play a significant role in the efficient design of systems with optimal component sizes and configurations, and in the development of control strategies to manage these systems. The framework of the dynamic modeling approach is selected through careful consideration of external constraints that limit the usefulness of a particular framework with regard to system design or control development. In the system design phase, the model should accurately predict the performance and behavior of a particular system configuration. The emphasis on accuracy in the design phase has lead to the use of complex models that provide the flexibility to capture the intricate behavior of the fluid flow and heat transfer phenomena common to vapor compression systems. In contrast, the model that is most beneficial for control design is the least complex model that still retains sufficient accuracy to capture the gross dynamic behavior of the system. For control design, it is critical to strike a delicate balance between dynamic complexity and accuracy in the model.

In general, there are three distinct modeling paradigms that have been applied to vapor compression systems. The details of the three modeling paradigms and the reasoning behind the selection of lumped parameter moving boundary model framework are presented in Section 3.1. A discussion of the heat exchanger and mass flow models is presented in Sections 3.2 and 3.3. In Section 3.4, a preliminary approach to including the effects of evaporator frosting into the heat exchanger models is presented. Section 3.5 details the modeling of refrigerant leaks and valve actuation faults in the modeling framework. The chapter concludes by discussing the limitations of the models and the potential opportunities for future development.

3.1 Modeling Paradigms

In general, the three modeling paradigms that have been applied to vapor compression systems are lumped parameter, finite volume (or discretized), and moving boundary models. Lebrun [50] and Bendapudi [9] provide literature reviews of the relevant vapor compression system modeling efforts. Bendapudi notes that the most important task in modeling vapor compression systems is effectively capturing the behavior of the heat exchangers, since they determine the dynamic behavior of the system. Therefore the remainder of this discussion will address the application of each modeling paradigm to the heat exchangers in a vapor compression system.

3.1.1 Lumped Parameter Models

Lumped parameter heat exchanger models attempt to capture the behavior of a heat exchanger with lumped heat transfer parameters. These models are commonly presented in textbooks, as illustrated in Incropera and DeWitt [40]. Often lumped parameter models are used to model vapor compression systems in conjunction with some other component (e.g. the cabin of a car or a room in a building). In this case the focus of the modeling effort is not on the dynamics of the vapor compression system, but on the cooling of the attached space. The simplicity of lumped parameter models tends to be insufficient to capture the dynamic response of some important system outputs (e.g. superheat), and therefore their use in control design is limited. It should be noted that lumped parameter models have found traction in steady state FDD algorithms, as demonstrated in [55].

3.1.2 Finite Volume Models

Finite volume and discretized approaches to the dynamic modeling of vapor compression systems decompose the heat exchanger geometry to a finite set of small regions, allowing spatial effects to be captured by the model. The governing partial differential equations are applied to each region, resulting in high order dynamic models. The complexity of the models is primarily used to capture the spatially varying fluid and heat transfer phenomena that occur in compact heat exchangers. Commercial software packages are available that use a finite volume approach to modeling vapor compression systems (e.g. E-Thermal [2], Modelica [80], EASY5 [1], or SINDA/FLUENT [20]).

3.1.3 Moving Boundary Models

Moving boundary models attempt to capture the dynamics of multiple phase flows within a heat exchanger by allowing the effective position(s) of phase change to vary as a function of time. The parameters for each fluid phase region in the heat exchanger are lumped, resulting in a model of fairly low dynamic order. This approach was first presented by Wedekind and Stoecker, who proposed using a mean void fraction to develop a transient model of evaporating and condensing flows [87]. The mean void fraction approach has been widely accepted by other researchers developing moving boundary models, and a number of experimental correlations have been proposed to capture the behavior of a variety of flow conditions and fluids. Summaries of well know correlations can be found in [64] and [90].

The dominant dynamics associated with the multi-phase flow within the heat exchanger are captured by the changing interface of the moving boundary approach. As a result, the moving boundary framework provides models that can accurately predict the behavior of important system outputs that must be controlled to obtain efficient system operation (i.e. superheat and pressure). The lumped parameter nature of each fluid phase region ensures that the overall dynamic model complexity remains low enough to permit the application of classical control design techniques.

3.2 Heat Exchanger Models

The refrigerant circuit dynamics of vapor compression systems are dominated by the dynamics within the heat exchangers. Modeling of heat exchangers is complicated by the presence of two-phase fluid flow and complex internal and external geometry. The moving boundary approach is based on the assumption of 1-dimensional fluid flow with effective diameter, flow length, and surface areas. The approach also assumes equal pressure throughout the heat exchanger. The heat exchanger is divided into regions based on the fluid phase, and the effective parameters are lumped in each region. The interface between fluid phase regions is allowed to be a dynamic variable. The following two subsections provide an overview of moving boundary models of an evaporator and a condenser with receiver. A comprehensive derivation of the models, as well as other heat exchanger configurations in presented in [63].

3.2.1 Evaporator

The evaporator model assumes a two phase flow condition at the heat exchanger inlet that transitions to a single phase flow at specific point within the heat exchanger. The location of the interface between these two phase regions is allowed to be a dynamic variable. The governing ordinary differential equations (ODEs) are obtained by integrating the governing partial differential equations (PDEs) (Eqs. 3.1-3.3) along the length of the heat exchanger and assuming lumped parameters in each fluid region.

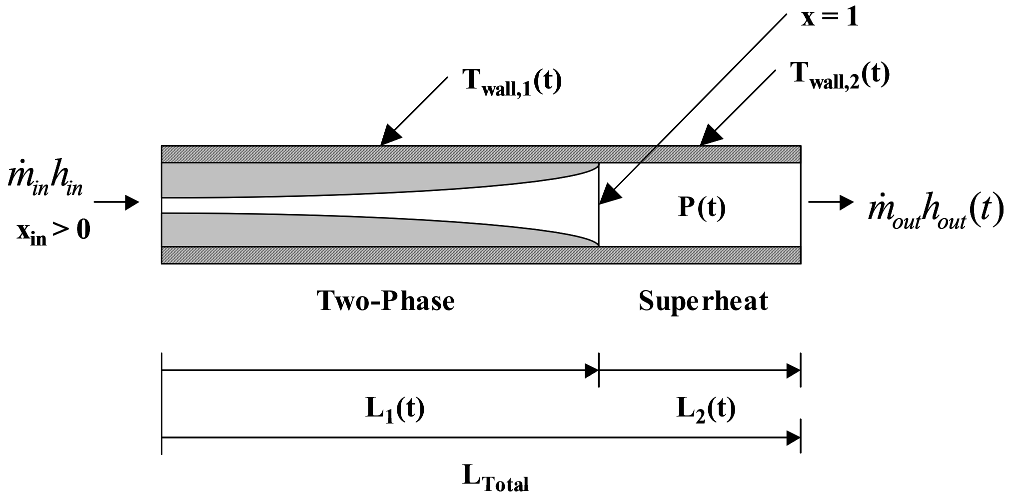


Figure 3.1 - Diagram of the Evaporator with two fluid regions

Several assumptions are made regarding the lumped parameters of the evaporator model. The air temperature used to determine the heat transfer between the walls of the heat exchanger and the air is assumed to be a weighted average of the inlet and outlet air temperatures, $T_a = T_{a,in}(\mu) + T_{a,out}(1-\mu)$. In the two phase region, the fluid properties are determined by assuming a mean void fraction, for example $\rho_1 = \rho_f(1-\bar{\gamma}) + \rho_g(\bar{\gamma})$. In the superheat region, average properties between the inlet and outlet refrigerant state are used, i.e. $h_2 = \frac{h_g + h_{out}}{2}$, $T_{r2} = T(P_e, h_2)$, and $\rho_2 = \rho(P_e, h_2)$. For the evaporator model, the time derivative of the mean void fraction is neglected. This assumption is valid not only because the change in mean void fraction tends to be small during transients considered, but also because its time dependence is related to dynamic modes that are much faster than the dominant system dynamics. Thus any mean void fraction dynamics can be replaced with their instantaneous, algebraic equivalents.

The governing partial differential equations for the conservation of refrigerant mass, refrigerant energy, and heat exchanger wall energy in a fluid region are given by Eqs. 3.1-3.3.

$$\frac{\partial(\rho A_{cs})}{\partial t} + \frac{\partial(\dot{m})}{\partial z} = 0 \tag{3.1}$$

$$\frac{\partial(\rho A_{cs} h - A_{cs} P)}{\partial t} + \frac{\partial(\dot{m}h)}{\partial z} = p_i \alpha_i (T_w - T_r) \tag{3.2}$$

$$(C_p \rho A)_w \frac{\partial(T_w)}{\partial t} = p_i \alpha_i (T_r - T_w) + p_o \alpha_o (T_a - T_w) \tag{3.3}$$

The integration of Eqs. 3.1-3.3 over the two phase and superheat regions of the evaporator results in the ODEs given by Eqs. 3.4-3.9.

Conservation of Refrigerant Mass (Two-Phase and Superheat Regions)

$$\left(\frac{d\rho_f}{dP_e}(1-\bar{\gamma})+\frac{d\rho_g}{dP_e}(\bar{\gamma})\right)A_{cs}L_1\dot{P}_e+\left(\rho_f-\rho_g\right)\left(1-\bar{\gamma}\right)A_{cs}\dot{L}_1=\dot{m}_{in}-\dot{m}_{\mathrm{int}} \tag{3.4}$$

$$\begin{aligned}&\left[\left(\left.\frac{\partial\rho_2}{\partial P_e}\right|_{h_2}\right)+\frac{1}{2}\left(\left.\frac{\partial\rho_2}{\partial h_2}\right|_{P_e}\right)\left(\frac{dh_g}{dP_e}\right)\right]AL_2\dot{P}_e+\frac{1}{2}\left(\left.\frac{\partial\rho_2}{\partial h_2}\right|_{P_e}\right)AL_2\dot{h}_{out}\\&+\left(\rho_g-\rho_2\right)A\dot{L}_1=\dot{m}_{\mathrm{int}}-\dot{m}_{out}\end{aligned} \tag{3.5}$$

Conservation of Refrigerant Energy (Two-Phase and Superheat Regions)

$$\begin{aligned}&\left(\frac{d\left(\rho_f h_f\right)}{dP_e}(1-\bar{\gamma})+\frac{d\left(\rho_g h_g\right)}{dP_e}(\bar{\gamma})-1\right)A_{cs}L_1\dot{P}_e+\left(\rho_f h_f-\rho_g h_g\right)\left(1-\bar{\gamma}\right)A_{cs}\dot{L}_1\\&=\dot{m}_{in}h_{in}-\dot{m}_{\mathrm{int}}h_{\mathrm{int}}+\alpha_{i1}A_i\left(\frac{L_1}{L_{Total}}\right)\left(T_{w1}-T_{r1}\right)\end{aligned} \tag{3.6}$$

$$\begin{aligned}&\left[\left(\left(\left.\frac{\partial\rho_2}{\partial P_e}\right|_{h_2}\right)+\left(\frac{1}{2}\right)\left(\frac{dh_g}{dP_e}\right)\left(\left.\frac{\partial\rho_2}{\partial h_2}\right|_{P_e}\right)\right)h_2+\left(\frac{1}{2}\right)\left(\frac{dh_g}{dP_e}\right)\rho_2-1\right]A_{cs}L_2\dot{P}_e\\&+\left[\left(\left.\frac{\partial\rho_2}{\partial h_2}\right|_{P_e}\right)h_2+\rho_2\right]\left(\frac{1}{2}\right)A_{cs}L_2\dot{h}_{out}+\left(\rho_g h_g-\rho_2 h_2\right)A_{cs}\dot{L}_1\\&=\dot{m}_{\mathrm{int}}h_{\mathrm{int}}-\dot{m}_{out}h_{out}+\alpha_{i2}A_i\left(\frac{L_2}{L_{Total}}\right)\left(T_{w2}-T_{r2}\right)\end{aligned} \tag{3.7}$$

Conservation of Wall Energy (Two-Phase and Superheat Regions)

$$\left(C_p\rho V\right)_w\dot{T}_{w1}=\alpha_{i1}A_i\left(T_{r1}-T_{w1}\right)+\alpha_o A_o\left(T_a-T_{w1}\right) \tag{3.8}$$

$$\left(C_p\rho V\right)_w\left[\dot{T}_{w2}-\left(\frac{T_{w2}-T_{w1}}{L_2}\right)\dot{L}_1\right]=\alpha_{i2}A_i\left(T_{r2}-T_{w2}\right)+\alpha_o A_o\left(T_a-T_{w2}\right) \tag{3.9}$$

The resulting six differential equations for conservation of refrigerant mass, refrigerant energy, and wall energy for the two-phase and superheat regions only contain five explicit time derivatives: $\dot{L}_1$, $\dot{P}_e$, $\dot{h}_{out}$, $\dot{T}_{w1}$, and $\dot{T}_{w2}$. The equations can be combined to eliminate the variable $\dot{m}_{\mathrm{int}}$. This results in Eq. 3.10, which is of the $Z(x,u)\cdot\dot{x}=f(x,u)$ form, with states $x=\left[L_1 \quad P_e \quad h_{out} \quad T_{w1} \quad T_{w2}\right]^T$, and where the elements of the $Z(x,u)$ matrix are given in Table 3.1.

$$\begin{bmatrix} z_{11} & z_{12} & 0 & 0 & 0 \\ z_{21} & z_{22} & z_{23} & 0 & 0 \\ z_{31} & z_{32} & z_{33} & 0 & 0 \\ 0 & 0 & 0 & z_{44} & 0 \\ z_{51} & 0 & 0 & 0 & z_{55} \end{bmatrix} \begin{bmatrix} \dot{L}_1 \\ \dot{P}_e \\ \dot{h}_{out} \\ \dot{T}_{w1} \\ \dot{T}_{w2} \end{bmatrix} = \begin{bmatrix} \dot{m}_{in}\left(h_{in} - h_g\right) + \alpha_{i1} A_i \left(\dfrac{L_1}{L_{Total}}\right)\left(T_{w1} - T_{r1}\right) \\ \dot{m}_{out}\left(h_g - h_{out}\right) + \alpha_{i2} A_i \left(\dfrac{L_2}{L_{Total}}\right)\left(T_{w2} - T_{r2}\right) \\ \dot{m}_{in} - \dot{m}_{out} \\ \alpha_o A_o \left(T_a - T_{w1}\right) - \alpha_{i1} A_i \left(T_{w1} - T_{r1}\right) \\ \alpha_o A_o \left(T_a - T_{w2}\right) - \alpha_{i2} A_i \left(T_{w2} - T_{r2}\right) \end{bmatrix} \qquad (3.10)$$

Table 3.1 - Matrix Elements of $Z(x,u)$ for the Evaporator

z_{11}	$\left[\rho_f\left(h_f - h_g\right)\right]\left(1-\bar{\gamma}\right)A_{cs}$
z_{12}	$\left[\left(\dfrac{d\left(\rho_f h_f\right)}{dP_e} - \dfrac{d\rho_f}{dP_e} h_g\right)\left(1-\bar{\gamma}\right) + \left(\dfrac{d\left(\rho_g h_g\right)}{dP_e} - \dfrac{d\rho_g}{dP_e} h_g\right)\left(\bar{\gamma}\right) - 1\right] A_{cs} L_1$
z_{21}	$\rho_2\left(h_g - h_2\right)A_{cs}$
z_{22}	$\left[\left(\left(\left.\dfrac{\partial\rho_2}{\partial P_e}\right\vert_{h_2}\right) + \left(\dfrac{1}{2}\right)\left(\left.\dfrac{\partial\rho_2}{\partial h_2}\right\vert_{P_e}\right)\left(\dfrac{dh_g}{dP_e}\right)\right)\left(h_2 - h_g\right) + \left(\dfrac{\rho_2}{2}\right)\left(\dfrac{dh_g}{dP_e}\right) - 1\right] A_{cs} L_2$
z_{23}	$\left[\left(\dfrac{1}{2}\right)\left(\left.\dfrac{\partial\rho_2}{\partial h_2}\right\vert_{P_e}\right)\left(h_2 - h_g\right) + \left(\dfrac{\rho_2}{2}\right)\right] A_{cs} L_2$
z_{31}	$\left[\left(\rho_g - \rho_2\right) + \left(\rho_f - \rho_g\right)\left(1-\bar{\gamma}\right)\right]A_{cs}$
z_{32}	$\left[\left[\left(\left.\dfrac{\partial\rho_2}{\partial P_e}\right\vert_{h_2}\right) + \dfrac{1}{2}\left(\left.\dfrac{\partial\rho_2}{\partial h_2}\right\vert_{P_e}\right)\left(\dfrac{dh_g}{dP_e}\right)\right] L_2 + \left[\left(\dfrac{d\rho_f}{dP_e}\right)\left(1-\bar{\gamma}\right) + \left(\dfrac{d\rho_g}{dP_e}\right)\left(\bar{\gamma}\right)\right] L_1\right] A_{cs}$
z_{33}	$\dfrac{1}{2}\left(\left.\dfrac{\partial\rho_2}{\partial h_2}\right\vert_{P_e}\right) A_{cs} L_2$
z_{44}	$\left(C_p \rho V\right)_w$
z_{51}	$\left(C_p \rho V\right)_w \left(\dfrac{T_{w1} - T_{w2}}{L_2}\right)$
z_{55}	$\left(C_p \rho V\right)_w$

The nonlinear model presented in Eq. 3.10 can be linearized around a particular system operating condition. This results in a linear evaporator model that can be used for control design and to study the parametric sensitivity of the system. The full derivation of the linear model is presented in [63].

3.2.2 Condenser with Receiver

The condenser portion of the condenser with receiver model assumes that superheated vapor enters the condenser and transitions to a two phase mixture. The outlet from the condenser is assumed to be near the saturated liquid condition, and the mean void fraction is used to account for deviations from the saturated liquid condition.

The outlet of the receiver is a saturated liquid at the constant pressure that is assumed for the entire condenser with receiver model. The governing ordinary differential equations (ODEs) are obtained by integrating the governing partial differential equations (PDEs) (Eqs. 3.1-3.3) along the length of the heat exchanger and assuming lumped parameters in each fluid region.

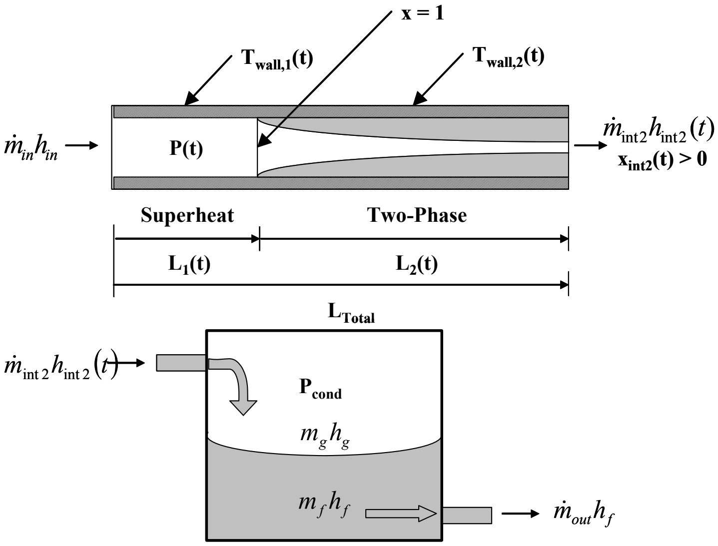


Figure 3.2 - Condenser with two fluid regions and a receiver

The condenser with receiver model uses many of the same assumptions presented in the evaporator model derivation. The air temperature used to determine the heat transfer between the walls of the heat exchanger and the air is assumed to be a weighted average of the inlet and outlet air temperatures, $T_a = T_{a,in}(\mu) + T_{a,out}(1-\mu)$. In the two-phase region, the fluid properties are determined by assuming a mean void fraction, for example $\rho_1 = \rho_f(1-\bar{\gamma}) + \rho_g(\bar{\gamma})$. As in the evaporator model, average properties are assumed in the superheat region. For this derivation, the time derivative of the mean void fraction is not neglected and is used to capture dynamic information regarding the evaporator outlet quality. For this application a "Slip-Ratio" void fraction correlation, $\gamma = \frac{x}{x + (1-x)\mu_S}$, is assumed, where $\mu_S = \left(\frac{\rho_g}{\rho_f}\right)S$. Mean void fraction is calculated by integrating the expression given by Eqs. 3.11 and 3.12.

$$\bar{\gamma} = \frac{1}{(1-\mu_S)} + \frac{\mu_S}{x_\Delta(1-\mu_S)^2}\ln\left[\frac{\mu_S + x_{in}(1-\mu_S)}{\mu_S + x_{in}(1-\mu_S) + x_\Delta(1-\mu_S)}\right] \qquad (3.11)$$

$$x_\Delta = x_{int2} - x_{int1} \qquad (3.12)$$

For the condenser, the fluid entering the two-phase region is assumed to be saturated vapor, $x_{int1} = 1$. If the fluid exiting the condenser is saturated liquid, then $x_{int2} = 0$, and $x_\Delta = -1$. For small deviations around this condition, the mean void fraction expression can be approximated as Eq. 3.13. Thus, solving for the evaporator outlet quality yields Eq. 3.14, where $a = \dfrac{\mu_S}{1-\mu_S}$ and $b = \dfrac{1}{1-\mu_S}$.

$$\bar{\gamma} = \frac{1}{(1-\mu_S)} - \frac{\mu_S}{(1-\mu_S)^2}\ln\left[\frac{1}{\mu_S + x_{\text{int}2}(1-\mu_S)}\right] \tag{3.13}$$

$$x_{\text{int}2} = b e^{\left(\frac{\bar{\gamma}-b}{ab}\right)} - a \tag{3.14}$$

The governing ODEs for the conservation of refrigerant mass, refrigerant energy, and heat exchanger wall energy are given as follows:

Conservation of Refrigerant Mass (Superheat and Two-phase Regions)

$$\left(\frac{\partial\rho_1}{\partial P_c} + \frac{1}{2}\frac{\partial\rho_1}{\partial h_1}\frac{dh_g}{dP_c}\right)A_{cs}L_1\dot{P} + \left(\frac{1}{2}\frac{\partial\rho_1}{\partial h_1}\right)A_{cs}L_1\dot{h}_i + \left(\rho_1 - \rho_g\right)A_{cs}\dot{L}_1 = \dot{m}_i - \dot{m}_{\text{int}1} \tag{3.15}$$

$$\begin{aligned}&(\rho_g - \rho_f)(1-\bar{\gamma})A_{cs}\dot{L}_1 + \left(\frac{d\rho_f}{dP_c}(1-\bar{\gamma}) + \frac{d\rho_g}{dP_c}(\bar{\gamma})\right)A_{cs}L_2\dot{P}_c \\ &+ (\rho_g - \rho_f)A_{cs}L_2\dot{\bar{\gamma}} = \dot{m}_{\text{int}1} - \dot{m}_{\text{int}2}\end{aligned} \tag{3.16}$$

Conservation of Refrigerant Energy (Superheat and Two-phase Regions)

$$\begin{aligned}&\left[\left(\frac{\partial\rho_1}{\partial P_c} + \frac{1}{2}\frac{\partial\rho_1}{\partial h_1}\frac{dh_g}{dP_c}\right)h_1 + \left(\frac{1}{2}\frac{dh_g}{dP_c}\right)\rho_1 - 1\right]A_{cs}L_1\dot{P}_c + \left[\left(\frac{1}{2}\frac{\partial\rho_1}{\partial h_1}\right)h_1 + \left(\frac{1}{2}\right)\rho_1\right]A_{cs}L_1\dot{h}_i \\ &+ (\rho_1 h_1 - \rho_g h_g)A_{cs}\dot{L}_1 = \dot{m}_i h_i - \dot{m}_{\text{int}1}h_g + \alpha_{i1}A_i\left(\frac{L_1}{L_{Total}}\right)(T_{w1} - T_{r1})\end{aligned} \tag{3.17}$$

$$\begin{aligned}&(\rho_g h_g - \rho_f h_f)(1-\bar{\gamma})A_{cs}\dot{L}_1 + \left(\frac{d(\rho_f h_f)}{dP_c}(1-\bar{\gamma}) + \frac{d(\rho_g h_g)}{dP_c}(\bar{\gamma}) - 1\right)A_{cs}L_2\dot{P}_c \\ &+ (\rho_g h_g - \rho_f h_f)A_{cs}L_2\dot{\bar{\gamma}} = \dot{m}_{\text{int}1}h_g - \dot{m}_{\text{int}2}h_{\text{int}2} + \alpha_{i2}A_i\left(\frac{L_2}{L_{Total}}\right)(T_{w2} - T_{r2})\end{aligned} \tag{3.18}$$

Conservation of Wall Energy (Superheat and Two-phase Regions)

$$(C_P\rho V)_w\left(\dot{T}_{w1} + \frac{T_{w1} - T_{w2}}{L_1}\dot{L}_1\right) = \alpha_{i1}A_i(T_{r1} - T_{w1}) + \alpha_o A_o(T_a - T_{w1}) \tag{3.19}$$

$$(C_P\rho V)_w\dot{T}_{w2} = \alpha_{i2}A_i(T_{r2} - T_{w2}) + \alpha_o A_o(T_a - T_{w2}) \tag{3.20}$$

Conservation of Mass (Receiver)

$$\dot{m}_{rec} = \dot{m}_{\text{int}2} - \dot{m}_o \tag{3.21}$$

Conservation of Energy (Receiver)

$$\begin{aligned}&\left[\frac{d\rho_g}{dP_c}V_g u_g + \frac{d\rho_f}{dP_c}V_f u_f + m_g\frac{du_g}{dP_c} + m_f\frac{du_f}{dP_c} - \left(\frac{\rho_g u_g - \rho_f u_f}{\rho_g - \rho_f}\right)\left(\frac{d\rho_g}{dP_c}V_g + \frac{d\rho_f}{dP_c}V_f\right)\right]\dot{P}_c \\ &+ \left(\frac{\rho_g u_g - \rho_f u_f}{\rho_g - \rho_f}\right)\dot{m}_{rec} = \dot{m}_{\text{int}2}h_{\text{int}2} - \dot{m}_o h_f - UA_{rec}(T_{rec} - T_{amb})\end{aligned} \tag{3.22}$$

These eight differential equations are combined to eliminate the variables $\dot{m}_{\text{int}1}$ and $\dot{m}_{\text{int}2}$. This combination results in a model with six states: L_1, P_c, $\bar{\gamma}$, m_{rec}, T_{w1}, and T_{w2}. The resulting model is given in Eq. 3.23 and is of the form $Z(x,u)\cdot\dot{x} = f(x,u)$, with the elements of the Z matrix given in Table 3.2.

$$\begin{bmatrix} z_{11} & z_{12} & 0 & 0 & 0 & 0 \\ z_{21} & z_{22} & z_{23} & z_{24} & 0 & 0 \\ z_{31} & z_{32} & z_{33} & z_{34} & 0 & 0 \\ 0 & z_{42} & 0 & z_{44} & 0 & 0 \\ z_{51} & 0 & 0 & 0 & z_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & z_{66} \end{bmatrix} \begin{bmatrix} \dot{L}_1 \\ \dot{P} \\ \dot{\bar{\gamma}} \\ \dot{m}_{rec} \\ \dot{T}_{w1} \\ \dot{T}_{w2} \end{bmatrix} = \begin{bmatrix} \dot{m}_i\left(h_i - h_g\right) + \alpha_{i1}A_i\left(\dfrac{L_1}{L_{Total}}\right)\left(T_{w1} - T_{r1}\right) \\ \dot{m}_o\left(h_g - h_{\text{int}2}\right) + \alpha_{i2}A_i\left(\dfrac{L_2}{L_{Total}}\right)\left(T_{w2} - T_{r2}\right) \\ \dot{m}_i - \dot{m}_o \\ \dot{m}_o\left(h_{\text{int}2} - h_f\right) - UA_{rec}\left(T_{rec} - T_{amb}\right) \\ \alpha_{i1}A_i\left(T_{r1} - T_{w1}\right) + \alpha_o A_o\left(T_a - T_{w1}\right) \\ \alpha_{i2}A_i\left(T_{r2} - T_{w2}\right) + \alpha_o A_o\left(T_a - T_{w2}\right) \end{bmatrix} \tag{3.23}$$

Table 3.2 - Matrix Elements of $Z(x,u)$ for the Condenser with Receiver

z_{11}	$\left(\rho_1 h_1 - \rho_1 h_g\right)A_{cs}$
z_{12}	$\left[\left(\dfrac{\partial\rho_1}{\partial P} + \dfrac{1}{2}\dfrac{\partial\rho_1}{\partial h_1}\dfrac{dh_g}{dP}\right)\left(h_1 - h_g\right) + \left(\dfrac{1}{2}\dfrac{dh_g}{dP}\right)\rho_1 - 1\right]A_{cs}L_1$
z_{21}	$\left(\rho_f h_g - \rho_f h_f\right)\left(1-\bar{\gamma}\right)A_{cs}$
z_{22}	$\left(\dfrac{d\left(\rho_f h_f\right)}{dP}\left(1-\bar{\gamma}\right) + \dfrac{d\left(\rho_g h_g\right)}{dP}\bar{\gamma} - \dfrac{d\rho_f}{dP}h_g\left(1-\bar{\gamma}\right) - \dfrac{d\rho_g}{dP}h_g\bar{\gamma} - 1\right)A_{cs}L_2$
z_{23}	$\left(\rho_f h_g - \rho_f h_f\right)A_{cs}L_2$
z_{24}	$\left(h_{\text{int}2} - h_g\right)$
z_{31}	$\left[\rho_1 - \rho_f\left(1-\bar{\gamma}\right) - \rho_g\bar{\gamma}\right]A_{cs}$
z_{32}	$\left[\left(\dfrac{d\rho_f}{dP}\left(1-\bar{\gamma}\right) + \dfrac{d\rho_g}{dP}\bar{\gamma}\right)L_2 + \left(\dfrac{\partial\rho_1}{\partial P} + \dfrac{1}{2}\dfrac{\partial\rho_1}{\partial h_1}\dfrac{dh_g}{dP}\right)L_1\right]A_{cs}$
z_{33}	$\left(\rho_g - \rho_f\right)A_{cs}L_2$
z_{34}	1
z_{42}	$\left[\dfrac{d\rho_g}{dP}V_g u_g + \dfrac{d\rho_f}{dP}V_f u_f + m_g\dfrac{du_g}{dP} + m_f\dfrac{du_f}{dP} - \left(\dfrac{\rho_g u_g - \rho_f u_f}{\rho_g - \rho_f}\right)\left(\dfrac{d\rho_g}{dP}V_g + \dfrac{d\rho_f}{dP}V_f\right)\right]$
z_{44}	$\left(\dfrac{\rho_g u_g - \rho_f u_f}{\rho_g - \rho_f} - h_{\text{int}2}\right)$

z_{51}	$\left(c_p \rho V\right)_w \left(\frac{T_{w1} - T_{w2}}{L_1}\right)$
z_{55}	$\left(c_p \rho V\right)_w$
z_{66}	$\left(c_p \rho V\right)_w$

The nonlinear model presented in Eq. 3.23 can be linearized around a particular system operating condition. This results in a linear condenser with receiver model that can be used for control design and to study the parametric sensitivity of the system. The full derivation of the linear model is presented in [63].

3.3 Mass Flow Device Models

The dynamics of the mass flow devices within a refrigeration circuit are typically an order of magnitude faster than the dynamics of the vapor compression system, which evolve on the order of seconds or minutes [63]. This time scale separation allows the mass flow devices to be modeled using static relationships.

3.3.1 Compressor

Two algebraic relationships are used to model the compressor. Mass flow rate is calculated in Eq. 3.24 where $\rho_k = \rho\left(P_{k,in}, h_{k,in}\right)$, and a volumetric efficiency, η_{vol}, is assumed. Additionally, compression is assumed to be an adiabatic process with an isentropic efficiency, and therefore the relationship between the entrance and exit enthalpies is given in Eq. 3.25, where $h_{out,isentropic} = h\left(P_{out}, s_k\right)$ and $s_k = s\left(P_{in}, h_{in}\right)$. For implementation, this is rearranged to give Eq. 3.26. Both the volumetric and isentropic efficiencies are assumed to change with operating condition, and are given by semi-empirical maps (Eqs. 3.27 and 3.28). For simulation purposes, the change of compressor speed is rate limited to reflect the limitations of a real compressor.

$$\dot{m}_k = \omega_k V_k \rho_k \eta_{vol} \tag{3.24}$$

$$\frac{h_{out,isentropic} - h_{in}}{h_{out} - h_{in}} = \eta_k \tag{3.25}$$

$$h_{out} = \frac{1}{\eta_k}\left[h_{out,isentropic} + h_{in}\left(\eta_k - 1\right)\right] \tag{3.26}$$

$$\eta_{vol} = f_1\left(P_{ratio}, \omega_k\right) \tag{3.27}$$

$$\eta_k = f_2\left(P_{ratio}, \omega_k\right) \tag{3.28}$$

The nonlinear compressor model can be linearized around a particular system operating condition. This results in a linear compressor model of the form $y = f(u)$ that can be used for control design and to study the parametric sensitivity of the system. The full derivation of the linear model is presented in [63].

3.3.2 Electronic Expansion Valve

Two algebraic relationships are used to model the electronic expansion valve (EEV). Mass flow rate is calculated assuming standard orifice flow (Eq. 3.29), and using a semi-empirical map for the discharge coefficient (Eq. 3.30). The discharge coefficient is assumed to be a function of valve input, u_v, and pressure differential,

$\Delta P = (P_{in} - P_{out})$. Additionally, expansion is assumed to be an isenthalpic process (Eq. 3.31). For simulation purposes, the change in electronic input is rate limited to reflect the limitations of a real expansion valve.

$$\dot{m}_v = C_d \sqrt{\rho(P_{in} - P_{out})} \tag{3.29}$$

$$C_d = f_1(u_v, \Delta P) \tag{3.30}$$

$$h_{v,in} = h_{v,out} \tag{3.31}$$

The nonlinear EEV model can be linearized around a particular system operating condition. This results in a linear EEV model of the form $y = f(u)$ that can be used for control design and to study the parametric sensitivity of the system. The full derivation of the linear model is presented in [63].

3.4 Evaporator Frost Growth Model

In Section 2.2.3 it was noted that the dominant system level effect of evaporator frosting appears to be a reduction in the flow rate of air over the coil, while the net energy transfer between fluids decreases slightly. The heat transfer in the evaporator is not significantly impacted by the reduction in air flow rate because the increase in surface roughness and air velocity next to the plate has beneficial effects that improve heat transfer to the air flowing by the heat exchanger. To capture this behavior it will be important to develop a model that will accurately capture the frost growth on the heat exchanger and its impact on air flow rate and heat transfer.

One of the primary obstacles in applying the model presented in [48] is that the model assumes a constant air mass flow rate across the evaporator as frost growth occurs. In a physical vapor compression system it is unlikely that the evaporator fan will be able to account for reductions in free flow area within the heat exchanger by increasing the fan speed to retain the unobstructed air mass flow rate. To accurately predict the impact of frost formation, the model should vary the flow rate of air over the coil as the layer of frost grows. This creates a situation where the air mass flow rate is a function of the pressure drop across the heat exchanger, which in turn is a function of the free flow area and air mass flow rate. In order to avoid an algebraic loop within the model, the interdependence between the air mass flow rate and the pressure drop across the heat exchanger must be eliminated. Subsection 3.4.2 presents two methods that can be used to eliminate this interdependence between the air mass flow rate and the pressure drop across the evaporator. A general outline of the frost growth model is presented in Figure 3.3.

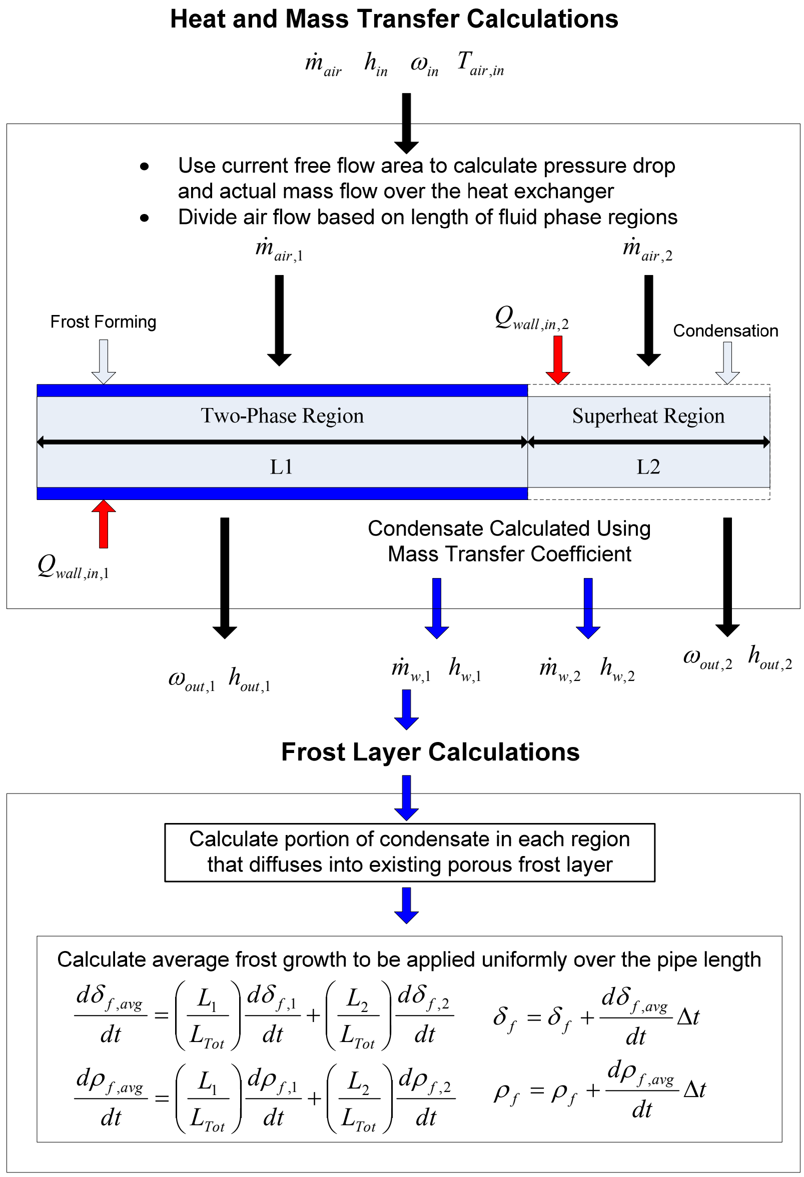


Figure 3.3 - Overall structure of frost model when the wall temperature in the two phase region is below 0° C

3.4.1 Frost Growth Model

The frost growth model impacts the wall energy balances of the original evaporator model, given in Eqs. 3.8 and 3.9. The main difference is that the term relating to the energy transfer from the air to the wall must be altered to account for the new heat transfer interactions that occur as a layer of frost forms. These interactions can be divided into three distinct sections; the air-frost interface, the existing frost layer, and finally the heat exchanger wall. To generate the frost model the following assumptions will be used:

1. Frost density is uniform across the thickness of the frost layer
2. The frost layer is porous, therefore the water vapor is divided into two parts
    a. A portion that diffuses into the existing frost layer
    b. A portion that increases the thickness of the frost layer

3. A small thickness and frost density is assumed at the start of the frosting process to avoid singularity, therefore the frost model does not capture the initial crystallization process at the beginning of a frost formation
4. Water vapor added to the frost layer remains in a solid state (no melting)
5. The frost layer can be considered to be at the same temperature as the heat exchanger wall temperature for a particular fluid region, therefore the thermal capacitance of the frost layer can be lumped with the heat exchanger wall
6. The frost growth occurs uniformly over the external surface of the heat exchanger, and is calculated as the average of the frost growth within the different fluid phase regions

It should be noted that previous studies that use similar frost models have found that these models are valid until approximately 50% of the free flow area in the heat exchanger has been eliminated, therefore the model results are only meant to approximate the frost formation behavior after the initial crystallization phase until there is a 50% blockage in the heat exchanger.

3.4.1.1 The Air-Frost Interface

At the air frost interface it is assumed that the sensible and latent heat transfer occurs simultaneously. Thus, the heat transferred from the air to the frost layer is given by Eqs. 3.32 and 3.33, where h_{sb} is the enthalpy of sublimation, $\dot{m}_w$ is the rate water is removed from the air, ω_{in} and ω_{out} are the air inlet and outlet specific humidity, $\dot{m}_\rho$ is the portion of the water vapor that contributes to increasing the density of the frost layer, and $\dot{m}_\delta$ is the portion of water vapor that increases the thickness of the frost layer.

$$\dot{Q} = \alpha_o A_o \left(T_a - T_w\right) + \dot{m}_w h_{sb} \qquad (3.32)$$

$$\dot{m}_w = \dot{m}_a \left(\omega_{in} - \omega_{out}\right) = \dot{m}_\rho + \dot{m}_\delta \qquad (3.33)$$

3.4.1.2 The Frost Layer

The total energy transferred from the moist air to the frost layer can also be written as the sum of the heat being conducted inside the frost layer and the energy added by the water vapor that sublimated into the porous frost layer [68], as shown in Eq. 3.34.

$$\dot{Q} = k_f A_o \left(\frac{dT}{dx}\right) + \dot{m}_\rho h_{sb} \qquad (3.34)$$

The mass flow rate of water vapor diffusing through the porous layer is given by Eq. 3.35 [48].

$$\dot{m}_\rho = A_o D_s \left[\frac{1 - \left(\dfrac{\rho_f}{\rho_{ice}}\right)}{1 + \left(\dfrac{\rho_f}{\rho_{ice}}\right)^{0.5}}\right] \frac{d\rho_v}{dx} \qquad (3.35)$$

The low partial pressure of the water vapor in the air permits the water vapor on the frost surface to be described by the ideal gas state equation, and therefore the following relation can be obtained [92].

$$\frac{d\rho_v}{dx} = \frac{1}{RT_w}\left(\frac{dP_v}{dT_w} - \frac{P_v}{T_w}\right)\frac{dT_w}{dx} \tag{3.36}$$

Where P_v is the partial pressure of the water vapor, R is the ideal gas constant for water vapor, and T_w is the temperature of the wall. Using the Clapeyron-Clausius Equation, the relation in Eq. 3.37 can be obtained.

$$\frac{dP_v}{dT_w} = \frac{h_{sb}}{T_w\left(v_g - v_{ice}\right)} \tag{3.37}$$

An empirical equation for the molecular diffusivity of water in a frosting process is given by Eckert and Drake [21].

$$D_s = \left(\frac{2.302\left(0.98\times10^5\right)}{P_a}\right)\left(\frac{T_w}{256}\right)\times10^{-5} \tag{3.38}$$

Combining Eqs. 3.35-3.37 produces Eq. 3.39.

$$\dot{m}_\rho = A_t D_s\left[\frac{1-\left(\dfrac{\rho_f}{\rho_{ice}}\right)}{1+\left(\dfrac{\rho_f}{\rho_{ice}}\right)^{0.5}}\right]\frac{1}{RT_w}\left(\frac{h_{sb}}{T_w\left(v_g - v_{ice}\right)} - \frac{P_v}{T_w}\right)\frac{dT_w}{dx} \tag{3.39}$$

Combining Eqs. 3.34 and 3.39 the following expression can be obtained.

$$\dot{m}_\rho = \frac{\dot{Q}}{D} \tag{3.40}$$

$$D = \frac{k_f}{D_s}\left(\frac{1+\left(\dfrac{\rho_f}{\rho_{ice}}\right)^{0.5}}{1-\left(\dfrac{\rho_f}{\rho_{ice}}\right)}\right)\left(\frac{RT_w^2\left(v_g - v_{ice}\right)}{h_{sb} - P_v\left(v_g - v_{ice}\right)}\right) + h_{sb} \tag{3.41}$$

3.4.1.3 Wall-Frost Interaction

The energy balances for the evaporator wall must now account for the impact of the frost layer on the effective thermal capacitance of the heat exchanger wall, as shown in Eq. 3.42.

$$\left(C_{pf}\rho_f A_o\delta + C_p\left(\rho V\right)_w\right)\dot{T}_w = \alpha_i A_i\left(T_r - T_w\right) + \dot{Q} \tag{3.42}$$

3.4.2 Pressure Drop and Air Mass Flow Rate

There are two ways to solve for the decrease in air mass flow rate that results from the growth of a frost layer on the evaporator. In the first method, a volume of air is used to break the algebraic loop that results from the interdependence between the air mass flow rate and the pressure drop across the heat exchanger. In the second method, detailed information of the fan behavior is used to eliminate the algebraic loop, as demonstrated in [38].

3.4.2.1 Plenum Solution Method

The plenum solution method assumes a fixed volume of air exists between the evaporator fan and the coil, as shown in Figure 3.4. The air mass flow rate entering the plenum from the fan is determined using a fan law. To use the fan law, an initial air mass flow rate must be assumed. The initial pressure of the plenum is calculated by using the pressure drop correlation for the heat exchanger and the initial air mass flow rate. The pressure drop correlation is used to determine the mass flow rate of air over the evaporator throughout the simulation. For the present analysis the pressure drop correlation for the heat exchanger (Eq. 3.43) was obtained from [45].

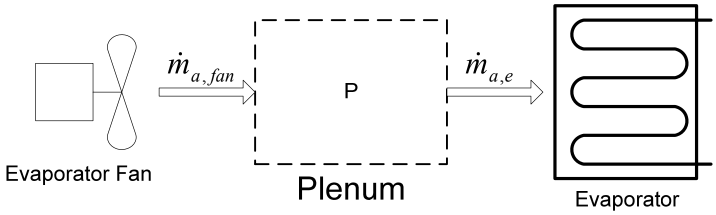


Figure 3.4 - A fixed volume plenum located between the fan and the evaporator.

For the following derivation, the following definitions are used.

A_o is the external surface area of the heat exchanger

A_{ff} is the free flow area through the heat exchanger

A_{front} is the frontal area of the heat exchanger

P_{atm} is the atmospheric pressure (at the fan inlet and evaporator outlet)

P is the pressure in the plenum

μ_a is the viscosity of the air

ρ_a is the density of the air

v_i is the specific volume of the air at the evaporator inlet

v_o is the specific volume of the air at the evaporator outlet

The pressure drop correlation is given by Eqs. 3.43-3.46.

$$\Delta P_a = \frac{G^2 v_i}{2}\left[\left(1+\sigma^2\right)\left(\frac{v_o}{v_i}-1\right)+f\left(\frac{A_o v_m}{A_{ff} v_i}\right)\right] \quad (3.43)$$

$$v_m = \frac{v_i + v_o}{2} \quad (3.44)$$

$$G = \frac{\dot{m}_a}{\sigma A_{front}} \quad (3.45)$$

$$\sigma = \frac{A_{ff}}{A_{front}} \quad (3.46)$$

The friction factor, *f*, is calculated from a correlation specific to the heat exchanger geometry, and is a function of Reynolds number. Substituting Eq. 3.43 into 3.45 and rearranging terms produces Eq. 3.47.

$$\dot{m}_a = \sqrt{\frac{\left(A_{ff}\right)^2 \Delta P_a}{C_1}} \tag{3.47}$$

$$C_1 = \frac{v_i}{2}\left[\left(1+\sigma^2\right)\left(\frac{v_o}{v_i}-1\right)+f\left(\frac{A_o v_m}{A_{ff} v_i}\right)\right] \tag{3.48}$$

The pressure drop in Eq. 3.47 can be calculated by taking the difference between the pressure of the plenum and the pressure of the atmospheric air at the exit of the evaporator, given by Eq. 3.49.

$$\Delta P_a = P - P_{atm} \tag{3.49}$$

The initial air mass flow rate and pressure drop are used in the fan law to obtain Eq. 3.50.

$$\dot{m}_{a,fan} = \dot{m}_{a,0}\sqrt{\frac{P_0 - P_{atm}}{P - P_{atm}}} \tag{3.50}$$

If the air in the fixed volume plenum is treated as an ideal gas with no leakage, then the rate of change in pressure within the plenum can be calculated using Eq. 3.51.

$$\dot{P} = \frac{RT}{V}\frac{\partial m}{\partial t} + \frac{mR}{V}\frac{\partial T}{\partial t} \tag{3.51}$$

R is the ideal gas constant for air

T is the air inlet temperature

V is the volume of the plenum

m is the mass of air in the plenum

If we assume that the air inlet temperature remains relatively constant, an assumption made in many air conditioning applications since the environmental conditions typically change gradually, Eq. 3.52 is obtained.

$$\frac{\partial T}{\partial t} \approx 0 \tag{3.52}$$

The conservation of mass for the plenum is given by Eq. 3.53.

$$\frac{\partial m}{\partial t} = \dot{m}_{a,fan} - \dot{m}_{a,e} \tag{3.53}$$

Combining Eqs. 3.51-3.53, the expression in Eq. 3.54 is obtained.

$$\dot{P} = \frac{RT}{V}\left(\dot{m}_{a,fan} - \dot{m}_{a,e}\right) \tag{3.54}$$

At each time step, the model uses the plenum pressure and the free flow area calculated by the frost growth model to solve for a new mass flow rate of air over the evaporator. It is interesting to note the effect of the volume, *V*, of the plenum has on the pressure response. If a small volume is used the pressure state will be highly responsive, and therefore will react significantly to small differences in air mass flow rate between the inlet and the outlet of the plenum.

3.4.2.2 *Empirical Fan Parameter Solution Method*

This method was proposed in [38], and requires specific manufacturer data on the fan behavior under various backpressure conditions. The basic idea is that fan manufacturers have operating data on the fan performance under different backpressure conditions. Using the information provided by the manufacturer for a specific fan, the volumetric flow rate of air through the fan for a backpressure condition can be calculated. Using a pressure drop correlation for a heat exchanger, such as the one presented in Eqs. 3.43-3.46, a system characteristic curve can be calculated that forms a relationship between volumetric flow rate and pressure drop across the heat exchanger. The system must operate at the intersection of the system characteristic curve and the fan backpressure performance curve. Figure 3.5 provides an example of the fan parameter solution method, where t1, t2, and t3 are system characteristic curves at three different times during the frost formation process, relating to different free flow areas. The diagram also includes fan backpressure performance curves for two different fan types.

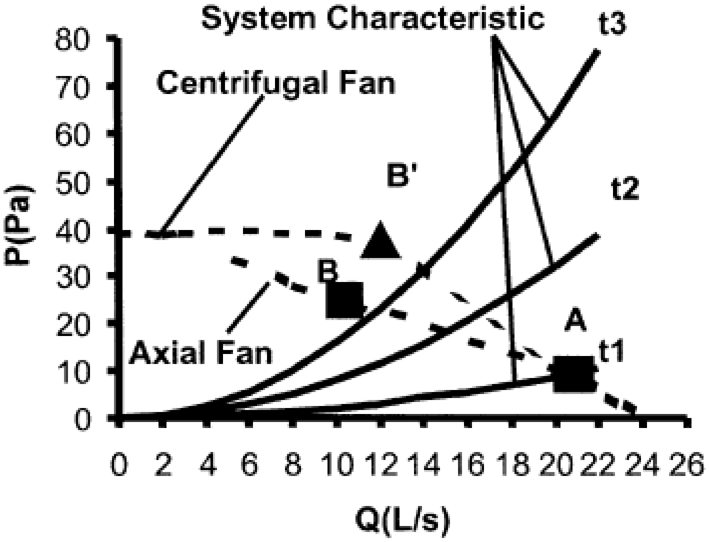


Figure 3.5 - Proposed solution method provided by [38]

3.4.3 Heat Transfer Coefficient

The external heat transfer coefficient at each time step is calculated using a j-factor correlation. To calculate the current free flow area, it is assumed that a layer of frost forms with a uniform thickness over the entire surface of the evaporator. This assumption is required since the lumped parameter moving boundary framework of the heat exchanger model limits the spatial effects that can be modeled. The j-factor calculation is discussed in detail in [22]. In the model, a new free flow area, hydraulic diameter, and Reynolds number is input to the j-factor calculation at each time step to reflect the impact that the growing frost layer has on the air flowing over the heat exchanger. A diagram of this equivalent frost growth process is presented in Figure 3.6.

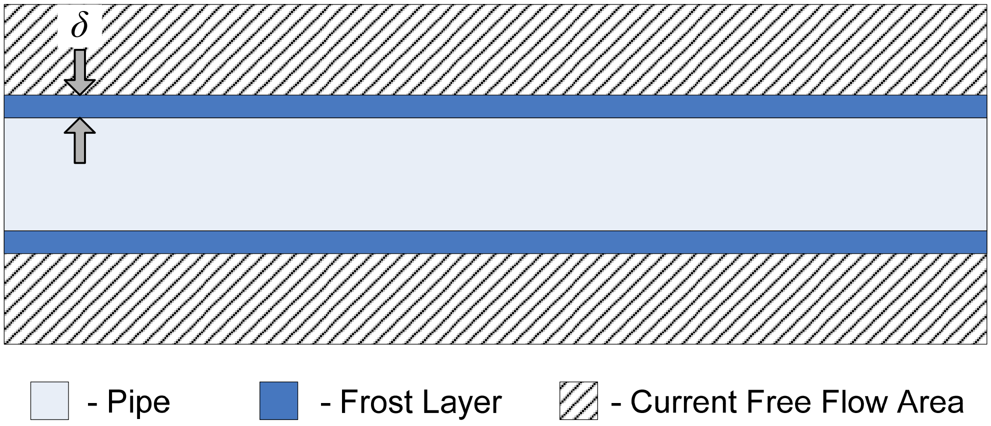


Figure 3.6 - Impact of frost growth on free flow area used to calculate the external heat transfer coefficient

3.4.4 Humid Air Calculations

In vapor compression systems with compact heat exchangers the local air temperature near the cooling coil is frequently below the saturation temperature of the air entering the system. The air near the cooling coil will be near the temperature of the coil, and therefore condensation may occur. In typical thermodynamics textbooks the case of condensation in a heat exchanger is handled using the ideal process of condensation. In the ideal condensation process, the air temperature is decreased uniformly to the dew point and any further cooling of the air results in condensation. In compact heat exchangers, local conditions will result in condensation when the average temperature of the exiting air flow is above the dew point, therefore another approach must be used to capture the condensation effect. The ideal and actual condensation processes are presented on a psychometric chart in Figure 3.7.

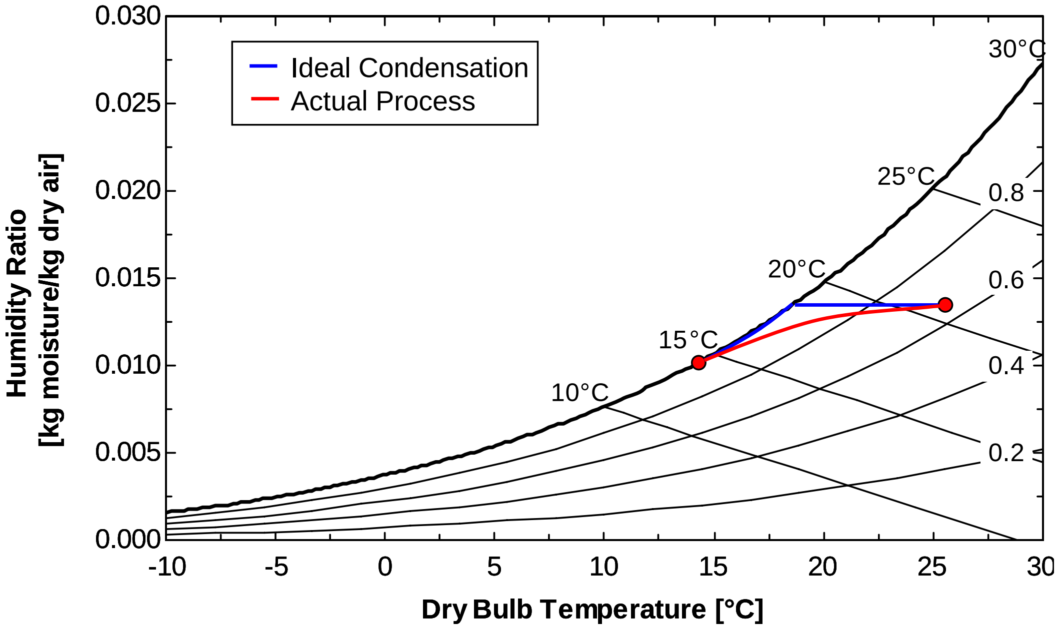


Figure 3.7 - Psychometric chart and associated dehumidification process in a compact heat exchanger

The mass transfer coefficient can be used to determine the amount of condensation that will occur when humid air passes through a compact heat exchanger [21,38,68]. Following the approach presented in [56], Eq. 3.55 can be used to solve for the specific humidity of the air at the evaporator outlet. ω_{wall} is the specific humidity of saturated air at the temperature of the wall in the heat exchanger.

$$\dot{m}_w = \dot{m}_a\left(\omega_{in} - \omega_{out}\right) = h_m A_o \eta_o \left(\frac{\left(\omega_{in} - \omega_{wall}\right) - \left(\omega_{out} - \omega_{wall}\right)}{\ln\left(\frac{\omega_{in} - \omega_{wall}}{\omega_{out} - \omega_{wall}}\right)} \right) \tag{3.55}$$

Assuming heat and mass transfer similarity for the heat exchange process within the evaporator causes the mass transfer coefficient to be a function of the heat transfer coefficient [21], as shown in Eq. 3.56.

$$h_m = \frac{\alpha_o}{\rho_a C_{p,a}} \tag{3.56}$$

The overall heat exchanger surface efficiency, η_o, is given by Eq. 3.57, and the fin efficiency, η_f, is obtained using Eqs. 3.58 and 3.59.

$$\eta_o = 1 - \frac{A_{fins}}{A_o}\left(1 - \eta_f\right) \tag{3.57}$$

$$\eta_f = \frac{\tanh\left(mL_c\right)}{mL_c} \tag{3.58}$$

$$m = \sqrt{\frac{\alpha_o P_{fin}}{k_{fin} A_c}} \tag{3.59}$$

The effective length, L_c, can be obtained using Eq. 3.60, which assumes an adiabatic fin tip and straight heat exchanger fins [40]. P_{fin} is the perimeter of the fin, t_{fin} is the thickness of the fin, L_{fin} is the length of the fin, A_c is the cross sectional area of the fin, and k_{fin} is the conductivity of the fin.

$$L_c = L_{fin} + \frac{t_{fin}}{2} \tag{3.60}$$

3.5 Modeling Refrigerant Leaks and Valve Faults

The impact of refrigerant leaks and valve actuation faults were also modeled to aid in FDD algorithm development. Subsections 3.5.1 and 3.5.2 detail the specifics of the fault models and their inclusion in the overall modeling framework.

3.5.1 Refrigerant Leaks

Refrigerant leaks are relatively simple to model within the Thermosys modeling framework. As an illustrative example, assume the leak occurs on the high side of the vapor compression system after the receiver. Figure 3.8 depicts the position of the leak at the outlet of the condenser with receiver model. Eqs. 3.61 and 3.62 detail the changes in the conservation of mass and conservation of energy equations for the receiver as a result of the refrigerant leak.

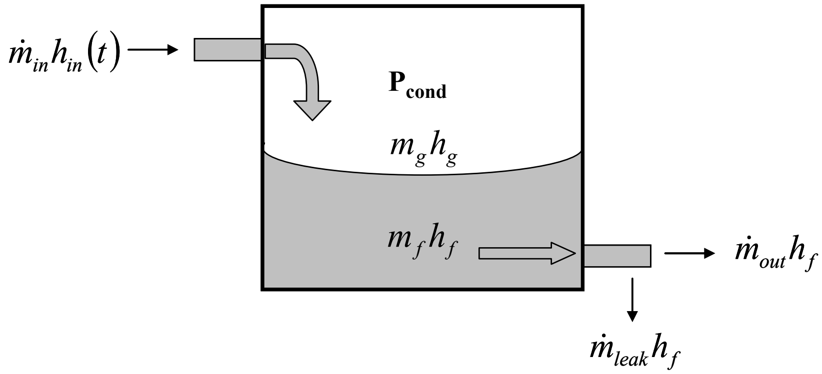


Figure 3.8 - Refrigerant leak at the outlet of the high side receiver

Conservation of Mass in the Receiver:

$$\dot{m}_{rec} = \dot{m}_{\text{int}\,2} - \dot{m}_o - \dot{m}_{leak} \tag{3.61}$$

Conservation of Energy in the Receiver:

$$\left[\frac{d\rho_g}{dP_c}V_g u_g + \frac{d\rho_f}{dP_c}V_f u_f + m_g\frac{du_g}{dP_c} + m_f\frac{du_f}{dP_c} - \left(\frac{\rho_g u_g - \rho_f u_f}{\rho_g - \rho_f}\right)\left(\frac{d\rho_g}{dP_c}V_g + \frac{d\rho_f}{dP_c}V_f\right)\right]\dot{P}_c$$
$$+\left(\frac{\rho_g u_g - \rho_f u_f}{\rho_g - \rho_f}\right)\dot{m}_{rec} = \dot{m}_{\text{int}\,2}h_{\text{int}\,2} - \dot{m}_o h_f - \dot{m}_{leak}h_f - UA_{rec}\left(T_{rec} - T_{amb}\right) \tag{3.62}$$

By combining the $\dot{m}_o$ and $\dot{m}_{leak}$ terms from Eqs. 3.61 and 3.62, the quantity would be equivalent to the total refrigerant exiting the heat exchanger. If it is assumed that the refrigerant leaks at the same thermal state as the exiting refrigerant, the enthalpy of the liquid for both the leak and the receiver outlet would be the same. Hence, by simply modifying the output mass flow rate quantity that is input to the condenser with receiver model, the model would capture the impact of the leak. This process could be done outside the condenser with receiver model, and could use a variety of leak models ranging from a simple steady leak to a leak based on the pressure of the condenser. Therefore, for all practical purposes it should be sufficient to model refrigerant leaks in the pipe models that interconnect all of the cycle components.

To include refrigerant leaks in the pipe model the simplest approach is to pick an end of the pipe where the leak occurs. This choice simplifies matters because the model already contains variables representing the enthalpy of the refrigerant at the inlet and the outlet of the pipe. For example, the leak at the outlet of the receiver could be introduced by adding the mass flow of the leak from the valve mass flow input to the condenser with receiver model. In this case the enthalpy of the leaking refrigerant would be equivalent to that of the refrigerant exiting the condenser with receiver model. A diagram of the model changes that would need to be included to introduce a constant leak at the outlet of the receiver is presented in Figure 3.9.

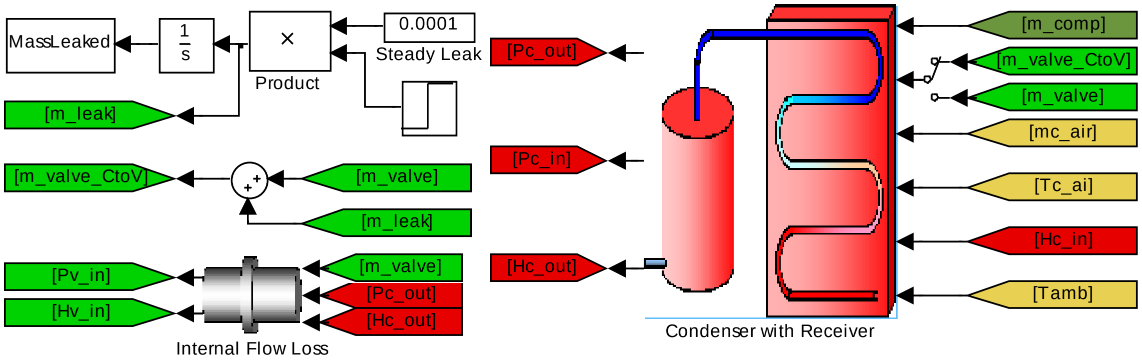


Figure 3.9 - Leak introduction into the simulation framework

3.5.2 Valve Actuation Faults

Industrial sponsors in the ACRC at the University of Illinois mentioned that when electronic expansion valves are used in air conditioning and refrigeration systems there is a significant likelihood that the valve will become stuck in a fixed position at some time during system operation. This fault condition can be simply modeled in Simulink by using a sample and hold on the valve command signal. At some point during simulation the valve actuation command is sampled and the value is held for the remainder of the simulation, regardless of the controller output. This effectively captures the impact of a valve actuation fault on a vapor compression system.

3.6 Model Limitations

The main limitation of the heat exchanger models is that they currently cannot handle the creation and annihilation of fluid regions during system transients. The heat exchanger models also use lumped heat transfer coefficients for each fluid region that are frequently difficult to identify. The parametric limitations of the lumped parameter models could be overcome if an effective means of identification or model optimization could be implemented.

The humid air model requires the use of a mass transfer coefficient that is specific to a given heat exchanger geometry, and therefore finding the proper empirical relationship from the literature can limit the effectiveness of the model. The frost growth model is also unable to account for spatial effects, whereas in an actual refrigeration unit the frost will grow at a more rapid rate near the evaporator refrigerant inlet port. The local rapid frost growth results in an increased level of blockage around the coldest heat exchanger walls in the evaporator. These spatial effects are difficult to address in the lumped parameter moving boundary model framework, and therefore the focus of the frost model is on system level impact rather than accurate spatial prediction of frost growth. The addition of the plenum also introduces a small amount of numerical instability if large step sizes are used in the simulation, therefore care must be used when selecting the simulation step size for a frost growth model.

Chapter 4. System Simulation

The Thermosys Toolbox was originally developed at the University of Illinois at Urbana-Champaign as a tool for simulating the transient performance of sub-critical and trans-critical vapor compression systems within Matlab/Simulink® [63]. The software package has progressed through a series of releases since its inception in 2002, and had reached a critical point where a significant restructuring was required to resolve a number of compatibility and development issues that plagued the previous versions. The result of the restructuring of Thermosys is what will be referred to as Thermosys Academic.

4.1 Introduction to Thermosys Academic

Thermosys Academic was created to alleviate some of the fundamental problems with the original Thermosys Toolbox. The new framework was designed to facilitate the inclusion of new/custom models to the existing software tool, and in the process eliminate compatibility problems with future releases of Thermosys and different versions of Matlab. This required a shift in the foundation of Thermosys from Simulink-based dynamic models to a loose framework that relies on Matlab functions. In the new framework, Simulink is used to handle the solution of the dynamic models while the Matlab functions contain the actual component models.

The component functions used for dynamic system simulation rely on global parameter structures that are created prior to simulation and stored in the Matlab workspace. To create a custom simulation the user will enter the physical parameters and operating conditions relating to their system into global parameter structures using a predefined naming structure. Using the operating conditions and physical parameters, initial condition solvers are used to calculate the initial states of the dynamic models. Once a global parameter structure with the initial states has been created, the user will enter the appropriate function name in the GUI associated with the various components of their system. Figure 4.1 depicts the general structure and information flow for Thermosys Academic.

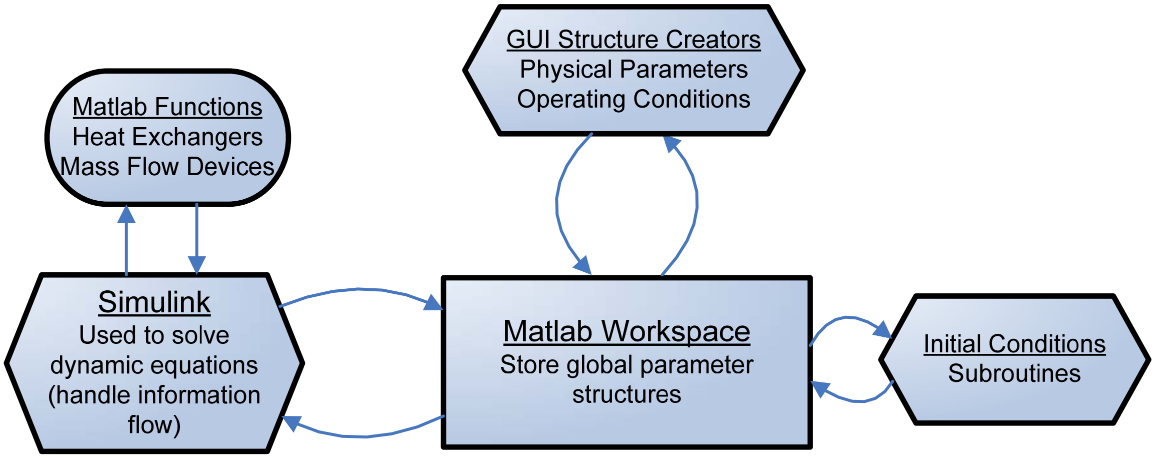


Figure 4.1 - General model structure and information flow for Thermosys Academic

4.2 Library Structure

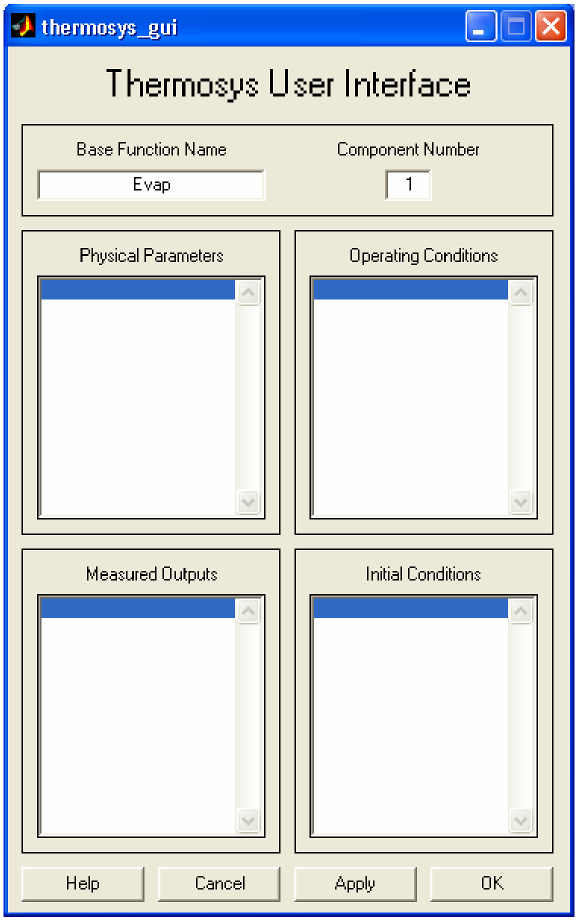
Thermosys Academic is a library of models and tools for simulating vapor compression systems. Vapor compression system models are simulated using the visual programming package Simulink®, while making extensive use of the commands and capabilities of MATLAB. These simulation tools can be easily accessed from the Simulink® Library Browser. The tools are organized into two directories: Components and Fluid Properties.

4.3 Thermosys Academic Simulink Blocks

Thermosys Academic employs three Simulink blocks to encompass all of the required functionality to dynamically simulate a vapor compression system. The blocks are quite general, and rely on the user to select the appropriate behavior (i.e. evaporator or condenser) to make the generic blocks match the actual components of the specific system the user would like to simulate. In the following subsections the three block structures and their associated graphic user interfaces are described.

4.3.1 Dynamic Model Block

The dynamic model block in the Thermosys Academic library is located under the components tab in the Thermosys Academic toolbox. This block can be used in a simulation to represent any dynamic component that is a part of the Thermosys Academic software. The flexibility is accessed by double clicking on the block when it is placed in a simulation, which opens the graphic user interface (GUI) for the dynamic model block. In this GUI, the user enters the base function name of one of the vapor compression component models which are included in the library. A list of the available functions is maintained at the Thermosys website. Figure 4.2 presents a screen shot of the GUI for the dynamic model block of Thermosys Academic.



Base Function Name: Evap

Component Number: 1

Figure 4.2 - Dynamic and static GUI for Thermosys Academic

The user also has the option to specify the component number in the GUI to allow for simulations of vapor compression systems with multiple components of the same type. The GUI contains four scroll boxes that will display the global parameter values associated with the component. The global parameter structures can be created using provided guides, and follow the naming convention where the function name concatenated with the component number is the primary subheading in the global parameter structure.

The inner workings of the dynamic model block can be accessed by a right click on the block and a selection of the “Look under mask” option. This opens the Simulink diagram presented in Figure 4.3. The dynamic model function for the component is placed in the Matlab Function block titled Model by the GUI. The GUI also writes the value of the Component Number field in the GUI to the block titled Num in Figure 4.3. A function is called by the GUI that sets the appropriate dimensions for the demux block and the number of outputs of all the Matlab Function blocks located within the dynamic model.

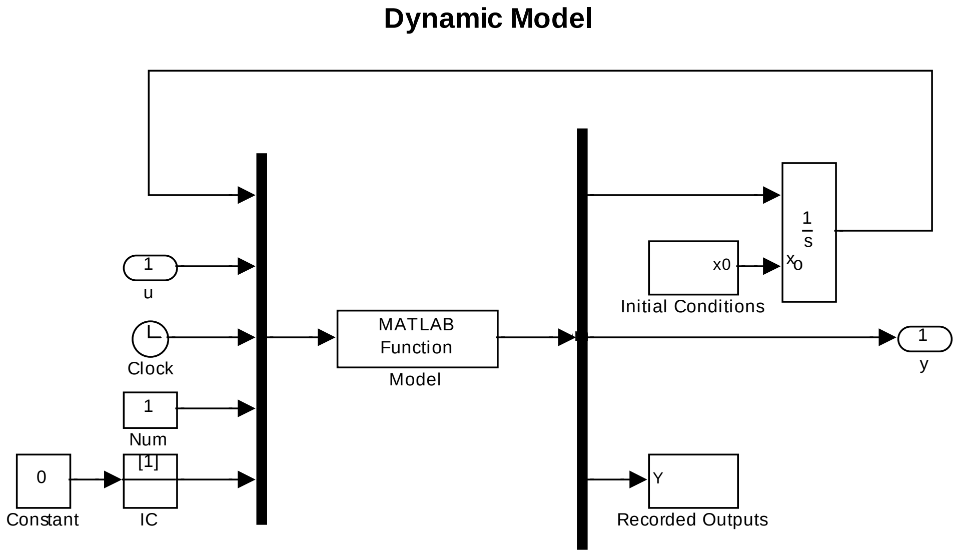


Figure 4.3 - Basic dynamic model layout for Thermosys Academic

If the user double clicks on the Initial Conditions subsystem in Figure 4.3, the Simulink diagram shown in Figure 4.4 will open. This is an enabled subsystem that runs at the very start of a simulation and outputs the initial conditions to the integrator block of Figure 4.3. By double clicking on the Enabled Subsystem block, the Simulink diagram depicted in Figure 4.5 will open. The GUI writes an appropriate function name using the root name provided by the user to the X0 block in Figure 4.5.

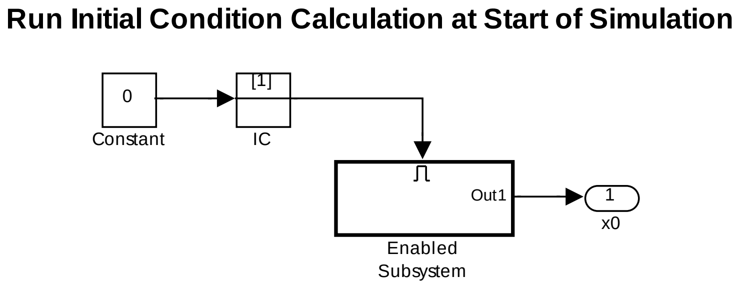


Figure 4.4 - Initial Conditions subsystem

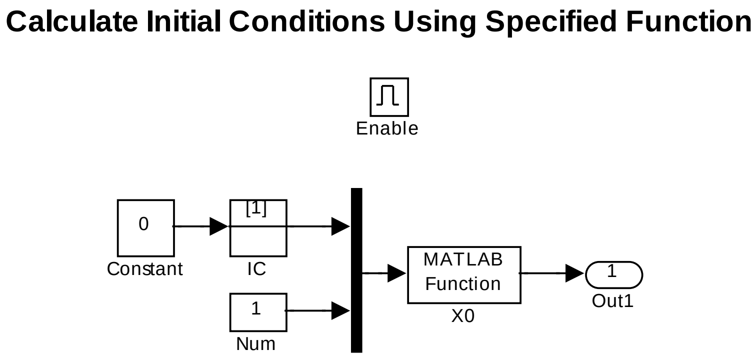


Figure 4.5 - The Enabled Subsystem used to calculate the initial conditions

Double clicking on the Recorded Outputs subsystem of Figure 4.3 opens the Simulink diagram shown in Figure 4.6. This subsystem contains a Record Function block that will store the simulation output in a global structure titled Output in the Matlab workspace. The output uses a naming structure consistent with that of the global structures created by the user prior to model simulation.

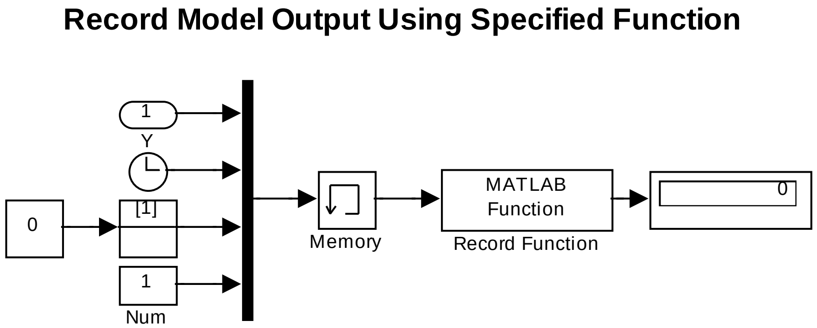


Figure 4.6 - The Recorded Outputs subsystem

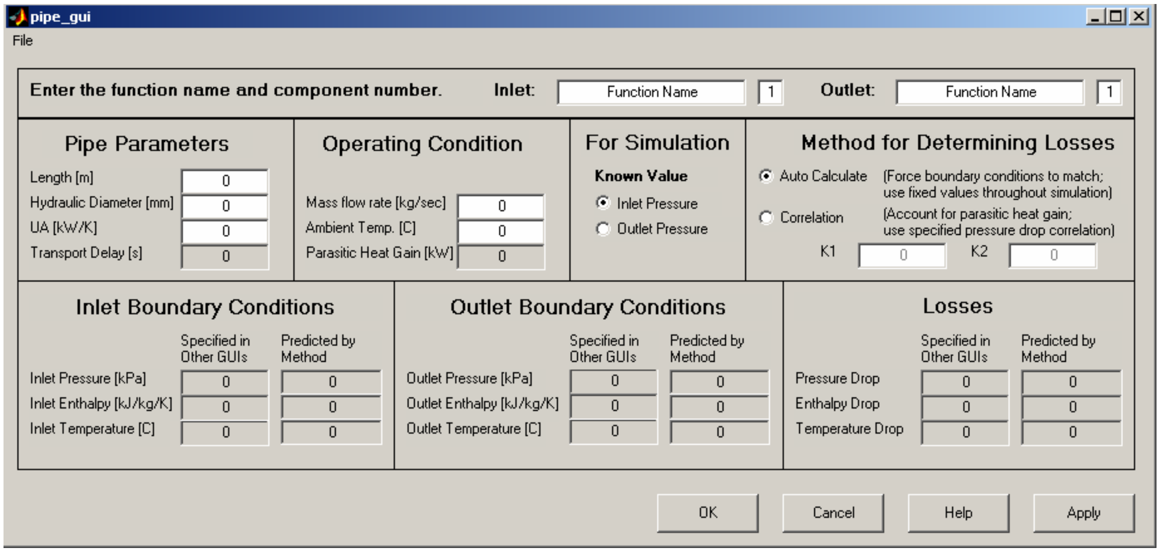
4.3.2 Static Model Block

The static model block uses the same GUI as the dynamic model GUI shown in Figure 4.2. Again, the user will enter the base function name for the static model they would like to use in the simulation. In general, static models are used for the mass flow devices, since the dynamics associated with these components are far faster than the heat exchanger models. The general structure of the model under the mask is quite similar to that presented in Section 4.3.1. The main difference is that the dynamic portion of the model has been removed.

4.3.3 Pipe Loss Block

In vapor compression systems there are some losses associated with the pipes that interconnect the primary system components. To capture these losses and account for any discrepancy in the operating conditions between components specified by the user, a pipe loss block has been included in the Thermosys Academic library. In its most rudimentary form, the pipe loss block simply calculates the pressure and energy drop between various components in the system. These calculated pressure and energy drops are then maintained throughout the simulation. The pipe loss block also allows the user to use predefined correlations to calculate the losses during the simulation. A detailed description of the correlations can be found on pages 64-68 and 167-169 of Rajat Shah's M.S. Thesis at the University of Illinois at Urbana-Champaign [69].

The GUI in Figure 4.7 will open when the pipe loss block is double clicked. The user must enter the function name and number of the component preceding and following the pipe, with respect to refrigerant flow direction. The user should then select the known pressure value in the For Simulation subheading within the GUI. In the modeling framework it is always assumed that the pressure is known in the heat exchanger, therefore the user needs to simply select if the heat exchanger is at the inlet or outlet of the pipe loss block.



Enter the function name and component number. **Inlet:** Function Name 1 **Outlet:** Function Name 1

Pipe Parameters

Length [m] 0
Hydraulic Diameter [mm] 0
UA [kW/K] 0
Transport Delay [s] 0

Operating Condition

Mass flow rate [kg/sec] 0
Ambient Temp. [C] 0
Parasitic Heat Gain [kW] 0

For Simulation

Known Value
● Inlet Pressure
○ Outlet Pressure

Method for Determining Losses

● Auto Calculate (Force boundary conditions to match; use fixed values throughout simulation)
○ Correlation (Account for parasitic heat gain; use specified pressure drop correlation)
K1 0 K2 0

Inlet Boundary Conditions

	Specified in Other GUIs	Predicted by Method
Inlet Pressure [kPa]	0	0
Inlet Enthalpy [kJ/kg/K]	0	0
Inlet Temperature [C]	0	0

Outlet Boundary Conditions

	Specified in Other GUIs	Predicted by Method
Outlet Pressure [kPa]	0	0
Outlet Enthalpy [kJ/kg/K]	0	0
Outlet Temperature [C]	0	0

Losses

	Specified in Other GUIs	Predicted by Method
Pressure Drop	0	0
Enthalpy Drop	0	0
Temperature Drop	0	0

Figure 4.7 - The Pipe Loss Block GUI

4.4 Global Parameter Structures

Thermosys Academic uses global parameter structures that are accessed by the various Matlab functions to perform a variety of tasks in the modeling process. The two basic structures that the user must input are named Physical_Parameter and Operating_Conditions. These two structures provide the models with the necessary heat exchanger geometry and the initial simulation operating condition required to calculate the initial conditions of the dynamic and static models. Initial condition subroutines are used prior to system simulation to calculate the Initial_Conditions global parameter structure. In general there are multiple ways to calculate the initial conditions for a particular model. These methods are incorporated into the same initial condition function for the component, and the function will print out a brief description of the assumption used to calculate the initial conditions when it is used. The user will need to decide which initial condition calculation method is most appropriate for their system, and supply the appropriate inputs to the subroutine to generate the desired initial conditions.

4.5 Thermosys Function Details

In Section 4.3 a significant number of references were made to locations where Matlab functions would be called by Simulink during model simulation. This section begins with a brief description of the basic functions that have been included in Thermosys Academic. New functions are continuously being added to the library as they are created by researchers at the University of Illinois, Texas A&M, or any others willing to share their models with the vapor compression system community. Subsection 4.5.1 contains all of the general information required to use the basic functions. The remaining subsections in this section provide further details on the model implementation and are primarily useful for more advanced users, or users who want to create their own models to incorporate into the software.

4.5.1 Functions for Basic Users

Currently there are a limited number of validated heat exchanger models that are included in the Thermosys Academic Toolbox. To keep this section concise, and since future development will likely add new functions, users should check the Thermosys website for more up to date information on available functions. For now this discussion will focus on the basic subcritical component models that are available.

There are four main subcritical heat exchangers included in the Thermosys Academic software package. All of the models are lumped parameter moving boundary models. The four heat exchangers types are listed below and their function name is included in parenthesis following the description.

- Evaporator (Evap.m)
- Condenser (Condenser.m)
- Evaporator with Receiver (EvapwRec.m)
- Condenser with Receiver (CondwRec.m)

If the user types “help” followed by one of the function names at the command prompt (e.g. “help evap”) a brief description of the model will appear. The description includes all of the required inputs to the model, lists the states of the dynamic model, and provides the user with the naming convention used to record outputs from the model. It is important that the user “Mux” the inputs to the model in the order listed in the help section, since this order is assumed internally by the function (see a Thermosys Academic sample system for an example). By specifying one of the appropriate function names in the GUI within Simulink, the m-file associated with the GUI will carry out all of the necessary model set-up to prepare the model for simulation. It is important to note that there is nothing to stop a user from entering a dynamic model function name into the static model block of Thermosys Academic. Therefore whenever a static model block is used, Thermosys Academic will print a friendly message at the Matlab command prompt reminding the user that the function name that they have entered should be a static model.

The four heat exchanger models listed above have a number of modeling assumptions that can be broken through inappropriate parameter specifications or extreme system transients. In the original Thermosys, the simulation would simply stop when the user violated these modeling assumptions, and the user would have to identify what went wrong on their own. In Thermosys Academic when a modeling assumption is violated the simulation will still stop, but Thermosys will also print exactly which assumption has been violated at the command prompt. This will streamline the trouble shooting procedure, and should help make the software more accessible.

4.5.2 Creating New Model Functions

This section details some of the programming details that are involved in Thermosys Academic. Ideally it will serve as a basic reference on the creation of new Thermosys Academic model functions. There are a number of challenges associated with the switch to a function based format in Simulink, and therefore some of these details need to be understood before a user attempts to add their own model to the Thermosys library.

In reality, the function name that the user enters in the GUI for the dynamic model block is only the base name of one of the four functions required to run a simulation. The base name that was entered is the actual dynamic model, but three other functions that have strings appended to the end of the file name are also used. For example, if the base function name the user entered was Evap, for the nonlinear evaporator model, then the GUI would place EvapIC and EvapRec in the Simulink function blocks associated with the initial conditions and recorded output subsystems of the dynamic model block. The GUI also calls a function called EvapDim that modifies the parameters of some of the blocks under the model mask. The following subsections explain the code associated with each of the four functions. It is recommended that the user consult one of the functions of Thermosys Academic as a reference when reading the following sections. If the reader does not have access to the toolbox, the code for a nonlinear evaporator model that can model frost growth is included in Appendix B. (Note: The EvapFrostMassTrans.m function is significantly more complicated than a basic function, but contains the same basic elements that are included in the basic Thermosys heat exchanger functions)

4.5.2.1 The Base Function

The top portion of the Evap.m m-file contains a large section of commented code, and the first block of commented code is the portion that will appear when the user types “help Evap” at the command prompt. After the help information, the creators have included a function history block where users or the creators of Thermosys Academic can detail changes that have been made to the function.

Each function begins by declaring the global and persistent variables that will be required for execution. In general the creators declared all of the potential global structures, as future models may rely on fields from different structures. Persistent variables are variables that will remain in the function workspace after the function has finished executing. Any variable that has not been declared persistent will be lost between the steps that Simulink takes as it solves the differential equations.

In the second section of the m-file the function input is broken into the component parts that will be used in the calculations. Since this is a nonlinear evaporator model, the model has 5 states and these states are placed in the variable x. The nonlinear evaporator also has 5 inputs and these are placed in the variable u. The function also uses the HX_num, simulation time, and a flag to detect the start of the simulation.

The third section contains the code that will be executed on the very first time step of the simulation. The primary use of the third section in the m-file is to import the required fields from the global parameter structures and store those values in a persistent parameter vector called ModParam. It should be noted that even if the user chooses a fixed time step in the simulation parameters, if the user selects a higher order solver the simulation will actually take smaller steps in between the time step specified by the user. This was one of the primary challenges with programming the functions for Thermosys Academic. Therefore when a user is writing their own model functions they should be aware of the fact that the number of steps between fixed time steps will vary based on the solver selected by the user.

The fourth section of the m-file contains the actual model. In a nonlinear model, this portion will calculate the Z(x,u) and f(x,u) terms from page 55 in [63]. A left matrix division is then used to calculate the new x_dot term. The user can type “help mldivide” at the command prompt for further details about this mathematical operation. The x_dot term will enter the integrator block in Figure 4.3, and is used to calculate the current state of the system.

The fifth section of the code contains checks for violations of the modeling assumptions associated with the model. In the case of the Evap.m file, there are three primary concerns. Since the Evap.m file represents a lumped parameter moving boundary model with two fluid regions, the model will become invalid if one of the two fluid regions disappears. The other potential error is a backflow condition, where refrigerant is flowing in the opposite direction through the heat exchanger. The model was not designed to handle refrigerant backflow, and a stop has been placed to detect this condition. If the user wishes, they can remove the model stops from the function, although in that case the user should be aware that the modeling assumptions can be violated and the model may produce unrealistic results.

The sixth and final section of the code creates the output vector that the function will return. The output vector contains the derivatives of the states, the outputs that are to be passed to other system components (inlet and outlet pressures and outlet enthalpy), and the individual outputs from the component that are to be recorded.

4.5.2.2 The Initial Condition Function

The EvapIC.m file from the Thermosys Academic library is fairly simple and contains only one section. The function is meant to be called at the first time step of the simulation and it should return the initial states of the evaporator model. Recalling Figures 4.4 and 4.5, the Simulink Matlab function block is contained within a triggered subsystem that will only be triggered on the first time step. This means that this m-file will not be called by Simulink after the first time step of the simulation. This was done to speed simulation execution since, after the first time step has finished, the states of the system are fixed and there is no longer a need to output the initial states to the integrator block. The m-file simply looks up the initial conditions created by the EvapIC_Creator function that were stored in the global Initial_Conditions structure.

4.5.2.3 The Record Function

The EvapRec.m function stores the component output in the global structure called Output. This function is necessary to record the model output only on the actual time steps of the simulation, and not the intermediate steps used by the various numerical solvers. In Figure 4.6, a memory block has been included prior to the input to the function. This prevents the output from intermediate and repeat time steps from entering the function. In this manner the recorded outputs will actually match the settings specified by the user in the Simulink parameters. A display has been placed at the output of the function block that indicates when the simulation is running. It should be noted that due to the memory block the very last time step of the simulation will not be recorded.

4.5.2.4 The Simulink Model Block Setup Function

The EvapDim.m function is called by the GUI, and changes some of the parameters of the Simulink diagrams contained under the mask of the dynamic model block. The function first changes the output dimensions parameter of the three Matlab function blocks to match the actual signals from the model selected by the user. In the case of the nonlinear evaporator model, there are 17 outputs from the Evap function block, 5 initial states are output from the EvapIC function, and 1 output from the EvapRec function. The EvapDim function also specifies how the signals output from the Evap function should be divided at the “Demux” block. Therefore it splits the output signal, sending the 5 state derivatives to the integrator block, the 3 block outputs to other components in the system, and the 9 recorded outputs to the EvapRec function.

4.6 Simulink Model Limitations

As with the previous releases of Thermosys, Thermosys Academic models are only valid when the assumptions applied in the modeling approach are valid. This means that the basic models are still unable to handle the creation and annihilation of fluid regions within the heat exchangers. In previous versions of Thermosys simulation stops were used to prevent the model from operating outside of regions where the modeling assumptions are valid. Unfortunately the model would simply stop without notifying the user of the modeling assumption that was being violated. When modeling assumptions are violated in Thermosys Academic the software automatically prints an error describing the nature of the violation, which is helpful for trouble shooting model failures. There are also limitations with regard to the fluid property tables that are addressed in [63].

Chapter 5. Experimental System

This chapter details the experimental air conditioning and refrigeration system that was used for dynamic model development, control design, and fault detection algorithm development at the University of Illinois at Urbana-Champaign. A detailed discussion of the physical system components is provided in [63]. This chapter will focus on the noise attenuation, data acquisition, and control framework of the system.

5.1 General System Description

The experimental system is a dual-evaporator “trainer” system, with tube-and-fin heat exchangers, semi-hermetic compressor, liquid line receiver, suction line accumulator, internal heat exchanger, an assortment of expansion devices, and a full suite of sensors. Individual components on the system can be included or excluded in the refrigerant circuit using an extensive set of manual valves. The valves allow the two evaporator circuits to be completely isolated, enabling the system to mimic the behavior of a single evaporator system. The system lacks the ability to control the air temperature entering each of the heat exchanger coils, limiting the ability to simulate the ambient conditions common to physical installations of similarly sized vapor compression systems.

The original system is pictured in Figure 5.8. A number of significant modifications have been made to the original system in order to provide the control authority and sensing capabilities required for transient model development, control design, and FDD algorithm implementation. A recent photo of the system is presented in Figure 5.9. Some important recent additions include a refrigerant recovery cylinder used to simulate refrigerant leaks, and a rudimentary humidity control system used for preliminary dynamic modeling efforts of condensation and frost in the evaporator.

Figure 5.8 - Photo of original system

Figure 5.9 - Photo of modified system

A schematic of the system is shown in Figure 5.10. Major components are identified by name, while an explanation of the valve acronyms is given in Table 5.3. The component models and manufacturers are listed in Table 5.4.

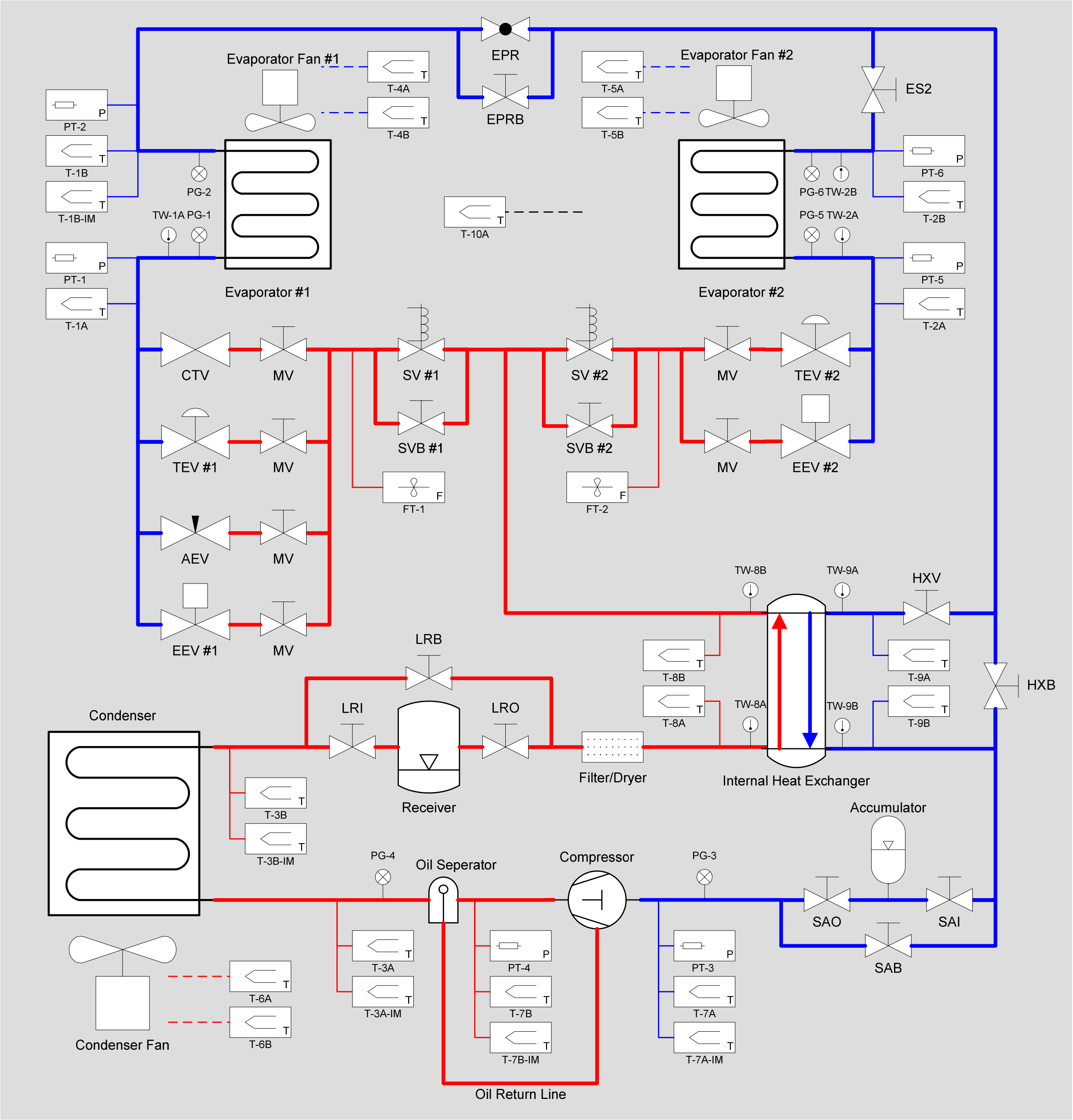


Figure 5.10 - Schematic of Experimental System

Table 5.3 - Valve Designations

Valve Designation	Description
CTV	Capillary Tube Valve
TEV	Thermostatic Expansion Valve
AEV	Automatic Expansion Valve
EEV	Electronic Expansion Valve
SV	Solenoid Valve
SVB	Solenoid Valve Bypass
EPR	Evaporator Pressure Regulating Valve
EPRB	Electronic Pressure Regulating Valve Bypass

Valve Designation	Description
ES2	Evaporator Side #2 Valve
HXV	Internal Heat Exchanger Valve
HXB	Internal Heat Exchanger Bypass
LRI	Liquid Line Receiver Inlet
LRO	Liquid Line Receiver Outlet
LRB	Liquid Line Receiver Bypass
SAI	Suction Line Accumulator Inlet
SAO	Suction Line Accumulator Outlet
SAB	Suction Line Accumulator Bypass
MV	Manual Valve

Table 5.4 - Component Name, Manufacturer, Model, and URL

Component	Manufacturer	Model	URL
Evaporator Fan and Casing	Larkin (HeatCraft)	VAK-17A	www.heatcraftrpd.com
Evaporator #1	HeatCraft	52601301 (VAK-17A)	www.heatcraftrpd.com
Evaporator #2	Blissfield	BH517	www.blissfield.com
Condenser and Fan	Tecumseh (Blissfield)	50803-1 (66001-3)	www.blissfield.com
Fan Control Boards	Control Resources Inc.	Nimbus 240BJW00	www.controlres.com
Internal Heat Exchanger	Superior (Sherwood)	HXSV-1/2	www.sherwoodvalve.com
Liquid Line Receiver	AC&R Components, Inc. (Henry Tech.)	S-8064	www.henrytech.com
Suction Line Accumulator	AC&R Components, Inc. (Henry Tech.)	S-7043	www.henrytech.com
Oil Separator	AC&R Components, Inc. (Henry Tech.)	S-5581	www.henrytech.com
Compressor	Copeland	KANA-006E-TAC-800	www.copeland-corp.com
Variable Frequency Drive	Baldor	ID15J101-ER	www.baldor.com
Capillary Tubing	Sealed Unit Parts Co., Inc.	BC-4	www.supco.com
TEV	Sporlan Valve Co.	FJ ¼ C	www.sporlan.com
AEV	Parker-Hannefin	A2	www.parker.com
EEV	Sporlan Valve Co.	SEI-0.5	www.sporlan.com
EEV Control Board	Sporlan Valve Co.	IB1, TCB	www.sporlan.com
EPR	Sporlan Valve Co.	ORIT 6-0/50-1/2”	www.sporlan.com
Manual Valves	Mueller Brass Co.	14838, 14841	www.muellerindustries.com
Filter-Dryer	Sporlan Valve Co.	C-052	www.sporlan.com
Sight Glasses	Sporlan Valve Co.	SA-14S, SA-12FM	www.sporlan.com
Pressure Transducers	Cole-Palmer	07356-53, 07356-54	www.coleparmer.com
Pressure Gauges	Ritchie Engineering Co., Inc.	49051, 49052	www.yellowjacket.com
Mass Flow Transducers	McMillan Company	102-6P	www.mcmflow.com
Mass Flow Gauges	Brooks Instrument (Emerson)	1350 EPIPMEAIA	www.emersonprocess.com/brooks
Immersion Thermocouple	Omega	GTMQSS-062U-6	www.omega.com
Welded Thermocouple	Omega	FF-T-20-100	www.omega.com
Watt Meter	Ohio Semitronics, Inc.	GW5-019D	www.ohiosemitronics.com
Pressure Switches	Ranco	010-1402, 011-1711	www.ranco.invensys.com
Line Reactors	MTE (Galco Industrial Electronics)	MTE RL-00402	www.galco.com
Analog Input Board	Measurement Computing, Inc.	PCI-DAS1200/JR	www.measurementcomputing.com
Analog Output Board	Measurement Computing, Inc.	PCI-DDA-08/12	www.measurementcomputing.com
Thermocouple Board	Measurement Computing, Inc.	PCI-DAS-TC	www.measurementcomputing.com
Terminal Boards	Measurement Computing, Inc.	CIO-MINI50	www.measurementcomputing.com
Signal Conditioners	Omega	OM5 Series	www.omega.com

5.2 Sensors

The experimental system is equipped with a wide range of sensing capabilities. Temperature measurements are obtained using type T thermocouples, and include air, surface, and refrigerant measurements. The presence of thermowells also provides the ability to take manual measurements using a thermometer. Refrigerant pressure is measured using strain-gauge based pressure transducers, while needle-based pressure gauges allow for visual corroboration. Turbine-based mass flow sensors are used to measure liquid mass flow prior to the expansion device. Finally, electric power consumed by the compressor is measured using a watt-meter. Table 5.5 details the location of the surface thermocouples, immersion thermocouples, thermowells, pressure transducers, pressure gauges, and flow transducers.

Table 5.5 - Temperature, Pressure and Flow Measurement Designations

Surface Thermocouple	**Immersion Thermocouple**	**Thermowell**	**Location**
T-1A		TW-1A	Evaporator #1 Inlet
T-1B	T-1B-IM	TW-1A (removed)	Evaporator #1 Outlet
T-2A		TW-2A	Evaporator #2 Inlet
T-2B		TW-2B	Evaporator #2 Outlet
T-3A	T-3A-IM	TW-3A (removed)	Condenser Inlet
T-3B	T-3B-IM	TW-3B (removed)	Condenser Outlet
T-4A			Evaporator #1 Air Inlet
T-4B			Evaporator #1 Air Outlet
T-5A			Evaporator #2 Air Inlet
T-5B			Evaporator #2 Air Outlet
T-6A			Condenser Air Inlet
T-6B			Condenser Air Outlet
T-7A	T-7A-IM	TW-7A (removed)	Compressor Inlet
T-7B	T-7B-IM	TW-7B (removed)	Compressor Outlet
T-8A		TW-8A	Internal Heat Exchanger – Liquid Inlet
T-8B		TW-8B	Internal Heat Exchanger – Liquid Outlet
T-9A		TW-9A	Internal Heat Exchanger – Vapor Inlet
T-9B		TW-9B	Internal Heat Exchanger – Vapor Outlet
T-10A			Ambient
Pressure Transducer	**Pressure Gage**	**Flow Transducer**	**Location**
PT-1	PG-1		Evaporator #1 Inlet
PT-2	PG-2		Evaporator #1 Outlet
PT-3	PG-3		Compressor Inlet
PT-4	PG-4		Compressor Outlet
PT-5	PG-5		Evaporator #2 Inlet
PT-6	PG-6		Evaporator #2 Outlet
		FT-1	Expansion Device #1 Inlet
		FT-2	Expansion Device #2 Inlet

5.3 VFD Noise Considerations

The variable frequency drive (VFD) dumps a decaying signal into the ground at the chopping frequency of the VFD (10 kHz). This signal was observed using an oscilloscope connected to the ground, and is the driving factor for the inclusion of optical isolation modules within the sensor and actuator wiring paths. The signal does not cause significant problems for higher level signals (signals measured in volts) if proper wiring is used on the VFD. To attenuate the signal strength as much as possible, the length of the leads from the VFD to the compressor was minimized. Line reactors were also included in the three phase power flow path to smooth the choppy signal from the VFD, helping to eliminate electrical reflection that can amplify the VFD's noise signal.

Unfortunately these modifications do not eliminate the signal's impact on the thermocouple measurements. The Type T thermocouples on the system output a signal in mV that is proportional to the temperature of the junction between the copper and constantan. The mV signal must be amplified before it is sampled on the thermocouple card. The amplification of the signal in the circuitry on the thermocouple card has an integrating effect on high frequency signals. If the decaying high frequency signal dumped by the VFD enters the thermocouple card through a ground loop with the copper piping of the system, the signal will be integrated, resulting in a steady state offset in the measurement. The magnitude of the offset is speed dependent, since the magnitude of the dumped signal from the VFD varies with actuating frequency. Figure 5.11 details the impact of the VFD noise signal on a single thermocouple measurement when an electrical connection exists between the VFD and the PC for various lengths of wire connecting the VFD to the compressor. The offset in thermocouple measurement in Figure 5.11 occurs when the VFD is powered on and set in forward mode with a speed of 0 RPM.

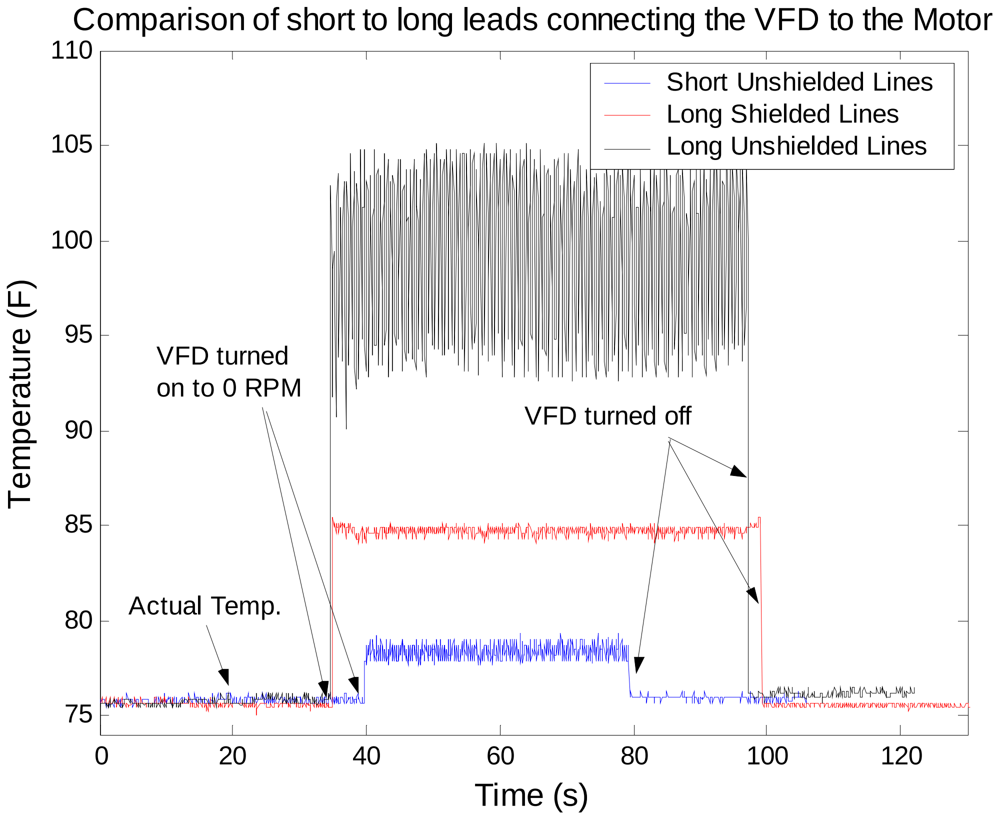


Figure 5.11 - Impact of the noise signal from the VFD on thermocouple measurements

5.4 Sensor Wiring

The extensive number of sensors and actuators on the experimental test stand result in a fairly complicated physical implementation. The implementation complexity was exacerbated by the high frequency signal dumped into the system ground by the variable frequency drive. This section details the reasoning behind the overall wiring architecture, and provides the important wiring diagrams that govern the operation of all of the components on the system. Figure 5.12 depicts a general overview of the flow paths that signals take from the PC to the system components.

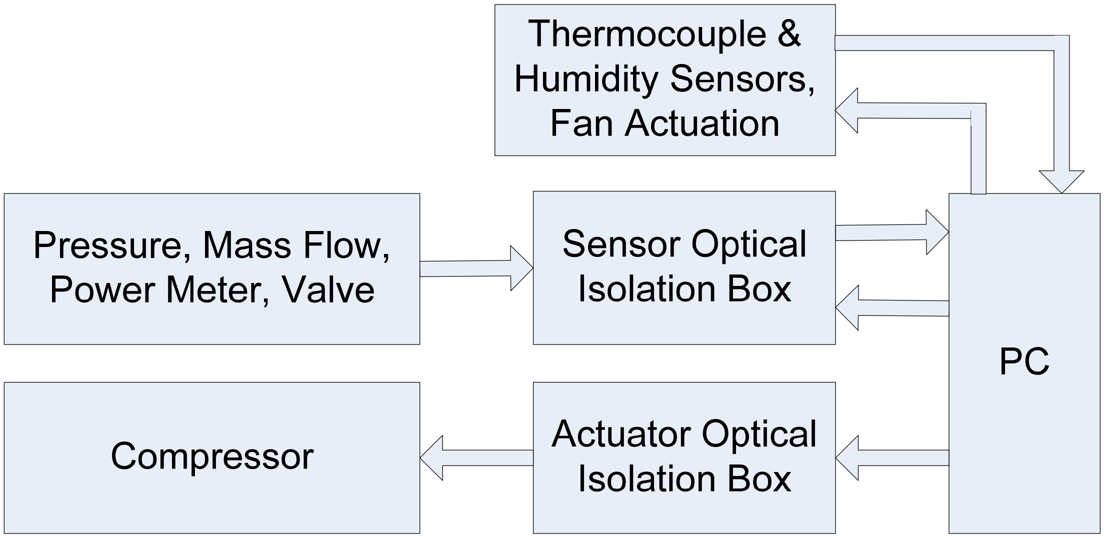


Figure 5.12 - Overall wiring structure for the experimental system

5.4.1 Thermocouples and Humidity Sensors

The thermocouples and humidity sensors do not use optical isolation units in the sensor signal path. The humidity sensors are powered from the 5 V power supply within the PC and the thermocouples do not require any power. A fuse is included within the computer case to protect the humidity sensors from spikes in voltage or current that could damage the sensors. The thermocouple PCI board is able to measure signals from 16 type T thermocouples. Since there are more thermocouples on the system than slots available on the thermocouple board, the sensors most relevant to each test are attached through screw terminals on the rear of the test stand, as depicted in Figure 5.13.

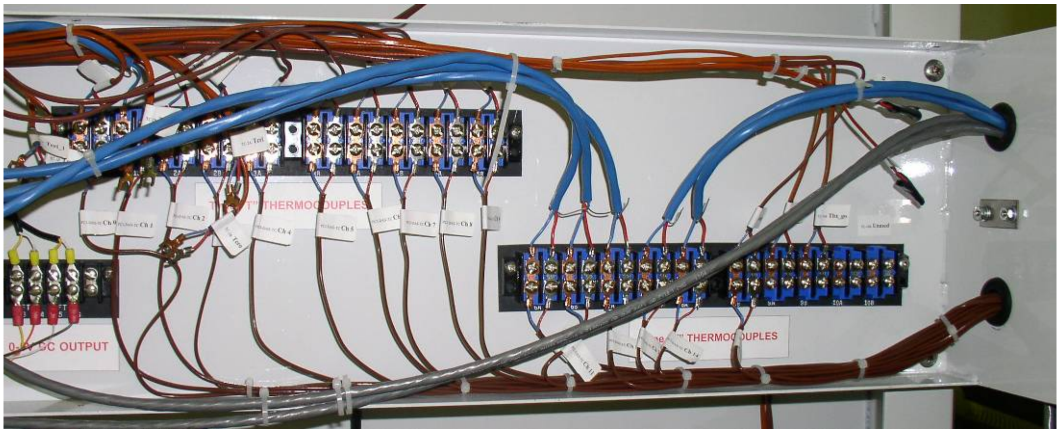


Figure 5.13 - Thermocouple screw terminal connections

5.4.2 Other Sensors and Valve Actuation

The pressure sensors on the system output 4-20 mA signals proportional to the gauge pressure on the diaphragm in the sensor. The 4-20 mA signal is converted to a 0-10 V signal at the screw terminals on the rear of the system as shown in Figure 5.14. The mass flow devices output a 0-5 V signal, and are wired to screw terminals to the right of the pressure sensors. The power meter is located in the power path between the wall outlet and the VFD. This means that the power meter is powered on the same side of the optical isolators as the VFD, and therefore contains the high frequency noise signal generated by the voltage chopping in the VFD. The power meter outputs a -10 to 10 V signal that must be optically isolated before it is sampled by the A/D PCI board in the PC.

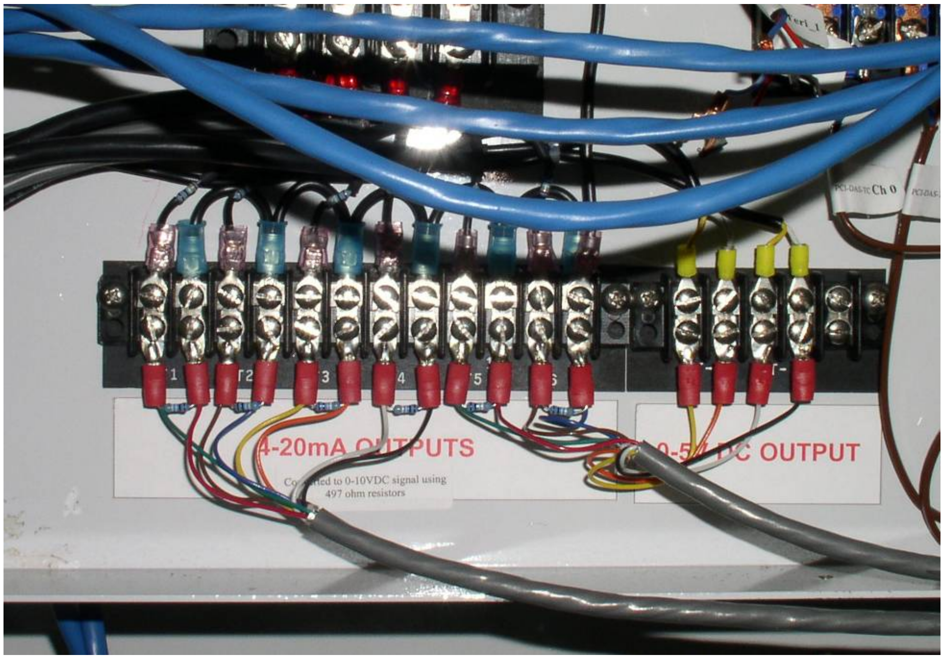


Figure 5.14 - The pressure and mass flow sensor screw terminals

The pressure, mass flow, and power meter signals are currently all routed through the sensor optical isolation box located behind the test stand. A picture of the sensor optical isolation box is presented in Figure 5.15. The sensor signals enter the box at the screw terminals in the lower right corner of Figure 5.15. The signals are passed back to the optical isolation units, and the isolated signals leaving the optical signal conditioners pass through the ribbon cable on the left wall of the box to the green screw terminal board. The eight isolation modules (7 blue with black top and 1 white with blue top) on the left of the box convert a -10 to 10 V signal to a 0 to 5 V signal. The two signal conditioners in the back right corner (blue with red tops) are output modules that take in a 0-5 V signal from the PC and output a 4-20 mA signal to the EEV control boards. A wiring diagram detailing the connections between the green screw terminal board and the 16 channel optical isolation backplane is presented in Figure 5.16.

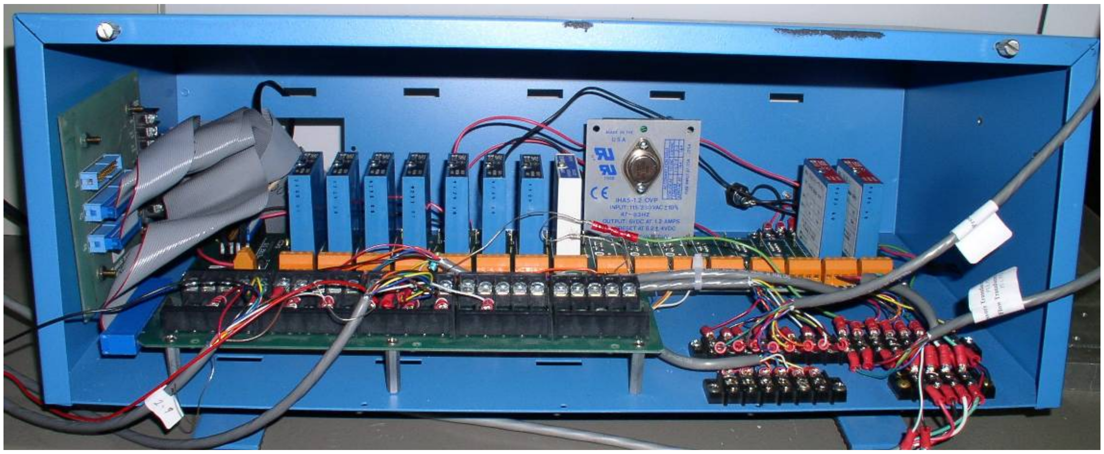


Figure 5.15 - The sensor optical isolation box

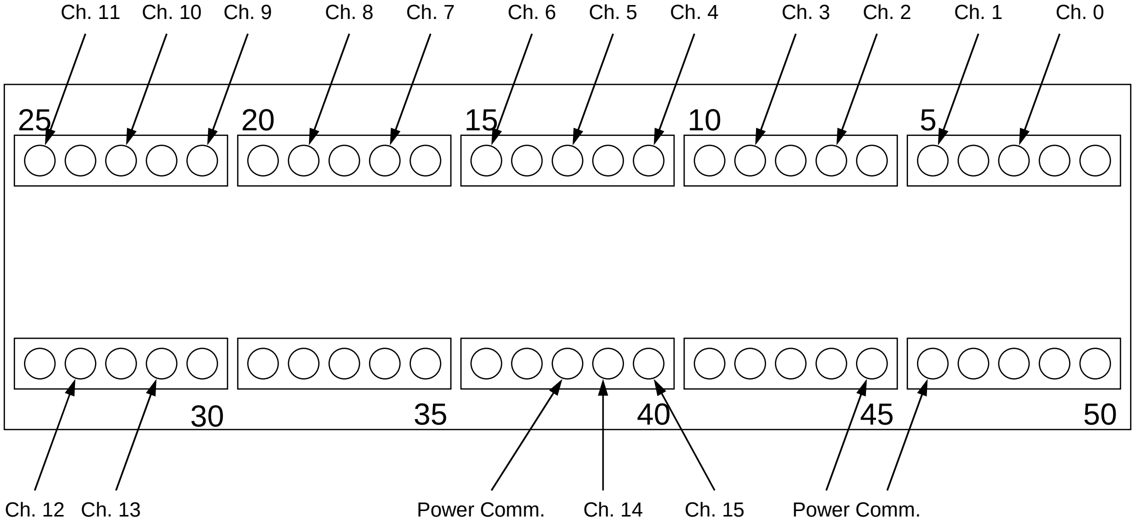


Figure 5.16 - Wiring diagram for connections between the green screw connection board and the optical isolation backplane

The 4-20 mA signal output from the red topped optical isolation units in Figure 5.15 are passed to the ICB board used to control EEV 1, and the TCB board used to control EEV 2. The wiring diagrams for the ICB and TCB are presented in Figures 5.17 and 5.18.

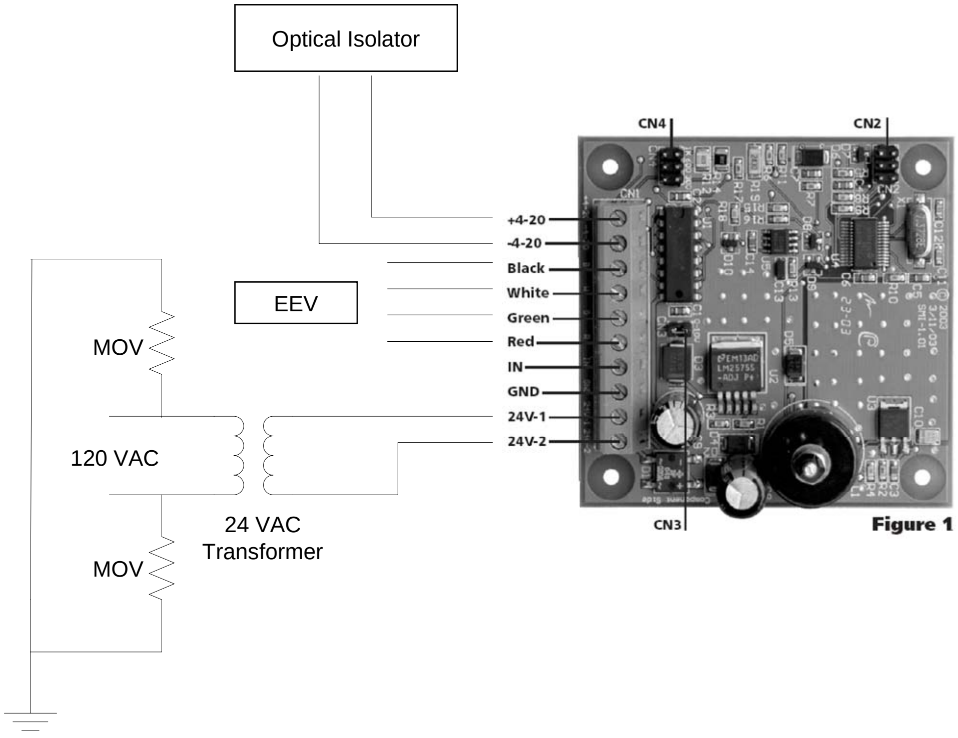


Figure 5.17 - Wiring diagram for the ICB

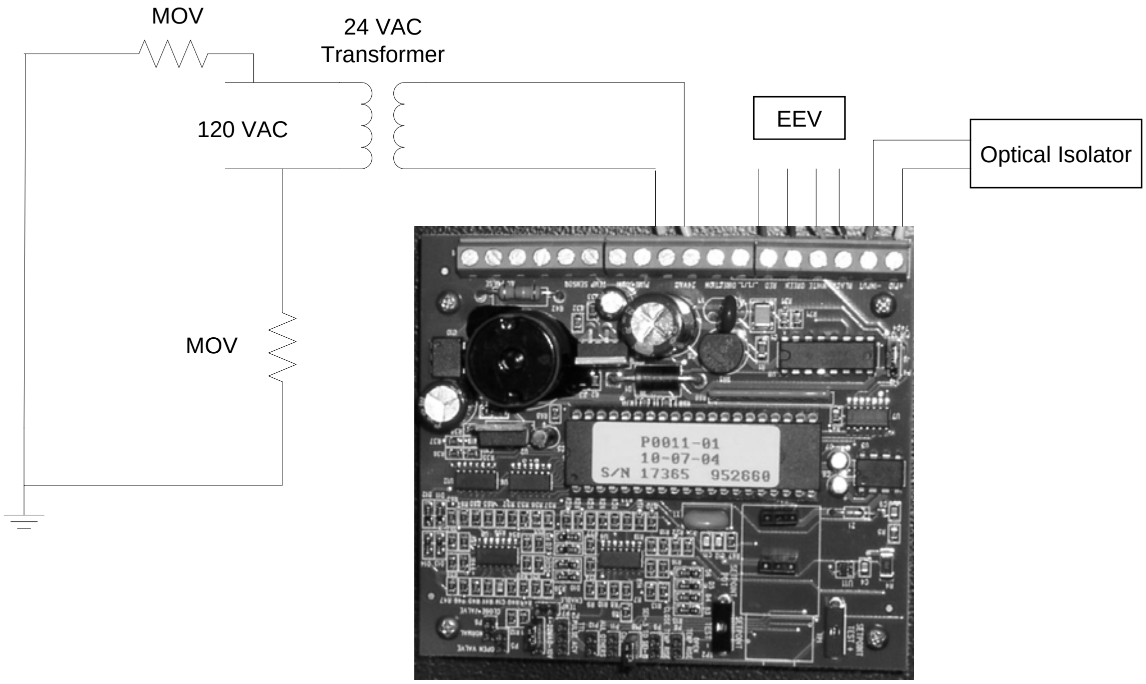


Figure 5.18 - Wiring diagram for the TCB

5.4.3 Compressor Actuation

The compressor speed is controlled through a -5 to 5 V signal output from the D\A PCI board. The -5 to 5 V signal enters an optical isolator located in the electronics box that contains the power meter for the compressor, and shown in Figure 5.19. The optical isolator outputs a 0 to 5 V signal to the VFD to control the compressor speed. It should be noted that if the VFD is set to remote speed control and the PC is restarted, a 0 V signal will be output by the D\A board in the PC, which translates through the optical isolator to a 2.5 V forward signal (approx. 1500 RPM) entering the VFD. Therefore, when the computer is restarted the VFD should be turned off or in the local control setting to avoid a situation where the compressor runs while the remainder of the system is turned off.

Figure 5.19 - The actuator optical isolation box

5.5 Data Acquisition

The PC is a Dell PowerEdge 400SC computer with a Pentium 4 - 2.8 GHz processor and 1 Gb of memory. Three PCI boards are contained within the computer and are used to sample the thermocouples, read the analog inputs, and output analog signals to the actuators. The data acquisition system is linked to Matlab using the Real Time Workshop kernel and WinCon v3.2. The drivers that govern the operation of the PCI boards are located in the Matlab6p1/rtw/c/dos/devices folder on the C:\ drive. Copies of the drivers are stored in the Matlab6p1/work/Drivers folder on the C:\ drive.

5.5.1 Thermocouple Board

The thermocouple input board is a PCI-DAS-TC board from Measurement Computing. The board can read 16 temperature measurements at a time, and is the limiting factor on the sample rate of the data acquisition system. The thermocouple board can not cycle through the 16 thermocouple channels faster than approximately 13 Hz, therefore the system is generally set to sample at 10 Hz. The PCI_DAS_TC.tlc driver contains the main code that governs the operation of the device when WinCon is running. The PCI_DAS_TC.tlc file contains three main sections; the Start function sets the registers of the card for the desired mode of operation and is run when the model is built, the Outputs function steps through the channel sampling at each time step during system operation, and the Terminate function is run when the Stop button is pressed on the WinCon server. The PCI_DAS_TC.h file contains definitions of the register values and functions called by the PCI_DAS_TC.tlc file. The PCI_DAS_TC.c file pulls important parameters from the Simulink file and the S-function block in Matlab to set up the communication of data during system operation.

5.5.2 The Analog Output Board

The analog output board is a PCI-DDA-08/12 board from Measurement Computing. The board can output 8 analog voltage signals, and is used to control the actuators on the system. The pci08da.tlc driver contains the same three basic functions as the thermocouple board; a Start function, an Output function, and a Terminate function. The pci08da.h file contains the definitions of the register values and functions for the pci08da.tlc file. As with the thermocouple board, the pci08da.c file pulls important parameters from the Simulink file and the S-function block in Matlab to set up the communication of data during system operation.

5.5.3 The Analog Input Board

The analog input board is a PCI-DAS-1200/Jr board from Measurement Computing. The board can read 8 dual channel inputs or 16 single channel inputs depending on the mode of operation specified by the drivers. The PCI12TAD.tlc driver governs the main operation of the board and contains the Start, Outputs, and Terminate functions that correspond to the WinCon modes of operation. The PCI1200.h file contains the definitions of the possible register values for the analog input board. The PCI12TAD.h file sets up the analog input board's mode of operation and contains functions called by the PCI12TAD.tlc driver. The current mode of operation is to read 8 dual channel inputs with a range of 0 to 5 V. The PCI12TAD.c file pulls important parameters from the Simulink file and the S-function block in Matlab to set up the communication of data during system operation.

It was determined that there is a difference between the voltage of the ground on the experimental system and within the PC by setting the board to dual channel mode and shorting the two terminals on the test stand. With shorted terminals the analog input board should read a 0 V difference between the two channels. Instead it read a constant offset as the result of a difference between the ground on the computer and the ground on the system. This voltage offset was accounted for within the voltage conversion function located in the PCI12TAD.h file. When the offset was cancelled the pressure measured by the data acquisition system matched that measured by the visual gauge on the system.

Chapter 6. Parameter Sensitivity of Control Oriented Models

In this chapter the linear models of the basic mass flow devices (EEV and compressor) are linked to heat exchangers that follow them in the refrigerant flow path to form a combined mass flow device/heat exchanger model. These combined models are useful for parameter sensitivity studies as they can be simply combined to form overall system models using linear fractional transformations. The resulting overall system model contains a large number of physical and empirical parameters that impact its dynamic response. Typically the parameters can be lumped into two categories:

i. physical parameters and operating conditions which are known, within a degree of uncertainty
ii. parameters which are immeasurable but are known to fall within physical bounds.

Due to the number and nature of the parameters involved in the modeling framework, the ideas of parameter sensitivity are useful to enhance the model generation process. Sensitivity methods were specifically developed to analyze the influence of parameters on the dynamic response of a system model [26]. Trajectory sensitivity functions are one such sensitivity method that is well suited to identify the dynamic importance of system parameters [19,88]. Knowledge of the dynamic importance of the system parameters will aid with: understanding and handling the immeasurable parameters common to vapor compression cycle models, the design of vapor compression systems to meet specific dynamic objectives, model tuning and validation, and the development of dynamic fault detection and diagnosis algorithms for these systems.

Clearly, once the dynamic impact of the system parameters is well understood, the model can be effectively tuned to accurately reflect a physical system. The sensitivity analysis also helps identify appropriate means to account for some of the un-modeled system components, allowing the dynamic effects of these components to be lumped into the heat exchanger models. With accurate models of vapor compression cycles, advanced control strategies will be able to obtain improved system performance. The improved model accuracy will also aid in the generation of fault detection and diagnosis algorithms that reduce the cost of system maintenance as well as ensure the system is operating efficiently [10].

6.1 Creating an Overall System Model

The following sections overview the derivation of the overall system model used in the parameter sensitivity analyses. For a detailed discussion of the equations involved in the derivation the interested reader should consult Appendix A. The first step to forming an overall linear system model is to combine the mass flow devices in the system with the heat exchangers that follow them, in the direction of refrigerant flow. For the system considered in this discussion, this results in a subsystem model of the valve attached to the evaporator and another subsystem model of the compressor attached to the condenser with receiver model. This interlinking of the individual component models is detailed in Figures 6.1 and 6.2.

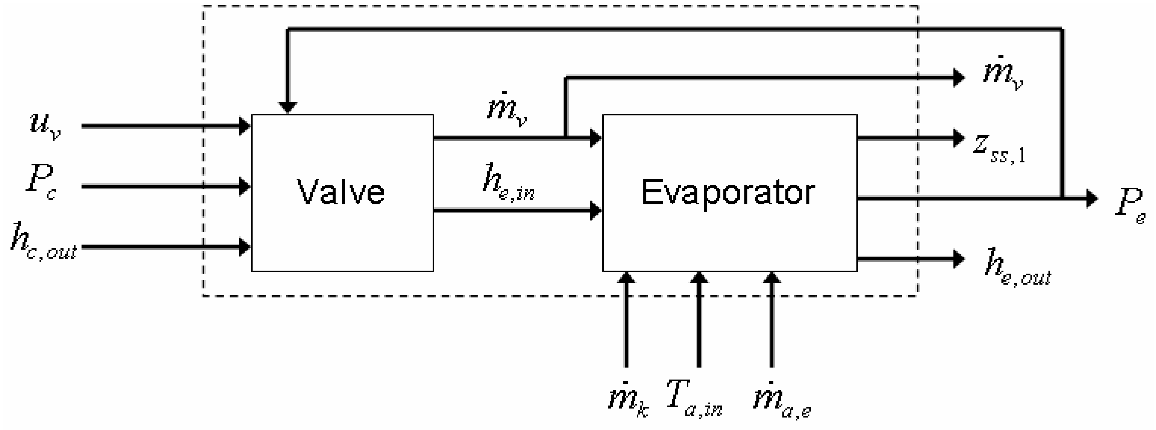


Figure 6.1 - Block diagram of the combined valve and evaporator model

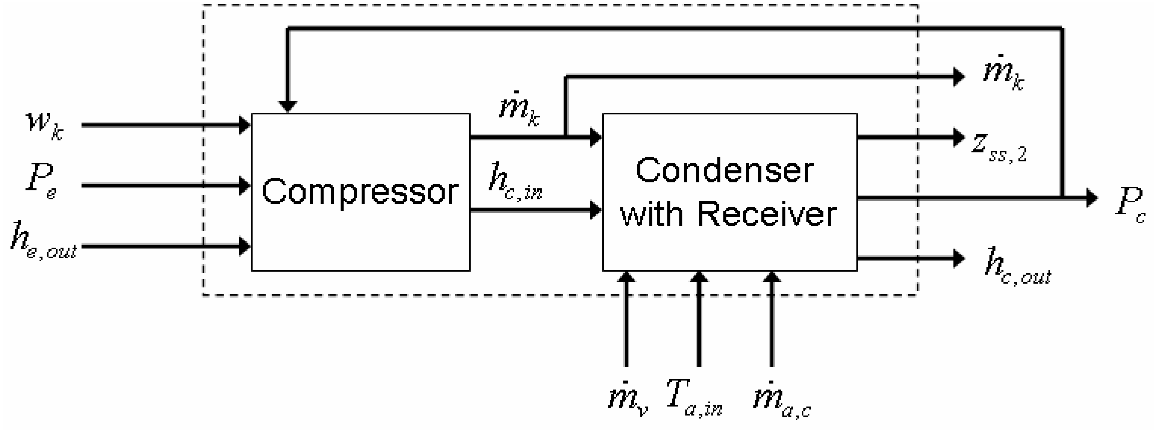


Figure 6.2 - Block diagram of the combined compressor and condenser with receiver model

Once the two subsystem models have been generated the models can be simply connected using the Redheffer star product. The block diagram for the star product is presented in Figure 6.3. For a detailed discussion of the procedure used to combine the system models and the details of the Redheffer star product, the reader is referred to [94].

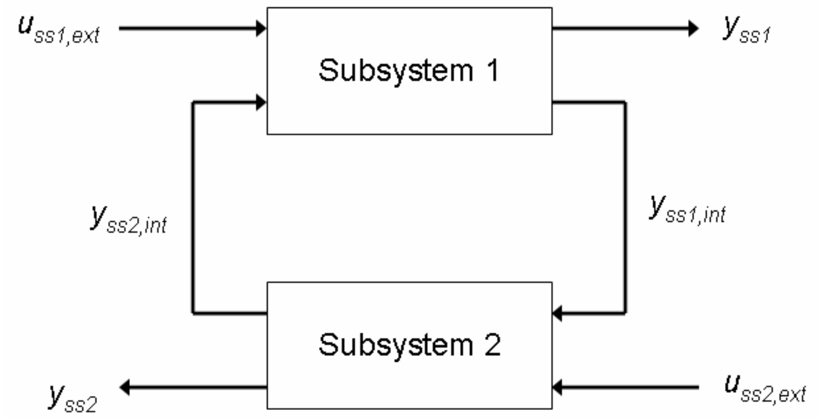


Figure 6.3 - Block diagram of the Redheffer star product

For simplicity the overall system model did not include the entire set of outputs possible from the combined subsystem models. Instead a subset of outputs was chosen to focus the sensitivity analysis on important signals within the system. The outputs of the valve/evaporator subsystem which were included in the total model were evaporator pressure, enthalpy at the exit of the evaporator, mass flow rate through the valve, evaporator air outlet temperature, and evaporator refrigerant outlet temperature. The outputs from compressor/condenser with receiver subsystem that were included in the total model were the condenser pressure, enthalpy at the exit of the condenser, mass flow rate through the compressor, condenser air outlet temperature, and condenser refrigerant outlet temperature.

Equations 6.1 through 6.6 provide the input and output vectors for Figure 6.3.

$$u_{ss1,ext} = \begin{bmatrix} u_v & T_{a,in} & \dot{m}_{a,e} \end{bmatrix} \tag{6.1}$$

$$u_{ss2,ext} = \begin{bmatrix} \omega_k & T_{a,in} & \dot{m}_{a,c} \end{bmatrix} \tag{6.2}$$

$$y_{ss1} = \begin{bmatrix} P_e & h_{e,out} & \dot{m}_v & T_{ea,out} & T_{er,out} \end{bmatrix} \tag{6.3}$$

$$y_{ss1,\text{int}} = \begin{bmatrix} \dot{m}_v & P_e & h_{e,out} \end{bmatrix} \tag{6.4}$$

$$y_{ss2} = \begin{bmatrix} P_c & h_{c,out} & \dot{m}_k & T_{ca,out} & T_{cr,out} \end{bmatrix} \tag{6.5}$$

$$y_{ss2,\text{int}} = \begin{bmatrix} \dot{m}_k & P_c & h_{c,out} \end{bmatrix} \tag{6.6}$$

Applying the format detailed in [94], the P and K transfer matrices for the system interconnection are formed from the state space definition of subsystem 1 and 2, and are given in Eqs. 6.7 and 6.8.

$$P = \begin{bmatrix} A_{ss1} & B_{ss1,1} & B_{ss1,2} \\ C_{ss1,1} & D_{ss1,11} & D_{ss1,12} \\ C_{ss1,2} & D_{ss1,21} & D_{ss1,22} \end{bmatrix} \tag{6.7}$$

$$K = \begin{bmatrix} A_{ss2} & B_{ss2,1} & B_{ss2,2} \\ C_{ss2,1} & D_{ss2,11} & D_{ss2,12} \\ C_{ss2,2} & D_{ss2,21} & D_{ss2,22} \end{bmatrix} \tag{6.8}$$

The P and K matrices were combined to form the overall system model using Eqs. 6.9 through 6.14. These equations are found on page 268 of [94].

$$\overline{A} = \begin{bmatrix} A_{ss1} + B_{ss1,1}\tilde{R}^{-1}D_{ss2,11}C_{ss1,2} & B_{ss1,2}\tilde{R}^{-1}C_{ss2,1} \\ B_{ss2,1}R^{-1}C_{ss1,2} & A_{ss2} + B_{ss2,1}R^{-1}D_{ss1,22}C_{ss2,1} \end{bmatrix} \tag{6.9}$$

$$\overline{B} = \begin{bmatrix} B_{ss1,1} + B_{ss1,2}\tilde{R}^{-1}D_{ss2,11}D_{ss1,21} & B_{ss1,2}\tilde{R}^{-1}D_{ss2,12} \\ B_{ss2,1}R^{-1}D_{ss1,21} & B_{ss2,2} + B_{ss2,1}R^{-1}D_{ss1,22}D_{ss2,12} \end{bmatrix} \tag{6.10}$$

$$\overline{C} = \begin{bmatrix} C_{ss1,1} + D_{ss1,12}D_{ss2,11}R^{-1}C_{ss1,2} & D_{ss1,12}\tilde{R}^{-1}D_{ss2,12} \\ D_{ss2,21}R^{-1}C_{ss1,2} & C_{ss2,2} + D_{ss2,21}R^{-1}D_{ss1,22}C_{ss2,1} \end{bmatrix} \tag{6.11}$$

$$\overline{D} = \begin{bmatrix} D_{ss1,11} + D_{ss1,12} D_{ss2,11} R^{-1} D_{ss1,21} & D_{ss1,12} \widetilde{R}^{-1} D_{ss2,12} \\ D_{ss2,21} R^{-1} D_{ss1,21} & D_{ss2,22} + D_{ss2,21} R^{-1} D_{ss1,22} D_{ss2,12} \end{bmatrix} \tag{6.12}$$

$$R = I - D_{ss1,22} D_{ss2,11} \tag{6.13}$$

$$\widetilde{R} = I - D_{ss2,11} D_{ss1,22} \tag{6.14}$$

6.2 Linearization With Respect to a Parameter

In order to implement trajectory sensitivity functions on the overall system, it is necessary to linearize the system model with respect to a particular parameter, α. Therefore this section details the linearization of Eqs. 6.9 through 6.14. It should be noted that Eqs. 6.13 and 6.14 reduce to the identity matrix for the overall system, due to the sparse nature of $D_{ss1,22}$ and $D_{ss2,11}$. The necessary linearizations are presented in Eqs. 6.15 through 6.30.

$$\frac{\partial \overline{A}_{11}}{\partial \alpha} = \frac{\partial A_{ss1}}{\partial \alpha} + \frac{\partial B_{ss1,2}}{\partial \alpha} D_{ss2,11} C_{ss1,2} + B_{ss1,2} \frac{\partial D_{ss2,11}}{\partial \alpha} C_{ss1,2} + B_{ss1,2} D_{ss2,11} \frac{\partial C_{ss1,2}}{\partial \alpha} \tag{6.15}$$

$$\frac{\partial \overline{A}_{12}}{\partial \alpha} = \frac{\partial B_{ss1,2}}{\partial \alpha} C_{ss2,1} + B_{ss1,2} \frac{\partial C_{ss2,1}}{\partial \alpha} \tag{6.16}$$

$$\frac{\partial \overline{A}_{21}}{\partial \alpha} = \frac{\partial B_{ss2,1}}{\partial \alpha} C_{ss1,2} + B_{ss2,1} \frac{\partial C_{ss1,2}}{\partial \alpha} \tag{6.17}$$

$$\frac{\partial \overline{A}_{22}}{\partial \alpha} = \frac{\partial A_{ss2}}{\partial \alpha} + \frac{\partial B_{ss2,1}}{\partial \alpha} D_{ss1,22} C_{ss2,1} + B_{ss2,1} \frac{\partial D_{ss1,22}}{\partial \alpha} C_{ss2,1} + B_{ss2,1} D_{ss1,22} \frac{\partial C_{ss2,1}}{\partial \alpha} \tag{6.18}$$

$$\frac{\partial \overline{B}_{11}}{\partial \alpha} = \frac{\partial B_{ss1,1}}{\partial \alpha} + \frac{\partial B_{ss1,2}}{\partial \alpha} D_{ss2,11} D_{ss1,21} + B_{ss1,2} \frac{\partial D_{ss2,11}}{\partial \alpha} D_{ss1,21} + B_{ss1,2} D_{ss2,11} \frac{\partial D_{ss1,21}}{\partial \alpha} \tag{6.19}$$

$$\frac{\partial \overline{B}_{12}}{\partial \alpha} = \frac{\partial B_{ss1,2}}{\partial \alpha} C_{ss2,1} + B_{ss1,2} \frac{\partial C_{ss2,1}}{\partial \alpha} \tag{6.20}$$

$$\frac{\partial \overline{B}_{21}}{\partial \alpha} = \frac{\partial B_{ss2,1}}{\partial \alpha} D_{ss1,21} + B_{ss2,1} \frac{\partial D_{ss1,21}}{\partial \alpha} \tag{6.21}$$

$$\frac{\partial \overline{B}_{22}}{\partial \alpha} = \frac{\partial B_{ss2,2}}{\partial \alpha} + \frac{\partial B_{ss2,1}}{\partial \alpha} D_{ss1,22} D_{ss2,12} + B_{ss2,1} \frac{\partial D_{ss1,22}}{\partial \alpha} D_{ss2,12} + B_{ss2,1} D_{ss1,22} \frac{\partial D_{ss2,12}}{\partial \alpha} \tag{6.22}$$

$$\frac{\partial \overline{C}_{11}}{\partial \alpha} = \frac{\partial C_{ss1,1}}{\partial \alpha} + \frac{\partial D_{ss1,12}}{\partial \alpha} D_{ss2,11} C_{ss1,2} + D_{ss1,12} \frac{\partial D_{ss2,11}}{\partial \alpha} C_{ss1,2} + D_{ss1,12} D_{ss2,11} \frac{\partial C_{ss1,2}}{\partial \alpha} \tag{6.23}$$

$$\frac{\partial \overline{C}_{12}}{\partial \alpha} = \frac{\partial D_{ss1,12}}{\partial \alpha} C_{ss2,1} + D_{ss1,12} \frac{\partial C_{ss2,1}}{\partial \alpha} \tag{6.24}$$

$$\frac{\partial \overline{C}_{21}}{\partial \alpha} = \frac{\partial D_{ss2,21}}{\partial \alpha} C_{ss1,2} + D_{ss2,21} \frac{\partial C_{ss1,2}}{\partial \alpha} \tag{6.25}$$

$$\frac{\partial \overline{C}_{22}}{\partial \alpha} = \frac{\partial C_{ss2,2}}{\partial \alpha} + \frac{\partial D_{ss2,21}}{\partial \alpha} D_{ss1,22} C_{ss2,1} + D_{ss2,21} \frac{\partial D_{ss1,22}}{\partial \alpha} C_{ss2,1} + D_{ss2,21} D_{ss1,22} \frac{\partial C_{ss2,1}}{\partial \alpha} \tag{6.26}$$

$$\frac{\partial \overline{D}_{11}}{\partial \alpha} = \frac{\partial D_{ss1,11}}{\partial \alpha} + \frac{\partial D_{ss1,12}}{\partial \alpha} D_{ss2,11} D_{ss1,21} + D_{ss1,12} \frac{\partial D_{ss2,11}}{\partial \alpha} D_{ss1,21} + D_{ss1,12} D_{ss2,11} \frac{\partial D_{ss1,21}}{\partial \alpha} \tag{6.27}$$

$$\frac{\partial \overline{D}_{12}}{\partial \alpha} = \frac{\partial D_{ss1,12}}{\partial \alpha} D_{ss2,12} + D_{ss1,12} \frac{\partial D_{ss2,12}}{\partial \alpha} \tag{6.28}$$

$$\frac{\partial \overline{D}_{21}}{\partial \alpha} = \frac{\partial D_{ss2,21}}{\partial \alpha} D_{ss1,21} + D_{ss2,21} \frac{\partial D_{ss1,21}}{\partial \alpha} \tag{6.29}$$

$$\frac{\partial \overline{D}_{22}}{\partial \alpha} = \frac{\partial D_{ss2,22}}{\partial \alpha} + \frac{\partial D_{ss2,21}}{\partial \alpha} D_{ss1,22} D_{ss2,12} + D_{ss2,21} \frac{\partial D_{ss1,22}}{\partial \alpha} D_{ss2,12} + D_{ss2,21} D_{ss1,22} \frac{\partial D_{ss2,12}}{\partial \alpha} \tag{6.30}$$

6.3 Parameter Sensitivity Analysis

Two methods were used to analyze the parametric sensitivity of the vapor compression cycle models. The first method applied the ideas of trajectory sensitivity functions as presented in [19] to gain an intuitive feel for a parameter's dynamic importance and impact on the time domain response of the system. The trajectory sensitivity analysis provided insight and understanding of the physical importance of various heat exchanger parameters. In the second method, the physical parameters of the model were perturbed and the resulting dynamic changes in the input-output system model were analyzed in the frequency domain. The second method provided direct insight into the effect deviations in a parameter had on the simulation model. The two methods were compared to identify the physical significance and dynamic impact parameter perturbations have on the overall nonlinear system model.

6.3.1 Trajectory Sensitivity Functions

Trajectory sensitivity functions provide a visual representation of the change in a system's time domain response due to a variation in a particular parameter [19]. Recall the linear state space model from Section 6.1, where **y** is the model output vector. The trajectory sensitivity functions, $\delta \mathbf{y}_i$, for the model can be thought of as a first order approximation of the parameter induced error, which is represented in Equations 6.31 and 6.32. $\boldsymbol{\alpha}_n$ is the nominal parameter vector, and α_i is the parameter whose sensitivity is being explored.

$$\delta \mathbf{y}_i(t, \alpha_i) = \boldsymbol{\eta}_i \delta \alpha_i \tag{6.31}$$

$$\boldsymbol{\eta}_i = \left(\frac{\partial \mathbf{y}}{\partial \alpha_i} \right)_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_n} \tag{6.32}$$

For the trajectory sensitivity analysis, $\delta\alpha_i$ was assumed to be 10% of the nominal parameter value. The value of 10% is somewhat arbitrary, but it was selected since it represents a parameter change that would be desirable to detect, as deviations in parameters larger than 10% may negatively impact the performance of a vapor compression system. The linearization of Section 6.2 was performed around the nominal values of system parameters that were derived from a preliminary model validation. After the linearization was completed, the trajectory sensitivity functions could be calculated using the state space representation presented in Equations 6.33 and 6.34.

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \dfrac{\partial \mathbf{A}}{\partial \alpha_i} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \dfrac{\partial \mathbf{B}}{\partial \alpha_i} \end{bmatrix} \mathbf{W}\mathbf{u} \qquad \boldsymbol{\lambda} = \frac{\partial \mathbf{x}}{\partial \alpha_i} \tag{6.33}$$

$$\begin{bmatrix} \mathbf{y} \\ \dfrac{\partial \mathbf{y}}{\partial \alpha_i} \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \dfrac{\partial \mathbf{C}}{\partial \alpha_i} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dfrac{\partial \mathbf{x}}{\partial \alpha_i} \end{bmatrix} + \begin{bmatrix} \mathbf{D} \\ \dfrac{\partial \mathbf{D}}{\partial \alpha_i} \end{bmatrix} \mathbf{W}\mathbf{u} \tag{6.34}$$

The weighting matrix, **W,** given in Equation 6.35, was initially selected to scale the inputs to the system to accurately reflect the physical magnitudes of the nominal input signals.

$$\mathbf{W} = 0.1\begin{bmatrix} u_{v,0} & 0 & 0 & 0 & 0 & 0 \\ 0 & T_{ae,in,0} & 0 & 0 & 0 & 0 \\ 0 & 0 & \dot{m}_{a,e,0} & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_{k,0} & 0 & 0 \\ 0 & 0 & 0 & 0 & T_{ac,in,0} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dot{m}_{a,c,0} \end{bmatrix} \qquad (6.35)$$

u_v = valve command [% Open]

$T_{ae,in}$ = evaporator air inlet temperature [C]

$\dot{m}_{a,e}$ = evaporator air mass flow rate [kg/s]

ω_k = compressor speed [rpm]

$T_{ac,in}$ = condenser air inlet temperature [C]

$\dot{m}_{a,c}$ = condenser air mass flow rate [kg/s]

The visual impact a parameter has on the dynamic response of the system can be obtained by applying an excitation to the $\delta\mathbf{y}_i$ generated with the trajectory sensitivity framework. The output can be thought of as an approximation of the difference between the response of the nominal system model and the perturbed system model, if both models start from the same state, as shown in Figure 6.4. By noting the shape of the response of $\delta\mathbf{y}_i$, the parameter can be identified as having a steady state or transient impact on the input-output response of the system.

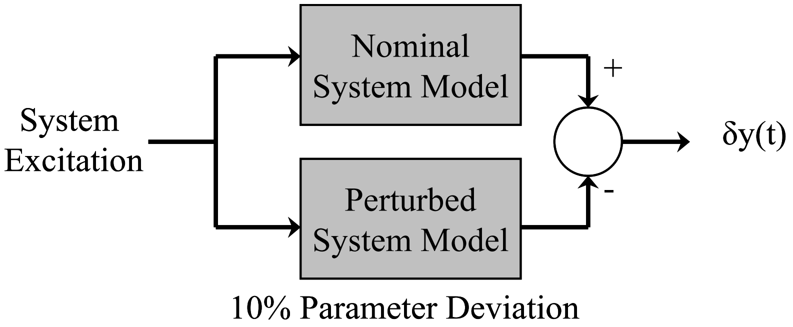


Figure 6.4 - Conceptual diagram of the approximate deviation in system output obtained through the trajectory sensitivity framework

6.3.2 Artificially Induced Parameter Perturbations

The second method for analyzing parameter sensitivity was a "brute force" approach that involved introducing small perturbations into parameters and recalculating the overall linear system model using the Thermosys framework. Starting with a baseline model of the vapor compression system, a parameter was selected and varied by 10% from its nominal value. The perturbed parameter value was then used to recalculate the overall system model with all other parameters and operating conditions remaining the same as the baseline model. The resulting linear system model that was calculated using the perturbed parameter was compared to the baseline model in the frequency domain to assess the magnitude of the dynamic changes caused by the parameter perturbation. In order to explore the nonlinearity of the system model, both positive and negative perturbations in the parameter were used.

Two metrics were used to identify the relative level of change in the input-output response of the dynamic model of the system, and are presented in Eqs. 6.36 through 6.38. Each metric was individually applied to each of the input-output pairs of the system model. The first metric was used to determine overall importance of a parameter by assessing the relative change in the maximum frequency response of the system. The second metric approximated the relative change in the steady state gain of the input-output transfer functions that resulted from a perturbation. Figure 6.5 depicts the frequency response characteristics identified by the two metrics when the mass of refrigerant in the receiver is perturbed.

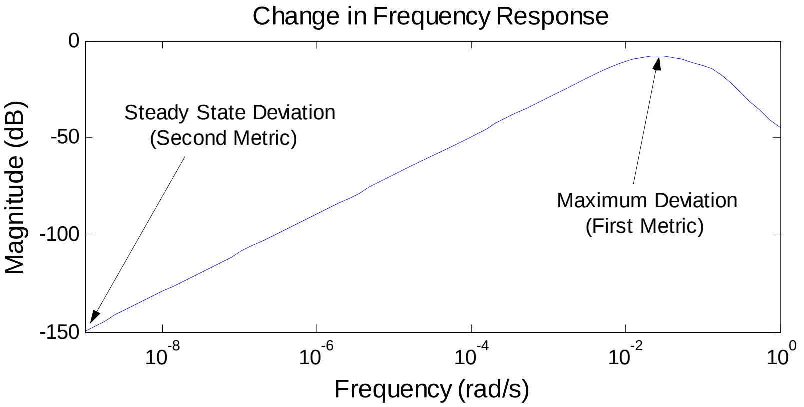


Figure 6.5 - Frequency response characteristics identified for the valve to condenser pressure I/O pair when the refrigerant mass in the receiver is perturbed

For Eqs. 6.36 through 6.38, G_0 is the original unperturbed transfer function for a single input-output pair, and $G_{perturbed}$ is the perturbed transfer function for the input-output pair.

$$\Delta G = \left(\|G_0\|_\infty\right)^{-1}\left(G_0 - G_{perturbed}\right) \tag{6.36}$$

$$Max \;\; Freq. \;\; response \;\; metric = \|\Delta G\|_\infty \tag{6.37}$$

$$Steady - State \;\; metric = \frac{\left|G_0(j\omega)\right| - \left|G_{perturbed}(j\omega)\right|}{\left|G_0(j\omega)\right|} \quad ; \quad \omega = 0 \tag{6.38}$$

Each metric was scaled by the magnitude of the metric for the original input-output pair to avoid unequal output weighting introduced by unit discrepancies. Since the metric was computed for each input-output pair, a matrix of values was generated for each of the metrics. Table 6.6 presents an example of the matrix of input-output Frequency Response metric values generated by a perturbation in the mass of refrigerant in the receiver. The induced 2-norm of the matrix of metric values was used to obtain a single number representing the overall dynamic impact of the parameter perturbation.

Table 6.6 - Example of I/O Frequency Response metric values

Max Frequency Response Metric Values for a 10% Decreasein the Initial Refrigerant Mass in the Receiver				
Outputs	**Inputs**			
	Valve Voltage (Vv)	**Compressor Speed (ω_k)**	**Evaporator Air Mass Flow Rate (m e_a)**	**Condenser Air Mass Flow Rate (m c_a)**
Evaporator Presure (Pe)	0.0043	0.0025	0.0044	0.0169
Evaporator Air Temp. Out (Te_ao)	0.0009	0.0019	0.0001	0.0335
Evaporator Ref. Temp. Out (Te_ro)	0.0024	0.0028	0.0013	0.0170
Condenser Pressure (Pc)	0.0174	0.0169	0.0156	0.0168
Condenser Air Temp. Out (Tc_ao)	0.0186	0.0169	0.0151	0.0152

6.3.3 Understanding Parameter Sensitivity

The two parameter sensitivity methods can be used to identify the dynamic impact and importance of the model parameters. This Subsection details the procedure used to glean information from each of the sensitivity analyses and provides a discussion of the differences and similarities between the results obtain through each method.

6.3.3.1 Trajectory Sensitivity Analysis

Figure 6.6 presents an example response of the trajectory sensitivity function for the internal surface area of the evaporator. In this case the trajectory sensitivity function was excited with a 10% change in the valve position, providing a time domain approximation of the deviation in evaporator pressure between the unperturbed and perturbed systems if they started at the same initial condition. The response indicates the impact perturbations in the internal surface area will have on the input-output relationship between the reference command of the electronic expansion valve and the evaporator pressure. Since the deviation in evaporator pressure response, δy, responds quickly and retains a non-zero steady state value, the parameter must have both transient and steady state impact. Due to the fact that this parameter retains a non-zero value at steady state, this parameter will significantly impact the steady state response of the input-output pair. This is in agreement with intuition, since the internal surface area has a direct impact on the amount of energy which is transferred between the refrigerant and the air.

To highlight the ease with which transient parameters can be identified, Figure 6.7 presents the response of the trajectory sensitivity function for cross sectional area of the refrigerant flow path in the evaporator excited by a 10% change in valve position. Upon first inspection the differences between the responses in Figures 6.6 and 6.7 are quite clear. The output deviation in evaporator pressure, δy, as a result of a perturbation in the cross sectional area decays to zero over time, implying that this parameter will have a minimal impact on the steady state response of the system (since the outputs of both the nominal and perturbed system models will be the same at steady state). The large initial peak in output deviation reveals that changes in cross sectional area cause the nominal and perturbed system models to have significantly different transient responses. The ease with which trajectory sensitivity functions allow the identification of these qualitative differences in a parameter's impact on the time domain response of the overall system model made them an ideal method for identifying tunable parameters.

Clearly, qualitative information on the impact of parameter perturbations was readily apparent from the trajectory sensitivity analysis, but the relative quantitative importance of the parameters must still be addressed. To this end two trajectory sensitivity metrics were used to determine the importance of a parameter on the steady state and transient system response. The simplest measures which could be obtained from the time response of the $\delta \mathbf{y}_i$ would be the maximum magnitude of the output deviation and the final value of the deviation in output between the nominal and perturbed system models.

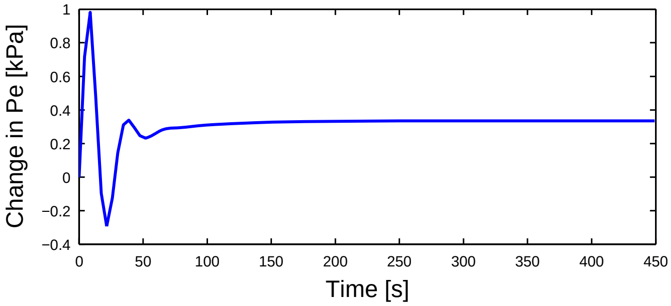


Figure 6.6 - Change in evaporator pressure response to a valve step due to a variation in the evaporator internal surface area.

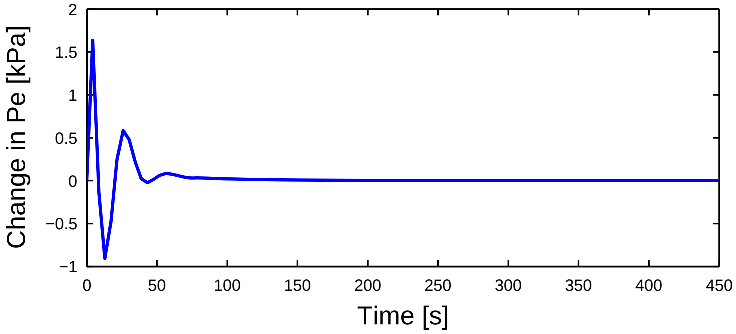


Figure 6.7 - Change in evaporator pressure response to a valve step due to a variation in the fluid flow cross sectional area in the evaporator.

Table 6.7 - Peak metric values for selected evaporator parameters

	Cp,w	**Ae,i**	**Ae,o**	**Ae,cs**
Pressures	1.16	2.21	2.13	4.69
Refrigerant Outlet Temps.	0.34	0.37	0.33	0.70
Air Outlet Temps.	0.07	0.33	0.29	0.30

Table 6.8 - Final value metric values for selected evaporator parameters

	Cp,w	**Ae,i**	**Ae,o**	**Ae,cs**
Pressures	0.05	1.15	2.13	0.06
Refrigerant Outlet Temps.	0.00	0.22	0.30	0.00
Air Outlet Temps.	0.00	0.31	0.28	0.00

Due to the fact that the outputs from the vapor compression cycle have units associated with them, the trajectory sensitivity metrics were applied to classes of outputs individually to avoid scaling issues. The pressures of the heat exchangers, refrigerant outlet temperatures from the heat exchangers, and air outlet temperatures from the heat exchangers are the three classes which are included in Tables 6.7 and 6.8. The experimental system has four controllable inputs, so for a given output, a trajectory sensitivity metric value was calculated for each input. In each output class there were two outputs, one for the evaporator and one for the condenser, creating a 2×4 matrix of trajectory sensitivity metric values. An example of the matrix for trajectory sensitivity metrics is presented in Table 6.9. The infinity norm of this matrix was used to compare the model parameters. The values of the infinity norm for the evaporator wall specific heat, $C_{p,w}$ (kJ/kg-K), the evaporator internal and external surface area, $A_{e,i}$ and $A_{e,o}$ (m^2) respectively, and the evaporator cross sectional area, $A_{e,cs}$ (m^2), are presented in Tables 6.7 and 6.8.

Table 6.9 - Peak and final value trajectory sensitivity metrics for the pressure output class when the evaporator cross sectional area perturbed

Outputs		**Inputs**			
		EEV Voltage	**Compressor RPM**	**Evap. Air Mass Flow Rate**	**Cond. Air Mass Flow Rate**
Evaporator Pressure	Peak	2.15	2.12	0.38	0.04
	Final Value	0.01	0.00	0.00	0.00
Condenser Pressure	Peak	0.59	0.51	0.15	0.03
	Final Value	0.03	0.00	-0.01	-0.01

From the qualitative information obtained from the shape of the deviation in output response approximated using the trajectory sensitivity framework, along with the magnitude of the trajectory sensitivity metrics presented in Tables 6.7 and 6.8, it is simple to determine which parameters could be used to tune the model to match experimental data. Clearly the refrigerant flow cross sectional area is a parameter well suited to tuning the transient response of the overall system. The output deviation introduced by a perturbation in cross sectional area decays to zero over time, as shown in column four of Table 6.8, indicating the parameter has nominal steady state impact. Yet the transient response of the system is highly sensitive to changes in this parameter, as indicated by the comparatively large transient output deviations indicated in column four of Table 6.7. Table 6.8 also indicates that either surface area of the evaporator could be used to improve the steady state response of the system.

6.3.3.2 Artificially Induced Parameter Perturbation Analysis

To ensure the results of the trajectory sensitivity analysis were reasonable, parameter perturbations were artificially induced into the simulation model using the method specified in Section 6.3.2. Using the same set of selected parameters as in the trajectory sensitivity analysis, Table 6.10 provides a summary of the results for both positive and negative artificially induced parameter perturbations. The Max Frequency Response metric provides information on the maximum deviation in the frequency response of the unperturbed and perturbed linear system models. The Steady-State metric indicates changes in low frequency response of the linear system models. By comparing the two frequency domain metrics both transient and steady state parameters can be identified. For example, perturbations in the evaporator cross sectional area result in zero values for the Steady-State metric while there are significant entries in the Frequency Response metric, indicating this parameter only impacts the transient response of the system model.

Table 6.10 - Metric values for both positive and negative perturbations in the system model

Metric		**Parameter**			
		Cp,w	**Ae,i**	**Ae,o**	**Ae,cs**
Max Frequency Response	+	0.23	0.25	0.00	0.25
	-	0.25	2.81	0.00	0.29
Steady State	+	0.00	1.34	0.00	0.00
		0.00	12.08	0.00	0.00

6.3.3.3 Comparison of Sensitivity Results

The two methods agree that the specific heat of the evaporator wall and the cross sectional area of the evaporator fluid flow path only impact the transient response of the system. The trajectory sensitivity method and the parameter perturbation method also agree that small changes in the cross sectional area of the refrigerant flow path in the evaporator cause larger deviations in the transient response of the system than changes in the specific heat of the evaporator wall. Both sensitivity approaches also identify the internal surface area of the evaporator as the more important steady state parameter.

One should note that the artificially induced parameter perturbation method indicates that the external surface area of the evaporator has little to no impact on the dynamic response of the system. Although this parameter would have a physical impact on the steady state response of the system, as shown by the trajectory sensitivity analysis, the effect is cancelled by the means which the simulation calculates the external heat transfer coefficient. The simulation uses an energy balance on the air-side of the heat exchanger to calculate the external heat transfer coefficient, so perturbations in the external area are cancelled by a resulting change in calculated external heat transfer coefficient such that the simulation energy balance is maintained.

The benefit of using two sensitivity analyses is clear from this discrepancy. Although trajectory sensitivity functions capture the actual physical sensitivity of a vapor compression system, it fails to catch cancellation effects which can occur in simulation implementation. Also by noting the differences between the impact of positive and negative perturbations the nonlinearity of the system model can be assessed. This provides added insight into the simulation which is not available through the trajectory sensitivity analysis.

6.4 Model Tuning

One important immeasurable model parameter is the mean void fraction. The mean void fraction is the ratio of the vapor volume to total volume in the two phase region of the heat exchanger. The mean void fraction is known to be a function of the slip ratio, which is the ratio of vapor velocity to liquid velocity in a two phase flow. The slip ratio, although unknown, must fall within reasonable physical bounds. The vapor will never be moving slower than the liquid, therefore the lower bound on the slip ratio is 1, and a reasonable upper bound is given by the Zivi correlation [64].

The trajectory sensitivity function excited with a 10% change in valve position for the evaporator mean void fraction parameter is presented in Figure 6.8. The deviation in output response indicates that this parameter, like the cross sectional area, has a significant impact on the transient response of the system.

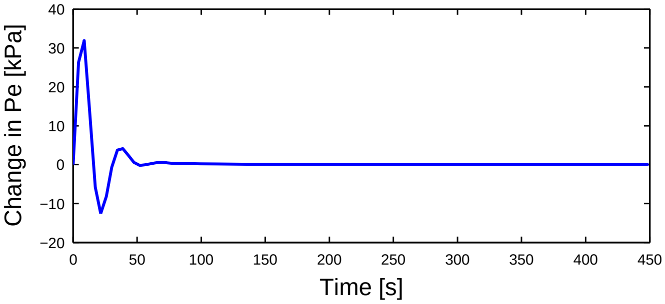


Figure 6.8 - Change in evaporator pressure response to a valve step due to a variation in the evaporator mean void fraction.

A simulation was run using values for physical parameters taken directly from manufacturer data on the experimental system described in Chapter 5. The initial operating conditions were measured using the sensors on the experimental test stand. The physical heat exchanger parameters used in the initial simulation were:

$M_e = 1.68$ kg	$M_c = 4.66$ kg
$C_{p,e} = 0.652$ kJ/kg-K	$C_{p,c} = 0.467$ kJ/kg-K
$A_{o,e} = 3.56$ m^2	$A_{o,c} = 2.79$ m^2
$D_{h,e} = 0.00733$ m	$D_{h,c} = 0.00810$ m
$L_{tot,e} = 11.87$ m	$L_{tot,c} = 10.69$ m
$A_{i,e} = 0.273$ m^2	$A_{i,c} = 0.275$ m^2
$A_{cs,e} = 4.22e\text{-}5$ m^2	$A_{cs,c} = 5.16e\text{-}5$ m^2
$Slip_e = 2$	$V_{rec} = 0.00287$ m^3
$Slip_c = 1.5$	

To simplify the data presentation, the improvement in three output variables will be presented. Simulation results and data will be compared for the evaporator pressure, condenser pressure, and condenser refrigerant outlet temperature. Figures 6.9 and 6.10 detail the inputs used to generate the transient system response, and the outputs are presented in Figures 6.11 through 6.13.

The output responses indicate that the initial model reacts significantly faster than the physical system. The transient model response can be significantly improved by simply tuning the slip ratio to more accurately reflect the system dynamics. In order to further improve the model performance, the cross sectional area can be increased to account for the extra volume within the pipes interconnecting the system components. Pressures build throughout this additional volume, slowing the response of the physical system. The cross sectional area was increased to reflect the volume in the pipes of the physical system. Simply tuning these two parameters significantly improved the transient response of the model. The slip and cross sectional area parameters used for the tuned simulation were:

$A_{cs,e} = 2.11e\text{-}4$ m^2	$A_{cs,c} = 2.57e\text{-}4$ m^2
$Slip_e = 4.27$	$Slip_c = 2.87$

Table 6.11 presents the root mean square error of the initial and tuned simulation output.

Table 6.11 - RMS error for the initial and tuned simulation output

Simulation	**Output**		
	Pe	Pc	Tc,ro
Initial	16.30	18.02	2.41
Tuned	8.91	13.75	1.32

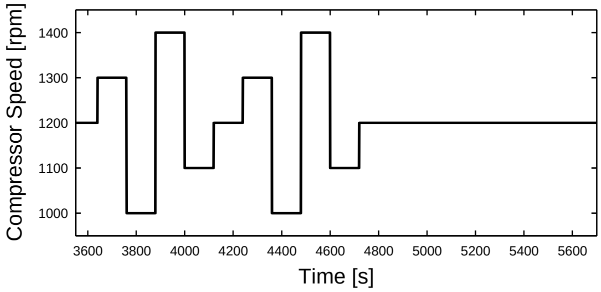


Figure 6.9 - Compressor speed input for the experimental trial.

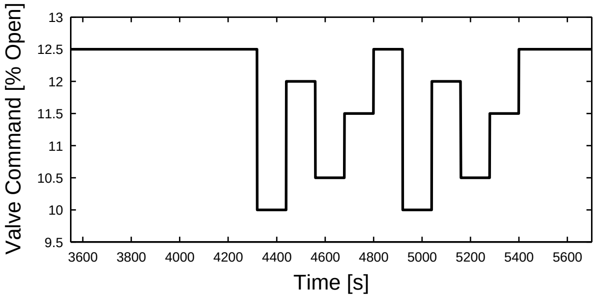


Figure 6.10 - Valve opening input for the experimental trial.

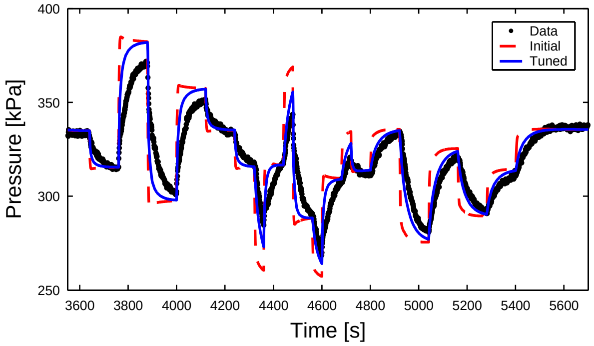


Figure 6.11 - Experimental and simulation output for the evaporator pressure.

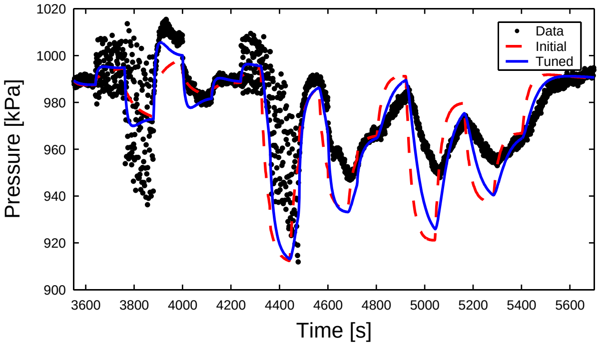


Figure 6.12 - Experimental and simulation output for the condenser pressure.

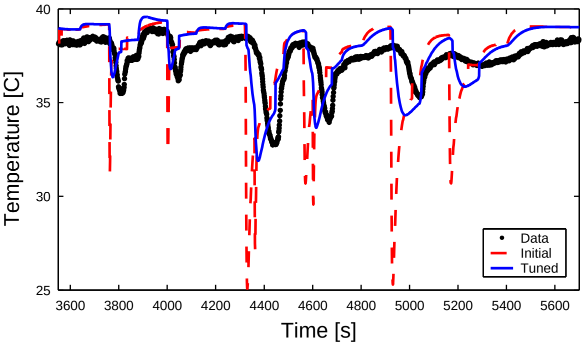


Figure 6.13 - Experimental and simulation output for the condenser refrigerant outlet temperature.

6.5 Identification of Sensitive Fault Signals

The information currently available in the literature on the dynamic impact of faults on vapor compression systems is quite limited. The ideas of parameter sensitivity are directly applicable to dynamic model-based FDD. Typically it is known how physical parameters are affected by the propagation of a particular fault in a vapor compression system, therefore a model sensitivity analysis that explores perturbations in the parameters can be used to identify the outputs well suited to FDD residual design.

6.5.1 Trajectory Sensitivity Scaling for FDD Sensitivity

Since the important signals in a vapor compression system have a variety of units, it is important to restructure the trajectory sensitivity framework to include a scaling factor associated with the outputs. The outputs were scaled to provide the most useful information from an FDD standpoint. Ideally, the analysis should provide the FDD designer with the system outputs that will have the strongest signal to noise ratio. This information can be extracted by scaling the outputs of the trajectory sensitivity analysis by the standard deviation in the measured signal. For example, on our system the evaporator pressure sensor has a standard deviation of $\sigma_{Pe} = 2.756$ kPa according to manufacturer data. The value was verified by running the system at a steady-state operating condition and using an unbiased estimator, resulting in a measured standard deviation of 2.651 kPa. The same verification procedure was used for the condenser pressure sensor, and a value of $\sigma_{Pc} = 8.268$ kPa was used in the scaling matrix. There was no manufacturer data on the uncertainty of the thermocouple measurements. Therefore, an unbiased estimator was used, and a value of $\sigma_T = 0.12°$ C was found to represent the uncertainty in the sensors. The scaling will modify trajectory sensitivity analysis of Section 6.3 to provide the number of standard deviations the faulty output will vary from the normal system output as the result of a 10% change in a particular system input, as shown in Eqs. 6.39 and 6.40.

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \dfrac{\partial \mathbf{A}}{\partial \beta_i} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \dfrac{\partial \mathbf{B}}{\partial \beta_i} \end{bmatrix} \mathbf{W_u u} \qquad \boldsymbol{\lambda} = \frac{\partial \mathbf{x}}{\partial \beta_i} \tag{6.39}$$

$$\begin{bmatrix} \mathbf{y} \\ \dfrac{\partial \mathbf{y}}{\partial \beta_i} \end{bmatrix} = \begin{bmatrix} \mathbf{W_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{W_y} \end{bmatrix} \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \dfrac{\partial \mathbf{C}}{\partial \beta_i} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \mathbf{W_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{W_y} \end{bmatrix} \begin{bmatrix} \mathbf{D} \\ \dfrac{\partial \mathbf{D}}{\partial \beta_i} \end{bmatrix} \mathbf{W_u u} \tag{6.40}$$

$\mathbf{W_u}$ is the weighting matrix given in Eq. 6.41, and $\mathbf{W_y}$ is the matrix given in Eq. 6.42.

$$\mathbf{W}_u = 0.1 \ diag\left\{ u_v,\ T_{ae,i},\ \dot{m}_{a,e},\ \omega_k,\ T_{ac,i},\ \dot{m}_{a,c} \right\} \tag{6.41}$$

$$\mathbf{W_y} = diag\left\{ \frac{1}{\sigma_{P_e}}, \frac{1}{\sigma_T}, \frac{1}{\sigma_T}, \frac{1}{\sigma_{P_c}}, \frac{1}{\sigma_T} \right\} \tag{6.42}$$

6.5.2 Condenser and Evaporator Air-Side Faults

The case of external fouling in the evaporator and condenser can be used as an example of how to use parameter sensitivity information to aid with FDD. The build up of a thermally insulating material on the external surface of a heat exchanger, such as frost on an evaporator or dirt on a condenser, will increase the thermal resistance between the refrigerant and the external fluid. As the layer of material increases in thickness it will impede the flow of the external fluid. The trajectory sensitivity framework can be used to explore the sensitivity of system outputs to changes in a physical parameter that will impact the overall system in a manner similar to the actual fault. In the case of external fouling, a primary effect is a reduction in heat transfer to the external fluid. Therefore, the sensitivity of the vapor compression system model to perturbations in the external heat transfer coefficients for both heat exchangers can be explored, and should provide output sensitivity information relevant to fouling.

Figure 6.14 presents the deviation in output response to a 10% step in the valve input command if the evaporator external heat transfer coefficient deviates by 10% from its nominal value, as predicted using the trajectory sensitivity framework. It is clear from the figure that particular system outputs, specifically the evaporator air outlet temperature, are more responsive to the formation of fouling on the external surface of the evaporator. Figure 6.15 presents the deviation in output response predicted using the trajectory sensitivity framework for a 10% step in the valve input command if the condenser external heat transfer coefficient is reduced by 10% from its nominal value. In this case the most responsive output from a signal to noise perspective is the condenser pressure. Figures 6.14 and 6.15 indicate the outputs that should be included in a structured or directional residual in an FDD algorithm. For example, if the FDD designer only wished to detect these two fault conditions, the algorithm would require only the measurement of evaporator air outlet temperature and condenser pressure. Although condenser pressure responds to both faults, the evaporator air outlet temperature will only respond to a fault in the evaporator heat transfer coefficient. Hence, the increase in condenser pressure could be used to detect a fault, and the evaporator air outlet temperature signal would indicate the location of the fault.

The reader should also note the difference in scale between the two simulated deviation responses in Figures 6.14 and 6.15. Clearly, the system outputs explored in this study are more sensitive to changes in the evaporator external heat transfer coefficient. This would imply that a frosted evaporator would be easier to detect than a fouled condenser.

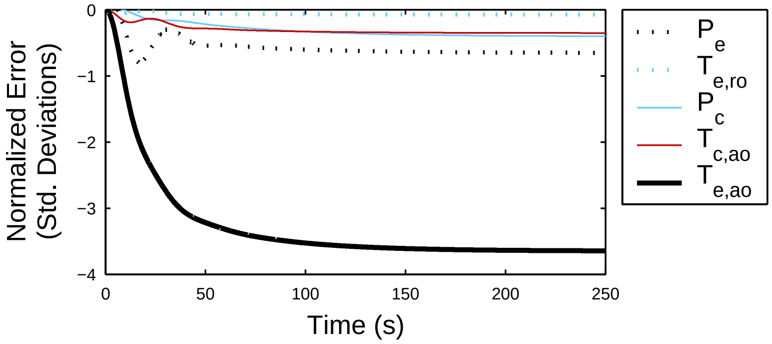


Figure 6.14 - Change in system response to a valve step as the result of external evaporator fouling.

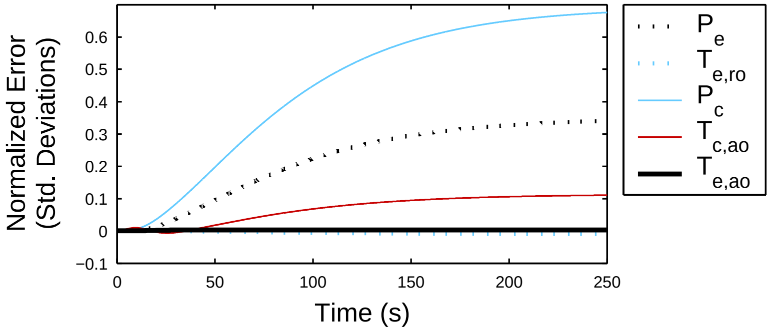


Figure 6.15 - Change in system response to a valve step as the result of external condenser fouling.

6.5.3 Experimental Validation of Air Flow Fault Sensitivity

This section details an experimental exploration of output sensitivity of the evaporator and condenser to air flow faults. A fault in the air mass flow rate over a heat exchanger could be caused by a variety of physical conditions in the system, such as a blockage of the air intake (fouling/frosting), or in a more severe case, a fault in the fan motor. The dynamic and steady state impact of this fault would propagate throughout the system. Therefore, in setting up an FDD algorithm the designer would need to know which system outputs are most sensitive to the fault.

One of the key effects of a reduction in air mass flow rate over a heat exchanger is a decrease in the heat transfer coefficient between the heat exchanger wall and the air. The average air temperature passing over the evaporator coil also decreases, further hindering the heat transfer from the refrigerant to the air. Two experimental approaches were used to explore which system outputs are sensitive to the air flow fault. The first method captures the changes in system state that are introduced as the result of the reduction in air flow. The second method explores changes in the input-output response of the system when a change in valve position is used to excite the system.

6.5.3.1 Air Flow Reduction

To explore the impact of a reduction in evaporator air mass flow rate, the system was set to run at a steady-state operating condition. With all other inputs held constant, the evaporator air mass flow rate was decreased by 10%. The resulting deviation in output response from the steady-state set point is presented in Figure 6.16. In this case, the air mass flow rate decrease occurred at t = 200 seconds, and the outputs were scaled by their sensors standard deviation as in Subsection 6.5.1. This provides the FDD designer with the strongest signals to identify an air mass flow fault in the evaporator. It is clear from the figure that the evaporator air outlet temperature and the evaporator refrigerant outlet temperature respond significantly to an air mass flow fault.

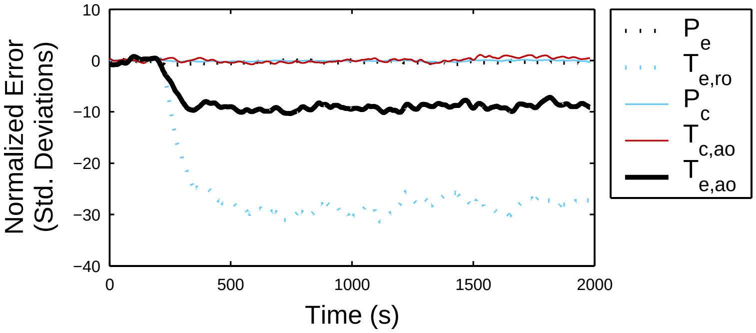


Figure 6.16 - Output deviation from a 10% decrease in evaporator air mass flow rate.

In a similar manner to the evaporator air mass flow fault, a 10% fault in the condenser air mass flow rate can be introduced. The level of fault was controlled by reducing the power supplied to the fan, which was correlated to a reduction in the mass flow of air over the heat exchanger. Thus, a specific level of air mass flow fault could be gradually introduced into the condenser. Figure 6.17 presents the output deviation resulting from a 10% decrease in condenser air mass flow rate. In general, a vapor compression system is less responsive to a condenser air mass flow fault, therefore the signals were low pass filtered to enhance the visual identification of signal sensitivity. From a visual inspection of Figure 6.17, the condenser pressure appears to exhibit a consistent upward drift as a result from the decrease in air mass flow rate, and a zoomed view of the pressure responses is presented in Figure 6.18. The evaporator air outlet temperature and refrigerant outlet temperature seem to respond in a more oscillatory manner, though this may be partially due to variations in ambient conditions. It should be noted that, as was predicted in Subsection 6.5.2, the size of the deviations as the result of a reduction in condenser air mass flow rate are smaller than those from an equivalent air mass flow rate fault in the evaporator. This would again imply that fouling faults in the evaporator will be easier to detect than a fault of comparable magnitude in the condenser.

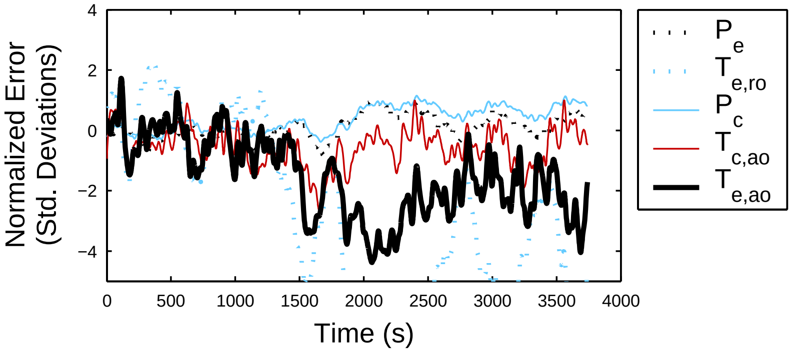


Figure 6.17 - Output deviation from a 10% decrease in condenser air mass flow rate.

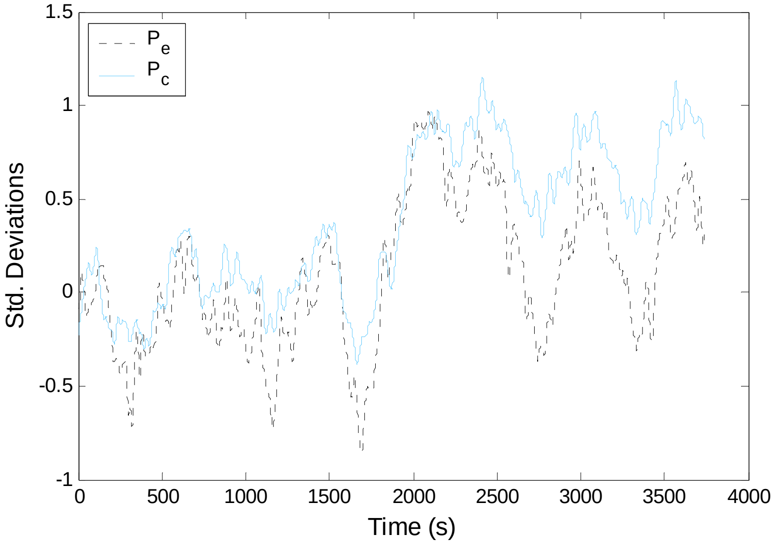


Figure 6.18 - Output deviation from a 10% decrease in condenser air mass flow rate (zoomed view of pressures)

6.5.3.2 System Response to Valve Excitation

From Subsection 6.5.3.1 it was determined that there are a number of outputs that will respond significantly to a reduction in air mass flow rate. The next question is whether similar changes will appear in the input-output response of the system when it is subjected to some form of excitation. To capture changes in input-output information the steady state changes in system operating condition would need to be removed, allowing the deviation in the dynamic response to be observed. Initially the system was run without any air flow fault condition with a superheat of 10° C. Once the system reached a steady state, the valve was closed by 1.6% (approximately 10% of its nominal opening). Then 10% air flow faults were individually introduced into each heat exchanger and again the system was brought to a 10° C superheat condition. The valve was then closed by 1.6% again, and the new output response was recorded. The excitation signals for the no fault and fault conditions are presented in Figure 6.19. To compare the dynamic responses for the different fault and no fault scenarios, the steady-state output prior to the valve step was subtracted from the outputs so that only the deviation from steady state would remain. Figures 6.20-6.22 present the output deviations for the evaporator and condenser air flow fault conditions.

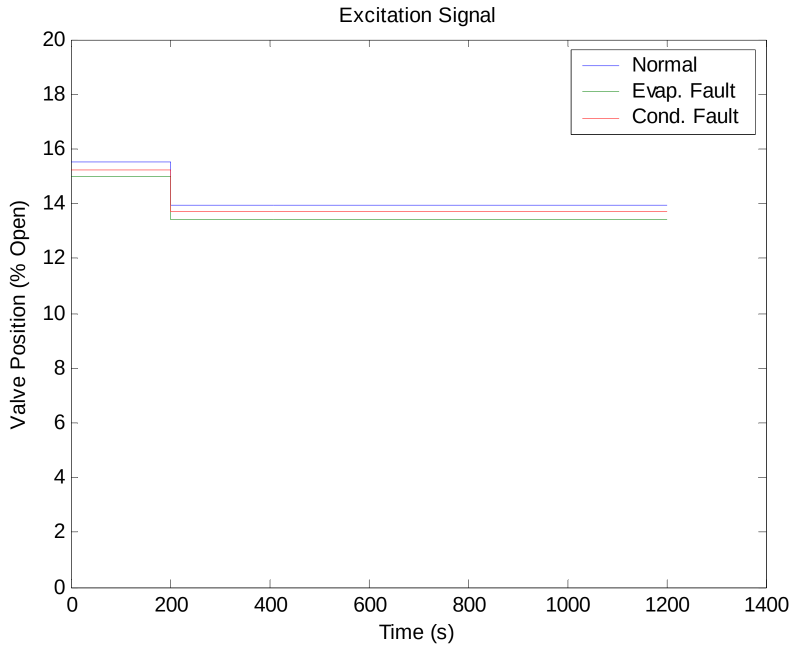


Figure 6.19 - Valve excitation signals used for the normal, evaporator air flow fault, and condenser air flow fault experimental trials

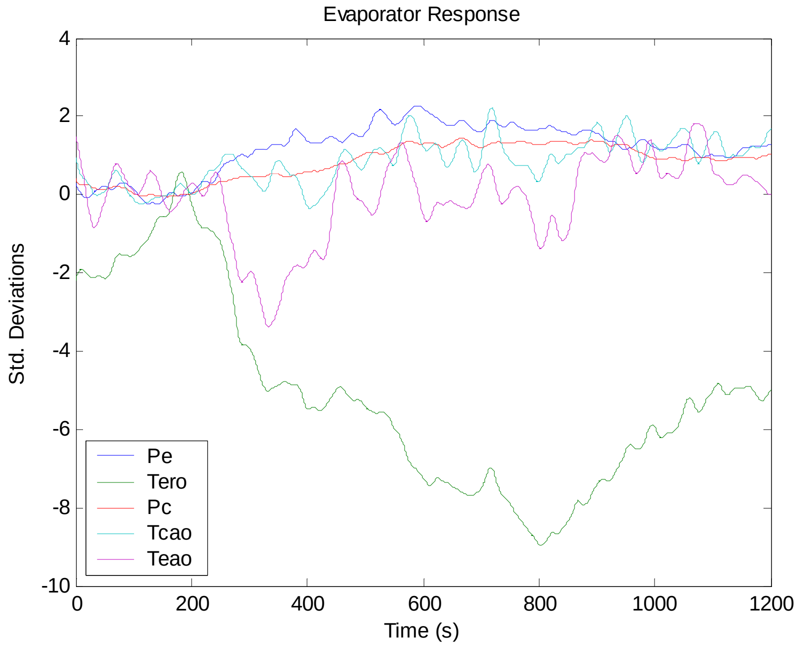


Figure 6.20 - Output deviation in the input-output response of the system to valve excitation when a 10% decrease in air flow has occurred in the evaporator

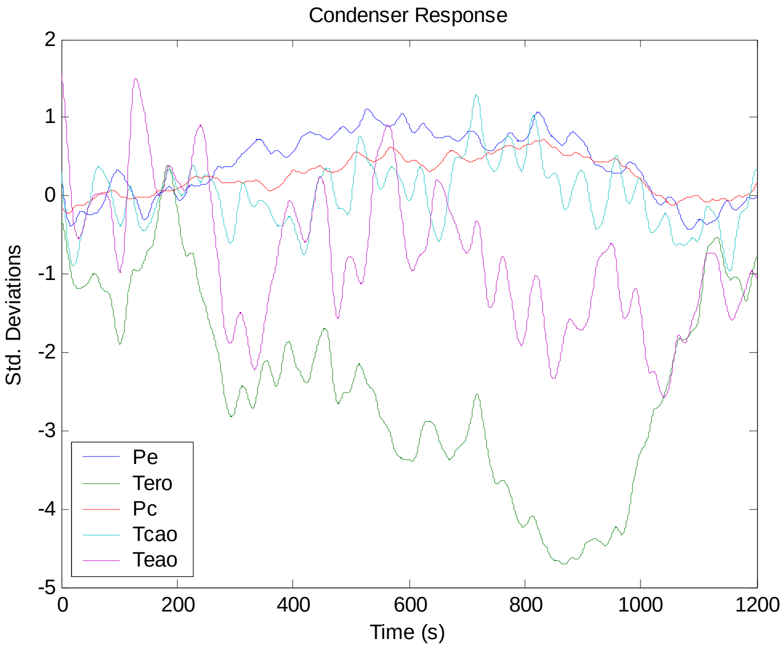


Figure 6.21 - Output deviation in the input-output response of the system to valve excitation when a 10% decrease in air flow has occurred in the condenser

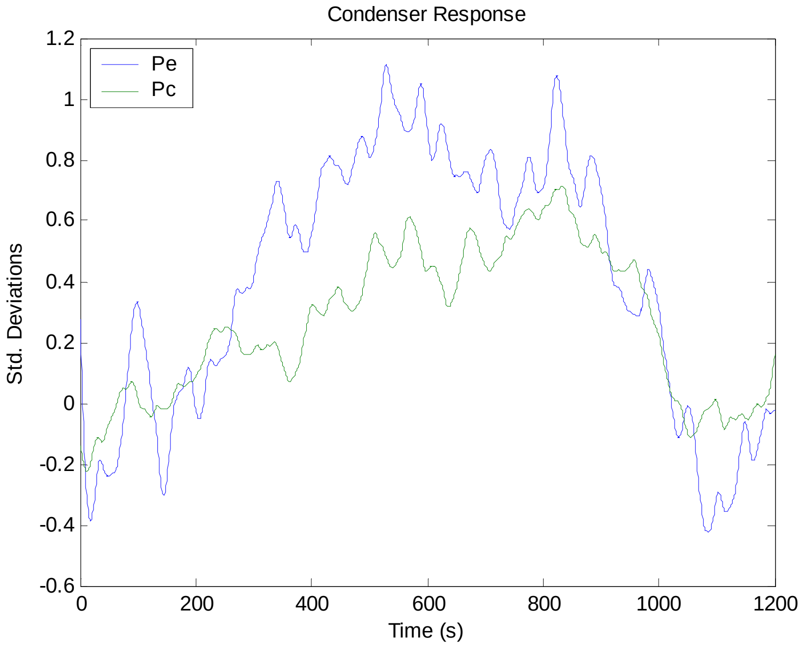


Figure 6.22 - Output deviation in the input-output response of the system to valve excitation when a 10% decrease in air flow has occurred in the condenser (zoomed pressure view)

Figures 6.20-6.22 indicate that there are limited differences in the steady state gain of the system response to a valve step for many of the system outputs. The lack of gain change is apparent in Figures 6.20-6.22 by noting that the deviation between the nominal and faulty system response returns to 0 as time progresses. It appears that for the evaporator air flow fault condition the refrigerant outlet temperature, the evaporator pressure, and the condenser pressure exhibit some lasting change in steady-state gain that may be detectable and could potentially be used in an FDD algorithm. To pick up a condenser air flow fault with the gain of the system model will likely be of limited use, yet there are still some significant deviations in evaporator pressure, condenser pressure, and evaporator refrigerant outlet temperature that occur during the transient portion of the response which may be useful for indicating condenser air flow faults. Again it appears that evaporator air flow faults may cause more significant deviations in system response and therefore may be easier to detect with an FDD algorithm.

6.5.4 FDD Sensitivity Conclusions

External heat exchanger fouling will impact a vapor compression system in two distinct ways. The build up of a layer of thermally insulating material on the surface of the heat exchanger will increase the thermal resistance between the air and the refrigerant. As the layer increases in size on a conventional tube and fin heat exchanger, the pressure drop across the heat exchanger will increase from the reduction in free flow area through the fins. This will decrease the total mass flow rate of air passing over the heat exchanger. The sensitivity analysis suggested that the condenser pressure and evaporator air outlet temperature would be sufficient to detect the build up of a thermally insulating material on the external surface of the two heat exchangers. From an experimental investigation of the impact of a reduction in air mass flow rate, it was seen that these two outputs do in fact respond significantly to the air flow fault. The sensitivity analysis did not capture all of the signals that would respond, as seen by comparing the refrigerant outlet temperature response in Figures 6.14, 6.16, and 6.20. This is likely due to the fact that the sensitivity analysis introduces a change in a single physical parameter to approximate the combined parametric and input effects that result from an actual fault.

Although the sensitivity analysis may underestimate the effects of certain input-output variables, it is important to note that it was successful in identifying the following characteristics:

Condenser fouling is harder to detect than evaporator fouling.

The evaporator air outlet temperature and condenser pressure can be used to detect and distinguish fouling faults in the evaporator and condenser.

The FDD designer could use the sensitivity information to create a structured or directional residual, and with the appropriate residual structure the algorithm will be able to distinguish between these two faults in the vapor compression system.

Chapter 7. Control Design

The benefits of variable speed capacity control have been known for several decades, yet implementation of variable speed technology has been limited by several technological constraints. Qureshi and Tassou [62] identified the four major constraints as: insufficient development and integration of compressors and variable speed drives, relatively high cost of variable speed drives, insufficient information from manufacturers of variable speed systems, and poor reliability due to unsophisticated and inadequately developed control systems. Recent technological advances in DC compressor technology, combined with the improved performance and reduced cost of the variable speed drives have significantly eased the component level constraints in variable speed systems. The advancements in compressor and drive technology have enabled variable speed capacity control systems to become economically viable in today's market. The creation of simple, effective control strategies to manage the coupled dynamics of these systems will result in air conditioning and refrigeration systems capable of high performance rapid cooling and energy efficient operation.

7.1 Introduction and Background

As discussed in Section 1.3, a basic vapor compression system is designed to operate with a small level of superheat at the compressor inlet and generally contains some method for varying the capacity of the system. The controllable inputs vary significantly among vapor compression systems, and are frequently dependent on application and system quality. For example, in automotive vapor compression systems the compressor is often powered by the engine shaft and the condenser air mass flow rate is dependent on the vehicle speed. This system configuration removes the ability to control the compressor speed, and therefore the engine speed must be treated as a disturbance. The variation in air mass flow rate over the condenser is also a disturbance to the automotive vapor compression system. The capacity of an automotive vapor compression system powered by the engine shaft is regulated by clutching on and off compressor power. With the advent of hybrid vehicles, there is substantial opportunity to include electrically driven compressors in automotive vapor compression systems. Including control authority over the compressor speed would improve the efficiency of the air conditioning system, resulting in improved fuel economy.

Traditionally, the standard industry approach is to regulate the superheat of the system with a thermostatic expansion valve and use an on/off cycle to control the system capacity. The remainder of this section will compare the standard system to a system with a variable speed compressor and an electronic expansion valve. A review of control approaches for variable speed systems is presented, and recent developments in multivariable control approaches are addressed.

7.1.1 Superheat Regulation with Expansion Devices

A thermostatic expansion valve (TEV) uses mechanical feedback to regulate the superheat of a vapor compression system using what is essentially a proportional controller with a first-order sensor dynamic [51]. A bulb filled with refrigerant of the same type as the system is placed at the outlet of the evaporator and connected to one side of a diaphragm in the TEV. As the temperature of the pipe at the evaporator outlet varies, the pressure of the two-phase refrigerant within the bulb changes. This pressure change will open or close the valve appropriately to alter the mass flow rate of refrigerant through the system, maintaining the desired superheat. A spring with user specified initial compression can be used to fine tune the regulation point of the system. A diagram of a basic TEV is presented in Figure 7.1.

An alternative to the TEV is the electronic expansion valve (EEV), which typically uses a stepper motor to control the valve opening. This provides the system designer with control authority over the opening in the valve, allowing for advanced control strategies that coordinate the valve opening with other actuators and respond with feed-forward control actions to measured disturbances [46]. The EEV requires a stepper motor control board and a thermocouple to operate. A diagram of the EEV connection to the evaporator is presented in Figure 7.2.

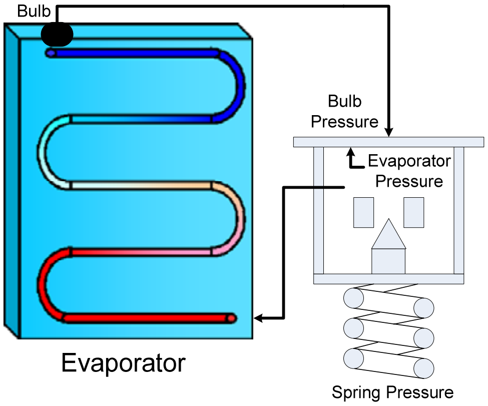


Figure 7.1 - Diagram of a Thermostatic Expansion Valve (TEV)

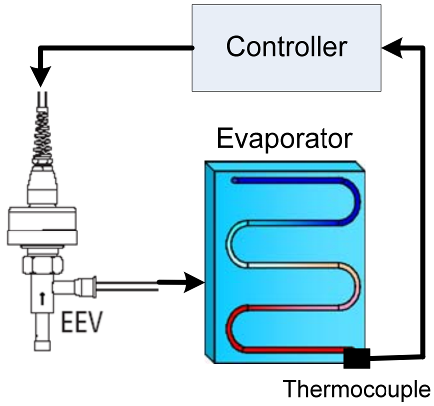


Figure 7.2 - Diagram of an Electronic Expansion Valve (EEV)

7.1.2 Capacity Control

The ability to match the capacity of a vapor compression system to the heat load with continuous system operation has significant energy benefits [62]. In the case of the experimental system detailed in Chapter 5, the benefits of continuous versus cycled operation are summarized in Figure 7.3. The system is sized to provide a cooling capacity of approximately 1.6 kW. The on/off control results were obtained by using a fixed cycle length (5, 10, 15, or 30 minutes) and varying the duty cycle (20, 40, 60, or 80%) with a fixed compressor speed of 2000 RPM during the on-cycle. As an example, an on/off control system operating with a 60% duty cycle has an average compressor power of 0.33 kW. The average cooling capacity for the 60% duty cycle tests is 0.86 kW. At that same average compressor power, the vapor compression system running continuously (at a lower speed) has an average capacity of 1.30 kW, a 50% increase over the cycled system performance. If only 0.86 kW are required to reject the current heat load, the system could operate continuously at an average compressor power of 0.2 kW, a 40% reduction in compressor power.

The necessary cooling capacity can only be estimated prior to system installation. Therefore, the system controller must have sufficient control authority to drive the system to a new operating condition that can reject the heat load. The most basic approach for varying the cooling capacity of the system is to use the valve to regulate the superheat of the system and the compressor to control the cooling capacity. In this case the coupled dynamics of the vapor compression system are neglected, and each control loop acts without regard to the other system actuator. Measuring the cooling capacity of the system is frequently infeasible since it would require multiple sensors; therefore the standard approach in the literature is to use the evaporation temperature (or pressure) as the feedback signal for the compressor control loop. If the evaporation temperature (or pressure) is reduced while the superheat at the exit of the evaporator is held constant then the walls of the evaporator will become colder, creating a stronger driving potential for heat transfer to the external fluid. A diagram of this basic controller configuration is presented in Figure 7.4.

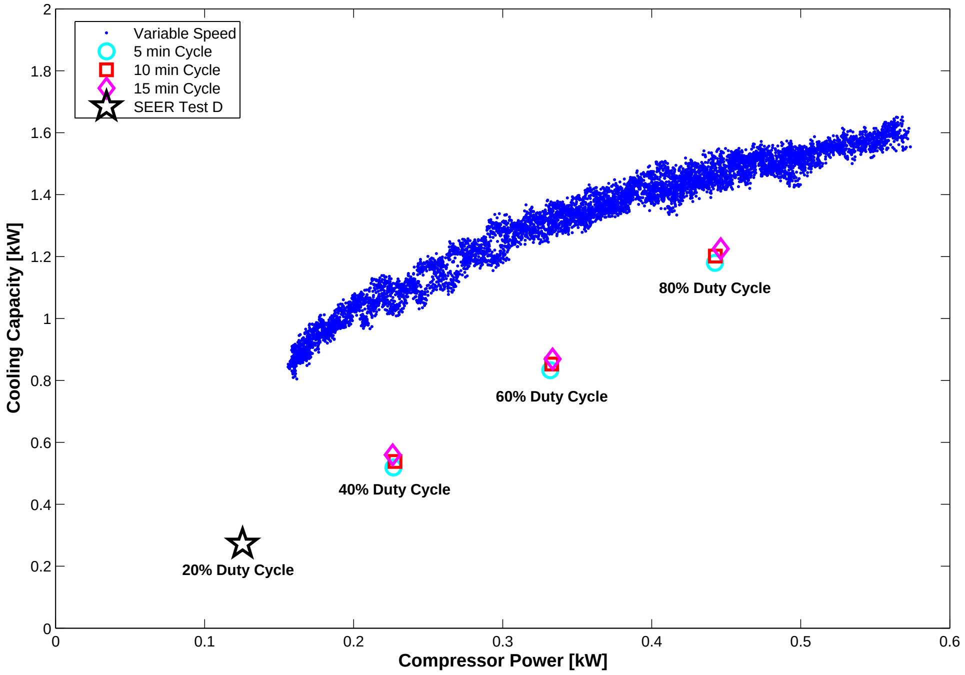


Figure 7.3 - Continuous versus cycled system cooling capacity and power consumption

Since movements in the valve position and compressor speed both significantly impact the evaporator pressure and superheat, the control scheme of Figure 7.4 has difficulty regulating to a specific set point. Even a carefully tuned controller on the experimental system from Chapter 5 has difficulty maintaining a set evaporator pressure, as demonstrated in Figure 7.5.

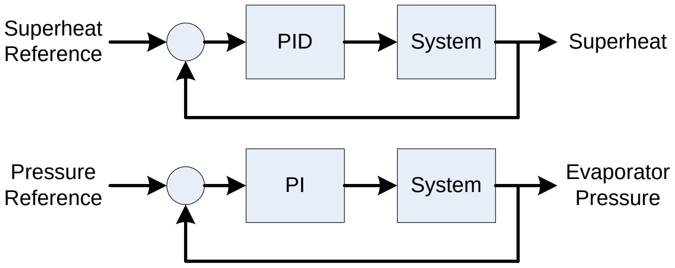


Figure 7.4 - Basic variable speed capacity control configuration

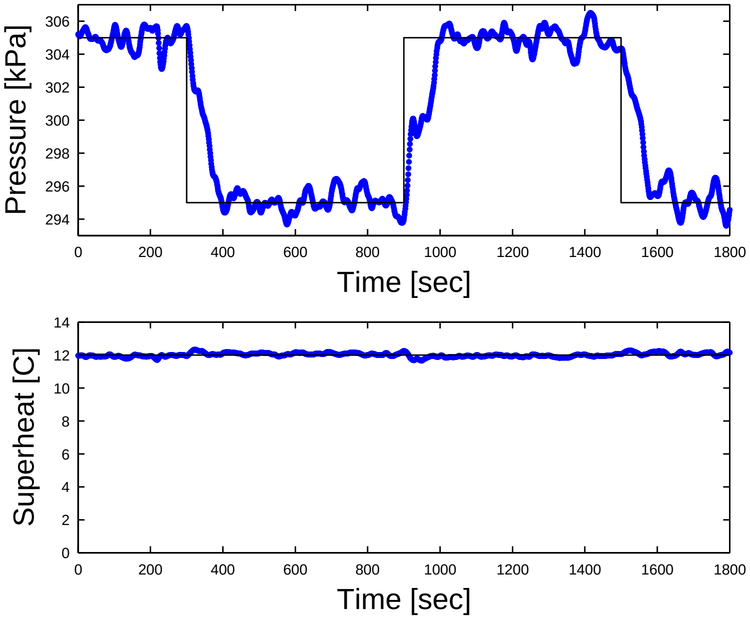


Figure 7.5 - Superheat and evaporator pressure regulation with SISO control

Multivariable control approaches have been suggested as an effective means to improve the regulation of variable speed vapor compression systems [35,63]. The first model-based MIMO approach to regulate superheat and evaporator pressure was proposed by He et. al. [36]. Since this initial effort, He and Asada have gone on to employ LQG [36], gain scheduled LTR [53], gain scheduled LQG [35], and feedback linearization [91] control approaches to regulate superheat and evaporator pressure. The papers by He and Asada generally contain limited experimental validation of the various control strategies. Recently Rasmussen has applied a gain scheduled control approach to vapor compression systems and experimentally demonstrated the controllers performance at off-design conditions [63].

In the remainder of this Chapter three different control approaches are presented. The first approach is a simple decentralized framework, where individual control loops are used to regulate superheat and a system pressure. The second approach uses a static decoupling technique to reduce the fighting between the control loops, providing improved control of system capacity. The final approach is a linear quadratic gaussian (LQG) multivariable control approach that is similar to controllers developed by He and Asada [35]. LQG is a multivariable control approach that can be used as a reference to analyze the performance of the simpler decentralized and decoupled control designs. The LQG control discussion also provides details on the design of an LQG controller that do not require a search over system weights to achieve reasonable controller performance. All of the control approaches are applied to two feedback frameworks that vary in the pressure signal that is used in the feedback loop. The reasoning behind the selection of the two pressure signals that are used in the feedback loop is provided in Section 7.2.

7.2 Coupling in Vapor Compression Systems

The strong coupling between superheat and evaporator pressure is the main limitation that prevents SISO control strategies from effectively regulating both outputs in a vapor compression system. In reality, superheat is a function of evaporator pressure since the saturation temperature at the evaporator outlet is directly related to the evaporator pressure. The tuning of optimal and robust control strategies frequently requires significant technical expertise that may not exist within many air conditioning and refrigeration companies, thus the development of simple control approaches that can be designed in simulation and retain the performance of complicated multivariable control approaches would greatly benefit the industry. In order to generate an effective control approach, a fundamental understanding of the open loop plant dynamics is required. If alternative signals for compressor control with reduced coupling are available within the vapor compression system a simple and effective control architecture could be designed to regulate cooling capacity and superheat.

7.2.1 Open Loop Plant Dynamics

The basic dynamic response of a vapor compression system can be identified using a simple time domain system identification procedure. Initially the system response to valve steps, compressor steps and pseudo-random binary combinations of compressor and valve steps around a particular operating condition was collected. Data for the response of superheat, evaporator pressure, and the pressure differential of the system is presented in Figures 7.6-7.8. The command signals were overlaid on the recorded data in order to visually highlight the individual impact each control actuation had on the system response. The most distinct characteristic of these three plots is the response of the pressure differential. While superheat and evaporator pressure respond comparably to valve and compressor steps, the pressure differential is far more responsive to compressor steps.

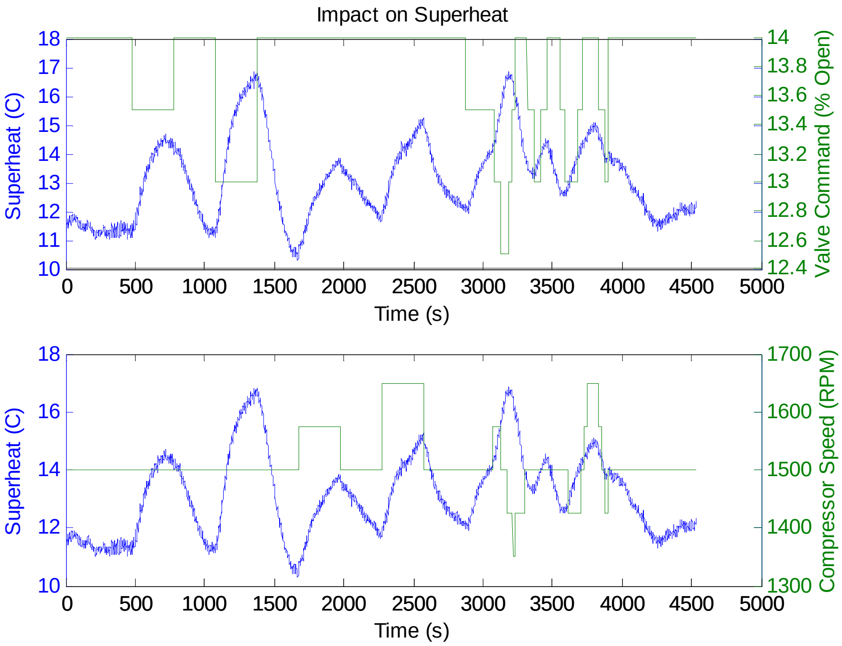


Figure 7.6 - Superheat response to valve and compressor step inputs

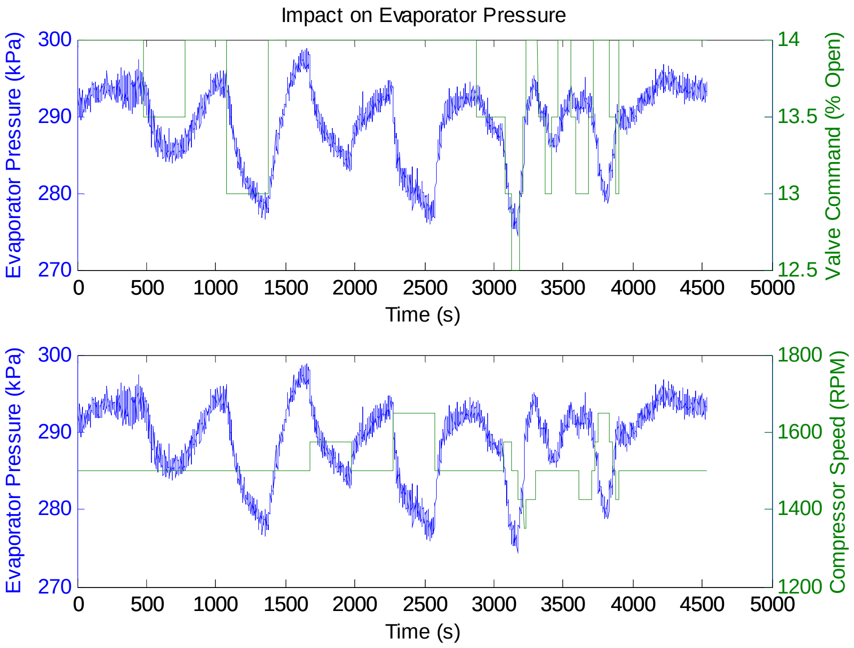


Figure 7.7 - Evaporator pressure response to valve and compressor step inputs

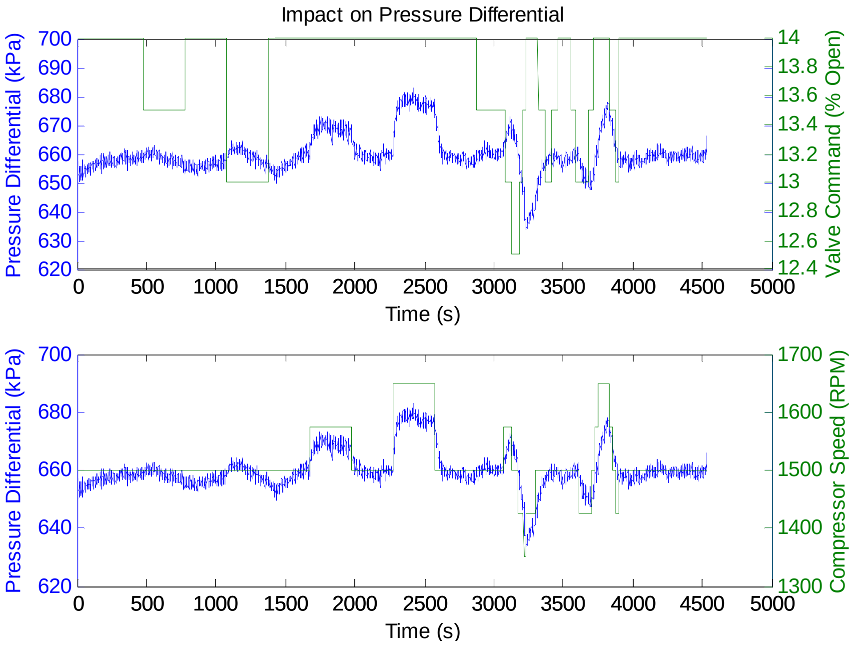


Figure 7.8 - Pressure differential response to valve and compressor step inputs

7.2.2 Identified Plant Models

By regulating the pressure differential and superheat of the system, the controller would still have the control authority to span the full capacity range. Using a standard prediction error/maximum likelihood system identification procedure [54], a second order dynamic model with superheat and either evaporator pressure or pressure differential as outputs can be identified. Figures 7.9 and 7.10 present a comparison of the model output and experimental data for the two proposed system models. It should be noted that the pressure outputs and compressor input were scaled prior to model identification in order to prevent unequal weighting of the error signals from the two outputs. A bode magnitude plot of the two identified system models is presented in Figure 7.11.

From Figure 7.11 it is clear that the two identification procedures identify similar input-output models for the superheat response. The main difference is that the valve to pressure differential transfer function has a significantly lower magnitude than that from the valve to evaporator pressure, indicating a reduction in coupling. An added benefit is that the compressor to pressure differential response has a higher bandwidth than the compressor to evaporator pressure response. This improved bandwidth can be used by the controller to achieve faster responses to changes in the pressure set point, improve set point regulation, and enable a faster recovery from off-design operating conditions.

7.2.3 RGA Analysis of Plant Models

The relative gain array (RGA) was originally introduced by Bristol [14] as a steady-state measure of closed loop interactions for decentralized (multiple SISO loop) control. For a non-singular square matrix, P, the relative gain array is defined by Eq. 7.1, where × denotes element by element multiplication (Schur product).

$$RGA(P) = \Lambda(P) = P \times \left(P^{-1}\right)^T \tag{7.1}$$

According to Skogestad and Postlethwaite [72], the RGA is a good indicator of:

- sensitivity to uncertainty in the input channels
- diagonal dominance
- the stability of decentralized control

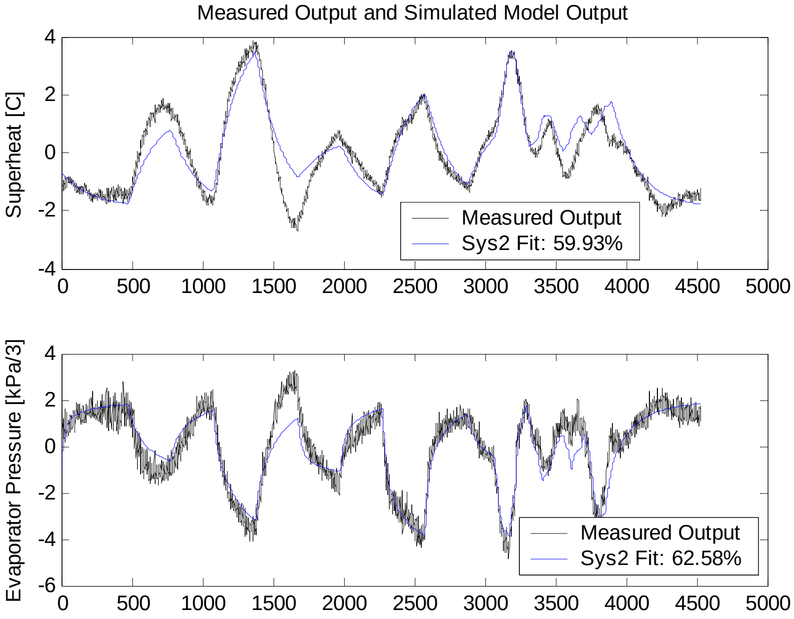


Figure 7.9 - Model fit for a 2nd order state space model with superheat and evaporator pressure as outputs

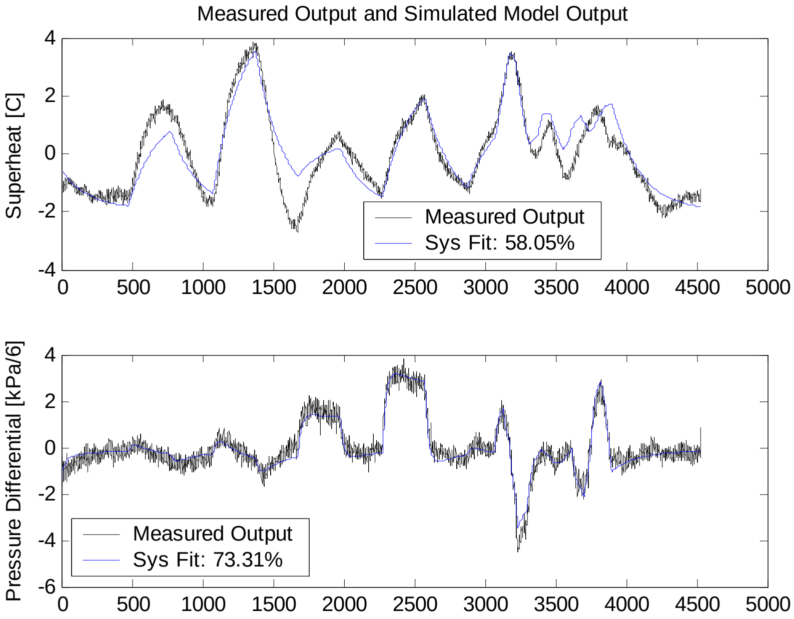


Figure 7.10 - Model fit for a 2nd order state space model with superheat and pressure differential as outputs

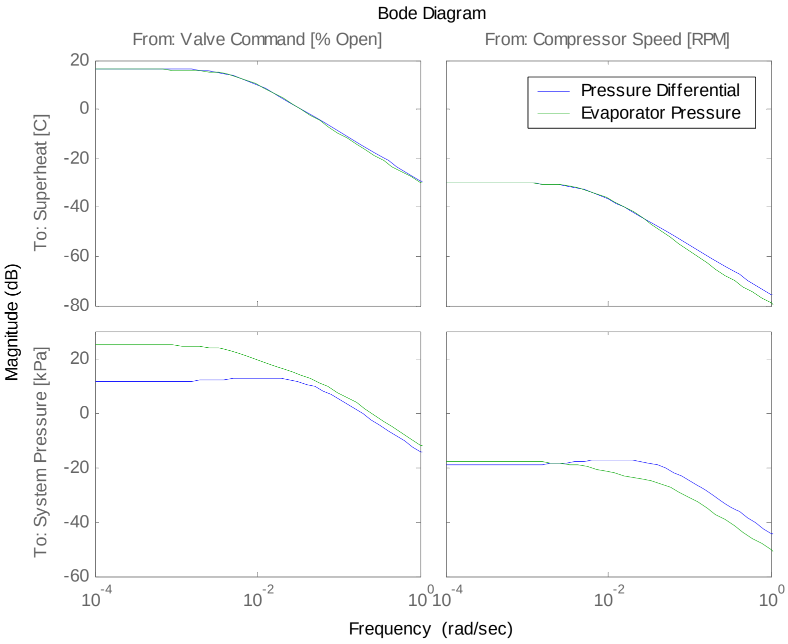


Figure 7.11 - Bode magnitude plot for the identified models with pressure differential and evaporator pressure as outputs

Uncertainty in the input channels is indicated by plants with large RGA elements around the crossover frequency, making these plants fundamentally difficult to control. A measure of the diagonal dominance of a plant, G, is obtained by calculating the RGA-number, given in Eq. 7.2.

$$RGA-number(G(\omega)) = \left\|\Lambda(|G(\omega)|) - I\right\|_{sum} \tag{7.2}$$

Taking the RGA of a diagonal or triangular plant results in an identity matrix, therefore the RGA-number of a diagonal or triangular plant is 0. If coupling between both inputs and outputs exists, the RGA will have non-zero terms in the off diagonal elements, indicating closed loop interactions that will limit decentralized control. Large RGA numbers are a clear indicator that the closed loop performance will be poor when decentralized control schemes are applied. Figures 7.12 and 7.13 present the RGA-number of the superheat/evaporator pressure and superheat/pressure differential plant models as a function of frequency. Clearly, the model with evaporator pressure as an output has a significantly higher RGA-number at all frequencies. This indicates that the superheat/evaporator pressure model has a higher degree of coupling between the controlled outputs. It is interesting to note that the RGA-number drops towards zero at a frequency of 0.004 rad/s in the superheat/pressure differential model. This due to the plant becoming triangular as the system pressure differential becomes unresponsive to changes in valve position (see Figures 7.13 and 7.14). This implies higher frequency valve movements will not significantly impact the pressure differential, resulting in nearly triangular plant that would be significantly easier to control using decentralized control approaches.

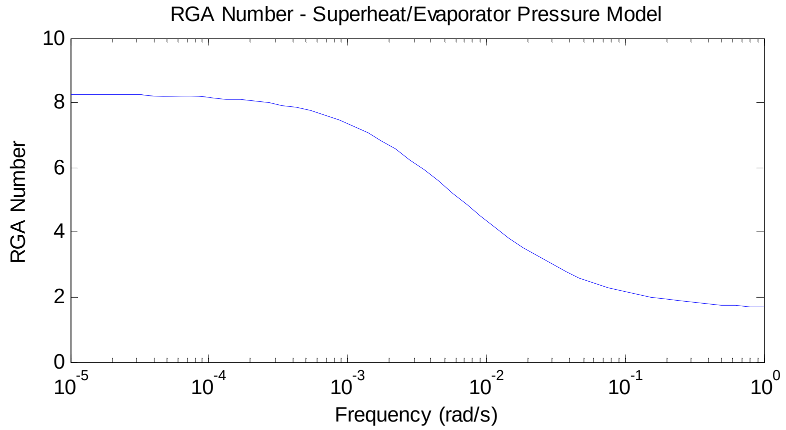


Figure 7.12 - RGA-number for the superheat/evaporator pressure model

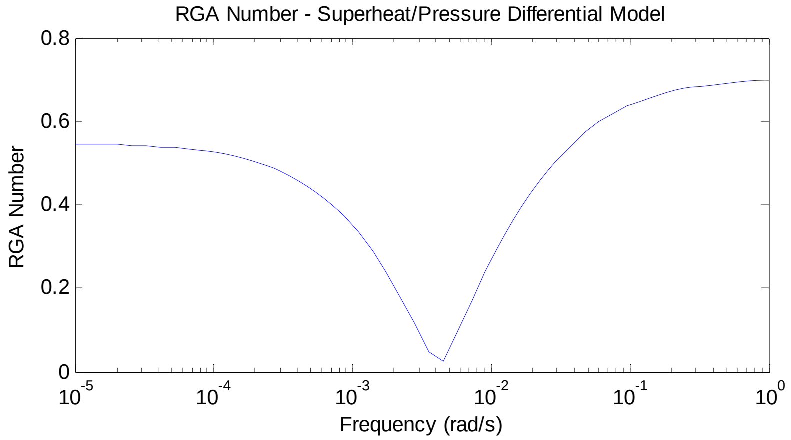


Figure 7.13 - RGA-number for the superheat/pressure differential model

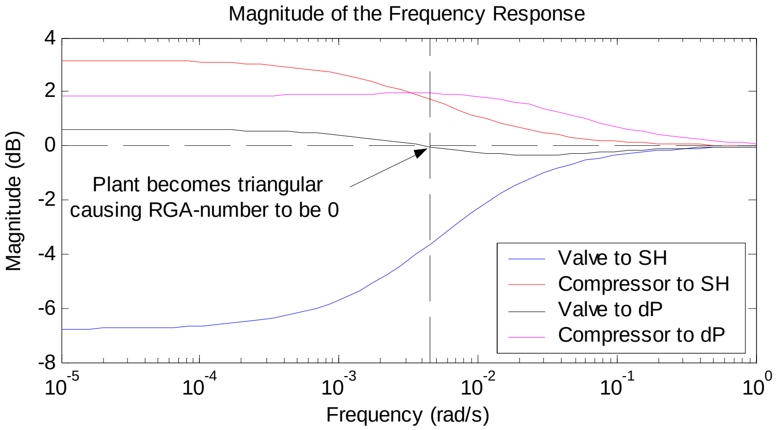


Figure 7.14 - Frequency response of the open loop input/output transfer functions for the superheat/pressure differential model

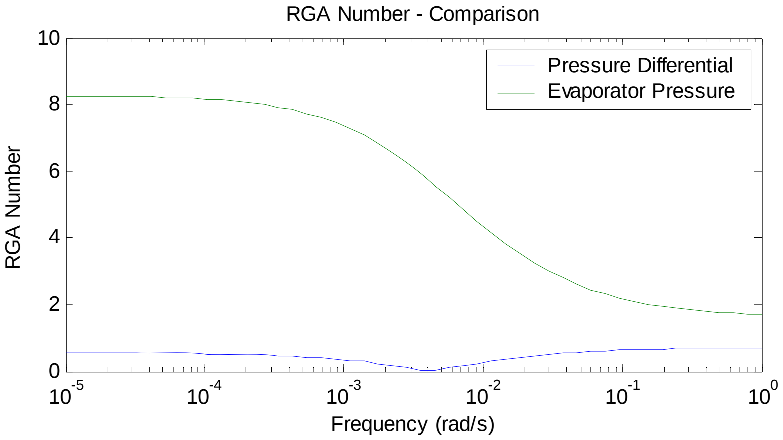


Figure 7.15 - Comparison of the RGA numbers for the superheat/pressure differential and superheat/evaporator pressure models

7.3 Decentralized PID Controller Performance

The most basic control approach is to use a decentralized scheme to manage the system outputs. In the following subsections the design of a PID controller for superheat and a PI controller for the pressure output is described. In each case the controllers were designed to have a 25 second rise time with limited overshoot. An analysis of the resulting closed loop performance is provided, and important system sensitivity functions are calculated. Experimental results of the closed loop controller performance on the lab system detailed in Chapter 5 are presented, and a discussion of the limitations of decentralized control is included.

7.3.1 Controller Design in Simulation

Using the identified models for the superheat/evaporator pressure and superheat/pressure differential output configurations, individual PID loops were tuned to obtain a closed loop system performance with a rise time of approximately 25 seconds and a settling time less than 100 seconds. The size of the reference change for each model was selected to require a comparable level of change in the compressor speed (~80 RPM). Figures 7.16 and 7.17 present the resulting output response and the required system inputs. The superheat/evaporator pressure controller used the following decentralized PID gains:

$$P_{sh} = 4 \quad I_{sh} = 0.01 \quad D_{sh} = 4 \quad P_{Pe} = 22 \quad I_{Pe} = 2.25$$

The superheat/pressure differential controller used the following decentralized PID gains:

$$P_{sh} = 2 \quad I_{sh} = 0.01 \quad D_{sh} = 4 \quad P_{dP} = 12 \quad I_{dP} = 0.85$$

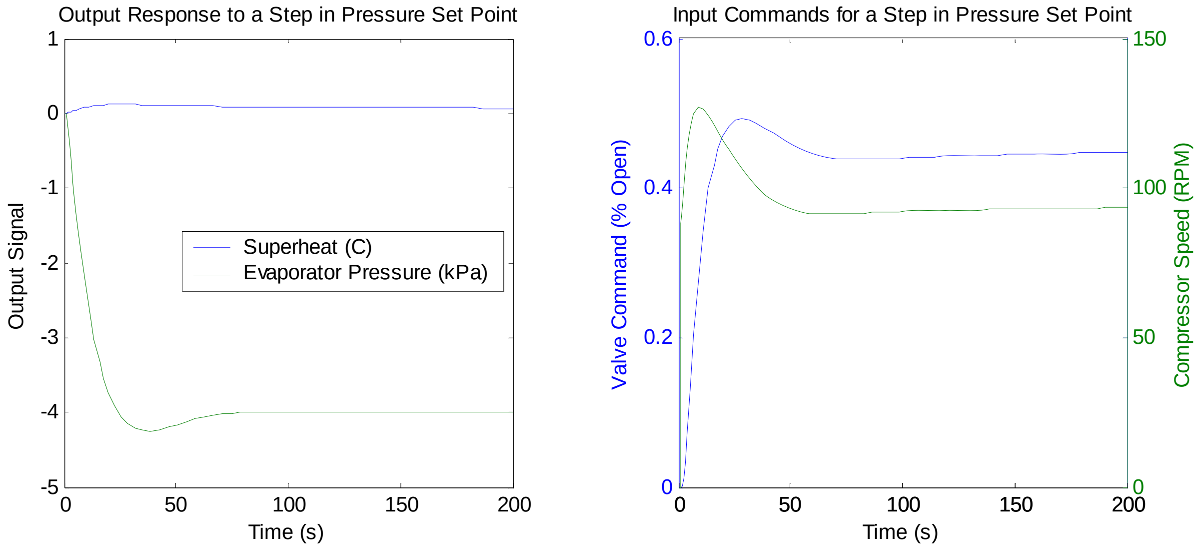


Figure 7.16 - Closed loop superheat/evaporator pressure output and input response

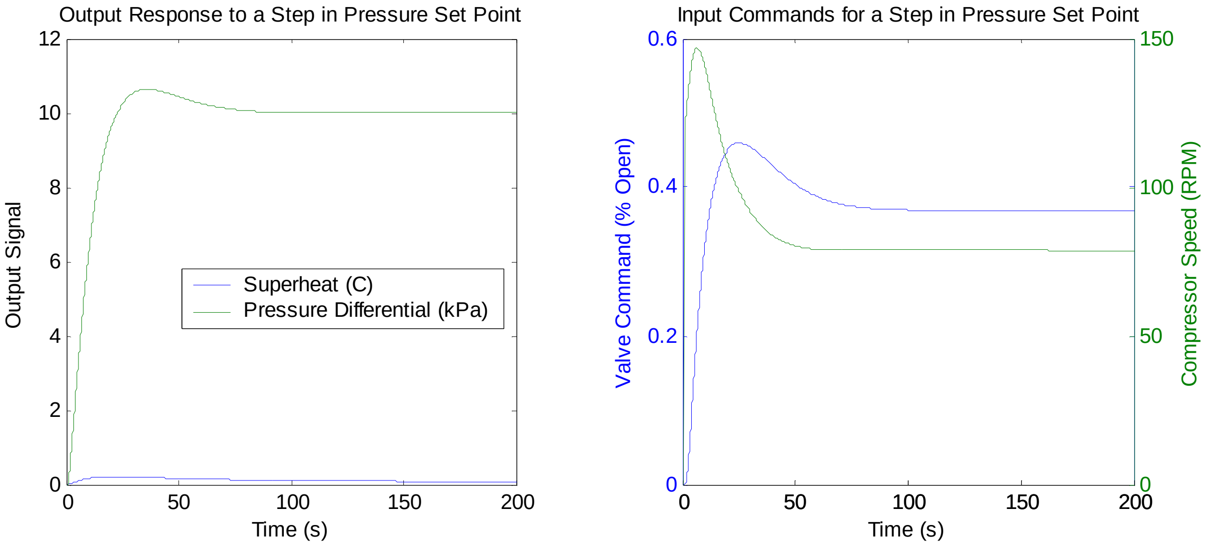


Figure 7.17 - Closed loop superheat/pressure differential output and input response

7.3.2 Closed Loop Analysis - Sensitivity Functions

A simple check on closed loop controller performance can be obtained by comparing the input-output sensitivity functions for the closed loop system. A block diagram of the closed loop system configuration is provided in Figure 7.18. The block diagram includes the controller, K, the plant model, G, and a filter in the feedback path, F. For decentralized control, K is a 2×2 diagonal transfer function matrix, and F is always a 2×2 diagonal transfer function matrix. Two uncontrolled exogenous inputs are assumed to enter the plant. An additive input disturbance, d, enters prior to the system plant and represents potential uncertainty in the plant model. Also a noise disturbance, n, is included in the feedback path to capture the effect of the sensor noise on the experimental system.

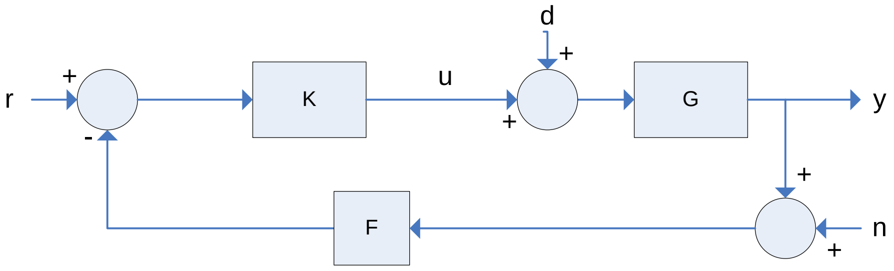


Figure 7.18 - Block diagram of the closed loop system used for the sensitivity analysis

From the block diagram in Figure 7.18 the closed loop relations in Eqs. 7.3-7.6 can be obtained.

Closed Loop System:

$$\frac{y}{r} = (I + GKF)^{-1} GK \tag{7.3}$$

Input Disturbance Sensitivity:

$$\frac{y}{d} = (I + GKF)^{-1} G \tag{7.4}$$

Output Noise Sensitivity:

$$\frac{y}{n} = (I + GKF)^{-1} GKF \tag{7.5}$$

Input Noise Sensitivity:

$$\frac{u}{n} = -(I + KFG)^{-1} KF \tag{7.6}$$

Bode magnitude plots of the relations in Eqs. 7.3-7.6 are provided in Figures 7.19-7.22. In Figure 7.19, the closed loop response of the diagonal elements between the two controllers is quite comparable, although there is a significant reduction in the off diagonal terms for the superheat/pressure differential model. Figure 7.20 reveals that both systems have good steady state disturbance rejection. Figures 7.21 and 7.22 identify the superheat/pressure differential controller as having significantly better noise attenuation characteristics.

7.3.3 Experimental Performance

The experimental performance of the two controllers is summarized in Figures 7.23 and 7.24. Comparing the results in parts a) and b), it is clear that the controller with evaporator pressure as a controlled variable oscillates around the pressure set point. This becomes readily apparent when comparing the zoomed pressure response to a reference step, shown in part d). The superheat/pressure differential controller settles at the new set point in about 25 seconds, whereas the superheat/evaporator response takes approximately 40 seconds just to reach the new set point with persistent oscillations that decay slightly over the next 100 seconds.

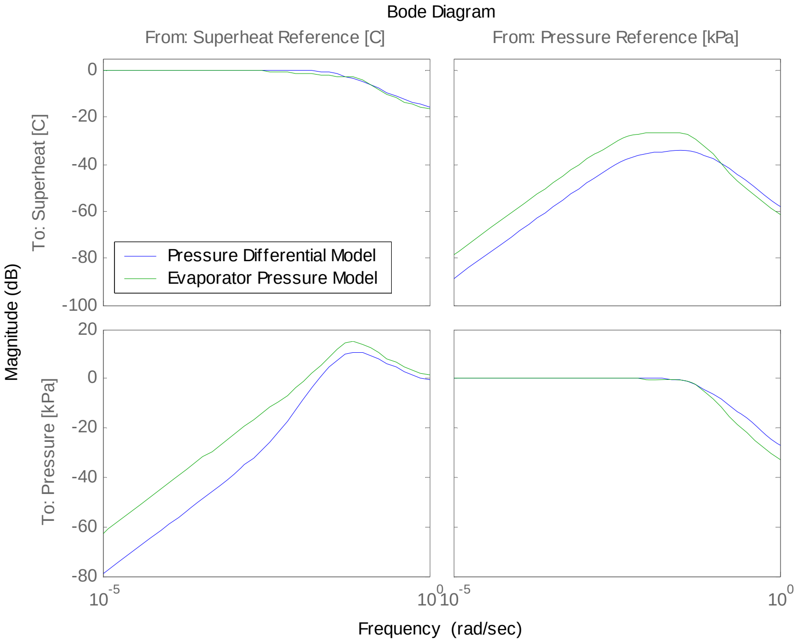


Figure 7.19 - Closed loop system response for decentralized controllers

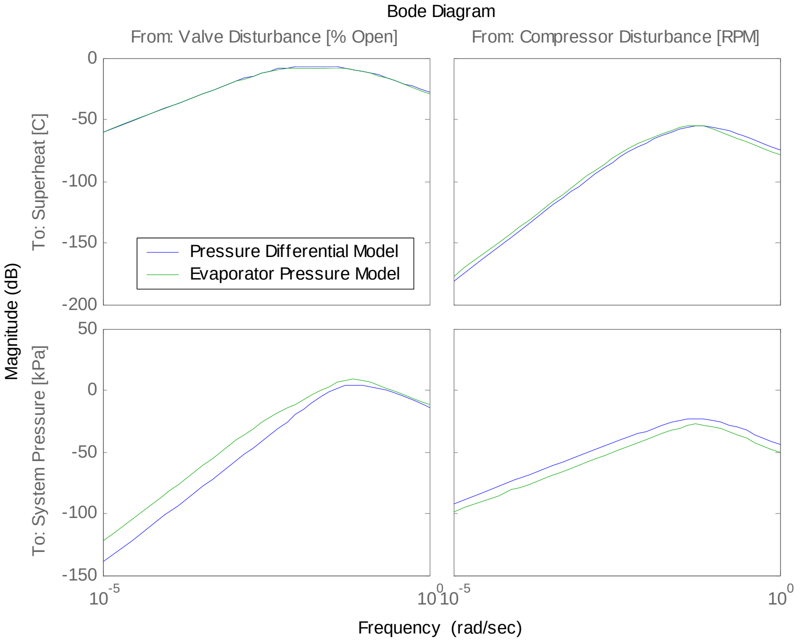


Figure 7.20 - Closed loop disturbance rejection of the decentralized controllers

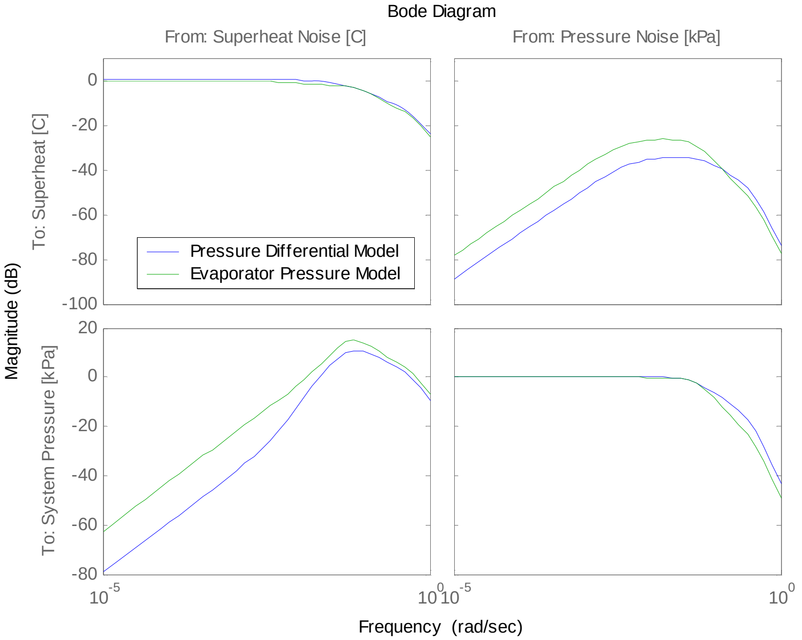


Figure 7.21 - Output sensitivity to a noise input for the decentralized controllers

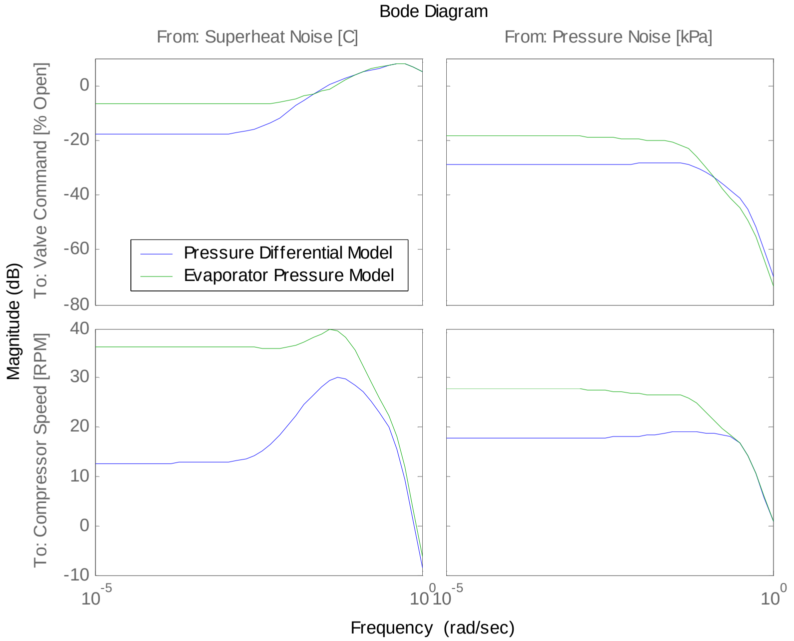
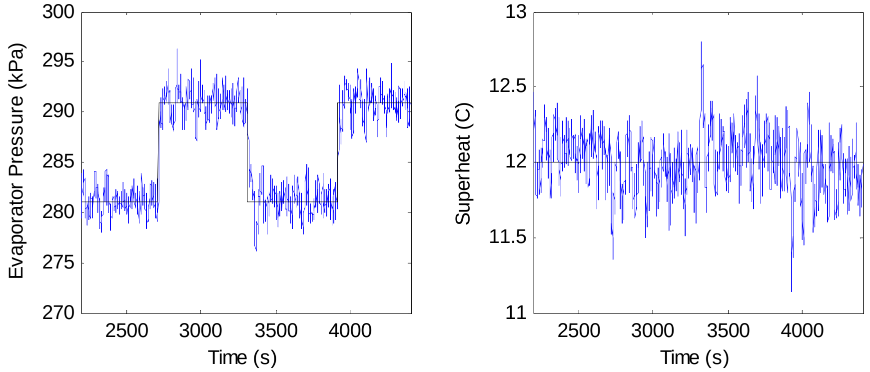
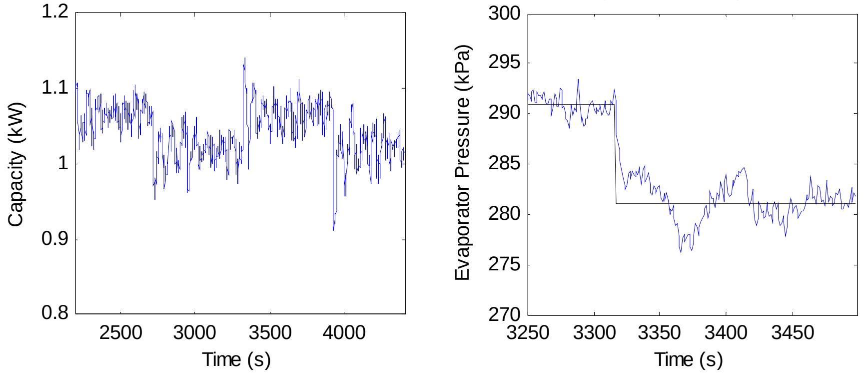


Figure 7.22 - Input sensitivity to a noise input for the decentralized controllers

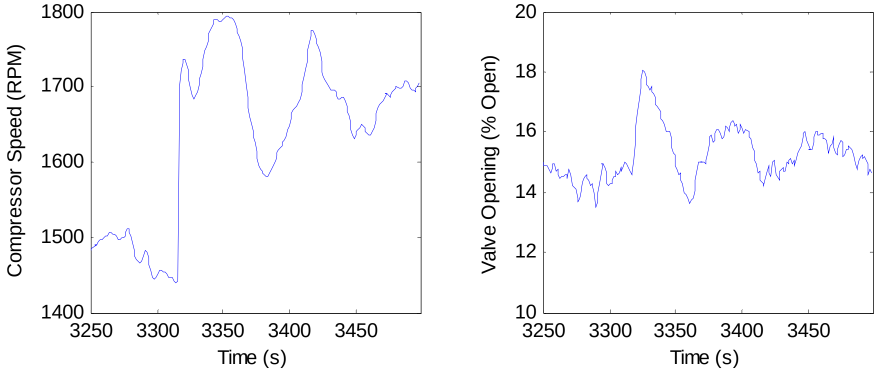
The fighting that results from the coupled superheat/evaporator pressure dynamics is perhaps best illustrated by comparing the actuator signals shown in parts e) and f). The compressor speed and valve actuation required by the superheat/evaporator pressure controller oscillate considerably after the initial step, whereas the reduced coupling of the superheat/pressure differential feedback configuration results in actuator performance comparable to that predicted by the simulation in Figure 7.17. It should be noted that although the superheat/pressure differential controller obtains better regulation, part c) indicates that the controller still induced oscillations in system capacity instead of cleanly transitioning to a new set point.


a) Pressure Response

b) Superheat Regulation


c) System Capacity

d) Pressure Response


e) Compressor Actuation

f) Valve Actuation

Figure 7.23 - Experimental response of the superheat/evaporator pressure decentralized PID controller

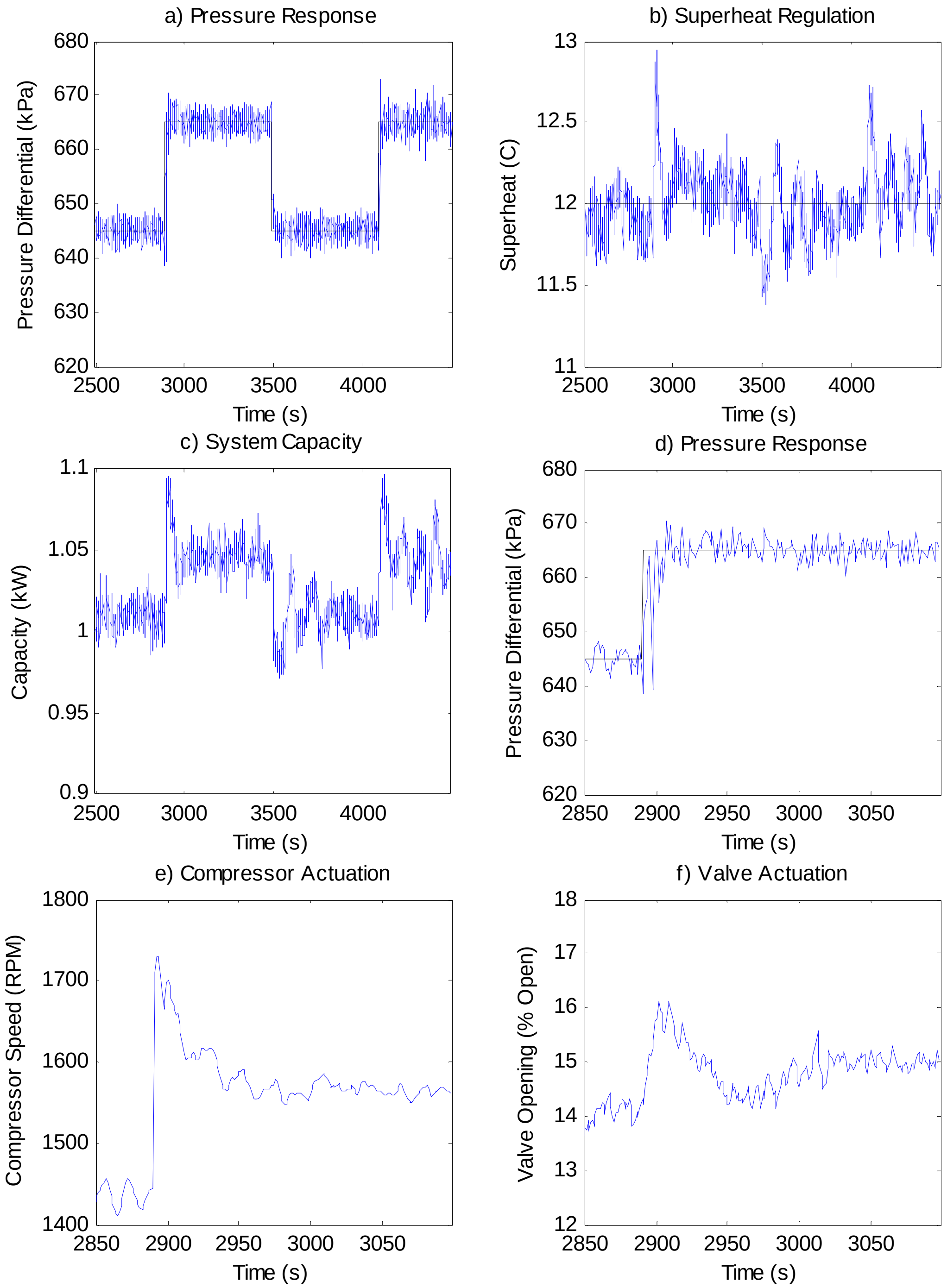


Figure 7.24 - Experimental response of the superheat/pressure differential decentralized PID controller

7.4 Static Decoupling

A static decoupler is a simple tool that can be used to improve the steady state performance of decentralized PID controllers by accounting for the static coupling inherent to the plant. The complete static decoupler design was originally presented by Astrom [5]. One important characteristic of this decoupled control architecture is that good performance will only be obtained if the bandwidth of the decoupled controller is sufficiently low. The static decoupler is found by using the system model shown in Eq. 7.7, where the *G*'s are the transfer functions from valve and compressor inputs to the superheat and pressure outputs, u is the valve input, ω is the compressor speed, T_{sh} is the evaporator superheat, and P_{sys} is either the evaporator pressure or the pressure differential of the system.

$$\begin{bmatrix} T_{sh} \\ P_{sys} \end{bmatrix} = \begin{bmatrix} G_{u\to sh}(s) & G_{\omega\to sh}(s) \\ G_{u\to P}(s) & G_{\omega\to P}(s) \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \tag{7.7}$$

If the magnitude of the transfer function matrix is evaluated at steady-state, the static decoupling matrix is given by Eq. 7.8.

$$D = G^{-1}(0) = \frac{1}{\det(G(0))} \begin{bmatrix} G_{\omega\to P}(0) & -G_{\omega\to sh}(0) \\ -G_{u\to P}(0) & G_{u\to sh}(0) \end{bmatrix} \tag{7.8}$$

Using the identified plant models from Section 7.2 the static decoupler for the superheat/evaporator pressure model is given by Eq. 7.9, and the static decoupler for the superheat/pressure differential model is given in Eq. 7.10.

$$D_{Peo} = \begin{bmatrix} -0.4777 & -0.1187 \\ -66.1344 & -24.3832 \end{bmatrix} \tag{7.9}$$

$$D_{dP} = \begin{bmatrix} -0.1272 & 0.0365 \\ 4.3256 & 7.7990 \end{bmatrix} \tag{7.10}$$

7.4.1 Controller Design in Simulation

As in Section 7.3, the decoupled controllers were tuned to obtain a closed loop system response with a rise time of approximately 25 seconds and a settling time less than 100 seconds. Figures 7.25 and 7.26 present the resulting output response and the required system inputs. The superheat/evaporator pressure controller used the following decoupled PID gains:

$$P_{sh} = 2 \quad I_{sh} = 0.01 \quad D_{sh} = 4 \quad P_{Pe} = 2.5 \quad I_{Pe} = 0.125$$

The superheat/pressure differential controller used the following decoupled PID gains:

$$P_{sh} = 2 \quad I_{sh} = 0.01 \quad D_{sh} = 4 \quad P_{dP} = 2.5 \quad I_{dP} = 0.11$$

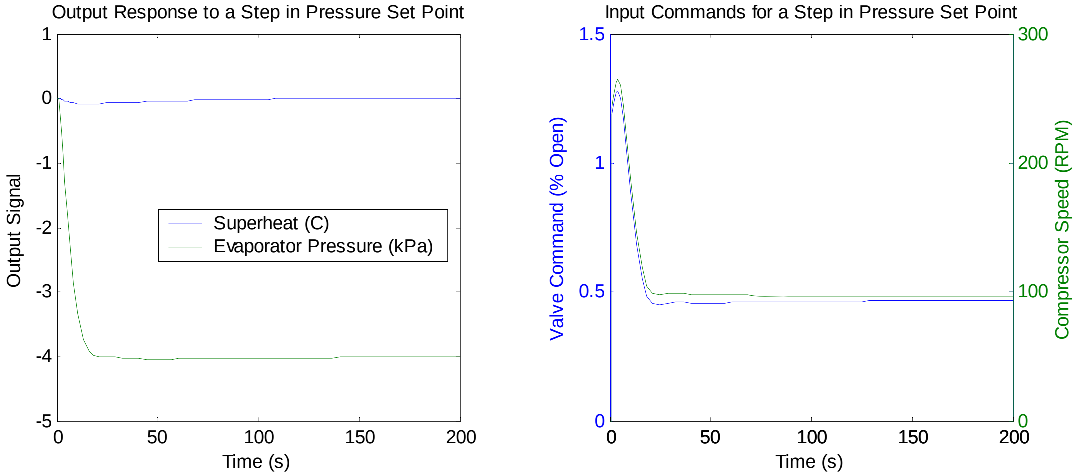


Figure 7.25 - Closed loop superheat/evaporator pressure output and input response

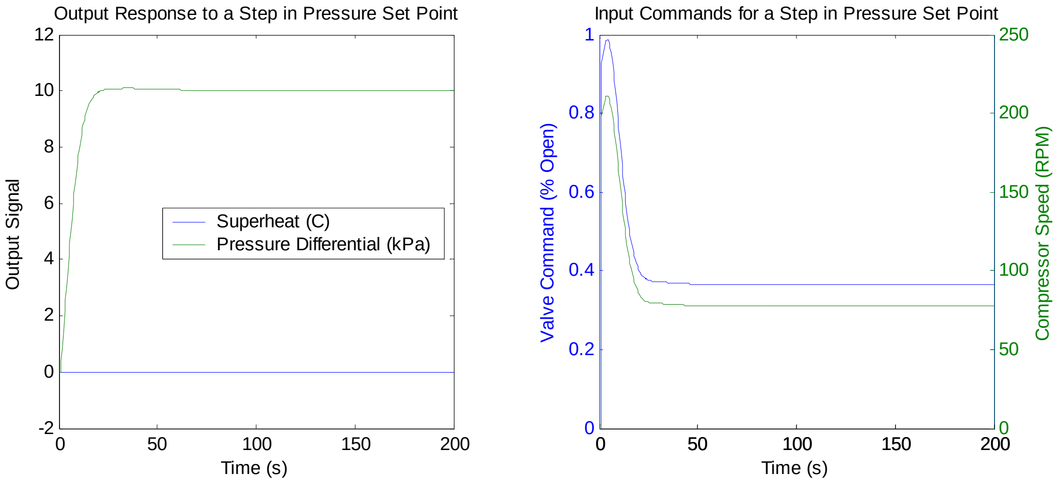


Figure 7.26 - Closed loop superheat/pressure differential output and input response

7.4.2 Closed Loop Analysis - Sensitivity Functions

The closed loop sensitivity functions defined in Eqs. 7.3-7.6 were evaluated for the decoupled PID controllers, and the bode magnitude plots of the resulting transfer functions are provided in Figures 7.27-7.30. The closed loop response in Figure 7.27 shows that both controllers should provide good steady state tracking. Both controllers also reject steady state disturbances quite well, since the disturbance sensitivity function in Figure 7.28 rolls off to zero at steady state. Figure 7.29 shows that the output noise does not get amplified and adversely affect the outputs of the system. The main difference between the responses of the two controllers can be seen in Figure 7.30, where the controller that regulates pressure differential instead of evaporator pressure has large reductions in the effect of sensor noise on the actuator response. The high gains from sensor noise to actuator response for the evaporator pressure model will result in oscillatory actuation on the actual system, which could induce oscillations in the regulated outputs.

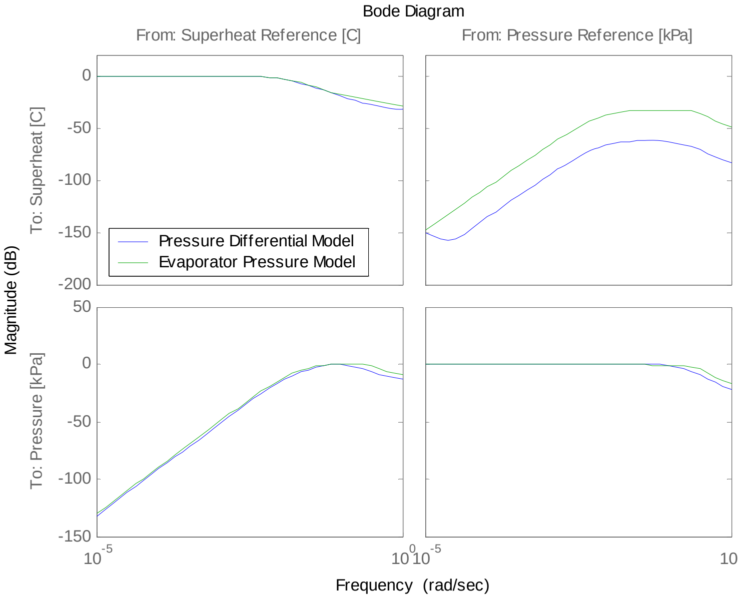


Figure 7.27 - Closed loop system response for the decoupled controllers

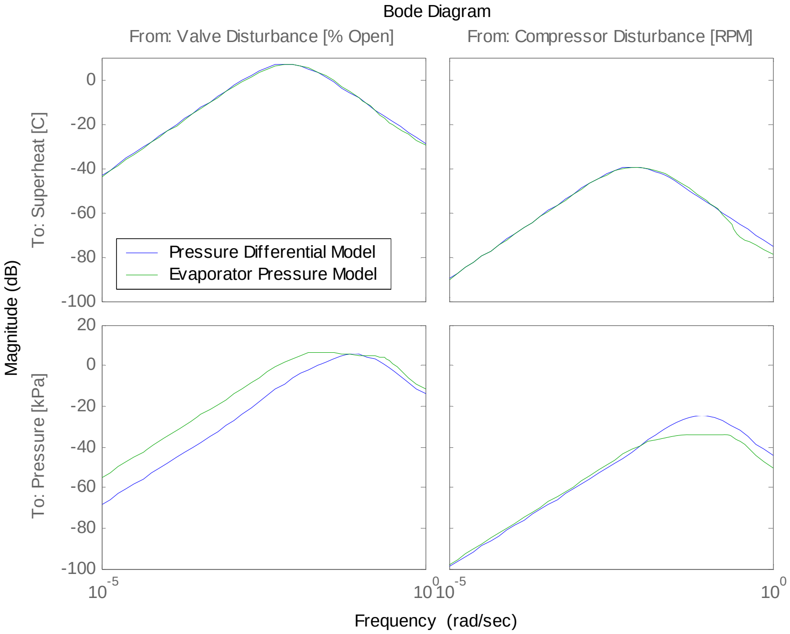


Figure 7.28 - Input disturbance rejection of the decoupled PID controllers

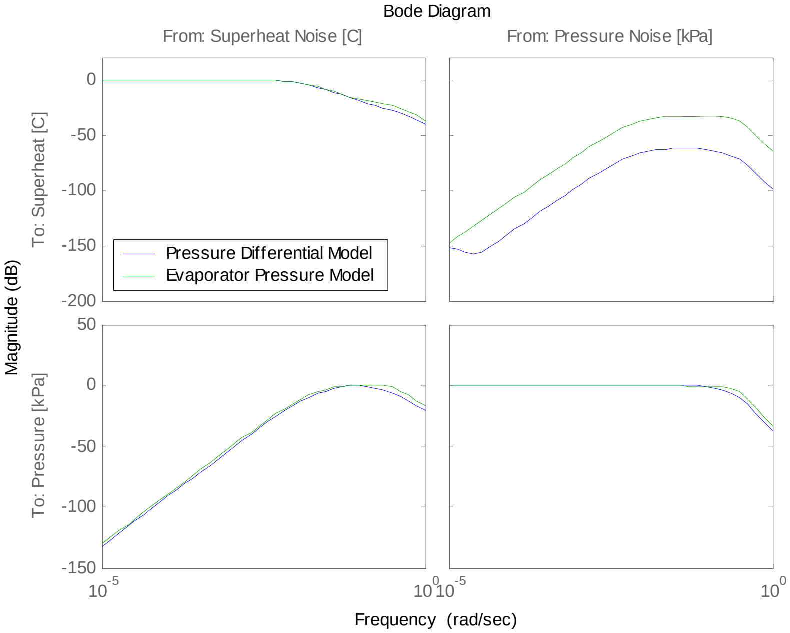


Figure 7.29 - Output noise attenuation of the decoupled PID controllers

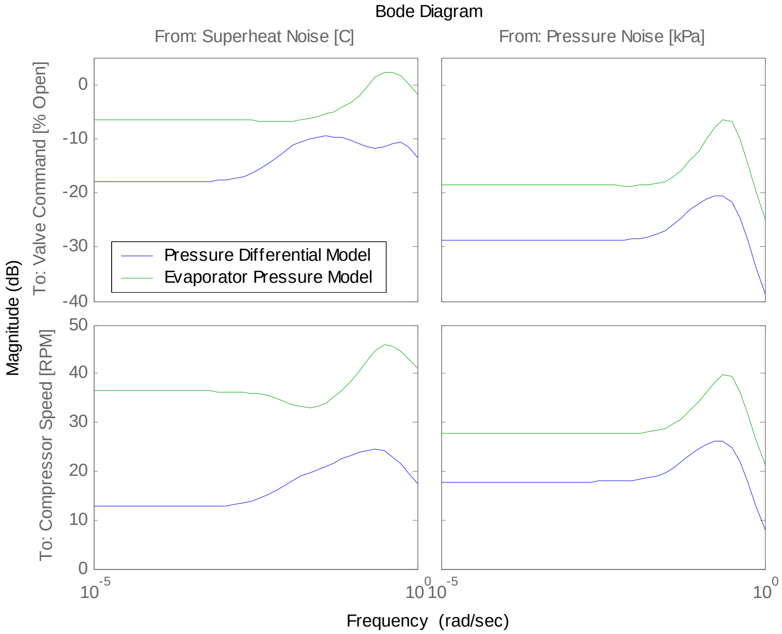


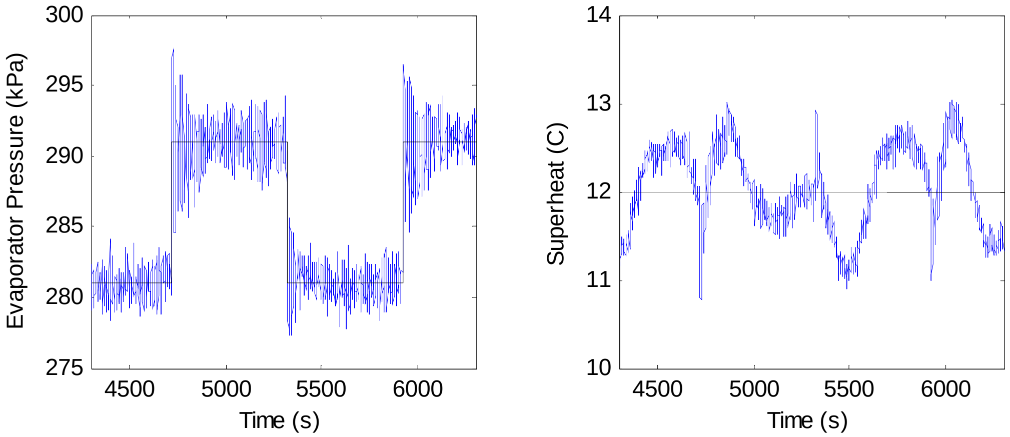
Figure 7.30 - Input response to sensor noise for the decoupled PID controllers

7.4.3 Experimental Controller Performance

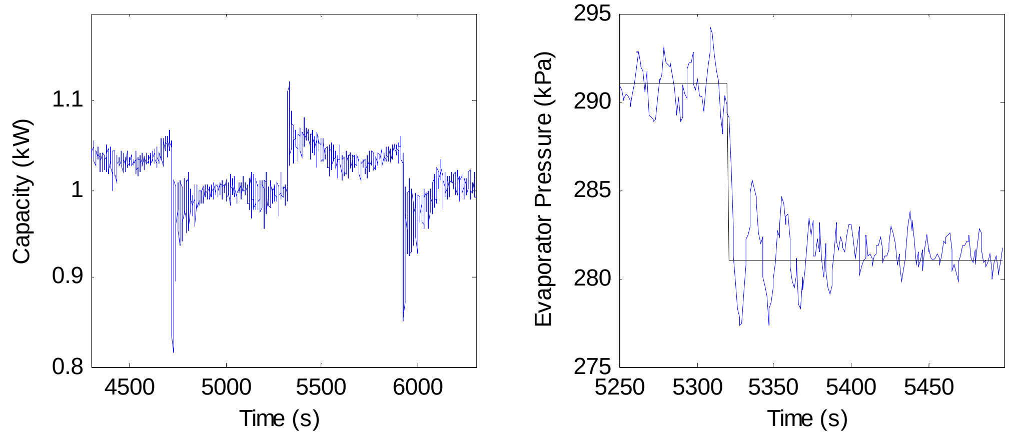
The experimental performance of the two decoupled controllers designed in Section 7.4.1 are summarized in Figures 7.31 and 7.32. The high bandwidth demands of the performance specifications create a situation where the highly coupled system with superheat and evaporator pressure outputs oscillates considerably due to poor dynamic decoupling. As predicted in Figure 7.30, the superheat/evaporator pressure controller causes the actuators oscillate as a result of noise in the system sensors. It is interesting to note that the decoupled controller of superheat and evaporator pressure switches system capacity considerably better than the decentralized control framework, as seen in part c) of Figures 7.23 and 7.31.

Both decoupled controllers do not regulate superheat as well as the decentralized controllers, although this may be due to the fact that in simulation the decoupling works exceptionally well for a wide range of PID gains on the superheat loop, making it difficult to assess which gains to use on the physical system. The decoupled superheat/pressure differential controller meets the performance specifications, and has a settling time around 25 seconds to a step change in pressure. The decoupled controller also appears to regulate system capacity better than the decentralized controller for the superheat/pressure differential case. This is likely due to the smooth actuator movement generated by the decoupled controller, as seen in parts e) and f) of Figure 7.32.

a) Pressure Response b) Superheat Regulation



c) System Capacity d) Pressure Response



e) Compressor Actuation f) Valve Actuation

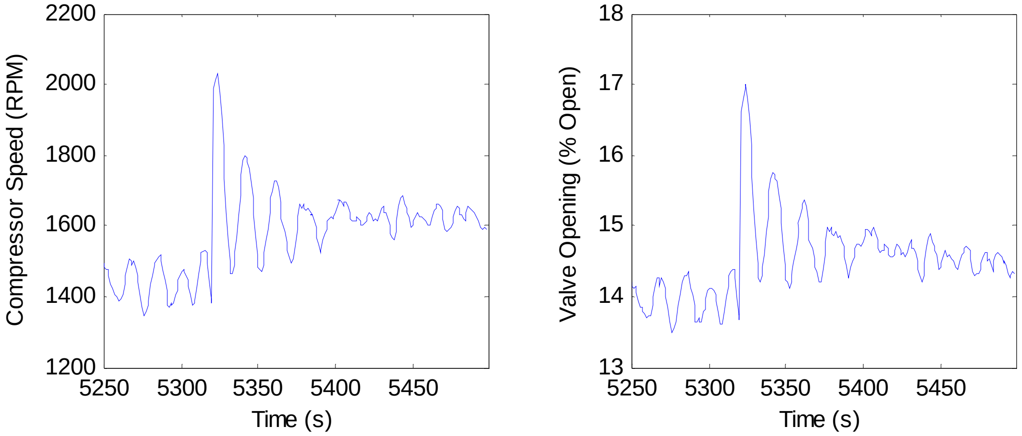


Figure 7.31 - Experimental response of a superheat/evaporator pressure decoupled PID controller

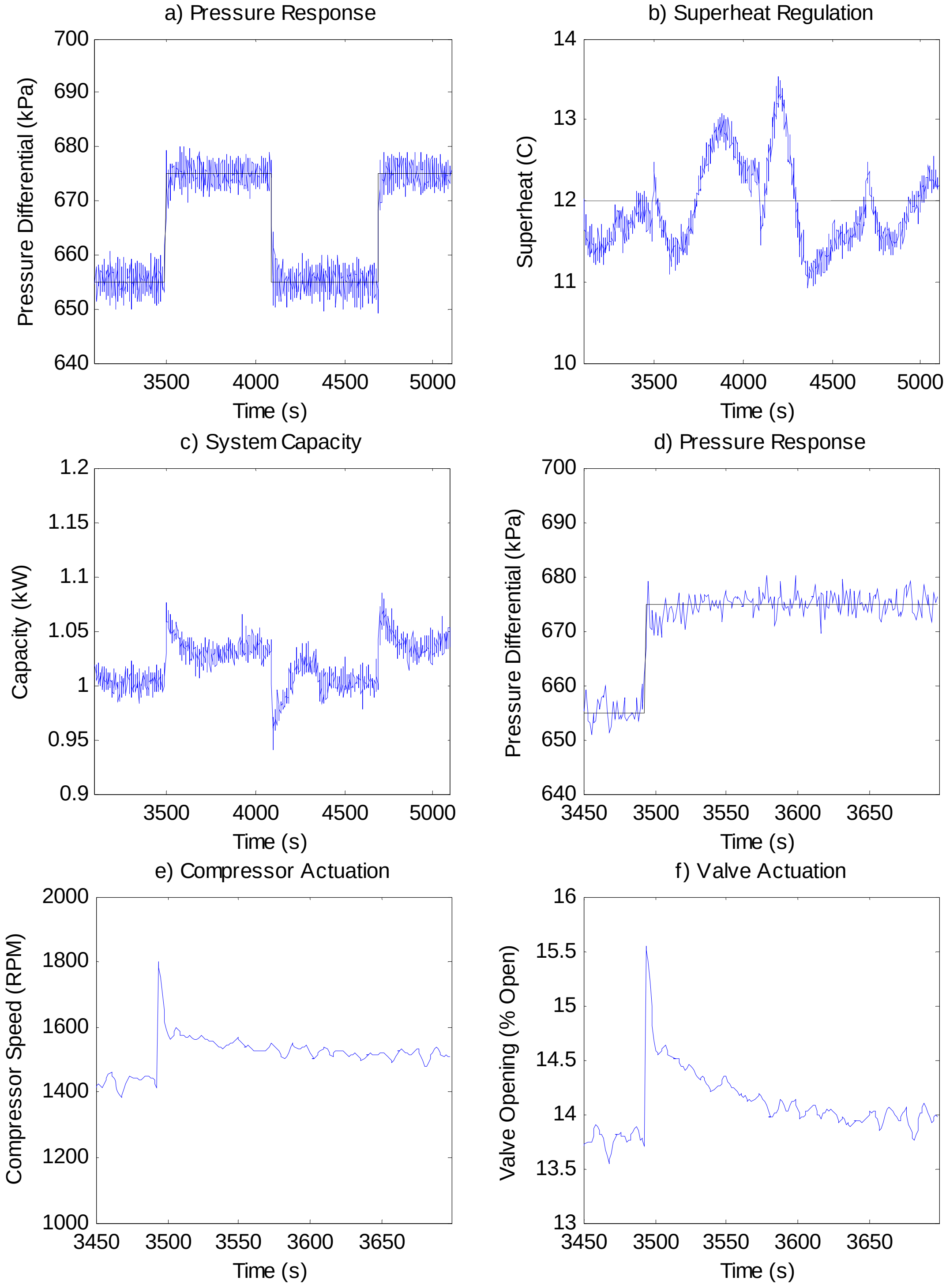


Figure 7.32 - Experimental response of the superheat/pressure differential decoupled PID controller

7.5 LQG Multivariable Control Design

Linear quadratic gaussian (LQG) multivariable controllers have been shown to provide high performance on vapor compression systems [35]. The demonstrated performance of LQG makes it an ideal controller to use as a baseline to assess the performance of the decentralized and decoupled PID control approaches for the two feedback frameworks. If a particular feedback configuration and simple controller can be realized that attains the same level of performance as the LQG controller, it will facilitate the adoption of continuously operated systems without requiring the use of an advanced MIMO control design framework. The LQG control design presented in this section also improves on the iterative search tuning method proposed in [35] by providing an intuitive approach that does not require significant iteration to obtain a high performance controller.

For linear quadratic gaussian (LQG) control, it is assumed that a linear plant model is known, and the measurement and process noise are stochastic with known statistical properties [72]. In mathematical terms, the system can be described by Eqs. 7.11 and 7.12, where w_d and w_n are assumed to be uncorrelated zero-mean Gaussian processes. The variance of w_d and w_n are given by Eqs. 7.13 and 7.14.

$$\dot{x} = Ax + Bu + w_d \tag{7.11}$$

$$y = Cx + w_n \tag{7.12}$$

$$E\left\{w_d(t)w_d(t)^T\right\} = W \tag{7.13}$$

$$E\left\{w_n(t)w_n(t)^T\right\} = V \tag{7.14}$$

In order to design an LQG controller, the separation principle is used to allow a linear quadratic regulator (LQR) and a Kalman observer to be designed independently. After the observer and controller have been created, they are combined to form the desired controller. The design of the appropriate state estimator is simplified when the model identification approach of Section 7.2 is used. As a byproduct of the model identification, values for both V and W can be calculated from the standard prediction error/maximum likelihood identification procedure. Further details of this calculation can be found in the supplied Matlab code in Appendix B.

Using the identified V and W, the resulting Kalman filter has the structure of an ordinary observer, given by Eq. 7.15.

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f\left(y - C\hat{x}\right) \tag{7.15}$$

The optimal choice of K_f, that minimizes the state error is given by Eq. 7.16, where $Y = Y^T$ is the unique positive-semidefinite solution of the algebraic Riccati equation, given in Eq. 7.17.

$$K_f = YC^TV^{-1} \tag{7.16}$$

$$YA^T + AY - YC^TV^{-1}CY + W = 0 \tag{7.17}$$

The standard LQR controller must be modified in order to appropriately track external references. A simple modification is summarized in [3], where the original plant model is augmented with integrated error states in order to force the system to track external references. A diagram of the augmentation is provided in Figure 7.33, and the mathematical description of the augmented plant is provided in Eq. 7.18.

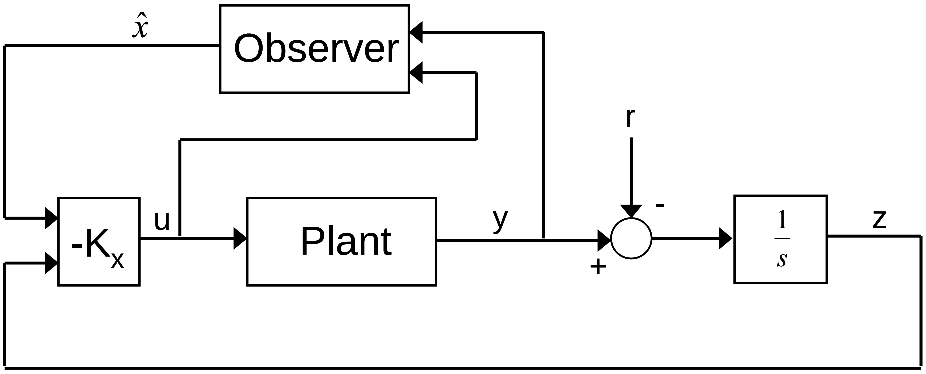


Figure 7.33 - Augmented LQG controller for reference tracking

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \qquad (7.18)$$

Using the augmented plant model, the Q and R weighting functions for the performance index in Eq. 7.19 can be selected to obtain the desired controller properties. The weights on the z states are used to provide the integral controller action that will ensure zero steady state error (similar to the integral gain in a PID controller).

$$V = \int_0^\infty \left(u^T R u + \bar{x}^T Q \bar{x} \right) dt \qquad (7.19)$$

$$\bar{x} = \begin{bmatrix} x \\ z \end{bmatrix} \qquad (7.20)$$

7.5.1 Evaporator Pressure LQG Controller Design

It is best to design the LQR controller and the Kalman observer for the scaled plant model. This aids in the selection of gains for the LQR controller, since the required inputs and outputs will be comparably weighted regardless of units. The observer design was relatively straight forward due to the model identification procedure that was used to generate the plant model. The variances of the plant and measurement noise used for the Kalman observer design are given in Eqs. 7.21 and 7.22. The identified variance values for the measurement noise in V correspond to an un-scaled uncertainty of ±0.12° C in the superheat measurement and ±2.756 kPa in the evaporator pressure measurement, which are the same as those used in Section 6.5.1. The plant uncertainty, W, was calculated from the model identification procedure.

$$V = \begin{bmatrix} 0.0144 & 0 \\ 0 & 0.8439 \end{bmatrix} \qquad (7.21)$$

$$W = \begin{bmatrix} 9.57e-5 & 9.51e-5 \\ 9.51e-5 & 9.51e-5 \end{bmatrix} \qquad (7.22)$$

When Eqs. 7.21 and 7.22 are used in Eq. 7.17, the resulting observer poles are given by Eq. 7.23.

$$\lambda_o = \begin{bmatrix} -11.65 \\ -0.045 \end{bmatrix} \qquad (7.23)$$

With the observer design complete the remaining task is to design the LQR weights to obtain the desired system performance. The *R* matrix can be used to penalize controller actuation, effectively limiting the actuator response to high frequency components in the feedback path. Increasing the gains in the *R* matrix will reduce the system sensitivity to noise, although large values limit the response time of the closed loop system. The *Q* matrix can be used to tune the proportional and integral gains to obtain a controlled system response. Reducing the gains on the augmented states decreases the effective integral gains of the controller. The resulting gains in the last two columns of the K_x matrix, shown in Figure 7.33, should be comparable in magnitude to those of the integral gain in a PID controller to ensure non-oscillatory performance. The gains in the *Q* matrix related to the observed system states can be used to limit the effective proportional gain of the system, resulting in more damped system responses.

Since the plant inputs and outputs are scaled, the weighting matrices in Eq. 7.24 were used for the initial controller, where the controller is given in Eq. 7.25. Figures 7.34 and 7.35 show the simulated system response to a change in pressure reference for the initial closed loop system without and with noise in the feedback path. It is clear from Figure 7.35 that the controller would produce rapid oscillations in the actuator inputs as a result of the sensor noise in the system. The noise induced oscillations combine with plant uncertainty, actuator limitations, and under damped dominant closed loop pole locations to produce undesirable highly oscillatory behavior on the experimental system.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.24}$$

$$K_x = \begin{bmatrix} -700.3 & 93.3 & -0.903 & -0.430 \\ -179.5 & -88.4 & -0.430 & -0.903 \end{bmatrix} \tag{7.25}$$

Increasing the gains on the input weighting, *R*, and decreasing the weighting on the integrated error signal in *Q* produces a less aggressive controller that is more robust to sensor noise. Tuned values of the *R* and *Q* weighting matrices are shown in Eq. 7.26, and the controller associated with these weights given by Eq. 7.27. The controller has significantly reduced integral terms, especially with regard to the impact of superheat error on valve actuation. The weights on the system states were also reduced as a result of the requirement that less actuation be used by the controller to meet the performance specification. The simulated system response without and with noise in the feedback path is presented in Figures 7.36 and 7.37.

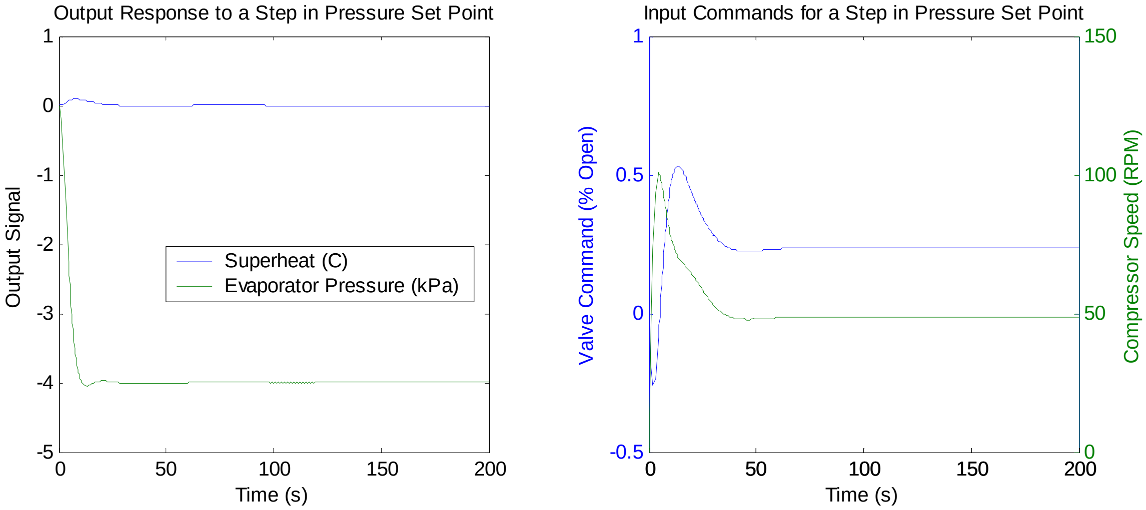


Figure 7.34 - Initial simulated system response without noise in the feedback path

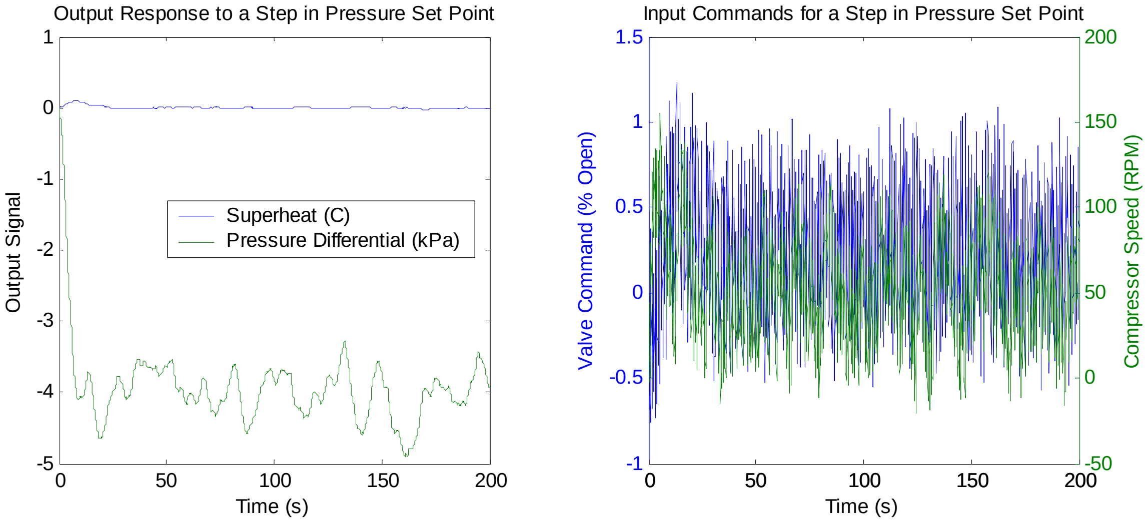


Figure 7.35 - Initial simulated system response with noise in the feedback path

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} \qquad R = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix} \tag{7.26}$$

$$K_x = \begin{bmatrix} -99.6 & 10.4 & -0.0315 & 0.0067 \\ -72.5 & -55.2 & -0.0095 & -0.223 \end{bmatrix} \tag{7.27}$$

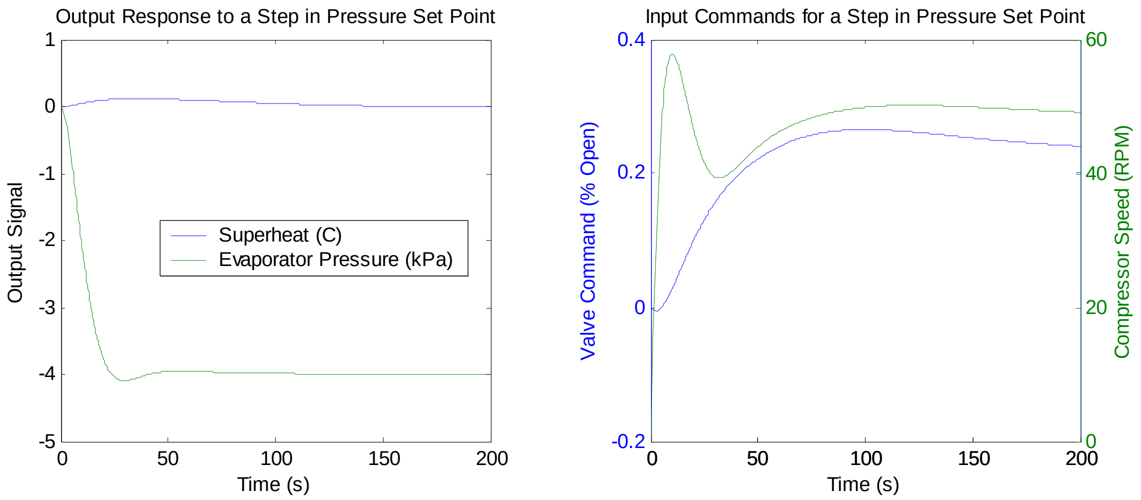


Figure 7.36 - Tuned simulated system response without noise in the feedback path

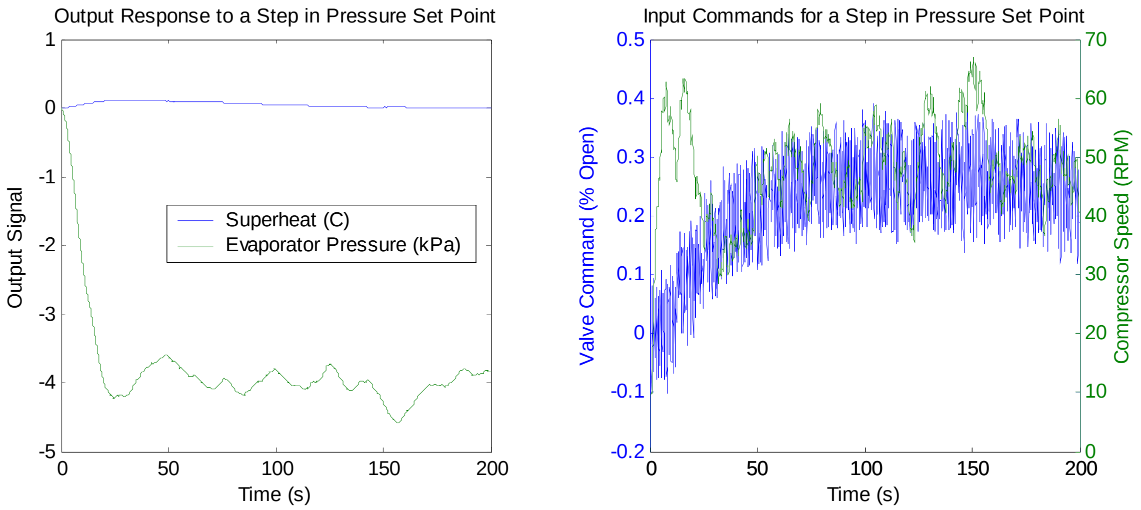


Figure 7.37 - Tuned simulated system response with noise in the feedback path

The controller given by Eq. 7.27 has been tuned so that the control action is within the bandwidth limitations of the actuators, since both the valve and compressor have physical limits to the rates at which they can change position or speed. When this controller is implemented on the system it will effectively track the pressure reference, but the superheat will have persistent oscillations of approximately ±0.75° C. These oscillations are caused by the under damped closed loop pole locations associated with the integral of the superheat error. The relation between the under damped closed loop poles and the augmented superheat error state is apparent by looking at the state vector, eigenvectors, and eigenvalues of the closed loop system, given in Eqs. 7.28-7.30.

$$x_{CL} = \begin{bmatrix} x & \hat{x} & z \end{bmatrix}^T \qquad (7.28)$$

$$\begin{bmatrix} -0.0007 & -0.0003+0.0002i & -0.0003-0.0002i & -0.0003+0.0003i & -0.0003-0.0003i & 0.0004 \\ 0.0156 & -0.0038+0.0034i & -0.0038-0.0034i & -0.0006+0.0005i & -0.0006-0.0005i & 0.0020 \\ -0.7142 & -0.0003+0.0002i & -0.0003-0.0002i & -0.0003+0.0003i & -0.0003-0.0003i & 0.0007 \\ -0.6978 & -0.0038+0.0034i & -0.0038-0.0034i & -0.0006+0.0005i & -0.0006-0.0005i & 0.0050 \\ 0.0161 & -0.0922+0.0368i & -0.0922-0.0368i & 0.9980 & 0.9980 & -0.3181 \\ -0.0499 & 0.9950 & 0.9950 & 0.0611+0.0186i & 0.0611-0.0186i & -0.9480 \end{bmatrix} \tag{7.29}$$

$$\begin{bmatrix} -11.6542 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.1107+0.1060i & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1107-0.1060i & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0196+0.0176i & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0196-0.0176i & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0454 \end{bmatrix} \tag{7.30}$$

By adding a proportional weight to the superheat output in the LQ design, the under damped closed loop poles associated with the integral of the superheat error can be moved onto the real axis, producing a well damped system response. Penalizing the system output in essence penalizes the derivative of the integral error state, increasing the damping associated with the integral error poles. The procedure to add proportional gains to the controller is to modify the controller weights to those shown in Eq. 7.31, with the resulting controller given by Eq. 7.32. Figure 7.38 summarizes the effect of the various controller modifications on the dominant closed loop poles of the system.

$$Q = \begin{bmatrix} (K_pC)^T(K_pC) & \mathbf{0} \\ \mathbf{0} & \begin{matrix} 0.01935 & 0 \\ 0 & 1 \end{matrix} \end{bmatrix} \qquad R = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix} \qquad K_p = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix} \tag{7.31}$$

$$K_x = \begin{bmatrix} -89.8 & 10.9 & -0.0137 & 0.0156 \\ 110 & -69.6 & -0.0068 & -0.3124 \end{bmatrix} \tag{7.32}$$

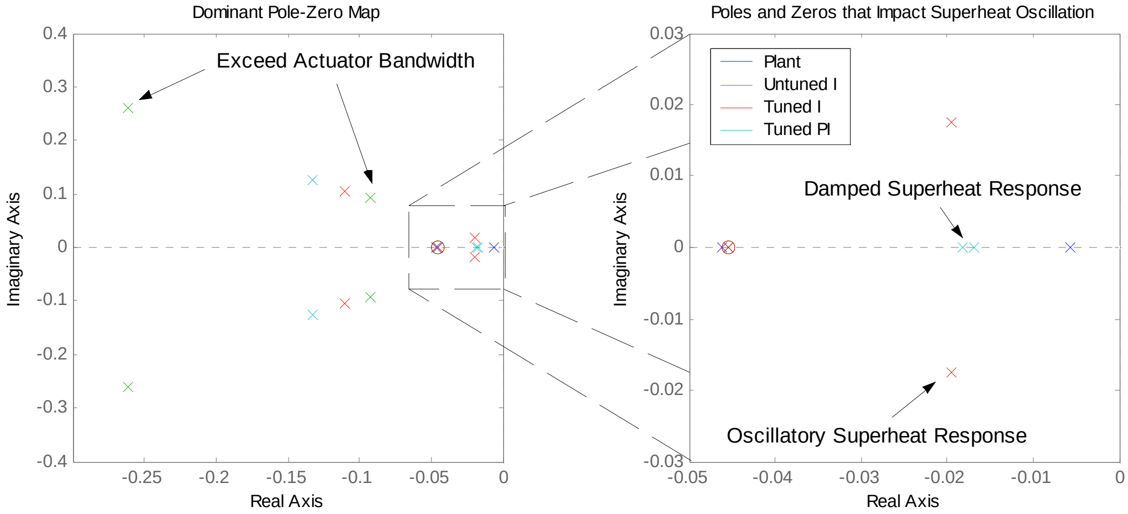


Figure 7.38 - Dominant closed loop pole locations for the un-tuned, tuned, and tuned PI controller configurations

7.5.2 Evaporator Pressure LQG Controller Performance

The performance of the tuned PI evaporator pressure LQG controller was verified experimentally and is summarized in Figure 7.39. The controller is able to track step changes in the evaporator pressure quite well while regulating superheat with maximum deviations of 1° C on the step changes and quickly returning to small errors around the set-point. It is clear that the LQG evaporator pressure controller is the best evaporator pressure controller designed in this chapter for producing clean steps in system capacity while minimizing system actuation, as shown in part c) of Figure 7.39.

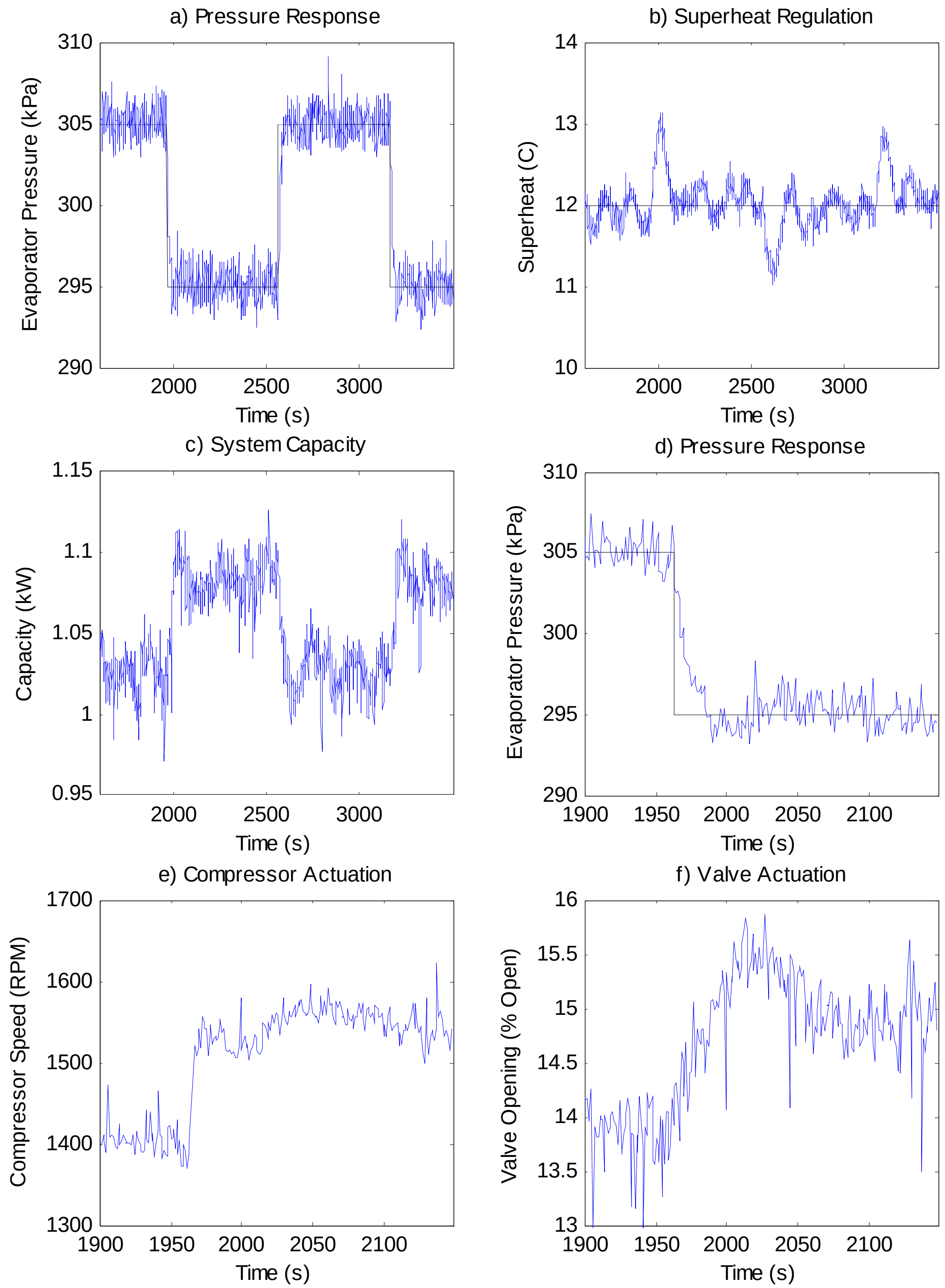


Figure 7.39 - Experimental performance of the tuned evaporator pressure LQG controller

7.5.3 Pressure Differential LQG Controller Design

The design procedure for the pressure differential/superheat LQG controller is basically the same as that of the evaporator pressure/superheat controller described in Section 7.5.1. For the pressure differential controller, the variances of the plant and measurement noise used for the Kalman observer design are given in Eqs. 7.33 and 7.34. The identified variance values for the measurement noise found in V correspond to an un-scaled uncertainty of ±0.12° C in the superheat measurement and ±8.268 kPa in the pressure differential measurement, which are the same as those used in Section 6.5.1. The plant uncertainty, W, was calculated from the model identification procedure.

$$V = \begin{bmatrix} 0.0144 & 0 \\ 0 & 1.898 \end{bmatrix} \tag{7.33}$$

$$W = \begin{bmatrix} 2.3e-6 & 1.51e-5 \\ 1.51e-5 & 1.15e-4 \end{bmatrix} \tag{7.34}$$

When Eqs. 7.33 and 7.34 are used in Eq. 7.17, the resulting observer poles are given by Eq. 7.35.

$$\lambda_o = \begin{bmatrix} -8.19 \\ -0.041 \end{bmatrix} \tag{7.35}$$

With the observer design complete the remaining task is to design the LQR weights to obtain the desired system performance. As in Section 7.5.1, the R matrix can be used to penalize controller actuation, effectively limiting the actuator response to high frequency components in the feedback path. Increasing the gains in the R matrix will reduce the system sensitivity to noise, although large values limit the response time of the closed loop system. A diagonal Q matrix will allow the designer to specify individual weights for both the system and augmented states. Reducing the gains on the augmented states decreases the effective integral gains of the controller. The resulting gains in the last two columns of the K_x matrix should be comparable in magnitude to those of the integral gain in a PID controller to ensure stable performance. The gains in the Q matrix related to the observed system states can be used to limit the effective proportional gain of the system, resulting in more damped system responses.

Since the plant inputs and outputs are scaled, the weighting matrices in Eq. 7.36 were used for the initial controller, where the controller is given in Eq. 7.37. Figures 7.40 and 7.41 show the simulated system response to a change in pressure reference for the initial closed loop system without and with noise in the feedback path. It is clear from Figure 7.41 that the controller would produce large oscillations in the actuator inputs as a result of the sensor noise in the system. The noise induced oscillations combine with plant uncertainty, actuator limitations, and under damped dominant closed loop pole locations to produce highly oscillatory behavior on the experimental system.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.36}$$

$$K_x = \begin{bmatrix} -643.8 & 17.8 & -0.998 & -0.0680 \\ -179.5 & 139.9 & -0.0680 & 0.998 \end{bmatrix} \quad (7.37)$$

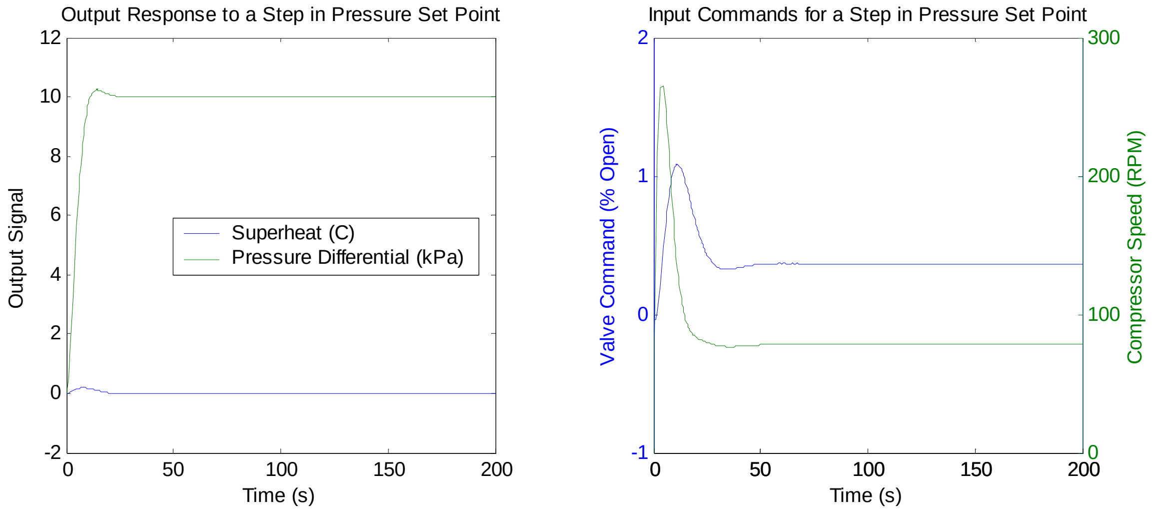


Figure 7.40 - Initial simulated system response without noise in the feedback path

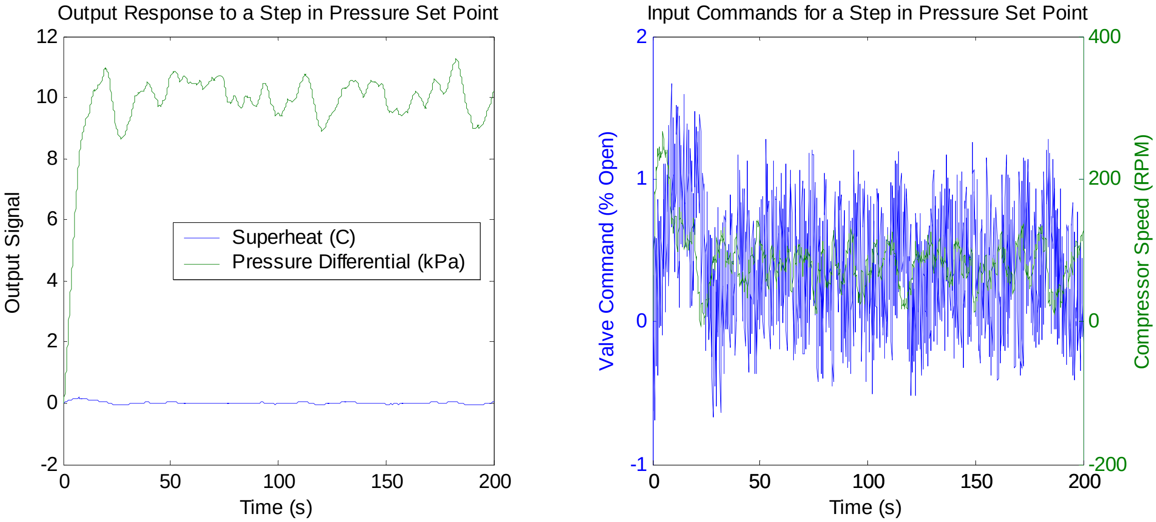


Figure 7.41 - Initial simulated system response with noise in the feedback path

By increasing the gains on the input weighting, R, and decreasing the weighting on the integrated error signal in Q produces a less aggressive controller that is more robust to sensor noise. Tuned values of the R and Q weighting matrices are shown in Eq. 7.38, and the controller associated with these weights given by Eq. 7.39. The controller has significantly reduced integral terms, especially with regard to the impact of superheat error on valve actuation. The simulated system response without and with noise in the feedback path is presented in Figures 7.42 and 7.43.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} \qquad R = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix} \tag{7.38}$$

$$K_x = \begin{bmatrix} -107.9 & 5.35 & -0.0315 & 0.0052 \\ -31.8 & 58.1 & 0.0073 & 0.223 \end{bmatrix} \tag{7.39}$$

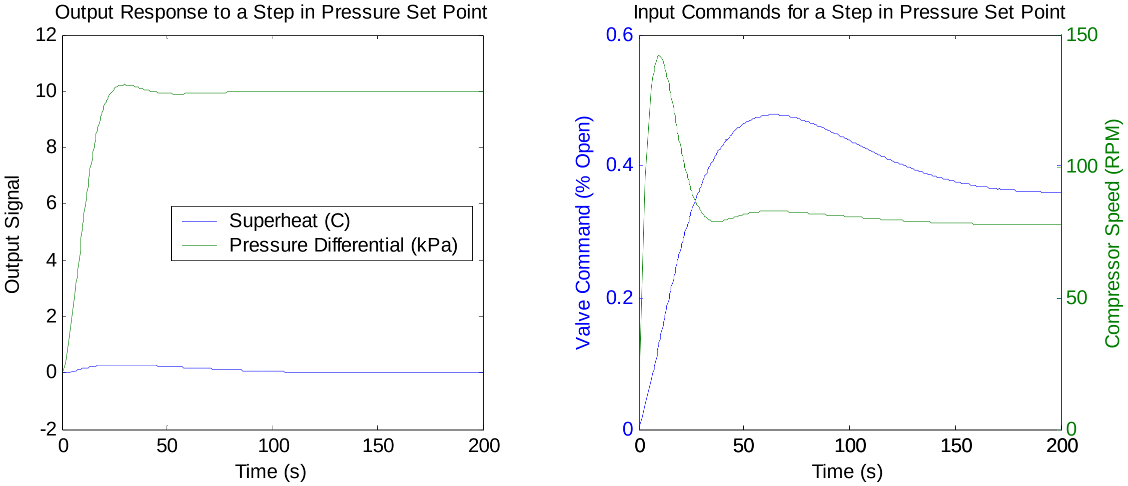


Figure 7.42 - Tuned simulated system response without noise in the feedback path

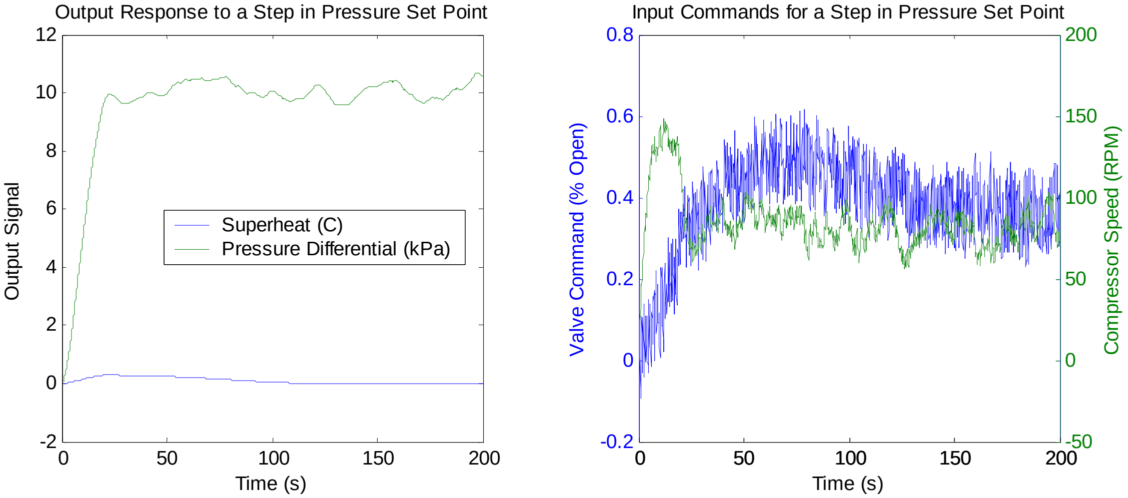


Figure 7.43 - Tuned simulated system response with noise in the feedback path

The controller given by Eq. 7.39 has been tuned so that the control action is within the bandwidth limitations of the actuators, since both the valve and compressor have physical limits to the rates at which they can change position or speed. When this controller is implemented on the system it will effectively track the pressure reference, but the superheat will have persistent oscillations of approximately ±0.75° C, as shown in Figure 7.45. These oscillations are caused by the under damped closed loop pole locations associated with the integral of the superheat error. The relation between the under damped closed loop poles and the augmented superheat error state appears in the state vector, eigenvectors, and eigenvalues of the closed loop system, given in Eqs. 7.40-7.42.

$$x_{CL} = \begin{bmatrix} x & \hat{x} & z \end{bmatrix}^T \qquad (7.40)$$

$$\begin{bmatrix} -0.0026 & -0.0002+0.0003i & -0.0002-0.0003i & -0.0002+0.0001i & -0.0002-0.0001i & -0.0002 \\ -0.0137 & -0.0039+0.0038i & -0.0039-0.0038i & -0.0000+0.0000i & -0.0000-0.0000i & 0.0011 \\ -0.7002 & -0.0002+0.0003i & -0.0002-0.0003 & -0.0002+0.0001i & -0.0002-0.0001i & -0.0002 \\ -0.7114 & -0.0039+0.0038i & -0.0039-0.0038i & -0.0000+0.0000i & -0.0000-0.0000i & 0.0041 \\ 0.0260 & 0.1479-0.0250i & 0.1479+0.0250i & 1.0000 & 1.0000 & 0.3643 \\ 0.0531 & 0.9887 & 0.9887 & -0.0043-0.0020i & -0.0043+0.0020i & -0.9313 \end{bmatrix} \qquad (7.41)$$

$$\begin{bmatrix} -8.1906 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.1294+0.1261i & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1294-0.1261i & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0124+0.0121i & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0124-0.0121i & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0409 \end{bmatrix} \qquad (7.42)$$

By adding a proportional weight to the superheat output in the LQ design, the under damped closed loop poles associated with the integral of the superheat error can be moved onto the real axis, producing a well damped system response. Penalizing the system output in essence penalizes the derivative of the integral error state, increasing the damping associated with the integral error poles. The procedure to add proportional gains to the controller is to modify the controller weights to those shown in Eq. 7.43, and the resulting controller is given in Eq. 7.44. Figure 7.44 summarizes the effect of the various controller modifications on the dominant closed loop poles of the system.

$$Q = \begin{bmatrix} (K_pC)^T(K_pC) & \mathbf{0} \\ \mathbf{0} & \begin{matrix} 0.021 & 0 \\ 0 & 1 \end{matrix} \end{bmatrix} \qquad R = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix} \qquad K_p = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix} \qquad (7.43)$$

$$K_x = \begin{bmatrix} -89.8 & 10.9 & -0.0137 & 0.0156 \\ 110 & -69.6 & -0.0068 & -0.3124 \end{bmatrix} \qquad (7.44)$$

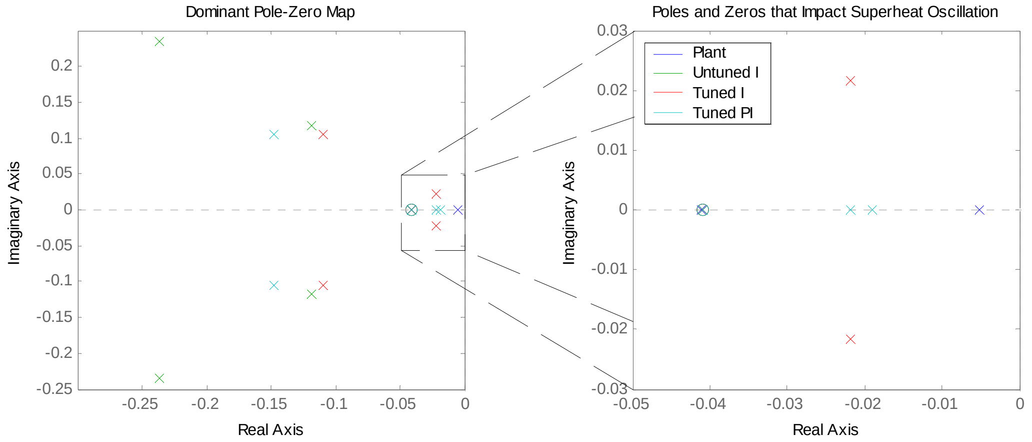


Figure 7.44 - Dominant closed loop pole locations for the un-tuned, tuned, and tuned PI pressure differential controller configurations

7.5.4 Pressure Differential LQG Controller Performance

The performance of the tuned I and tuned PI pressure differential LQG controllers were verified experimentally and are summarized in Figures 7.45 and 7.46. The tuned integral controller is able to track step changes in the pressure differential quite well while regulating superheat in an oscillatory manner between 11° and 13° C. By including the proportional gain in the LQG controller structure the dominant closed loop poles associated with the augmented superheat error state are moved onto the real axis. The addition of a proportional weight results in a more damped superheat response, as shown in Figure 7.46. The increased damping on the superheat response produces cleaner steps in system capacity, improving the controller's ability to match the heat load of a real system.

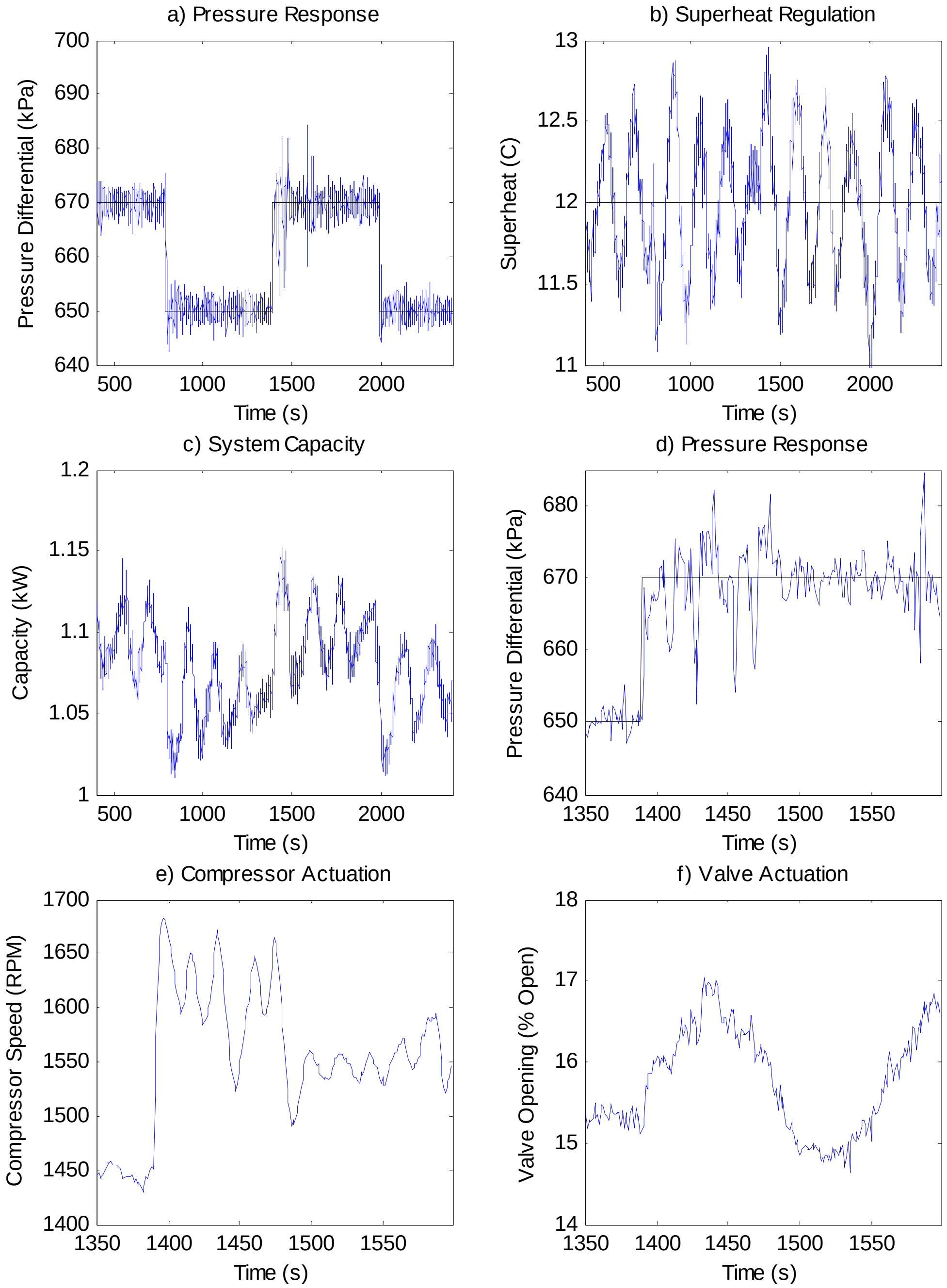


Figure 7.45 - Experimental performance of the tuned I pressure differential LQG controller

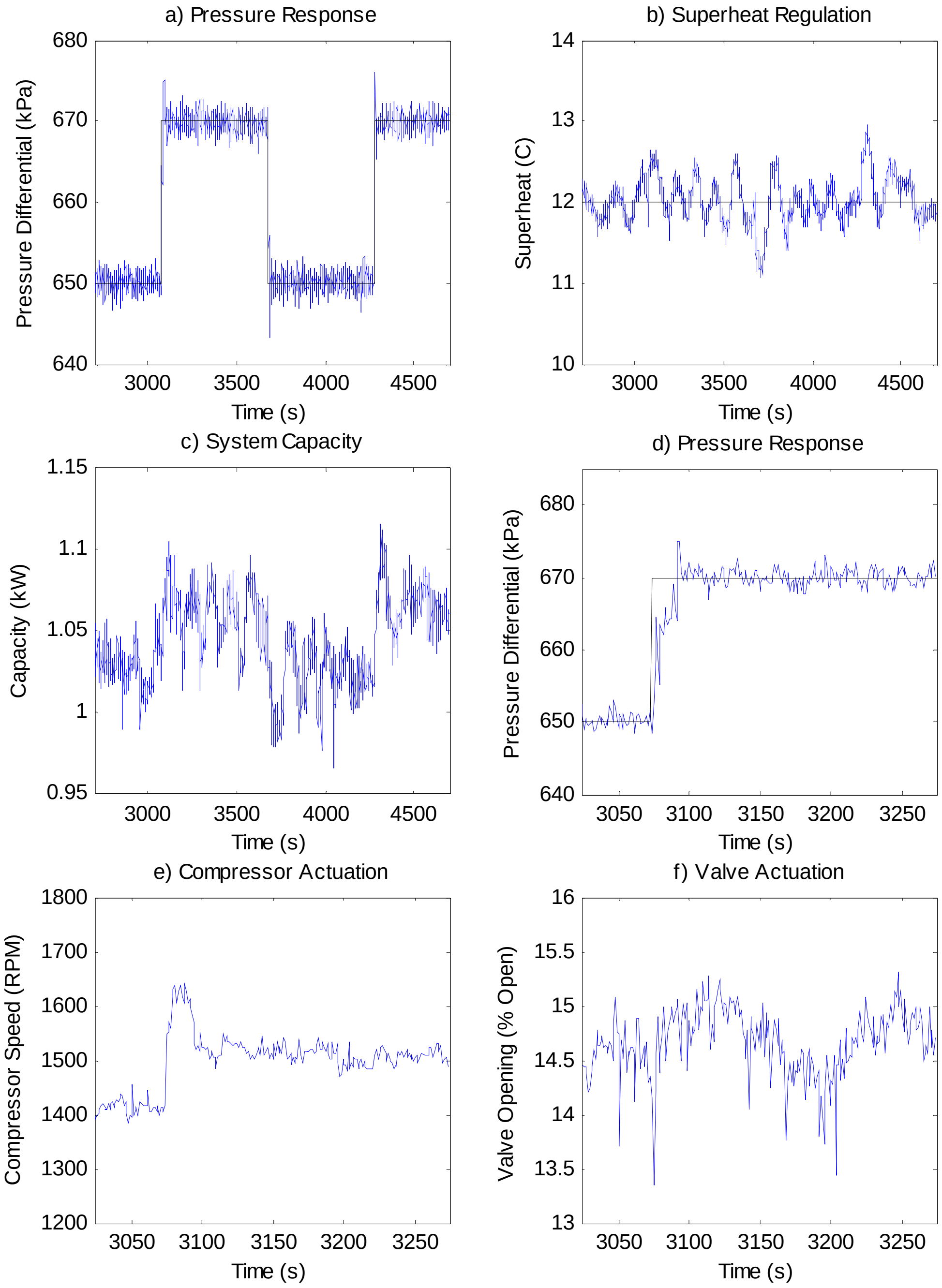


Figure 7.46 - Experimental performance of the tuned PI pressure differential LQG controller

7.6 Control Design Conclusions

As the limits of system efficiency are pressed by the conversion of AC&R systems to continuous operation configurations, it is imperative that the systems employ the proper controller architecture to effectively regulate the desired system outputs. In Section 7.2, it was shown that the performance of a decentralized control approach (SISO compressor and valve loops) will be greatly improved by selecting a feedback configuration that uses system outputs with reduced coupling. The decrease in system coupling between superheat and pressure differential vastly improves decentralized controller performance, as demonstrated in Section 7.3. The decentralized controller for the superheat/pressure differential feedback configuration obtained better output tracking because of the reduced fighting between the individual control loops, producing relatively smooth steps in system capacity.

With the addition of the static decoupler to the controller framework, detailed in Section 7.4, the step changes in pressure set point produced significantly faster changes in the system capacity. The improved response of the system capacity is depicted in part c) of Figures 7.47 and 7.48. The increased coupling of the superheat/evaporator pressure plant caused the high closed loop system bandwidth required by the performance specification to exceed the frequency range where the static decoupler would produce damped system responses. The oscillatory actuation of the decoupled superheat/evaporator pressure controller could be avoided by decreasing the desired speed of response (reduced closed loop bandwidth). The superheat/pressure differential controller performed quite well with the decoupled control architecture, producing a system response with non-oscillatory actuation and rapid steps in system capacity.

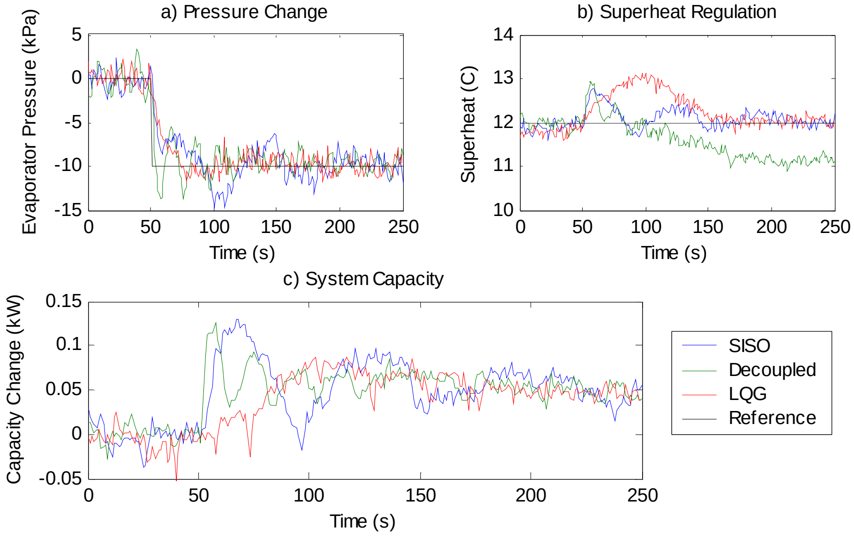


Figure 7.47 - Performance comparison for the superheat/evaporator pressure controllers

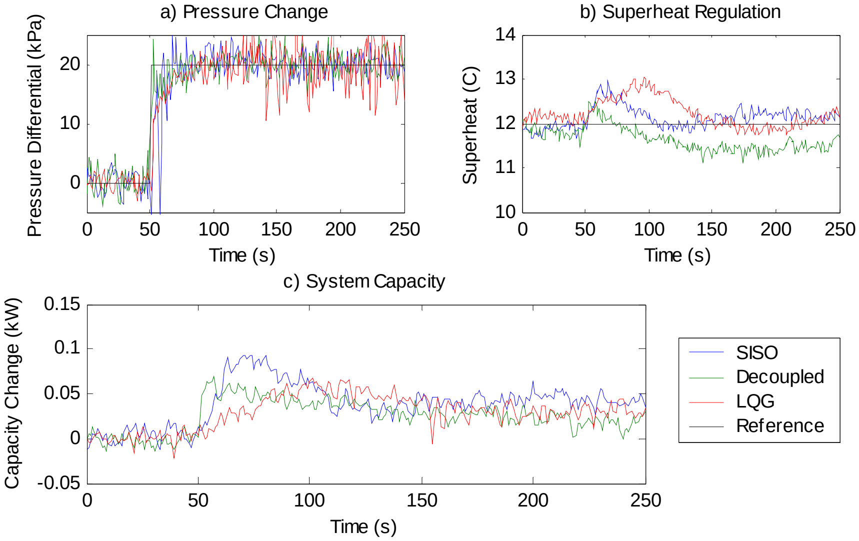


Figure 7.48 - Performance comparison for the superheat/pressure differential controllers

Another important controller characteristic that should be considered by the system designer is the level of actuation produced as the controller regulates the system. A common problem in industrial systems is an EEV failure where the EEV no longer responds to command signals. Frequent valve actuation increases the chances of EEV failures; therefore it is important that the controller minimizes oscillations in the valve movement. The controllers that were best at minimizing actuation were the LQG superheat/evaporator pressure controller and the decoupled PID superheat/pressure differential controller. By including the feedback error signal in the LQG observer design, a filtered error signal could be produced that would reduce the high frequency content of the valve and compressor actuation for both LQG controllers.

An overall review of the experimental system responses, summarized in Figures 7.47 and 7.48, provides a direct comparison between the three controller frameworks that were applied using the two feedback configurations. From Figure 7.47 the augmented superheat/evaporator pressure LQG controller from Section 7.5 is identified as the controller that created smooth steps in system capacity while effectively regulating the system to the desired pressure and superheat set-points with non-oscillatory system actuation. *Therefore, if the system designer chooses the highly coupled superheat/evaporator pressure feedback configuration for the controller, a MIMO control approach will be required to produce satisfactory system performance.*

In Figure 7.48, the performance of the LQG controller for the superheat/pressure differential plant is shown to be similar to that of the decoupled and decentralized superheat/pressure differential controllers. The LQG superheat/pressure differential controller was able to regulate superheat slightly better that the decoupled superheat/pressure differential controller over an extended series of pressure steps, but this improved regulation produced increased oscillations and slower changes in system capacity. The decentralized PID controller framework achieved comparable superheat regulation to the LQG superheat/pressure differential controller; although the oscillations in system capacity over multiple pressure steps were larger and the controller created oscillatory actuation. *In either case, the performance of the decentralized and decoupled control frameworks was quite good, implying that with a different feedback configuration high performance vapor compression system regulation can be achieved with simple controller frameworks*.

Chapter 8. Dynamic Fault Impact in Vapor Compression Systems

As was noted in Chapter 2, there is a distinct lack of information about the *dynamic changes* that occur in vapor compression systems under different fault conditions. This chapter attempts to identify a subset of scenarios where dynamic fault information could lead to more rapid fault identification, which could improve the performance of FDD algorithms in AC&R systems. The first section of this chapter discusses faults on the air side of the vapor compression system, and presents the simulated impact of frost formation, and a discussion of fan failures. The second section explores the impact of refrigerant leaks, and provides a discussion of effective means for detecting a leak condition in a system with a high side receiver. The last section of the chapter presents some conclusions and insights into the future use of dynamic FDD in vapor compression systems.

8.1 Air-Side Faults

Two fault conditions are simulated in this section: frost build up on the external surface of the evaporator, and fan failures occurring in each heat exchanger. Each fault was simulated and a discussion of the impact of the fault is presented in the subsection associated with the specific fault.

8.1.1 Frost Growth and Slow Forming Air Flow Faults

Frost growth and external particulate fowling are slow forming faults that impact the fluid flow and heat transfer properties of the external fluid. Using the dynamic frost growth model developed in Chapter 3, the Thermosys Academic toolbox was used to simulate a frost growth condition in the evaporator. To set up the frost simulation, the experimental system detailed in Chapter 5 was run at a steady state condition near the point where frost formation would occur to establish a reasonable initial condition for the model. In the simulation, the valve was closed from this initial near frosting state, bringing the wall temperature of the model below 0° C in the two phase region. This transition of the two phase wall temperature from above to below freezing begins the frosting process at the point indicated in Figure 8.1. Figure 8.2 details the properties of the growing layer of frost on the evaporator wall. The mass of the frost layer grows at a constant rate, as shown in Figure 8.3, which is in qualitative agreement with the growth behavior described by Rite [65], and Kondepudi and O'Neal [49].

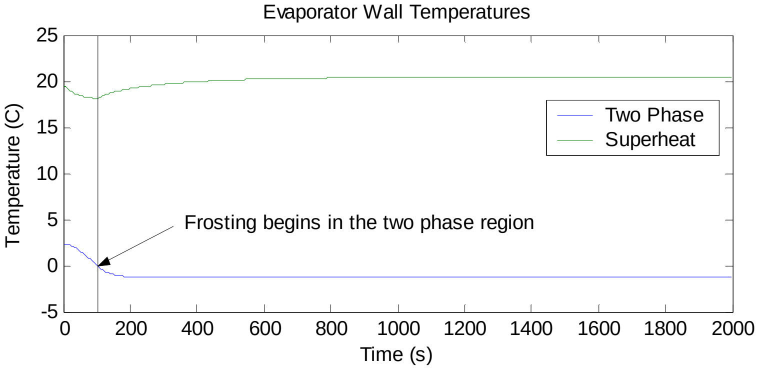


Figure 8.1 - Wall temperatures during the frost simulation

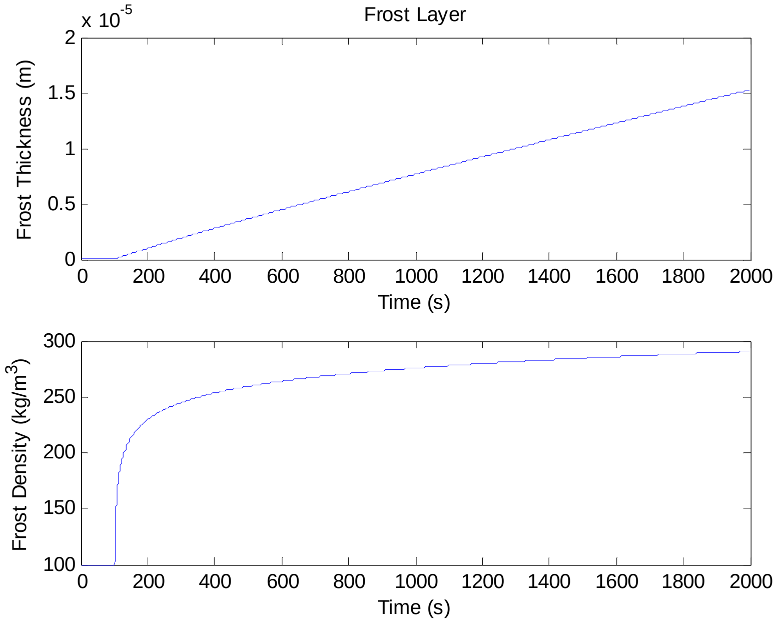


Figure 8.2 - Thickness and density of the growing frost layer on the evaporator

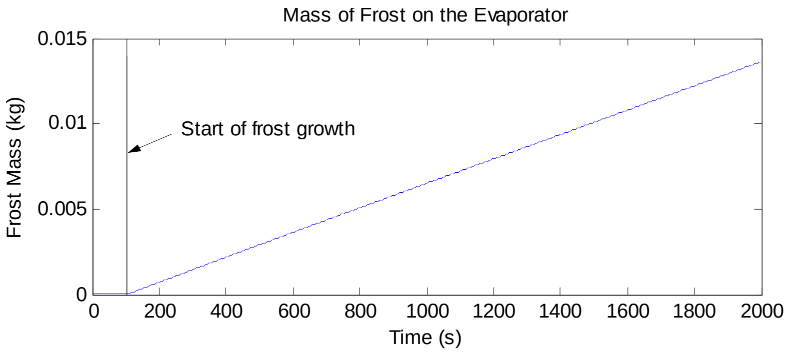


Figure 8.3 - Mass of frost that has accumulated on the evaporator

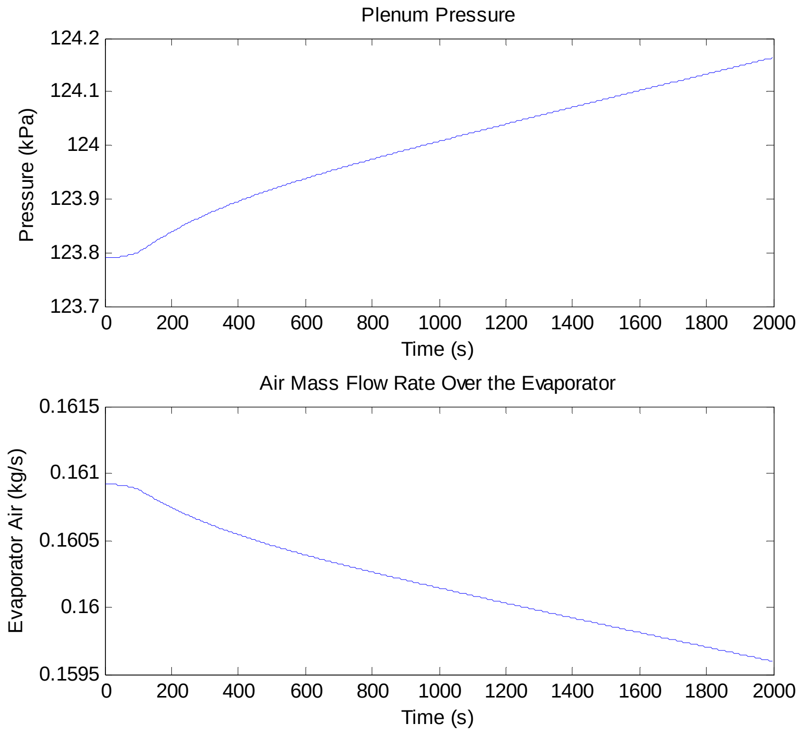


Figure 8.4 - Plenum pressure and the resulting air mass flow rate over the evaporator

The slow nature of the frost growth and particulate fouling processes produce very gradual changes in system operation. Since frost formation impacts both the flow rate (and pressure drop) of air across the heat exchanger, and the heat transfer through the wall to the refrigerant, the process will gradually change the static operating condition of the system. Simulation results and previous studies [78] have indicated that the static changes that occur will be more detectable on the air side of the heat exchanger. The reduced air mass flow rate introduced by the frost will cause the temperature drop of the air over the evaporator to increase, and is perhaps the best indicator of this fault in controlled systems (systems with an EEV or TXV).

In systems that operate with on/off cycles, the start-up and shut down transients may produce information that could indicate a frosted evaporator. As an example, consider a shut down condition where the evaporator is frosted. The thermal inertia of the melting frost will slow the rate at which the evaporator equilibrates with the ambient temperature around the evaporator, producing a transient signal that could be used to detect the frost condition.

8.1.2 Fan Failures and Rapid Changes in Air Flow

Fan failures and rapid blockages produce transient system responses that could be used to detect the presence of faults faster and perhaps with greater confidence than static FDD algorithms. A rapid partial blockage of the fan opening, perhaps caused by some obstruction such as leaves being pulled onto the air intake, would produce a system transient at the time of the fault. The transient would prevent static FDD algorithms from detecting the fault and, in some cases, after the transient the fault could potentially remain undetected if the system was unable to recover to a steady state condition.

As an example, different decreases in the air mass flow rate were introduced into the experimental system detailed in Chapter 5. The system was run at a constant compressor speed using an EEV with a PID controller regulating the superheat of the evaporator. Once the system attained a steady state, a step decrease in fan speed was applied introducing a transient into the system. The resulting responses for evaporator and condenser fan air flow reductions are summarized in Figures 8.5 and 8.6.

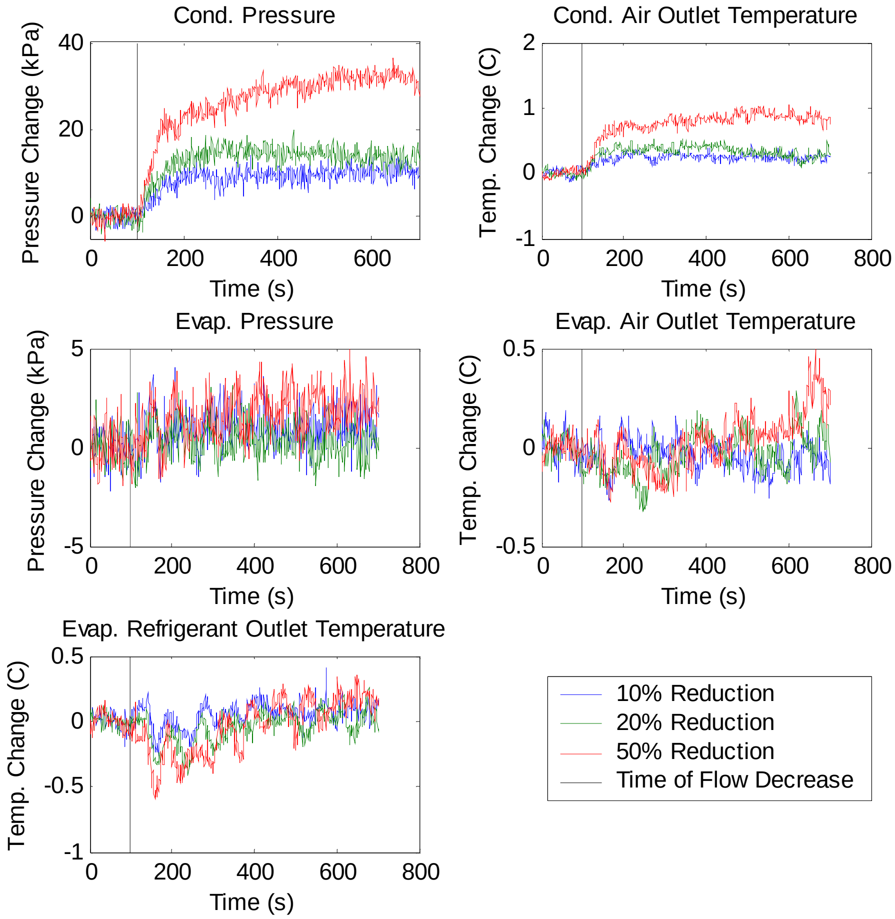


Figure 8.5 - Experimental response to a rapid decrease in the air mass flow rate over the condenser

The system response shown in Figure 8.5 indicates that the condenser pressure and condenser air outlet temperature are the only two signals that will respond significantly to a sudden reduction of air flow in the condenser. For a system operating at steady state, with all other inputs remaining fixed, the sudden increase in condenser pressure and air outlet temperature would indicate the presence of the air flow fault. It is clear that the system retains the steady state offset caused by the reduction in air flow, although the steady state detector may inhibit the algorithm from correctly identifying the fault until the system has attained a new steady state under the reduced air flow conditions.

Similar indicators are found when the evaporator air mass flow rate is rapidly reduced. In this case a large enough reduction in evaporator air mass flow rate introduces instability into the valve controller as well as alters the system state, as can be seen in Figure 8.6. The evaporator outputs (evaporator pressure, air outlet temperature, and refrigerant outlet temperature) all respond significantly to the air flow fault, distinguishing it from the condenser air flow fault. The instability introduced in the superheat controller can prevent the system from attaining a steady state for an extended period of time. This implies that a severe enough evaporator air mass flow fault may cause persistent oscillations in the system due to instability in the superheat controller, preventing a steady state detector from ever allowing a static FDD algorithm to diagnose the fault.

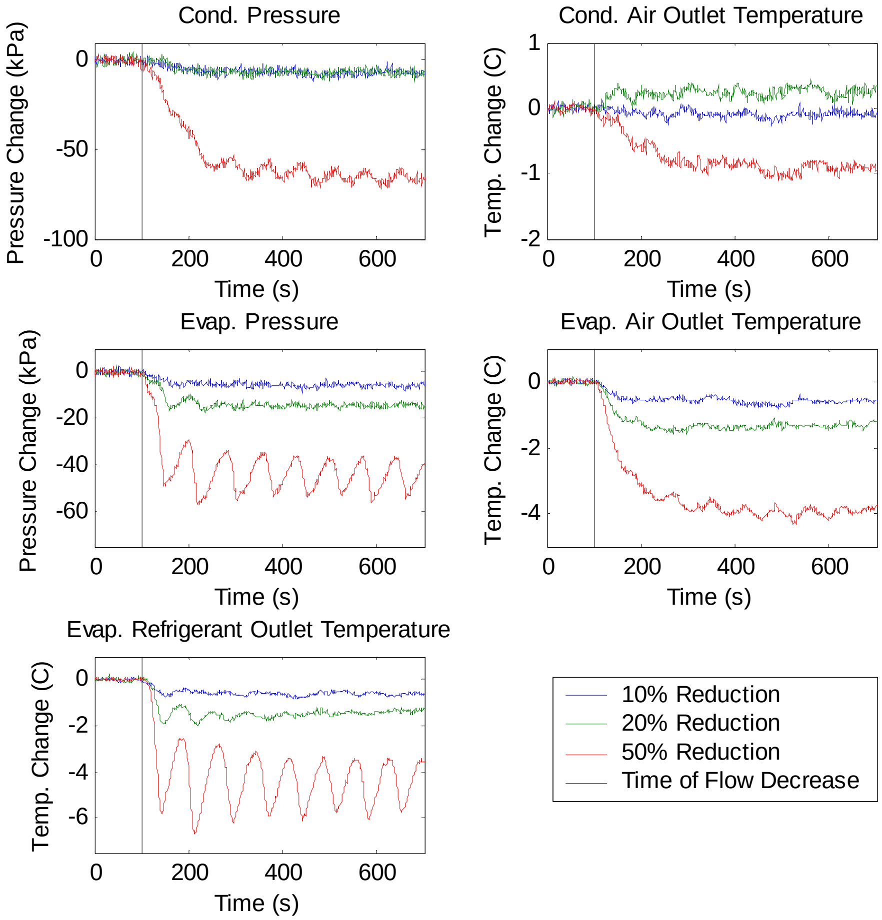


Figure 8.6 - Experimental response to a rapid decrease in the air mass flow rate over the evaporator

8.2 Refrigerant Leaks

The refrigerant leak modeling framework of Chapter 3 was used to simulate leaks at various locations in a vapor compression system. In each of the leak trials, a constant leak was introduced into the system 100 seconds into the simulation. In reality, most leaks in refrigeration systems are extremely slow, and it would take months or years for the refrigerant to fully leak out of the system. This results in prohibitively long simulations, even when a variable step solver is used; therefore leaks of 0.5 g/s and 0.1 g/s were simulated in order to identify trends in system behavior under two significant leak conditions.

For each leak magnitude, the leak was simulated individually in the 4 different pipes between the system components. The leak location was varied to identify any distinguishing characteristics in the resulting system response that could be used to indicate the location of the leak. Figures 8.7-8.11 show the response of a number of significant system outputs when a 0.5 g/s leak forms in the system 100 seconds into the simulation. The simulated leak was allowed to progress until the modeling assumptions were violated. In this case, the mass of refrigerant in the receiver went below the minimum allowable value (the refrigerant mass that would be in the receiver when it is completely filled with saturated vapor at the pressure of the condenser), and therefore the simulation was terminated. The magnitude of the leak is significant enough to cause a system transient, changing the steady state operating condition of the system. The shape of the transient is dependent on the location of the fault. Leaks on the low pressure side of the system seem to introduce larger transients in all of the system outputs presented in Figures 8.7-8.11. After the initial transient the refrigerant mass that is leaked by the system is effectively taken directly from the receiver at the outlet of the condenser.

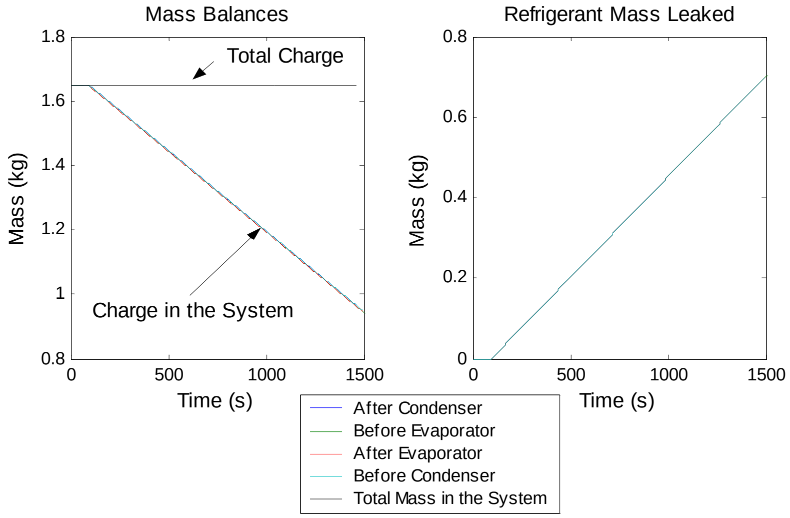


Figure 8.7 - System charge with a 0.5 g/s leak

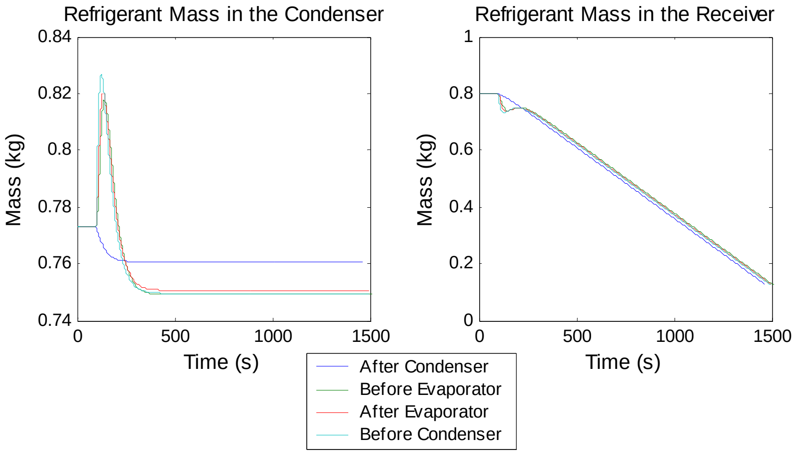


Figure 8.8 - Charge location in the condenser and receiver model with a 0.5 g/s leak

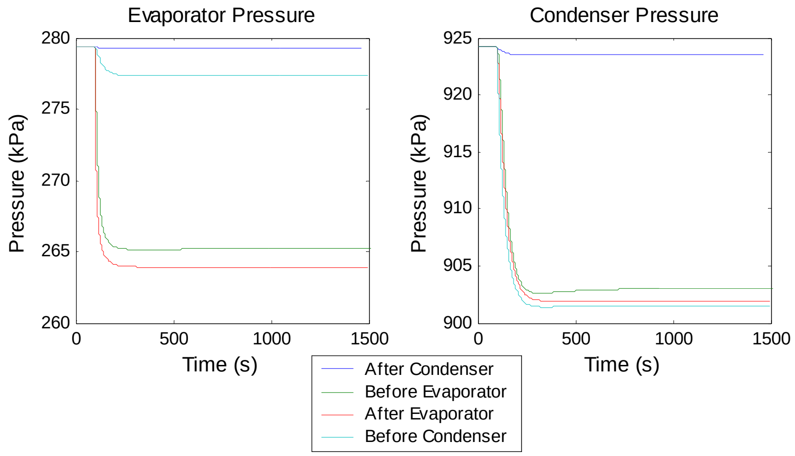


Figure 8.9 - System pressure response when a 0.5 g/s leak forms in the system

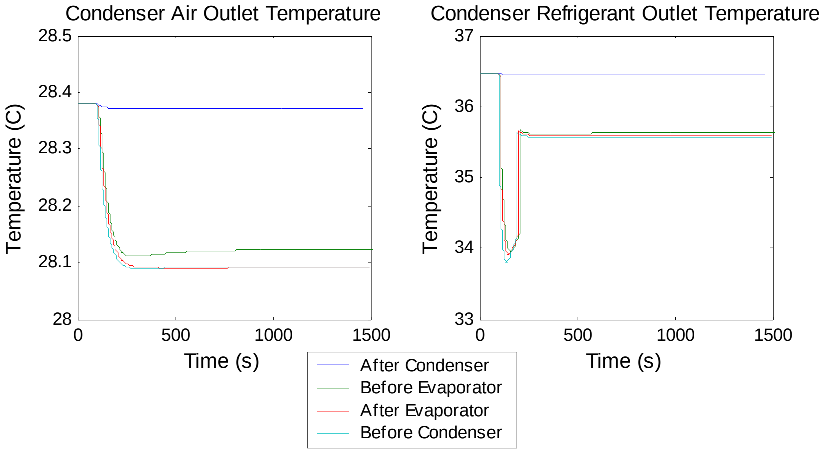


Figure 8.10 - Condenser temperature responses from a 0.5 g/s leak

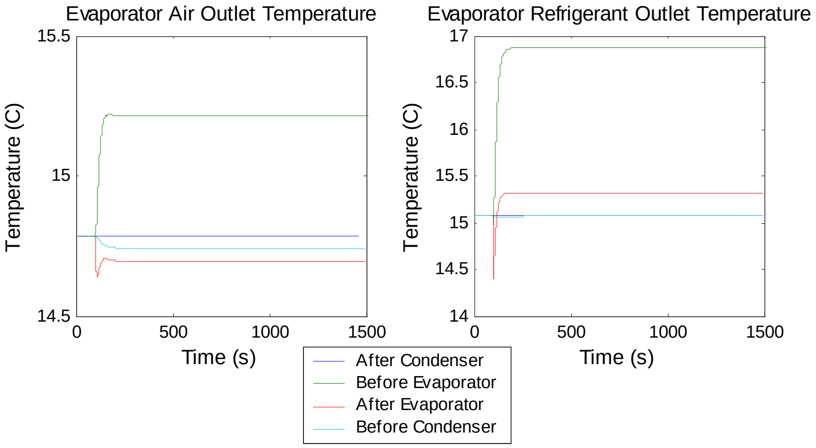


Figure 8.11 - Evaporator temperature responses from a 0.5 g/s leak

The transient introduced into the system is highly dependent on the magnitude of the leak. Figures 8.12-8.16 present the response of the system when a 0.1 g/s leak is introduced into the four potential leak locations. As with the 0.5 g/s leak simulation, the 0.1 g/s leak simulation was allowed to progress until a modeling assumption was violated. In this trial, just as with the 0.5 g/s leak, the simulation was terminated when the mass in the receiver became less than the minimum allowable value.

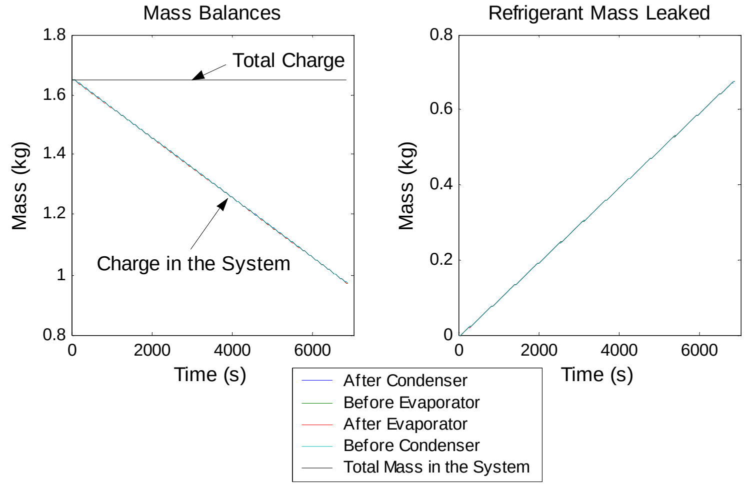


Figure 8.12 - System charge with a 0.1 g/s leak

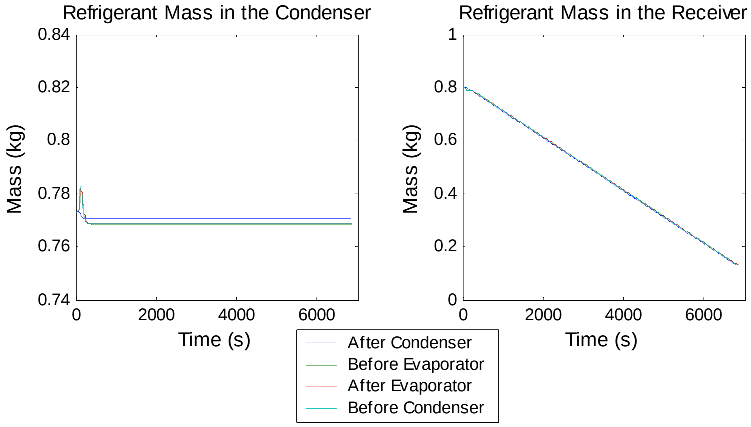


Figure 8.13 - Charge location in the condenser and receiver model with a 0.1 g/s leak

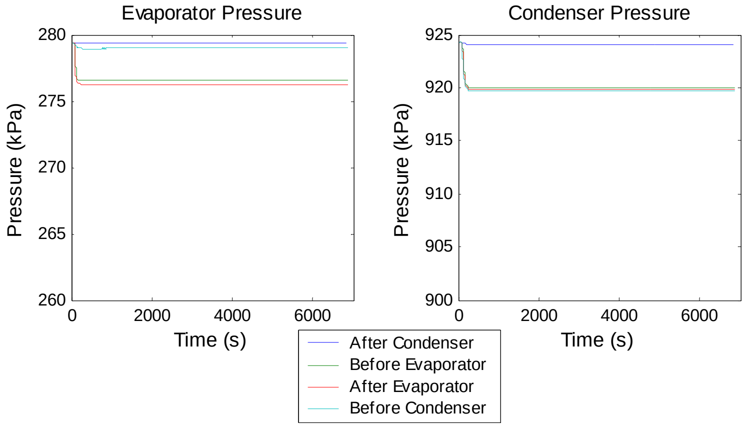


Figure 8.14 - System pressure response when a 0.1 g/s leak forms in the system

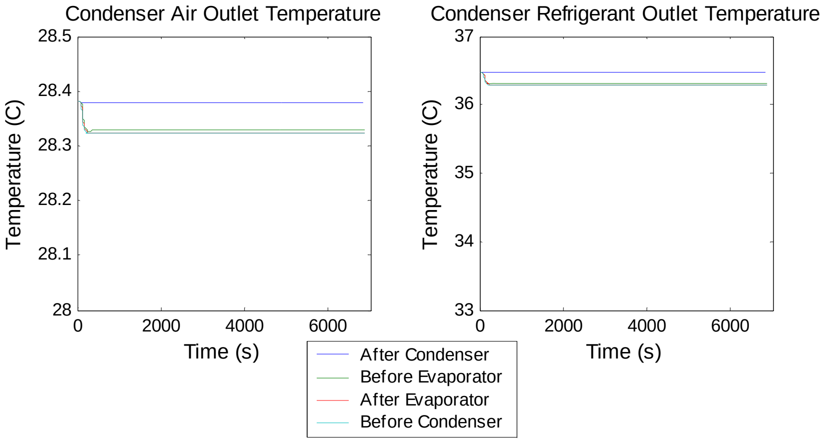


Figure 8.15 - Condenser temperature responses from a 0.1 g/s leak

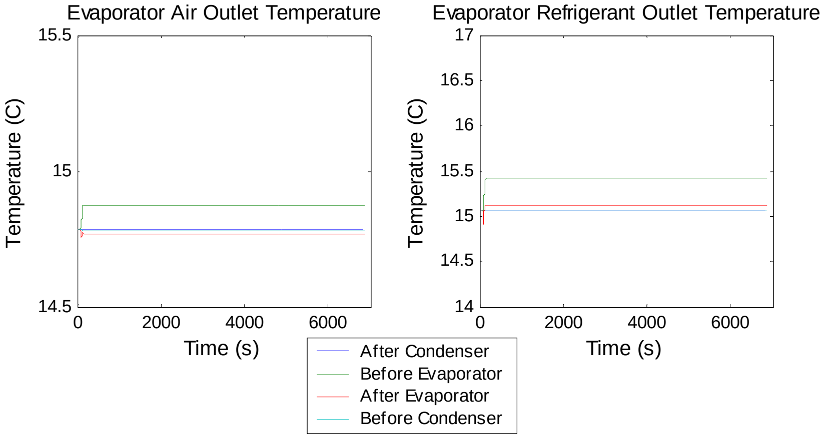


Figure 8.16 - Evaporator temperature responses from a 0.1 g/s leak

The magnitude of the system transients and resulting steady state operating condition introduced by the 0.1 g/s leak are significantly less than those induced by the 0.5 g/s leak. If the leak became comparable in size to a leak that would empty the system over a period of months or years, the transient that would form when the leak began would be so small that the change in system operating condition would be undetectable. This result is specific to a system that contains a liquid receiver at the condenser outlet, since the receiver is acting a buffer preventing the steady state system behavior from deviating until the receiver has been emptied.

For a system with a high side receiver that is charged to a point where the receiver is not completely full of refrigerant, the simulation results indicate that there will be limited static information to indicate a slow leak until the receiver of the system has been emptied. Yet the mass of refrigerant in the receiver should impact the transient response of the condenser, since the capacitance of the liquid in the receiver will slow the response of the condenser to a change in system set-point.

Using the trajectory sensitivity framework presented in Chapter 6, the impact of small deviations in the mass of refrigerant in the receiver on system outputs was explored. The sensitivity analysis was used to indicate any transient changes that should appear in the system that could be used to detect a slow refrigerant leak. As in Chapter 6, the outputs of the trajectory sensitivity analysis are scaled by the noise in the signal, to identify the signals with detectable information that would indicate the fault. For the analysis, $\delta\alpha$ was assumed to be -0.1 kg (a reduction of the mass in the receiver by 0.1 kg). Figure 8.17 presents the scaled output deviation introduced when the perturbed system is excited with a valve step. Figure 8.17 indicates that only transient changes are introduced by the reduction in receiver mass inventory, and that the condenser pressure is the richest signal from the experimental system detailed in Chapter 5 that could be used to detect a slow refrigerant leak. A close inspection of the y-axis in Figure 8.17 reveals that relatively small changes in transient behavior are introduced as the refrigerant mass in the receiver decreases, indicating that a significant loss of refrigerant will be required to accurately diagnose the leak condition by changes in the transient response.

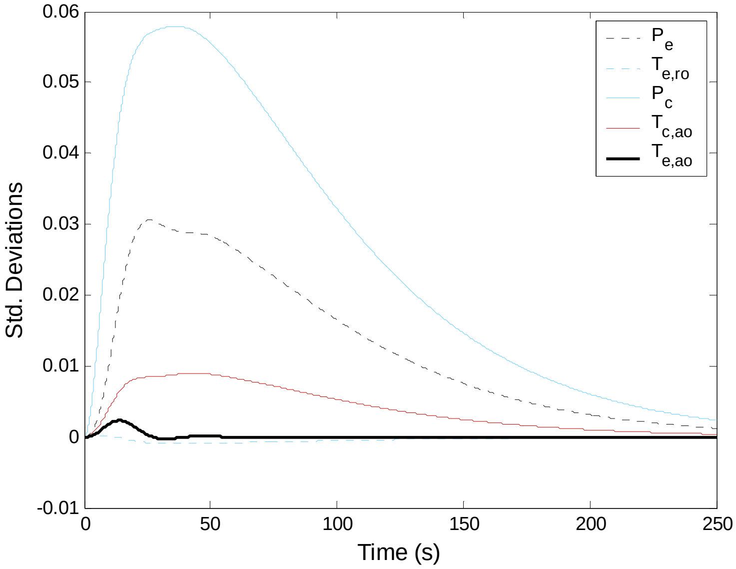


Figure 8.17 - Trajectory sensitivity analysis of the mass of refrigerant in the receiver

Figure 8.18 details the change in transient response of the system when the initial quality of the refrigerant in the receiver is varied from 0.05 (almost all liquid, 1.5 kg in the receiver) to 0.64 (0.2 kg in the receiver). It is clear that the change in refrigerant mass in the receiver does not impact the static response of the simulation, since for both conditions the same steady state point is achieved after the transients introduced by the system actuation have settled out. However, there are some distinct differences in the transient response of the condenser pressure and air outlet temperature. These transient differences would be the only indicator of a very slow system leak until the receiver had emptied and static fault information would again become available.

If a system with a high side receiver is initially charged to a point where the receiver is completely full of refrigerant, creating a subcool region in the condenser, then there would be some initial static fault information that could be used to detect a slow refrigerant leak. As the leak progressed the subcool region in the condenser would gradually disappear, and could be used as an early static indicator of a slow system leak. The static change in subcool would appear prior to the condition where only transient information would exist, although the subcool region may be more responsive to changes in ambient condition and therefore less reliable for indicating the leak.

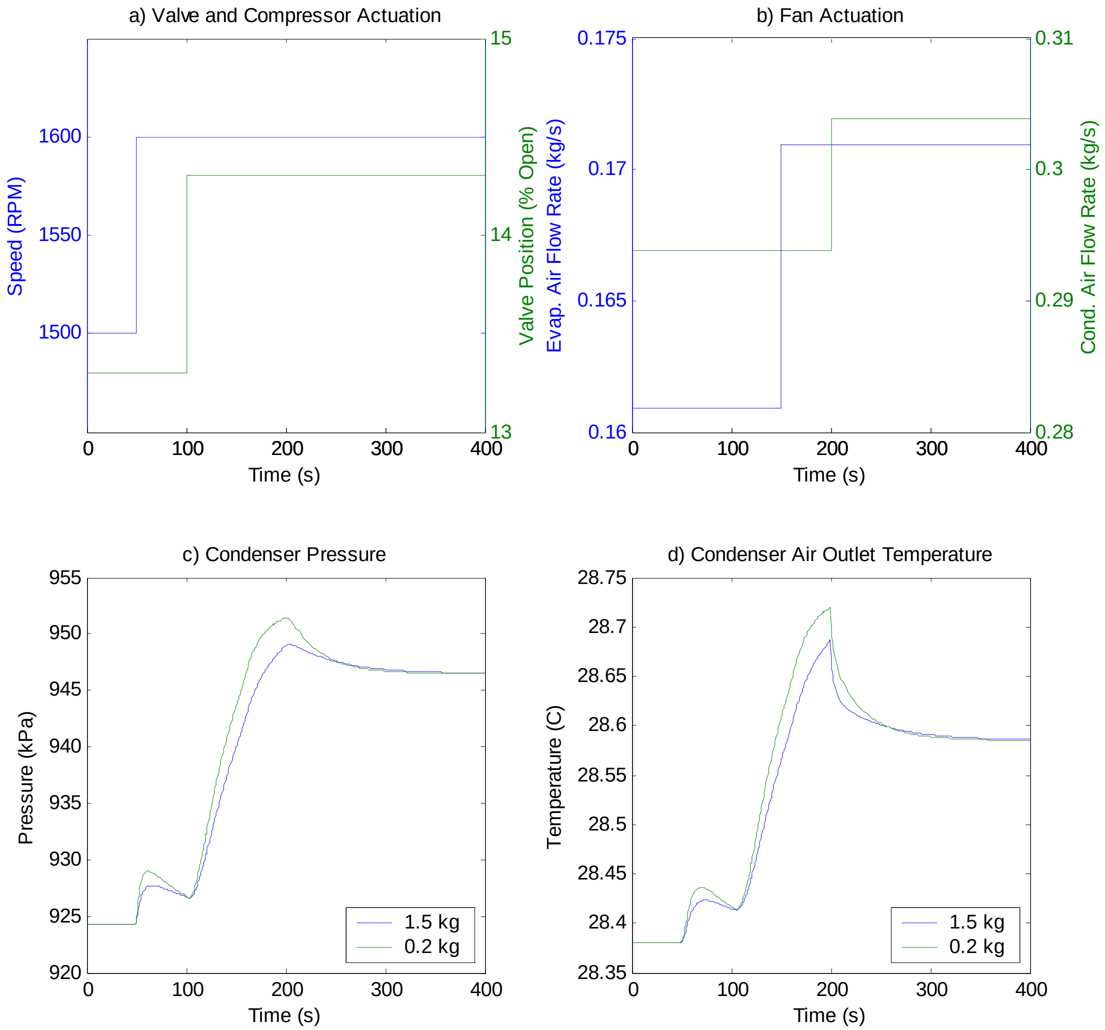


Figure 8.18 - Simulated change in the transient response of condenser pressure and air outlet temperature when the mass in the receiver is decreased from 1.5 kg to 0.2 kg

8.3 Conclusions

There are still many unanswered questions about the future role of dynamic model based FDD in vapor compression systems. This chapter has highlighted a few fault conditions and provided a basic exploration of the impact that air side faults and refrigerant leaks have on the dynamic behavior of a vapor compression system. The frost model has yet to be validated with experimental data, but the preliminary results agree qualitatively with previous results, producing a reasonable level of confidence in the model. The frost model could easily be expanded to other heat exchanger models in the Thermosys framework, and if a particulate accumulation rate was known, the frost model could be adapted to simulate external heat exchanger fouling.

It was also shown in Section 8.2 that receivers tend to mask the steady-state changes introduced by a slow refrigerant leak, effectively protecting the system from leaks until a large amount of refrigerant has escaped from the system. Although the receiver masks steady state changes, the change in the fluid capacitance within the receiver impacts the transient response of the condenser. This provides a unique condition where dynamic FDD has a distinct advantage over steady state approaches.

Another scenario where dynamic fault detection could indicate a fault earlier than steady state approaches is the case of EEV failure. If the expansion valve becomes stuck in a fixed position the input-output behavior of the valve command to superheat will change drastically. Using appropriately designed adaptive parameter estimation techniques, this change in the input-output model could be detected in a rather short time, enabling the rapid identification of a valve actuation fault.

Chapter 9. Conclusions and Future Work

9.1 Summary of Research Contributions

This thesis makes contributions to two distinct areas of vapor compression system research. First, an analysis of coupling between the outputs used in the controller feedback revealed an alternative feedback configuration that attains high performance with simple controller frameworks. Second, this thesis provides an initial analysis of the potential use of dynamic model based FDD algorithms in vapor compression systems and explores the impact a subset of faults has on the behavior of an experimental refrigeration system.

9.1.1 Control of Vapor Compression Systems

A significant contribution of this thesis is the analysis of the impact of output coupling on the performance of various control approaches in vapor compression systems. It was shown that controllers could be designed in simulation for a particular system operating condition using an identified system model. The performance of decentralized PID, statically decoupled PID, and augmented LQG controllers were compared for two potential feedback configurations. It was determined that a highly coupled feedback configuration motivates the use of advanced control techniques to achieve high performance in AC&R systems. If a relatively uncoupled output feedback configuration can be realized, then relatively simple control frameworks such as decentralized PID or statically decoupled PID controllers will achieve performance that is comparable to a more advanced multivariable approach.

9.1.2 Dynamic Fault Detection in Vapor Compression Systems

The other major contribution of this thesis was an initial exploration of the potential use of dynamic model based FDD algorithms in vapor compression systems. The dynamic modeling framework included in the Thermosys Toolbox was used with modified classical sensitivity techniques to develop a means to identify fault sensitive signals within a vapor compression system. The inclusion of various system faults into the Thermosys framework was discussed, and both refrigerant leak and frost formation faults were simulated in the dynamic modeling framework. The experimental test stand was also used to further the understanding of fault impact on a physical system. The inclusion of certain system components, such as a superheat regulating valve or a high side receiver, was shown to significantly increase the robustness of the system to the presence of different fault conditions. The sensitivity analysis and the simulated fault behavior revealed that the high side receiver will effectively mask the presence of a slow refrigerant leak if only steady state system behavior is considered. Also, it was demonstrated that slow forming air side faults tend to introduce larger static changes into the system operating condition than dynamic changes in the input-output response. Rapid forming air-side faults, such as fan failures, were shown to produce both static and dynamic characteristics that could be used to detect the fault.

9.2 Future Work

There are many areas and aspects of control and dynamic FDD that should be explored in vapor compression systems. The frost model developed in this thesis could be adapted to simulate the build up of particulate fouling on the external surface of the heat exchanger, adding another fault model to the Thermosys Toolbox. The frost model also could be more thoroughly validated using data from [65], although the model would have to be adapted to a flooded evaporator condition as was used in [65].

The benefits of a model based FDD approach should continue to be explored. An algorithm that could be adapted as components in the system were varied would eliminate the time consuming system identification process that is used in the vast majority of vapor compression system FDD algorithms. There is also still significant room for improvement of the control strategies in vapor compression systems. The controllers developed in this thesis were only designed for a single operating condition. The nonlinear nature of vapor compression systems motivates the use of a gain scheduled approach that will provide high performance across the wide range of operating conditions common to a commercial vapor compression system.

List of References

[1] "EASY5 Multiphase Fluid Library," http://www.mscsoftware.com/products/products_detail.cfm?PI=492, 2005.

[2] Anand, G., Mahajan, M., Jain, N., Maniam, B., and Tumas, T. M., "e-Thermal: Automobile Air Conditioning Module," *Society of Automotive Engineers 2004 World Congress,* Detroit, Michigan, 2004.

[3] Anderson, B. D. O. and Moore, J. B. *Optimal Control: Linear Quadratic Methods*, Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1990.

[4] Armstrong, P., "Model Identification with Application to Building Control and Fault Detection," Phd. Dissertation, Massachusetts Institute of Technology, 2004.

[5] Astrom, K. J., Johansson, K. H., and Wang, Q. G., "Design of decoupled PID controllers for MIMO systems," *2001 American Control Conference,* pp. 2015-2020, June, 2001.

[6] Bailey, M. B., "System performance characteristics of a helical rotary screw air-cooled chiller operating over a range of refrigerant charge conditions," *Proceedings of the 1998 ASHRAE Annual Meeting,* pp. 274-285, June, 1998.

[7] Bailey, M. B. and Kreider, J. F., "Creating an automated chiller fault detection and diagnostics tool using a data fault library," *ISA Transactions*, vol. 42, pp. 485-95, 2003.

[8] Bejan, A., Vargas, J. V. C., and Lim, J. S., "When to defrost a refrigerator, and when to remove the scale from the heat exchanger of a power plant," *International Journal of Heat and Mass Transfer*, vol. 37, no. 3, pp. 523-532, 1994.

[9] Bendapudi, S. and Braun, J. E., "A Review of Literature on Dynamic Models of Vapor Compression Equipment," ASHRAE Report #4036-5, May 2002.

[10] Braun, J. E., "Automated fault detection and diagnostics for vapor compression cooling equipment," *Transactions of the ASME, Journal of Solar Energy Engineering*, vol. 125, pp. 266-74, 2003.

[11] Breuker, M., Rossi, T., and Braun, J. E., "Smart maintenance for rooftop units," *ASHRAE Journal*, vol. 42, pp. 41-42, 2000.

[12] Breuker, M. S. and Braun, J. E., "Common faults and their impacts for rooftop air conditioners," *Int. J. HVAC&R Research*, vol. 4, pp. 401-425, 1998.

[13] Breuker, M. S. and Braun, J. E., "Evaluating the performance of a fault detection and diagnostic system for vapor compression equipment," *HVAC&R Research*, vol. 4, no. 4, pp. 401-425, 1998.

[14] Bristol, E. H., "On a new measure of interaction for multi-variable process control" *IEEE Transactions on Automatic Control*, vol. 11, no. 1, pp. 133-134, 1966.

[15] Castro, N. S. and Remington, G., "Performance evaluation of a reciprocating chiller using experimental data and model predictions for fault detection and diagnosis," *2002 ASHRAE Winter Meeting,* pp. 889-903, Jan., 2002.

[16] Chen, B. and Braun, J. E., "Simple rule-based methods for fault detection and diagnostics applied to packaged air conditioners," *ASHRAE Transactions*, vol. 107(1), pp. 847-857, 2001.

[17] Chen, J. and Patton, R. J., *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Norwell, MA: Kluwer Academic Publishers, 1999.

[18] Comstock, M. C., Braun, J. E., Groll, E. A., and Danks, R., "A survey of common faults for chillers," *2002 ASHRAE Winter Meeting,* pp. 819-825, Jan., 2002.

[19] Cruz, J. B., *Feedback Systems*, New York: McGraw-Hill, 1972.

[20] Cullimore, B. A. and Hendricks, T. J., "Design and Transient Simulation of Vehicle Air Conditioning Systems," *Society of Automotive Engineers World Congress,* Detroit, Michigan, 2004.

[21] Eckert, E. and Drake, R., *Analysis of Heat and Mass Transfer*, New York: McGraw-Hill, 1972.

[22] Eldredge, B., “Improving the Accuracy and Scope of Control-Oriented Vapor Compression Cycle System Models,” Dept. of Mechanical Engineering, University of Illinois, 2006.

[23] Epstein, N., "Fouling in heat exchangers," *Sixth International Heat Transfer Conference,* pp. 235-53, Aug., 1978.

[24] Farzad, M. and O'Neal, D. L., “System performance characteristics of an air conditioner over a range of charging conditions,” *International Journal of Refrigeration*, vol. 14, pp. 321-328, 1991.

[25] Farzad, M. and O'Neal, D. L., “Influence of the expansion device on air-conditioner system performance characteristics under a range of charging conditions,” *ASHRAE Transactions*, vol. 99, pp. 3-13, 1993.

[26] Frank, P. M., *Introduction to System Sensitivity Theory*, New York: Academic Press, 1978.

[27] Frank, P. M., “Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. A survey and some new results,” *Automatica*, vol. 26, pp. 459-474, 1990.

[28] Frank, P. M., “New developments using AI in fault diagnosis,” *Engng. Applic. Artif. Intell.*, vol. 10, pp. 3-14, 1997.

[29] Gertler, J., *Fault Detection and Diagnosis in Engineering Systems*, New York: Marcel Dekker, 1998.

[30] Gertler, J., “Survey of model-based failure detection and isolation in complex plants,” *IEEE Control Systems Magazine*, vol. 8, pp. 3-11, 1988.

[31] Grace, I. N., Datta, D., and Tassou, S. A., “Sensitivity of refrigeration system performance to charge levels and parameters for on-line leak detection,” *Applied Thermal Engineering*, vol. 25, no. 4, pp. 557-566, 2005.

[32] Grimmelius, H. T., Woud, J. K., and Been, G., “On-line failure diagnosis for compression refrigeration plants,” *International Journal of Refrigeration*, vol. 18, no. 1, pp. 31-41, 1995.

[33] Gustafsson, F., *Adaptive Filtering and Change Detection*, New York: John Wiley and Sons, 2000.

[34] Halm-Owoo, A. K. and Suen, K. O., “Applications of fault detection and diagnostic techniques for refrigeration and air conditioning: A review of basic principles,” *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 216, no. 3, pp. 121-132, 2002.

[35] He, X. D., Asada, H., Liu, S., and Itoh, H., “Multivariable Control of Vapor Compression Systems,” *HVAC&R Research*, vol. 4, pp. 205-230, 1998.

[36] He, X., Liu, S., and Asada, H., “Multivariable Feedback Design for Regulating Vapor Compression Cycles,” *Proceedings of the American Control Conference,* pp. 4331-4335, Jun., 1995.

[37] Himmelblau, D. M., *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*, New York: Elsevier Scientific Publishing Company, 1978.

[38] Hong Chen, Thomas, L., and Besant, R. W., “Fan supplied heat exchanger fin performance under frosting conditions,” *International Journal of Refrigeration*, vol. 26, no. 1, pp. 140-149, 2003.

[39] Inatsu, H., Matsuo, H., Fujiwara, K., Yamada, K., and Nishizawa, K., "Development of refrigerant monitoring system for automotive air-conditioning system," *International Congress and Exposition,* pp. 29-39, Feb., 1992.

[40] Incropera, F. P. a. D. D. P., *Introduction to Heat Transfer*, New York: John Wiley & Sons, 2002.

[41] Isermann, R., “Process Fault Detection Based on Modeling and Estimation Methods - A Survey,” *Automatica*, vol. 20, no. 4, pp. 387-404, 1984.

[42] Isermann, R. and Balle, P., “Trends in the application of model-based fault detection and diagnosis of technical processes,” *Control Engineering Practice*, vol. 5, pp. 709-719, 1997.

[43] Katipamula, S. and Brambley, M. R., “Methods for fault detection, diagnostics, and prognostics for building systems - A review, Part I,” *HVAC and R Research*, vol. 11, pp. 3-25, 2005.

[44] Katipamula, S. and Brambley, M. R., “Methods for fault detection, diagnostics, and prognostics for building systems - A review, Part II,” *HVAC and R Research*, vol. 11, pp. 169-187, 2005.

[45] Kays, W. M. and London, A. L., *Compact Heat Exchangers*, New York: McGraw-Hill Book Company, 1984.

[46] Keir, M. C., Rasmussen, B. P., and Alleyne, A. G., "Improving Energy Efficiency in Automotive Vapor Compression Systems through Advanced Control Design," *Proc. SAE World Congress,* Detroit, MI, 2006.

[47] Kim, M. and Kim, M. S., "Performance investigation of a variable speed vapor compression system for fault detection and diagnosis," *International Journal of Refrigeration*, vol. 28, no. 4, pp. 481-488, 2005.

[48] Kondepudi, S. N. and O'Neal, D. L., "Performance of finned-tube heat exchangers under frosting conditions: I. Simulation model," *International Journal of Refrigeration*, vol. 16, no. 3, pp. 175-180, 1993.

[49] Kondepudi, S. N. and O'Neal, D. L., "Performance of finned-tube heat exchangers under frosting conditions: II. Comparison of experimental data with model," *International Journal of Refrigeration*, vol. 16, no. 3, pp. 181-184, 1993.

[50] Lebrun, J. and Bourdouxhe, J.P., "Reference Guide for Dynamic Models of HVAC Equipment" ASHRAE Project 738-TRP, Atlanta, GA, 1998.

[51] Lenger, M. J., Jacobi A.M., and Hrnjak, P. S., "Superheat Stability of an Evaporator and Thermostatic Expansion Valve" ACRC Technical Report No. 138, 1998.

[52] Li, H. and Braun, J. E., "An improved method for fault detection and diagnosis applied to packaged air conditioners," *ASHRAE Transactions,* vol. 109(2), pp. 683-692, 2003.

[53] Liu, S. and He, X., "Gain-Scheduled Control Design Using the Loop Transfer Recovery Method for Vapor Compression Systems," *Proceedings of the American Control Conference*, pp. 3326-3330, Jun., 1997.

[54] Ljung, L., *System Identification Toolbox: For Use with Matlab*, Natick, MA: The Math Works Inc., 2001.

[55] McIntosh, I. B. D., Mitchell, J. W., and Beckman, W. A., "Fault detection and diagnosis in chillers - Part I: model development and application," *ASHRAE Transactions*, vol. 106, pp. 268-282, 2000.

[56] McQuiston, F. C., "Fin Efficiency with Combined Heat and Mass Transfer," *ASHRAE Transactions*, vol. 81, no. 1, pp. 350-355, 1975.

[57] Nimmo, I., "Adequately address abnormal situation operations," *Chemical Engineering Progress*, vol. 91, pp. 36-45, 1995.

[58] Patton, R. J., Fantuzzi, C., and Simani, S., *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*, New York: Springer-Verlag, 2003.

[59] Peitsman, H. C. and Bakker, V. E., "Application of black-box models to HVAC systems for fault detection," *ASHRAE Transactions*, vol. 102, no. 1, pp. 628-640, 1996.

[60] Proctor, J. P., "Field measurements of new residential air conditioners in Phoenix, Arizona," *Proceedings of the 1997 ASHRAE Annual Meeting*, pp. 406-415, Jan., 1997.

[61] Proctor, J. P. "AC Performance Associated with AB970," *Presentation to the California Energy Comission,* http://www.energy.ca.gov/ab970_standards/documents/presentations/2000-11-28_PROCTOR.PPT, 2000.

[62] Qureshi, T. Q. and Tassou, S. A., "Variable-speed capacity control in refrigeration systems," *Applied Thermal Engineering*, vol. 16, no. 2, pp. 103-113, 1996.

[63] Rasmussen, B. P., "Dynamic Modeling and Advanced Control of Air Conditioning and Refrigeration Systems," Dept. of Mechanical Engineering, University of Illinois, 2005.

[64] Rice, C. K., "Effect of Void Fraction Correlation and Heat Flux Assumption on Refrigerant Charge Inventory Predictions," *ASHRAE Transactions*, vol. 93, pp. 341-367, 1987.

[65] Rite, R. W., "The effect of frosting on the performance of domestic refrigerator-freezer finned tube evaporator coils" Dept. of Mechanical Eng., University of Illinois at Urbana-Champaign, 1990.

[66] Rossi, T. M. and Braun, J. E., "Statistical, rule-based fault detection and diagnostic method for vapor compression air conditioners," *HVAC&R Research*, vol. 3, no. 1, pp. 19-37, 1997.

[67] Rueda, E., Tassou, S. A., and Grace, I. N., “Fault detection and diagnosis in liquid chillers,” *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 219, no. 2, pp. 117-125, 2005.

[68] Seker, D., Karatas, H., and Egrican, N., “Frost formation on fin-and-tube heat exchangers. Part I. modeling of frost formation on fin-and-tube heat exchangers,” *International Journal of Refrigeration*, vol. 27, no. 4, pp. 367-374, 2004.

[69] Shah, R., “Dynamic Modeling and Control of Single and Multi-Evaporator Subcritical Vapor Compression Systems,” Dept. of Mechanical Engineering, University of Illinois, 2002.

[70] Siegel, J. A. and Nazaroff, W. W., “Predicting particle deposition on HVAC heat exchangers,” *Atmospheric Environment*, vol. 37, no. 39-40, pp. 5587-5596, 2003.

[71] Siegel, J. A. and Wray, C. P., "An evaluation of superheat-based refrigerant charge diagnostics for residential cooling systems," *ASHRAE Transactions*, vol. 108(2), pp. 965-975, 2002.

[72] Skogestad, S. and Postlethwaite, I., *Multivariable Feedback Control*, New York: John Wiley & Sons, 1996.

[73] Stouppe, D. E. and Lau, Y. S., “Air conditioning and refrigeration equipment failures,” *National Engineer*, vol. 93, pp. 14-17, 1989.

[74] Stylianou, M., "Application of classification functions to chiller fault detection and diagnosis," *ASHRAE Transactions*, vol. 103, pp. 645-656, 1997.

[75] Stylianou, M. and Nikanpour, D., “Performance monitoring, fault detection, and diagnosis of reciprocating chillers,” *ASHRAE Transactions*, vol. 102, no. 1, pp. 615-627, 1996.

[76] Thybo, C. and Izadi-Zamanabadi, R., "Development of fault detection and diagnosis schemes for industrial refrigeration systems - Lessons learned," *2004 IEEE International Conference on Control Applications, Sep 2-4 2004,* pp. 1248-1253, 2004.

[77] Thybo, C., Izadi-Zamanabadi, R., and Niemann, H., "Toward high performance in industrial refrigeration systems," *Proceedings of the 2002 IEEE International Conference on Control Applications,* pp. 915-920, Sept., 2002.

[78] Thybo, C., Rasmussen, B. D., and Izadi-Zamanabadi, R., "Detecting Air Circulation Faults in Refrigerated Display Cabinets," *Proc. of the IIF - IIR Commission D1/B1,* Urbana, IL, 2002.

[79] Tso, C. P., Cheng, Y. C., and Lai, A. C. K., “An improved model for predicting performance of finned tube heat exchanger under frosting condition, with frost thickness variation along fin,” *Applied Thermal Engineering*, vol. 26, no. 1, pp. 111-120, 2006.

[80] Tummescheit, H. and Eborn, J., "Design of a Thermo-Hydraulic Model Library in Modelica™," *Simulation: Past, Present and Future. 12th European Simulation Multiconference,* San Diego, CA, pp. 132-136.

[81] Turaga, M., Lin, P., and Fazio, P. P., “Correlations for heat transfer and pressure drop factors for direct expansion air cooling and dehumidifying coils,” *ASHRAE Transactions*, vol. 94, pp. 616-630, 1988.

[82] Venkatasubramanian, V., Rengaswamy, R., and Kavuri, S. N., “A review of process fault detection and diagnosis. II. Qualitative models and search strategies,” *Computers & Chemical Engineering*, vol. 27, pp. 313-326, 2003.

[83] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K., “A review of process fault detection and diagnosis. III. Process history based methods,” *Computers & Chemical Engineering*, vol. 27, pp. 327-346, 2003.

[84] Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N., “A review of process fault detection and diagnosis. I. Quantitative model-based methods,” *Computers & Chemical Engineering*, vol. 27, pp. 293-311, 2003.

[85] Wagner, J. and Shoureshi, R., “Failure detection diagnostics for thermofluid systems,” *Journal of Dynamic Systems, Measurement and Control,* vol. 114, no. 4, pp. 699-706, 1992.

[86] Wagner, J. and Shoureshi, R., “Robust failure diagnostic system for thermofluid processes,” *Automatica*, vol. 28, no. 2, pp. 375-381, 1992.

[87] Wedekind, G. L., Bhatt, B. L., and Beck, B. T., “A System Mean Void Fraction Model for Predicting Various Transient Phenomena Associated With Two-Phase Evaporating and Condensing Flows,” *International Journal of Multiphase Flow*, vol. 4, pp. 97-114, 1978.

[88] Wilkie, D. F. and Perkins, W. R., “Essential Parameters in Sensitivity Analysis,” *Automatica*, vol. 5, no. 2, pp. 191-197, 1969.

[89] Willsky, A. S., “A survey of design methods for failure detection in dynamic systems,” *Automatica*, vol. 12, pp. 601-611, 1976.

[90] Wilson, M. J., Newell, T. A., and Chato, J. C., “Experimental Investigation of Void Fraction during Horizontal Flow in Larger Diameter Refrigeration Applications,” ACRC Technical Report No.140, Jul., 1998.

[91] Xiang-Dong He and Asada, H. H., "A new feedback linearization approach to advanced control of multi-unit HVAC systems," *Proceedings of 2003 American Control Conference,* pp. 2311-2316, Jun., 2003.

[92] Yao, Y., Jiang, Y., Deng, S., and Ma, Z., “A study on the performance of the airside heat exchanger under frosting in an air source heat pump water heater/chiller unit,” *International Journal of Heat and Mass Transfer*, vol. 47, no. 17-18, pp. 3745-56, 2004.

[93] Yongzhong J. and Reddy, T. A., “Characteristic physical parameter approach to modeling chillers suitable for fault detection, diagnosis, and evaluation,” *Transactions of the ASME, Journal of Solar Energy Engineering*, vol. 125, no. 3, pp. 258-65, 2003.

[94] Zhou, K., Doyle, J. C., and Glover, K., *Robust and Optimal Control*, New Jersey: Prentice Hall, 1996.

[95] Zogg, D., Shafai, E., and Geering, H. P., "A fault diagnosis system for heat pumps," *Proceedings of the 2001 IEEE International Conference on Control Applications,* pp. 70-76, Sept., 2001.

Appendix A. Combined Mass Flow/Heat Exchanger Models

In this appendix the basic mass flow devices (EEV and compressor) are linked to heat exchangers to form a single state space model. These combined models are useful for parameter sensitivity studies as they can be simply combined to form overall system models. The procedure for interconnecting two mass flow/heat exchanger models to form an overall system model is also included in this appendix. For further details on the linearization of the individual component models the reader should consult [63].

A.1 Combined Valve/Evaporator Model

This section details the derivation of the combined valve/evaporator model. The combined model form is presented in Figure A.1. The $z_{ss,1}$ term is a vector containing any outputs from the evaporator model which are not passed on to other components in the overall system model.

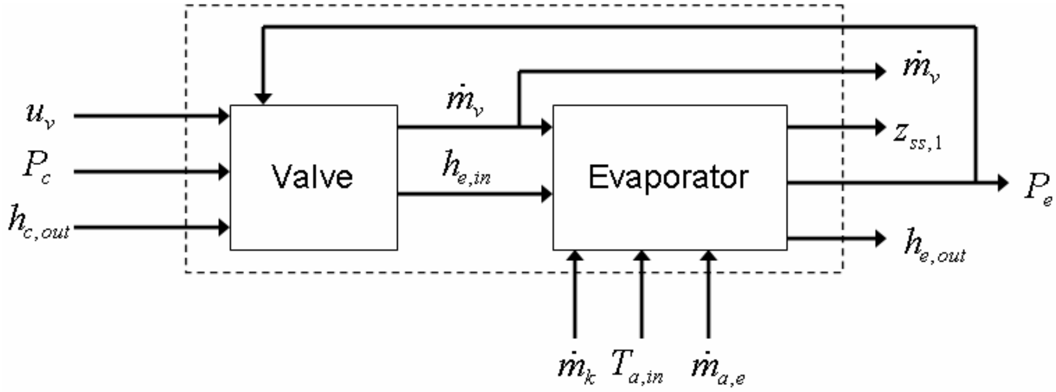


Figure A.1 - Block diagram of the combined valve and evaporator model

A.1.1 Original Component Model Equations

The governing equations of the linearized valve model are given by Equations A. through A..

$$u = \begin{bmatrix} u_v & P_{in} & P_{out} & h_{in} \end{bmatrix}^T \tag{A.1}$$

$$y = \begin{bmatrix} \dot{m}_v & h_{out} & T_{out} \end{bmatrix}^T \tag{A.2}$$

$$\frac{\partial f}{\partial u} = D = \begin{bmatrix} d_{11} & d_{12} & d_{13} & 0 \\ 0 & 0 & 0 & d_{24} \\ 0 & 0 & d_{33} & 0 \end{bmatrix} \tag{A.3}$$

$$y = Du \tag{A.4}$$

The linearized Evaporator model is given by Equations A. through A..

$$x_e = \begin{bmatrix} L_1 & P_e & h_{out} & T_{w1} & T_{w2} \end{bmatrix}^T \tag{A.5}$$

$$u_e = \begin{bmatrix} \dot{m}_{in} & \dot{m}_{out} & h_{in} & T_{a,in} & \dot{m}_a \end{bmatrix}^T \tag{A.6}$$

$$y_e = \begin{bmatrix} L_1 & P_e & h_{out} & T_{w1} & T_{w2} & T_{a,out} & T_{r,out} & T_{r,sh} & m_e \end{bmatrix}^T \tag{A.7}$$

$$Z = \begin{bmatrix} z_{11} & z_{12} & 0 & 0 & 0 \\ z_{21} & z_{22} & z_{23} & 0 & 0 \\ z_{31} & z_{32} & z_{33} & 0 & 0 \\ 0 & 0 & 0 & z_{44} & 0 \\ z_{51} & 0 & 0 & 0 & z_{55} \end{bmatrix} \tag{A.8}$$

$$\frac{\partial f}{\partial x} = F_x = \begin{bmatrix} f_{x,11} & f_{x,12} & 0 & f_{x,14} & 0 \\ f_{x,21} & f_{x,22} & f_{x,23} & 0 & f_{x,25} \\ 0 & 0 & 0 & 0 & 0 \\ f_{x,41} & f_{x,42} & 0 & f_{x,44} & f_{x,45} \\ f_{x,51} & f_{x,52} & f_{x,53} & f_{x,54} & f_{x,55} \end{bmatrix} \tag{A.9}$$

$$\frac{\partial f}{\partial u} = F_u = \begin{bmatrix} f_{u,11} & 0 & f_{u,13} & 0 & 0 \\ 0 & f_{u,22} & 0 & 0 & 0 \\ f_{u,31} & f_{u,32} & 0 & 0 & 0 \\ 0 & 0 & 0 & f_{u,44} & f_{u,45} \\ 0 & 0 & 0 & f_{u,54} & f_{u,55} \end{bmatrix} \tag{A.10}$$

$$\frac{\partial g}{\partial x} = G_x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ g_{x,61} & 0 & 0 & g_{x,64} & g_{x,65} \\ 0 & g_{x,72} & g_{x,73} & 0 & 0 \\ 0 & g_{x,82} & g_{x,83} & 0 & 0 \\ g_{x,91} & g_{x,92} & g_{x,93} & 0 & 0 \end{bmatrix} \tag{A.11}$$

$$\frac{\partial g}{\partial u} = G_u = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_{u,64} & g_{u,65} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_{u,93} & 0 & 0 \end{bmatrix} \tag{A.12}$$

$$\dot{x}_e = Z^{-1} F_x x_e + Z^{-1} F_u u_e \tag{A.13}$$

$$y_e = G_x x_e + G_u u_e \tag{A.14}$$

A.1.2 Valve/Evaporator Subsystem Equations

From Figure A.1 the overall states, inputs and outputs of the combined system model are given in Equations A. through A..

$$x_{ss1} = x_e \tag{A.15}$$

$$u_{ss1} = \begin{bmatrix} u_v & T_{a,in} & \dot{m}_{a,e} & \dot{m}_k & P_c & h_{c,out} \end{bmatrix}^T \tag{A.16}$$

$$y_{ss1} = \begin{bmatrix} P_e & h_{out} & \dot{m}_v & L_1 & T_{w1} & T_{w2} & T_{a,out} & T_{r,out} & T_{r,sh} & m_e \end{bmatrix}^T \tag{A.17}$$

The overall combined valve evaporator model will have the form presented in Equations A. and A..

$$\dot{x}_{ss1} = Z^{-1}\left(F_{x,ss1} + F_{i,ss1}\right)x_{ss1} + Z^{-1}F_{u,ss1}u_{ss1} \tag{A.18}$$

$$y_{ss1} = \left(G_{x,ss1} + G_{i,ss1}\right)x_{ss1} + G_{u,ss1}u_{ss1} \tag{A.19}$$

Where the $F_{i,ss1}$ and $G_{i,ss1}$ are terms which result from the internal feedback within the combined system of the state P_e. Carrying out the algebraic manipulations required to combine the two models, the representations of $F_{x,ss1}$, $F_{i,ss1}$, $F_{u,ss1}$, $G_{i,ss1}$, $G_{i,ss1}$, and $G_{i,ss1}$ are presented in Equations A. through A..

$$F_{x,ss1} = F_x \tag{A.20}$$

$$F_{i,ss1} = \begin{bmatrix} 0 & f_{u,11}d_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & f_{u,31}d_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.21}$$

$$F_{u,ss1} = \begin{bmatrix} f_{u,11}d_{11} & 0 & 0 & 0 & f_{u,11}d_{12} & f_{u,13}d_{24} \\ 0 & 0 & 0 & f_{u,22} & 0 & 0 \\ f_{u,31}d_{11} & 0 & 0 & f_{u,32} & f_{u,31}d_{12} & 0 \\ 0 & f_{u,44} & f_{u,45} & 0 & 0 & 0 \\ 0 & f_{u,54} & f_{u,55} & 0 & 0 & 0 \end{bmatrix} \tag{A.22}$$

$$G_{x,ss1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ g_{x,61} & 0 & 0 & g_{x,64} & g_{x,65} \\ 0 & g_{x,72} & g_{x,73} & 0 & 0 \\ 0 & g_{x,82} & g_{x,83} & 0 & 0 \\ g_{x,91} & g_{x,92} & g_{x,93} & 0 & 0 \end{bmatrix} \tag{A.23}$$

$$G_{i,ss1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & d_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.24}$$

$$G_{u,ss1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ d_{11} & 0 & 0 & 0 & d_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_{u,64} & g_{u,65} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & g_{u,93}d_{24} \end{bmatrix} \tag{A.25}$$

A.2 Combined Compressor/Condenser with Receiver Model

This section details the derivation of the combined compressor/condenser with receiver model. The combined model form is presented in Figure A.2. The $z_{ss,2}$ term is a vector containing any outputs from the condenser model which are not passed on to other components in the overall system model.

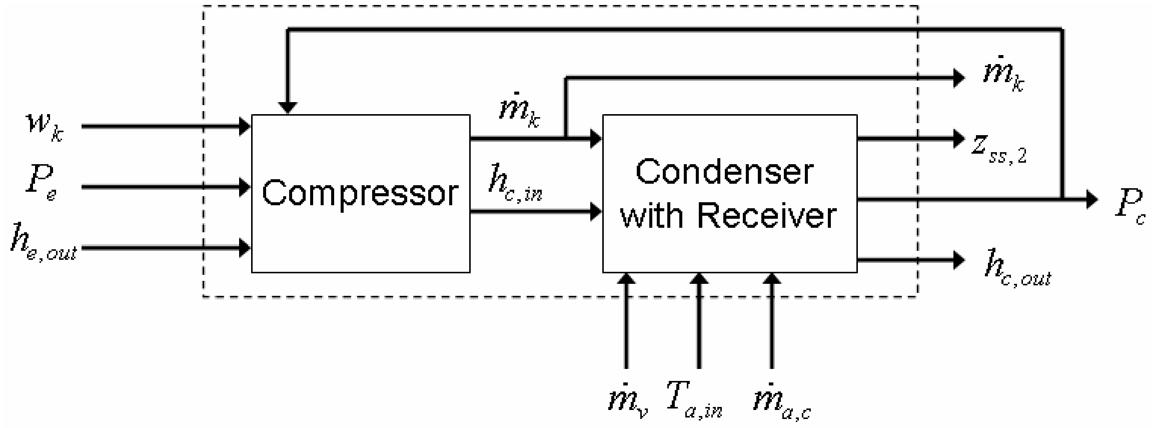


Figure A.2 - Block diagram of the combined compressor and condenser with receiver model

A.2.1 Original Component Model Equations

The equations of the original linearized compressor model are given by Equations A. through A..

$$u_k = \begin{bmatrix} \omega_k & P_{in} & P_{out} & h_{in} \end{bmatrix}^T \tag{A.26}$$

$$y_k = \begin{bmatrix} \dot{m}_k & h_{out} & T_{out} \end{bmatrix}^T \tag{A.27}$$

$$\frac{\partial f}{\partial u} = D_k = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \end{bmatrix} \tag{A.28}$$

$$y_k = D_k u_k \tag{A.29}$$

The equations for the original linearized condenser model are given in Equations A. through A..

$$x_c = \begin{bmatrix} L_1 & P_c & \bar{\gamma} & m_{rec} & T_{w1} & T_{w2} \end{bmatrix}^T \tag{A.30}$$

$$u_c = \begin{bmatrix} \dot{m}_{in} & \dot{m}_{out} & h_{in} & T_{a,in} & \dot{m}_a \end{bmatrix}^T \tag{A.31}$$

$$y_c = \begin{bmatrix} L_1 & P_c & x_{\text{int}} & h_{out} & T_{w1} & T_{w2} & T_{a,o} & T_{r,sh} & m_c & m_{rec} \end{bmatrix}^T \tag{A.32}$$

$$Z = \begin{bmatrix} z_{11} & z_{12} & 0 & 0 & 0 & 0 \\ z_{21} & z_{22} & z_{23} & z_{24} & 0 & 0 \\ z_{31} & z_{32} & z_{33} & z_{34} & 0 & 0 \\ 0 & z_{42} & 0 & z_{44} & 0 & 0 \\ z_{51} & 0 & 0 & 0 & z_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & z_{66} \end{bmatrix} \tag{A.33}$$

$$\frac{\partial f}{\partial x} = F_x = \begin{bmatrix} f_{x,11} & f_{x,12} & 0 & 0 & f_{x,15} & 0 \\ f_{x,21} & f_{x,22} & f_{x,23} & 0 & 0 & f_{x,26} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & f_{x,42} & f_{x,43} & 0 & 0 & 0 \\ f_{x,51} & f_{x,52} & 0 & 0 & f_{x,55} & f_{x,56} \\ f_{x,61} & f_{x,62} & 0 & 0 & f_{x,65} & f_{x,66} \end{bmatrix} \tag{A.34}$$

$$\frac{\partial f}{\partial u} = F_u = \begin{bmatrix} f_{u,11} & 0 & f_{u,11} & 0 & 0 \\ 0 & f_{u,22} & 0 & 0 & 0 \\ f_{u,31} & f_{u,32} & 0 & 0 & 0 \\ 0 & f_{u,42} & 0 & 0 & 0 \\ 0 & 0 & f_{u,53} & f_{u,54} & f_{u,55} \\ 0 & 0 & 0 & f_{u,64} & f_{u,65} \end{bmatrix} \tag{A.35}$$

$$\frac{\partial g}{\partial x} = G_x = \begin{bmatrix} g_{x,11} & 0 & 0 & 0 & 0 & 0 \\ 0 & g_{x,22} & 0 & 0 & 0 & 0 \\ 0 & g_{x,32} & g_{x,33} & 0 & 0 & 0 \\ 0 & g_{x,42} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_{x,55} & 0 \\ 0 & 0 & 0 & 0 & 0 & g_{x,66} \\ g_{x,71} & 0 & 0 & 0 & g_{x,75} & g_{x,76} \\ 0 & g_{x,82} & 0 & 0 & 0 & 0 \\ g_{x,91} & g_{x,92} & g_{x,93} & 0 & 0 & 0 \\ 0 & 0 & 0 & g_{x,10,4} & 0 & 0 \end{bmatrix} \tag{A.36}$$

$$\frac{\partial g}{\partial u} = G_u = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_{u,74} & g_{u,75} \\ 0 & 0 & g_{u,83} & 0 & 0 \\ 0 & 0 & g_{u,93} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.37}$$

$$\dot{x}_c = Z^{-1}F_x x_c + Z^{-1}F_u u_c \tag{A.38}$$

$$y_c = G_x x_c + G_u u_c \tag{A.39}$$

A.2.2 Compressor/Condenser with Receiver Subsystem Equations

From Figure A.2 the overall states, inputs and outputs of the combined system model are given in Equations A. through A..

$$x_{ss2} = x_c \tag{A.40}$$

$$u_{ss2} = \begin{bmatrix} w_k & T_{a,in} & \dot{m}_{a,c} & \dot{m}_v & P_e & h_{e,out} \end{bmatrix}^T \tag{A.41}$$

$$y_{ss2} = \begin{bmatrix} P_c & h_{c,out} & \dot{m}_k & L_1 & x_{\text{int}} & T_{w1} & T_{w2} & T_{a,o} & T_{r,sh} & m_c & m_{rec} \end{bmatrix}^T \tag{A.42}$$

The overall combined valve evaporator model will have the form presented in Equations A. and A..

$$\dot{x}_{ss2} = Z^{-1}\left(F_{x,ss2} + F_{i,ss2}\right)x_{ss2} + Z^{-1}F_{u,ss2}u_{ss2} \tag{A.43}$$

$$y_{ss2} = \left(G_{x,ss2} + G_{i,ss2}\right)x_{ss2} + G_{u,ss2}u_{ss2} \tag{A.44}$$

Where the $F_{i,ss2}$ and $G_{i,ss2}$ are terms which result from the internal feedback within the combined system of the state P_c. Carrying out the algebraic manipulations required to combine the two models, the representations of $F_{x,ss2}$, $F_{i,ss2}$, $F_{u,ss2}$, $G_{i,ss2}$, $G_{i,ss2}$, and $G_{i,ss2}$ are presented in Equations A. through A..

$$F_{x,ss2} = F_x \tag{ A.45 }$$

$$F_{i,ss2} = \begin{bmatrix} 0 & f_{u,11}d_{13} + f_{u,13}d_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & f_{u,31}d_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & f_{u,53}d_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{ A.46 }$$

$$F_{u,ss2} = \begin{bmatrix} f_{u,11}d_{11} + f_{u,13}d_{21} & 0 & 0 & 0 & f_{u,11}d_{12} + f_{u,13}d_{22} & f_{u,11}d_{14} + f_{u,13}d_{24} \\ 0 & 0 & 0 & f_{u,22} & 0 & 0 \\ f_{u,31}d_{11} & 0 & 0 & f_{u,32} & f_{u,31}d_{12} & f_{u,31}d_{14} \\ 0 & 0 & 0 & f_{u,42} & 0 & 0 \\ f_{u,53}d_{21} & f_{u,54} & f_{u,55} & 0 & f_{u,53}d_{22} & f_{u,53}d_{24} \\ 0 & f_{u,64} & f_{u,65} & 0 & 0 & 0 \end{bmatrix} \tag{ A47 }$$

$$G_{x,ss2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & g_{x,42} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_{x,32} & g_{x,33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ g_{x,71} & 0 & 0 & 0 & g_{x,75} & g_{x,76} \\ 0 & g_{x,82} & 0 & 0 & 0 & 0 \\ g_{x,91} & g_{x,92} & g_{x,93} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{ A.48 }$$

$$G_{i,ss2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_{u,83}d_{23} & 0 & 0 & 0 & 0 \\ 0 & g_{u,93}d_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{ A.49 }$$

$$G_{u,ss2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ d_{11} & 0 & 0 & 0 & d_{12} & d_{14} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_{u,74} & g_{u,75} & 0 & 0 & 0 \\ g_{u,83}d_{21} & 0 & 0 & 0 & g_{u,83}d_{22} & g_{u,83}d_{24} \\ g_{u,93}d_{21} & 0 & 0 & 0 & g_{u,93}d_{22} & g_{u,93}d_{24} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (\text{A.50})$$

A.3 Total System Model

Once the two subsystem models have been generated the models can be simply connected using the Reheffer star product. The block diagram for the star product is presented in Figure A.3. For a detailed discussion of the procedure used to combine the system models the reader is referred to [94].

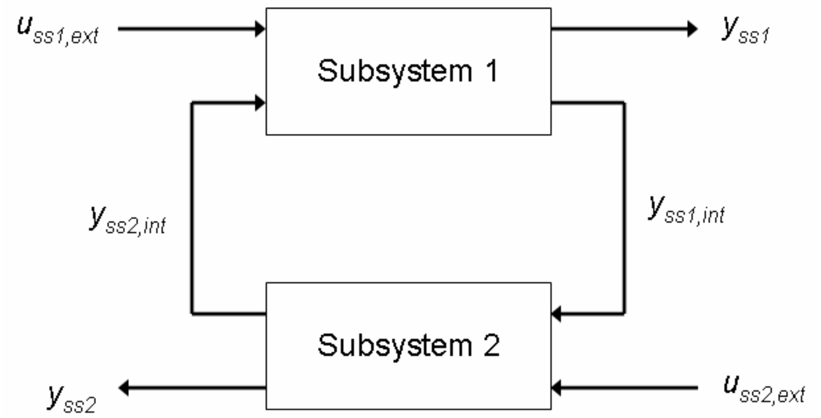


Figure A.3 - Block diagram of the star product

A.3.1 Reduced Output System Model

For simplicity the overall system model presented will not include the entire set of outputs possible from the combined subsystem models presented in Sections A.1 and A.2. Instead a subset of important parameters was chosen to illustrate the procedure. The outputs of Subsystem 1 which were included in the total model were evaporator pressure, enthalpy at the exit of the evaporator, mass flow rate through the valve, evaporator air outlet temperature, and evaporator refrigerant outlet temperature. The outputs from Subsystem 2 which were included in the total model were the condenser pressure, enthalpy at the exit of the condenser, mass flow rate through the compressor, condenser air outlet temperature, and condenser refrigerant outlet temperature.

Equations A. through A. provide the input and output vectors for Figure A.3.

$$u_{ss1,ext} = \begin{bmatrix} u_v & T_{a,in} & \dot{m}_{a,e} \end{bmatrix} \qquad (\text{A.51})$$

$$u_{ss2,ext} = \lfloor \omega_k \quad T_{a,in} \quad \dot{m}_{a,c} \rfloor \tag{A.52}$$

$$y_{ss1} = \lfloor P_e \quad h_{e,out} \quad \dot{m}_v \quad T_{ea,out} \quad T_{er,out} \rfloor \tag{A.53}$$

$$y_{ss1,\mathrm{int}} = \lfloor \dot{m}_v \quad P_e \quad h_{e,out} \rfloor \tag{A.54}$$

$$y_{ss2} = \lfloor P_c \quad h_{c,out} \quad \dot{m}_k \quad T_{ca,out} \quad T_{cr,out} \rfloor \tag{A.55}$$

$$y_{ss2,\mathrm{int}} = \lfloor \dot{m}_k \quad P_c \quad h_{c,out} \rfloor \tag{A.56}$$

Applying the format detailed in [94], the P and K transfer matrices of the star product are those given in Equations A. and A.5845.

$$P = \begin{bmatrix} A_{ss1} & B_{ss1,1} & B_{ss1,2} \\ C_{ss1,1} & D_{ss1,11} & D_{ss1,12} \\ C_{ss1,2} & D_{ss1,21} & D_{ss1,22} \end{bmatrix} \tag{A.57}$$

$$K = \begin{bmatrix} A_{ss2} & B_{ss2,1} & B_{ss2,2} \\ C_{ss2,1} & D_{ss2,11} & D_{ss2,12} \\ C_{ss2,2} & D_{ss2,21} & D_{ss2,22} \end{bmatrix} \tag{A.5845}$$

A.3.1.1 Equations for P Matrix from Subsystem 1

Following the format provided for interconnection, the resulting A, B, C, and D matrices for Subsystem 1 are given in Equations A.5946 through A.. The terms in these matrices are the same as those found in Section A.1.

$$A_{ss1} = F_{x,ss1} + F_{i,ss1} \tag{A.5946}$$

$$B_{ss1,1} = \begin{bmatrix} f_{u,11}d_{11} & 0 & 0 \\ 0 & 0 & 0 \\ f_{u,31}d_{11} & 0 & 0 \\ 0 & f_{u,44} & f_{u,45} \\ 0 & f_{u,54} & f_{u,55} \end{bmatrix} \tag{A.60}$$

$$B_{ss1,2} = \begin{bmatrix} 0 & f_{u,11}d_{12} & f_{u,13}d_{24} \\ f_{u,22} & 0 & 0 \\ f_{u,32} & f_{u,31}d_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.61}$$

$$C_{ss1,1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & d_{13} & 0 & 0 & 0 \\ g_{x,61} & 0 & 0 & g_{x,64} & g_{x,65} \\ 0 & g_{x,72} & g_{x,73} & 0 & 0 \end{bmatrix} \tag{A.62}$$

$$C_{ss1,2} = \begin{bmatrix} 0 & d_{13} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (A.63)$$

$$D_{ss1,11} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ d_{11} & 0 & 0 \\ 0 & g_{u,64} & g_{u,65} \\ 0 & 0 & 0 \end{bmatrix} \quad (A.64)$$

$$D_{ss1,12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & d_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (A.65)$$

$$D_{ss1,21} = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (A.66)$$

$$D_{ss1,22} = \begin{bmatrix} 0 & d_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (A.67)$$

A.3.1.2 Equations for K Matrix from Subsystem 2

Following the format provided for interconnection the resulting A, B, C, and D matrices for Subsystem 2 are given in Equations A. through A.. The terms in these matrices are the same as those found in Section A.2.

$$A_{ss2} = F_{x,ss2} + F_{i,ss2} \quad (A.68)$$

$$B_{ss2,1} = \begin{bmatrix} 0 & f_{u,11}d_{12} + f_{u,13}d_{22} & f_{u,11}d_{14} + f_{u,13}d_{24} \\ f_{u,22} & f_{u,21}d_{12} & f_{u,21}d_{14} \\ f_{u,32} & 0 & 0 \\ f_{u,42} & f_{u,41}d_{12} & f_{u,41}d_{14} \\ 0 & f_{u,53}d_{22} & f_{u,53}d_{24} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (A.69)$$

$$B_{ss2,2} = \begin{bmatrix} f_{u,11}d_{11} & 0 & 0 \\ f_{u,21}d_{11} & 0 & 0 \\ 0 & 0 & 0 \\ f_{u,41}d_{11} & 0 & 0 \\ 0 & f_{u,54} & f_{u,55} \\ 0 & f_{u,64} & f_{u,65} \\ 0 & f_{u,74} & f_{u,75} \end{bmatrix} \quad (\text{A.70})$$

$$C_{ss2,1} = \begin{bmatrix} 0 & 0 & d_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.71})$$

$$C_{ss2,2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & d_{13} & 0 & 0 & 0 & 0 \\ g_{x,81} & g_{x,82} & 0 & 0 & g_{x,85} & g_{x,86} & g_{x,87} \\ 0 & 0 & g_{x,93} & g_{x,94} & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.72})$$

$$D_{ss2,11} = \begin{bmatrix} 0 & d_{12} & d_{14} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.73})$$

$$D_{ss2,12} = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.74})$$

$$D_{ss2,21} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & d_{12} & d_{14} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.75})$$

$$D_{ss2,22} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ d_{11} & 0 & 0 \\ 0 & g_{u,84} & g_{u,85} \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.76})$$

A.3.2 Overall State Space System Model

The matrices provided in Subsection A.3.1 were combined to form the overall system model using Equations A. through A.. These equations are found on page 268 of [94].

$$\overline{A} = \begin{bmatrix} A_{ss1} + B_{ss1,1}\widetilde{R}^{-1}D_{ss2,11}C_{ss1,2} & B_{ss1,2}\widetilde{R}^{-1}C_{ss2,1} \\ B_{ss2,1}R^{-1}C_{ss1,2} & A_{ss2} + B_{ss2,1}R^{-1}D_{ss1,22}C_{ss2,1} \end{bmatrix} \tag{A.77}$$

$$\overline{B} = \begin{bmatrix} B_{ss1,1} + B_{ss1,2}\widetilde{R}^{-1}D_{ss2,11}D_{ss1,21} & B_{ss1,2}\widetilde{R}^{-1}D_{ss2,12} \\ B_{ss2,1}R^{-1}D_{ss1,21} & B_{ss2,2} + B_{ss2,1}R^{-1}D_{ss1,22}D_{ss2,12} \end{bmatrix} \tag{A.78}$$

$$\overline{C} = \begin{bmatrix} C_{ss1,1} + D_{ss1,12}D_{ss2,11}R^{-1}C_{ss1,2} & D_{ss1,12}\widetilde{R}^{-1}D_{ss2,12} \\ D_{ss2,21}R^{-1}C_{ss1,2} & C_{ss2,2} + D_{ss2,21}R^{-1}D_{ss1,22}C_{ss2,1} \end{bmatrix} \tag{A.79}$$

$$\overline{D} = \begin{bmatrix} D_{ss1,11} + D_{ss1,12}D_{ss2,11}R^{-1}D_{ss1,21} & D_{ss1,12}\widetilde{R}^{-1}D_{ss2,12} \\ D_{ss2,21}R^{-1}D_{ss1,21} & D_{ss2,22} + D_{ss2,21}R^{-1}D_{ss1,22}D_{ss2,12} \end{bmatrix} \tag{A.80}$$

$$R = I - D_{ss1,22}D_{ss2,11} \tag{A.81}$$

$$\widetilde{R} = I - D_{ss2,11}D_{ss1,22} \tag{A.82}$$

Appendix B. Matlab Code

This appendix contains the Matlab code that is used to create the evaporator frost growth model from Chapter 3 and the code used to generate the LQG controller in Chapter 7.

B.1 Evaporator with Frost Growth Model

There are four different m-files that are used to simulate the growth of frost on the evaporator in Thermosys Academic.

B.1.1 Main Simulation File

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  FILENAME:                EvapFrostMassTrans.m
%  WRITTEN BY:              Michael Keir
%                           University of Illinois at Urbana-Champaign
%
%  COMMENTS ON CODE STRUCTURE:  This code is a nonlinear evaporator model
%                     Inputs: 15
%                     Outputs: 26
%
%   This is a lumped parameter moving boundary model of a subcritical
%   evaporator. A detailed model derivation can be found in Michael Keir's
%   MS Thesis. The model includes the capability to model the growth of
%   frost on the surface of the evaporator. The current version of the
%   model does not allow for the frosting process to be reversed (no
%   melting)
%
%   The evaporator model has 6 states and 6 inputs. The six states are
%   length of the 2-phase region, evaporator pressure, outlet enthalpy,
%   the wall temperature of each evaporator region, and the pressure in
%   the plenum.
%
%   The 6 inputs are the refrigerant mass flow rate into the evaporator,
%   the refrigerant mass flow rate out of the evaporator, the enthalpy of
%   the refrigerant entering the evaporator, the air inlet temperature, the
%   nominal air mass flow rate over the evaporator, and the relative
%   humidity of the inlet air.
%
%   The evaporator model has 17 recorded outputs:
%   Le_1    - Length of the 2-phase region in the evaporator
%   Pe      - Evaporator pressure
%   He_ro   - Enthalpy of the refrigerant exiting the evaporator
%   Te_w1   - Temperature of the wall in the 2-phase region
%   Te_w2   - Temperature of the wall in the superheat region
%   Pa      - Pressure in the plenum
%   Te_ao   - Air outlet temperature
%   Te_ro   - Refrigerant outlet temperature
%   Te_sh   - Superheat temperature
%   minv_e  - Mass of refrigerant in the evaporator
%   Phie_ao - Relative humidity of outlet air
%   me_wo   - Mass flow rate of condensed water leaving evaporator
%   me_a    - Air flow over the evaporator
%   delta_fr- Thickness of the frost layer
%   rho_fr  - Density of the frost layer
%   ma_fan  - Air flow rate into the plenum
```

```
%   Aff     - Current free flow area in the evaporator
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ORIGINAL DATE WRITTEN:                  07 June 2006
%  DATE OF LAST MODIFICATION:       07 June 2006
%  MODIFICATION HISTORY:
%  DATE:                        COMMENT:
%  07 June 2006                - Original write of the stellar program...
%
%  PROBLEMS/MODIFICATIONS FOR FUTURE WORK:
%
%
%
%
%%%%%%%%%%%%%%%%%% START OF Michael Keir's Awesome CODE

function [output] = EvapFrostMassTrans(U)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1. Load Global Parameter Structures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global Initial_Conditions
global Physical_Parameter
global Operating_Conditions
global Measured_Outputs
global FluidProp
global External_FluidProp
persistent x0 u0 y0 Y0 index ModParam j Re_j rho_fr delta_fr Prev_Time ma_e frosting_flag1 frosting_flag2
Ta_delt

OPTIONS = OPTIMSET('Display','none','LargeScale','off');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2. Break Input Vector, U, into Proper Components
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x = U(1:6);
u = U(7:12);
Time = U(end-2);
HX_num = U(end-1);
flag = U(end);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3. Evaporator Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

StartTime = str2double(get_param(bdroot(gcb),'StartTime'));

if Time == StartTime    %Only run this code once at the start of the simulation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3.1 Set Default Physical Parameters of the Evaporator
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Mass_e;       %mass of the evaporator
% Cpw_e;        %specific heat of the evaporator wall
% Ae_o;         %external surface area of the evaporator
% Diameter_e; %hydraulic diameter of the evaporator tubes
% Le_total;     %total length of tube the refrigerant passes through in the evaporator
% Total_e;      %number of fluid paths in the evaporator
% Ae_i;         %internal surface are of the evaporator
% Ae_cs;        %Cross-sectional area

%set mass block
Mass_e = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Mass_e'));
%set Cp block
Cpw_e = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Cpw_e'));
%set ExtSA Block
Ae_o = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Ae_o'));
%set hydraulic diameter Block
Diameter_e = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Diameter_e'));
%set Tube Length Block
Le_total = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Le_total'));
%set Total # of Fluid Paths Block
Total_e = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Total_e'));
%set Internal Surface Area Block
Ae_i = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Ae_i'));
%set Cross Sectional Area Block
Ae_cs = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Ae_cs'));
%set Fin Area
Af = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Af'));
%set Fin Conductivity
kfin = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.kfin'));
%set Fin Thickness
tfin = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.tfin'));
%set Transverse Tube Pitch
Pt = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Pt'));
%set Longitudinal Tube Pitch
Pl = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Pl'));
%set Tube Inside Radius
r = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.r'));
%initial free flow area
Aff = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Aff'));
%set outer tube diameter
D_o = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.D_o'));
%frontal area
FA = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.FA'));
%sigma_hx
sigma_hx = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.sigma_hx'));
%j factor hydraulic diameter
Dh = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Dh'));
%Set Plenum Volume
Vol_PL = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Vol_PL'));

%load parameters associated with j factor
if isfield(eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num))),'j_factor_flag')== 1;
   if eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.j_factor_flag')) == 1;
      %j factor data points
      j = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.j'));
      %j factor Reynolds number data points
```

```
         Re_j = eval(strcat('Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Re'));
         %set flag
         j_factor_flag = 1;
      else
         j_factor_flag = 0;
      end
   else
      j_factor_flag = 0;
      disp('Warning: Frost model uses the heat transfer coefficient to calculate the mass transfer of water')
      disp('from the air. Using a j-factor based external heat transfer coefficient is recommended.')
   end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3.2 Define Default Initial Operating Conditions for the Evaporator
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   % me_air;      %mass flow rate of external fluid at the evaporator
   % mdot_e;      %mass flow rate of refrigerant through the evaporator
   % Pe;          %pressure of the refrigerant in the evaporator
   % He_ri;       %enthalpy of the refrigerant at the evaporator inlet
   % Te_ro;       %temperature of the refrigerant at the evaporator outlet
   % Te_ai;       %temperature of the air at the evaporator inlet
   % Te_ao;       %temperature of the air at the evaporator outlet
   % Slip_e;      %slip ratio in the evaporator
   % Phie_ai;     %inlet air relative humidity
   % P_air;       %air pressure

   me_air  = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.me_air'));
   ma_e    = me_air;
   mdot_e  = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.mdot_e'));
   Pe      = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Pe'));
   He_ri   = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.He_ri'));
   Te_ro   = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Te_ro'));
   Te_ai   = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Te_ai'));
   Te_ao   = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Te_ao'));
   Slip_e  = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Slip_e'));
   Phie_ai = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Phie_ai'));
   P_air   = eval(strcat('Operating_Conditions.EvapFrostMassTrans',num2str(HX_num),'.P_air'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3.3 Initial Conditions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   %Load necessary initial conditions
   % alpha_e_i1;          %refrigerant to HX wall heat transfer coefficient in the two phase region
   % alpha_e_i2;          %refrigerant to HX wall heat transfer coefficient in the single phase region
   % alpha_e_o;           %heat transfer coefficient between the HX wall and the external fluid
   % Te_a_mu;             %inlet/oulet weight for determining average air temperature
   % Pa_0;                %initial pressure of the air in the plenum at steady state

   alpha_e_i1  = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.alpha_e_i1'));
   alpha_e_i2  = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.alpha_e_i2'));
   alpha_e_o   = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.alpha_e_o'));
   Te_a_mu     = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Te_a_mu'));
   Pa_0        = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Pa_0'));
```

```
   % Initialize the frost layer
   Prev_Time = 0;             %  Initialize previous time variable to prevent problem on first simulation step
   frosting_flag1 = 0;        %  Assume frost is not forming at start of simulation (this will be checked later)
   frosting_flag2 = 0;        %  Assume frost is not forming at start of simulation (this will be checked later)
   delta_fr = 1e-8;           %  Assume initial frost thickness to avoid singularity
   rho_fr   = 100;            %  Assume initial frost density to avoid singularity

   % Inputs for nonlinear model
   ModParam = [Le_total Ae_cs Ae_i Ae_o Mass_e Cpw_e Te_a_mu Slip_e alpha_e_i1 alpha_e_i2 alpha_e_o
me_air Te_ao P_air Af kfin tfin Pt Pl r j_factor_flag Aff FA Dh Vol_PL Pa_0 delta_fr rho_fr];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4. Nonlinear Evaporator Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4.1 Fluid Properties, Current States, Model Inputs, and Required Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%         Density, Enthalpy, and Internal Energy
Rho_F      = interp1(FluidProp.Psat,FluidProp.Rhof,x(2));      %        Saturated liquid density
Rho_G      = interp1(FluidProp.Psat,FluidProp.Rhog,x(2));      %        Saturated vapor density
hF         = interp1(FluidProp.Psat,FluidProp.Hf,x(2));        %        Saturated liquid enthalpy
hG         = interp1(FluidProp.Psat,FluidProp.Hg,x(2));        %        Saturated vapor enthalpy
hi         = u(3);                                             %        Enthalpy of entering refrigerant
ho         = x(3);                                             %        Enthalpy of exiting refrigerant
He_r2      = (hG+ho)/2;                                   %   Average enthalpy in the second region
Rho_2      = interp2(FluidProp.H,FluidProp.P,FluidProp.Rho_ph,He_r2,x(2));    %        Average density of
refrigerant in the second region

%         Partial Derivatives
dRhoF_dP    = interp1(FluidProp.Psat,FluidProp.dRhof_dP,x(2));         %        Partial derivative of the saturated
liquid density with respect to pressure
dRhoG_dP    = interp1(FluidProp.Psat,FluidProp.dRhog_dP,x(2));         %        Partial derivative of the saturated
vapor density with respect to pressure
dhF_dP      = interp1(FluidProp.Psat,FluidProp.dHf_dP,x(2));           %        Partial derivative of saturated liquid
enthalpy with respect to pressure
dhG_dP      = interp1(FluidProp.Psat,FluidProp.dHg_dP,x(2));                        %        Partial derivative
of saturated vapor enthalpy with respect to pressure
dRho2_dP    = interp2(FluidProp.H,FluidProp.P,FluidProp.dRho_dP_H_ph,He_r2,x(2));   %        Partial derivative
of the average density in the second region with respect to pressure
dRho2_dh    = interp2(FluidProp.H,FluidProp.P,FluidProp.dRho_dH_P_ph,He_r2,x(2));   %        Partial derivative
of the average density in the second region with respect to enthalpy

%   Dry air specific heat
Cpair = 1.005;
%   Mean Void Fraction
He_fg   = hG-hF;

if hi<=hF
   H1=hF;
   x1=0;
else
   H1=hi;
   x1=(hi-hF)/He_fg;
```

```
end

if ho>=hG
   H2=hG;
   x2=1;
else
   H2=ho;
   x2=(ho-hF)/He_fg;
end

Slip_e = ModParam(9);
A_mvf=(Rho_G/Rho_F)*Slip_e;
B_mvf=1-A_mvf;
E_mvf=x2-x1;
C_mvf=A_mvf/((B_mvf^2)*E_mvf);
D_mvf=(B_mvf*x1+A_mvf)/(B_mvf*x2+A_mvf);

%Value of the Mean Void Fraction in the 2-phase region
Gamma = 1/B_mvf+(1/(x2-x1))*((A_mvf/(B_mvf^2))*log((B_mvf*x1+A_mvf)/(B_mvf*x2+A_mvf)));

%       Fluid Temperatures and Specific Heats
Tr1     = interp1(FluidProp.Psat,FluidProp.Tsat,x(2));        %Average refrigerant temperature in the first region
Tr2     = interp2(FluidProp.H,FluidProp.P,FluidProp.T_ph,He_r2,x(2));    %Average refrigerant temperature in
the second region
Tro     = interp2(FluidProp.H,FluidProp.P,FluidProp.T_ph,x(3),x(2));

%       Geometry
Ltotal    = ModParam(1);              %      Total length of heat exchanger "tubes" (irrespective of the
geometry, the heat exchanger is assumed to be a single tube)
A_cs      = ModParam(2);              %      Total cross sectional area for fluid flow
A_i       = ModParam(3);              %      Total internal surface area
A_o       = ModParam(4);              %      Total exterior surface area
m         = ModParam(5);              %      Mass of heat exchanger
Cpw       = ModParam(6);              %      Specific heat of the heat exchanger
mu_air    = ModParam(7);        %   Weighted average of inlet/outlet temp. of external fluid

%       Heat Transfer Coefficients, Outlet Air Temp, and heat transfer parameters/flags
alpha_i1   = ModParam(9);                          % Heat transfer coefficient between the refrigerant and
the tube wall in the first section
alpha_i2   = ModParam(10);                         % Heat transfer coefficient between the refrigerant and
the tube wall in the second section
Tao        = ModParam(13);              %   Outlet air temperature
P_air      = ModParam(14);              %   Air Pressure
Af         = ModParam(15);
kfin       = ModParam(16);
tfin       = ModParam(17);
Pt         = ModParam(18);
Pl         = ModParam(19);
r          = ModParam(20);
Aff_0      = ModParam(22);        %   Initial minimum free flow area for the heat exchanger
Pa_0       = ModParam(26);        %   Initial pressure in plenum at steady state
ma_0       = ModParam(12);        %   Initial air mass flow rate for pressure in plenum at steady state
Dh_0       = ModParam(24);        %   Initial external hydraulic diameter
FA         = ModParam(23);        %   Heat exchanger frontal area
Vol_PL     = ModParam(25);        %   Assumed volume of the plenum (only impacts the rate at which pressure
builds in the plenum)
```

```
delta_fr0   = ModParam(27);          %   Assumed initial frost thickness used to prevent singularity

j_factor_flag = ModParam(21);       %   Flag determining use of j factor

%           State Variables and Inputs
mi          = u(1);                  %          Mass flow rate entering the heat exchanger
mo           = u(2);                 %          Mass flow rate exiting the heat exchanger
ma           = u(5);                 %          Mass flow rate of air across the heat exchanger
Tai         = u(4);                  %          Inlet air temperature
Phi_in      = u(6);                  %          Inlet relative humidity
L1          = x(1);                  %          Length of the first region
Tw1           = x(4);                %          Average tube wall temperature in the first region
Tw2           = x(5);                %          Average tube wall temperature in the second region
Pa          = x(6);                  %          Pressure of air in plenum prior to heat exchanger

%           Calculated inputs
L2=Ltotal-L1;                                       %      Length of second region
Rho_1=Rho_F*(1-Gamma)+Rho_G*Gamma;                  %      Average density in the 2-phase region
hFG=hG-hF;                                          %      Enthalpy of Vaporization
dRhoFhF_dP = dRhoF_dP*hF+Rho_F*dhF_dP;              %      Partial derivative of the product of saturated liquid
density and enthalpy with respect to pressure
dRhoGhG_dP = dRhoG_dP*hG+Rho_G*dhG_dP;              %      Partial derivative of the product of saturated vapor
density and enthalpy with respect to pressure

%           Superheat Calculation
Tsh=2*(Tr2-Tr1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4.2 Model of Air Side Behavior (Flow Blockage, Heat Transfer)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%   Air Calculations
Ta = Tai*mu_air + Tao*(1-mu_air);

%   Current Geometric HX Data
Aff     = Aff_0 - 2*Ltotal*(delta_fr - delta_fr0);          %current minimum free flow area
FA = ModParam(23);                                          %frontal HX area
sigma_hx = Aff/FA;                                          %current HX constriction ratio

%   Note: The definition of the hydraulic diameter and the way it varies with
%   frost thickness will change depending on the j-factor correlation that is used
Dh = Dh_0 + 2*(delta_fr - delta_fr0);  %hydraulic diameter definition for Evap2 j-factor correlation

%   Air Mass Flow Calculations - Volume plenum approach (adds a dynamic state)
ma_fan  = ma_0*sqrt(((Pa_0-P_air)/(Pa-P_air))^1);   %Assume fan speed fixed, then as static pressure increases
mass flow will decrease (fan law)
R_a     = 0.287;                                    %Ideal Gas Constant for air [kJ/kg-K]
rho_air = Pa/(R_a*(Ta+273.15));                     %Density of air in evaporator
mu      = (1.458*10^-6*(Ta+273.15)^1.5)/(110.4 + Ta + 273.15);     %Viscosity of air in evaporator
v_i     = R_a*(Tai+273.15)/Pa;                      %specific volume of air at evaporator inlet
v_o     = R_a*(Tao+273.15)/P_air;                   %specific volume of air at evaporator outlet
v_m     = (v_i+v_o)/2;
G       = ma_e/Aff;                                 %maximum mass velocity
Re      = G*Dh/mu;                                  %Reynolds Number
f       = 0.724*Re^-0.423;                          %friction factor correlation for Evap2
C1      = v_i/2*((1+sigma_hx^2)*((v_o/v_i)-1)+f*(A_o*v_m/(Aff*v_i)));
```

```
ma_e    = sqrt((Aff^2*(Pa-P_air))/C1);                    %Mass flow rate over the evaporator

%   Calculate Plenum State Derivative
Pa_dot = (R_a*(Tai+273.15)/Vol_PL)*(ma_fan-ma_e);

%   Heat transfer calculation with current mass flow information
if j_factor_flag == 1
    alpha_o = j_factor(sigma_hx,Dh,ma_e,FA,Ta,Re_j,j);
else
    alpha_o = ModParam(11)*(u(5)/ModParam(12))^0.8;         %           Heat transfer coefficient between the tube
wall and the outside air scaled by change in mass flow of air
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4.3 Condensation and Frost Growth Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Find the dewpoint temperature
Pg_in = interp1(External_FluidProp.Tsat,External_FluidProp.Psat,Tai); %saturation pressure of water at inlet
temperature
Pv = Phi_in*Pg_in;                                                    %partial pressure of inlet vapor
Tdp = interp1(External_FluidProp.Psat,External_FluidProp.Tsat,Pv);    %dew point temperature

%   Split the airflow based on the region length
ma1 = (L1/Ltotal)*ma;
ma2 = (L2/Ltotal)*ma;

%   Check wall temperatures to determine if condensation or frosting is occuring in each region
if Tw1 < Tdp
    condensing_flag1 = 1;
    if Tw1 < 0,
        frosting_flag1 = 1;
    end
else
    condensing_flag1 = 0;
end

if Tw2 < Tdp
    condensing_flag2 = 1;
    if Tw2 < 0,
        frosting_flag2 = 1;
    end
else
    condensing_flag2 = 0;
end

%   Find values common to both regions or both conditions
hg_in = 2501.3 + 1.82*Tai;                                 %approximate inlet vapor enthalpy
win = .622*Phi_in*Pg_in/(P_air-Phi_in*Pg_in);              %inlet absolute humidity
hin = Cpair*Tai + win*hg_in;                               %inlet humid air enthalpy

Psat_w1 = interp1(External_FluidProp.Tsat,External_FluidProp.Psat,Tw1);      %water saturation pressure at wall
temperature 1
hg1 = interp1(External_FluidProp.Psat,External_FluidProp.Hg,Psat_w1);  %Saturated water vapor enthalpy
hf1 = interp1(External_FluidProp.Psat,External_FluidProp.Hf,Psat_w1);   %Saturated water liquid enthalpy
hfg1 = hg1-hf1;     %Enthalpy for condensation
```

```
Psat_w2 = interp1(External_FluidProp.Tsat,External_FluidProp.Psat,Tw2);      %water saturation pressure at wall
temperature 2
hg2 = interp1(External_FluidProp.Psat,External_FluidProp.Hg,Psat_w2);  %Saturated wtaer vapor enthalpy
hf2 = interp1(External_FluidProp.Psat,External_FluidProp.Hf,Psat_w2);   %Saturated water liquid enthalpy
hfg2 = hg2-hf2;       %Enthalpy for condensation

%   Region 1
if condensing_flag1 == 0              %Code for non-condensing conditions
   wout1 = win;                       %outlet absolute humidity, assuming no condensation
   mw1 = 0;                           %no condensation, set water mass flow rate to zero
   Qw_in1 = alpha_o*A_o*(L1/Ltotal)*(Ta-Tw1) + mw1*hfg1; %Heat transfer from air to wall in region 1
   hout1 = hin - Qw_in1/ma1;          %outlet humid air enthalpy
elseif condensing_flag1 == 1          %Code for condensing conditions
   XL = sqrt((Pt/2)^2 + Pl^2/2);                       %Geometric parameter
   XM = Pt/2;                                          %Geometric parameter
   Req = r*(1.27*XM/r*sqrt(XL/XM - .3));               %Effective radius for circular fins
   phi_fin = (Req/r - 1)*(1 + .35*log(Req/r));         %Fin efficiency correction factor
   m = sqrt((alpha_o)/(kfin*tfin));                    %Nameless parameter
   eta_f = tanh(m*r*phi_fin)/(m*r*phi_fin);            %Fin efficiency
   eta_o = 1 - Af/A_o*(1 - eta_f);                     %Overall surface efficiency
   rho_a = P_air/(.287*(Ta+273.15));                %Air density, assuming ideal gas behavior with R = .287
   hm = alpha_o/(rho_a*Cpair);                         %Mass transfer coefficient
   Pg_wall_1 = interp1(External_FluidProp.Tsat,External_FluidProp.Psat,Tw1);  %Saturation pressure of water at
wall temperature
   wwall_1 = .622*Pg_wall_1/(P_air - Pg_wall_1);   %Wall absolute humidity
   wout1 = (win - wwall_1)/(exp(hm*A_o*L1/Ltotal*eta_o/ma1)) + wwall_1;   %Outlet absolute humidity
   mw1 = ma1*(win-wout1);                              %Mass flow rate of water

   if frosting_flag1 == 0,
      %Calculate energy transfer and outlet conditions
      Qw_in1 = alpha_o*A_o*(L1/Ltotal)*(Ta-Tw1) + mw1*hfg1;     %Heat transfer from air to wall in region 1
      hw1 = interp1(External_FluidProp.Psat,External_FluidProp.Hf,Psat_w1);  %Enthalpy of condensate
      hout1 = hin - mw1*hw1/ma1 - Qw_in1/ma1;                          %Outlet enthalpy
   else
      %Required Ice Properties
      rho_ice = 921;                                       %density of ice [kg/m3]
      h_sb = interp1(External_FluidProp.Tsat_ice,External_FluidProp.Hsb,Tw1); %enthalpy of sublimation of ice
[kj/kg]
      Ts1 = Tw1+273.15;                     %surface temperature of pipe [K]
      kf = 0.001202*rho_fr^0.963;           %thermal conductivity of frost from Sanders [W/m-K]
      R_w = 0.4615;                         %Gas constant for water [kJ/kg-K]
      vg = R_w*Tai/Pg_in;                   %specific volume of water at inlet temperature (assume ideal gas
behavior) [m3/kg]
      v_ice = 1/rho_ice;                    %specific volume of ice
      %Energy transfer and air outlet enthalpy
      Qw_in1 = alpha_o*A_o*(L1/Ltotal)*(Ta-Tw1) + mw1*h_sb;                %Heat transfer from air to wall in
region 1
      hw1 = interp1(External_FluidProp.Tsat_ice,External_FluidProp.Hice,Tw1); %Enthalpy of frost
      hout1 = hin - mw1*hw1/ma1 - Qw_in1/ma1;                          %Outlet enthalpy
      %Break mw1 into portions contributing to increases in density and thickness
      D_s1 = (2.302*(0.98e5)/(Pa*10^3))*(Ts1/256)*10^-5;                   %Molecular diffusivity of water in a
frosting process (Eckert & Drake)
      D_fr1 = (kf/D_s1)*((1+(rho_fr/rho_ice)^0.5)/(1-(rho_fr/rho_ice)))*((R_w*Ts1^2*(vg-v_ice))/(h_sb-
Pg_wall_1*(vg-v_ice))) + h_sb;
      mdot_rho_fr1 = Qw_in1/D_fr1;    %mass flow rate of water that increases the density of the frost layer
```

```
        mdot_delta_fr1 = mw1 - mdot_rho_fr1;              %mass flow rate of water that increases the frost layer
thickness
        %Rates of frost growth in region 1
        deltaT = Time - Prev_Time;                         %time for frost growth
        Ddelta_fr1 = mdot_delta_fr1*deltaT/(A_o*rho_fr);   %change in frost thickness
        Vf = (L1/Ltotal)*A_o*delta_fr;                     %volume of accumulated frost
        Drho_fr1 = mdot_rho_fr1*deltaT/Vf;                 %change in frost density
    end
end

%   Region 2
if condensing_flag2 == 0 %Code for non-condensing conditions
    wout2 = win;                                   %outlet absolute humidity, assuming no condensation
    mw2 = 0;                                       %no condensation, set water mass flow rate to zero
    Qw_in2 = alpha_o*A_o*(L2/Ltotal)*(Ta-Tw2) + mw2*hfg2;          %Heat transfer from air to wall in
region 2
    hout2 = hin - Qw_in2/ma2;                          %outlet humid air enthalpy
    Ddelta_fr2 = 0;                                    %change in frost thickness
    Drho_fr2 = 0;                                      %change in frost density
elseif condensing_flag2 == 1    %Code for condensing conditions
    XL = sqrt((Pt/2)^2 + Pl^2/2);                  %Geometric parameter
    XM = Pt/2;                                     %Geometric parameter
    Req = r*(1.27*XM/r*sqrt(XL/XM - .3));          %Effective radius for circular fins
    phi_fin = (Req/r - 1)*(1 + .35*log(Req/r));    %Fin efficiency correction factor
    m = sqrt((alpha_o)/(kfin*tfin));               %Nameless parameter
    eta_f = tanh(m*r*phi_fin)/(m*r*phi_fin);       %Fin efficiency
    eta_o = 1 - Af/A_o*(1 - eta_f);                %Overall surface efficiency
    rho_a = P_air/(.287*(Ta+273.15));              %Air density, assuming ideal gas behavior with R = .287
    hm = alpha_o/(rho_a*Cpair);                    %Mass transfer coefficient
    Pg_wall_2 = interp1(External_FluidProp.Tsat,External_FluidProp.Psat,Tw2);   %Saturation pressure of water at
wall temperature
    wwall_2 = .622*Pg_wall_2/(P_air - Pg_wall_2);      %Wall absolute humidity
    wout2 = (win - wwall_2)/(exp(hm*A_o*L2/Ltotal*eta_o/ma2)) + wwall_2;        %Outlet absolute humidity
    mw2 = ma2*(win-wout2);                                     %Mass flow rate of water

    if frosting_flag2 == 0,
        Qw_in2 = alpha_o*A_o*(L2/Ltotal)*(Ta-Tw2) + mw2*hfg2;          %Heat transfer from air to wall in
region 2
        hw2 = interp1(External_FluidProp.Psat,External_FluidProp.Hf,Psat_w2);   %Enthalpy of condensate
        hout2 = hin - mw2*hw2/ma2 - Qw_in2/ma2;                 %Outlet enthalpy
        %Check if frost growth is occurring in only the two phase region
        Ddelta_fr2 = 0;                                %change in frost thickness
        Drho_fr2 = 0;                                  %change in frost density
    else
        %Required Ice Properties for Region 2
        h_sb = interp1(External_FluidProp.Tsat_ice,External_FluidProp.Hsb,Tw2); %enthalpy of sublimation of ice
[kj/kg]
        Ts2 = Tw2+273.15;                                  %surface temperature of pipe [K]
        %Energy transfer and air outlet enthalpy
        Qw_in2 = alpha_o*A_o*(L2/Ltotal)*(Ta-Tw2) + mw2*h_sb;          %Heat transfer from air to wall in
region 1
        hw2 = interp1(External_FluidProp.Tsat_ice,External_FluidProp.Hice,Tw2); %Enthalpy of frost
        hout2 = hin - mw2*hw2/ma2 - Qw_in2/ma2;                 %Outlet enthalpy
        %Break mw2 into portions contributing to increases in density and thickness
        D_s2 = (2.302*(0.98e5)/(Pa*10^3))*(Ts2/256)*10^-5;              %Molecular diffusivity of water in a
frosting process (Eckert & Drake)
```

```
      D_fr2 = (kf/D_s2)*((1+(rho_fr/rho_ice)^0.5)/(1-(rho_fr/rho_ice)))*((R_w*Ts2^2*(vg-v_ice))/(h_sb-
Pg_wall_2*(vg-v_ice))) + h_sb;
      mdot_rho_fr2 = Qw_in2/D_fr2;   %mass flow rate of water that increases the density of the frost layer
      mdot_delta_fr2 = mw2 - mdot_rho_fr2;                    %mass flow rate of water that increases the frost
layer thickness
      %Rates of frost growth for region 2
      Ddelta_fr2 = mdot_delta_fr2*deltaT/(A_o*rho_fr);        %change in frost thickness
      Vf2 = (L2/Ltotal)*A_o*delta_fr;                         %volume of accumulated frost
      Drho_fr2 = mdot_rho_fr2*deltaT/Vf2;                     %change in frost density
   end
end

%Calculate uniform increase in frost density and thickness for entire pipe length by combining the growth %rates
from region 1 and region 2 to obtain average growth rates
if frosting_flag1 == 1,
   Ddelta_fr_avg = (L1/Ltotal)*Ddelta_fr1 + (L2/Ltotal)*Ddelta_fr2;
   Drho_fr_avg = (L1/Ltotal)*Drho_fr1 + (L2/Ltotal)*Drho_fr2;
   delta_fr = delta_fr + Ddelta_fr_avg;                       %Total frost thickness
   rho_fr = rho_fr + Drho_fr_avg;                             %Total frost density
   %   Calculate thermal inertia of frost layer
   Cpf = 2.1; %Specific heat of ice at ~273 K (kJ/kg-K)
   Fr_therm = Cpf*A_o*delta_fr*rho_fr;
end

%Mixing of air and water from region 1 and region 2
mw = mw1 + mw2;
wout = (ma1*wout1 + ma2*wout2)/(ma1 + ma2);
hout = (ma1*hout1 + ma2*hout2)/(ma1 + ma2);

%Air temperature and relative humidity of mixed outlet air
if Time == StartTime,
   Tao = (hout - 2501.3*wout)/(Cpair + 1.82*wout);
   Tao_alt = (Ta - mu_air*Tai)/(1 - mu_air);
   Ta_delt = Tao_alt - Tao;   %Adjustment for rounding error in calculation that occurs (helps stabilize Pa)
end
Tao = (hout - 2501.3*wout)/(Cpair + 1.82*wout) + Ta_delt;              %outlet air temperature
Pg_out = interp1(External_FluidProp.Tsat,External_FluidProp.Psat,Tao);     %saturation pressure of water at outlet
temperature
Phi_out = (wout*P_air)/((0.622 + wout)*Pg_out);                        %outlet relative humidity

%   Save parameters required for the next time step
ModParam(13) = Tao;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4.4 Calculate State Derivatives
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%        Z matrix
Z = zeros(5,5);
Z(1,1)=Rho_F*(hF-hG)*(1-Gamma)*A_cs;
Z(1,2)=((dRhoFhF_dP-dRhoF_dP*hG)*(1-Gamma)+(dRhoGhG_dP-dRhoG_dP*hG)*(Gamma)-1)*A_cs*L1;
Z(2,1)=0.5*Rho_2*(hG-ho)*A_cs;
Z(2,2)=((dRho2_dP+0.5*dRho2_dh*dhG_dP)*(0.5*(ho-hG))+0.5*Rho_2*dhG_dP-1)*A_cs*L2;
Z(2,3)=(0.5*dRho2_dh*(0.5*(ho-hG))+0.5*Rho_2)*A_cs*L2;
Z(3,1)=((Rho_G-Rho_2)+(Rho_F-Rho_G)*(1-Gamma))*A_cs;
Z(3,2)=((dRho2_dP+0.5*dRho2_dh*dhG_dP)*L2+((dRhoF_dP)*(1-Gamma)+(dRhoG_dP)*(Gamma))*L1)*A_cs;
```

```
Z(3,3)=0.5*dRho2_dh*A_cs*L2;
Z(4,4)=Cpw*m;
Z(5,1)=Cpw*m*(Tw1-Tw2)/L2;
Z(5,5)=Cpw*m;
if frosting_flag1 == 1,
   Z(4,4)=Cpw*m + Fr_therm;
   Z(5,1)=(Cpw*m + Fr_therm)*(Tw1-Tw2)/L2;
   Z(5,5)=Cpw*m + Fr_therm;
end

%        F vector
f(1,1)=mi*hi-mi*hG+alpha_i1*(A_i/Ltotal)*L1*(Tw1-Tr1);
f(2,1)=mo*hG-mo*ho+alpha_i2*(A_i/Ltotal)*L2*(Tw2-Tr2);
f(3,1)=mi-mo;
f(4,1)=alpha_i1*A_i*(Tr1-Tw1)+alpha_o*A_o*(Ta-Tw1) + (Ltotal/L1)*mw1*hfg1;
f(5,1)=alpha_i2*A_i*(Tr2-Tw2)+alpha_o*A_o*(Ta-Tw2) + (Ltotal/L2)*mw2*hfg2;

%        Calculate HX Derivatives
x_dot1=Z\f;
x_dot = [x_dot1; Pa_dot];

%   Refrigerant Mass Inventory
Mass=A_cs*L1*Rho_1 + A_cs*L2*Rho_2;

%   Store last time value
Prev_Time = Time;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 5. Check for violations of modeling assumptions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if (x(1)/ModParam(1)) >= 1,
   disp('Modeling Assumptions Violated :-(')
   disp('Length of superheat region has gone to zero.')
   set_param(gcs,'SimulationCommand','stop')
end

if x(1) <= 0,
   disp('Modeling Assumptions Violated :-(')
   disp('Length of two-phase region has gone to zero.')
   set_param(gcs,'SimulationCommand','stop')
end

if u(1) < 0,
   disp('Modeling Assumptions Violated :-(')
   disp('Negative mass flow rate into heat exchanger is not allowed.')
   set_param(gcs,'SimulationCommand','stop')
end

if frosting_flag1 == 1,
   %check if initial frost density causes impossible start condition
   if mdot_delta_fr1 < 0,
      disp('Increase density of initial frost layer to prevent simulation from calculating a decrease in frost thickness')
      set_param(gcs,'SimulationCommand','stop')
   end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 6. Model Output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%          Outputs
y(1)=x(1);
y(2)=x(2);
y(3)=x(3);
y(4)=x(4);
y(5)=x(5);
y(6)=x(6);
y(7)=Tao;
y(8)=Tro;
y(9)=Tsh;
y(10)=Mass;
y(11)=Phi_out;
y(12)=mw;
y(13)=ma_e;
y(14)=delta_fr;
y(15)=rho_fr;
y(16)=ma_fan;
y(17)=Aff;

yout = [y(2) y(2) y(3)];    %Output Pin, Pout, Hout
output = [x_dot' yout y];
```

B.1.2 Initial Condition Simulation Function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  FILENAME:                    EvapFrostMassTransIC.m
%  WRITTEN BY:                  Michael Keir
%                               University of Illinois at Urbana-Champaign
%
%  COMMENTS ON CODE STRUCTURE:  This code calculates the ICs for a nonlinear evaporator with %frost
growth model. The code will only run once at the start of the simulation and it will input the %appropriate initial
conditions to the integrator block.
%                         Inputs: 2
%                         Outputs: 6
%
%  ORIGINAL DATE WRITTEN:          11 June 2006
%  DATE OF LAST MODIFICATION:   11 June 2006
%  MODIFICATION HISTORY:
%  DATE:                    COMMENT:
%  11 June 2006              - Original write of the stellar program...
%
%  PROBLEMS/MODIFICATIONS FOR FUTURE WORK:
%
%%%%%%%%%%%%%%%%%%% START OF Michael Keir's Awesome CODE%%%%%%%%%%%

function [output] = EvapFrostMassTransIC(U)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1. Pull in initial conditions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global Initial_Conditions
```

```
persistent x0

flag = U(1);
HX_num = U(2);

if flag == 1,

   Le_1    = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Le_1'));
   Pe      = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Pe'));
   He_ro   = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.He_ro'));
   Te_w1   = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Te_w1'));
   Te_w2   = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Te_w2'));
   Pa      = eval(strcat('Initial_Conditions.EvapFrostMassTrans',num2str(HX_num),'.Pa_0'));
end

x0 = [Le_1 Pe He_ro Te_w1 Te_w2 Pa];

output = x0;
```

B.1.3 Recording of Model Output

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  FILENAME:                 EvapFrostMassTransRec.m
%  WRITTEN BY:               Michael Keir
%                            University of Illinois at Urbana-Champaign
%
%  COMMENTS ON CODE STRUCTURE:  This code records the output from a nonlinear evaporator %model.
%                      Inputs: 14
%                      Outputs: 1 (indicator of data recording)
%
%  ORIGINAL DATE WRITTEN:       13 June 2006
%  DATE OF LAST MODIFICATION:   13 June 2006
%  MODIFICATION HISTORY:
%  DATE:                   COMMENT:
%  13 June 2006             - Original write of the stellar program...
%
%  PROBLEMS/MODIFICATIONS FOR FUTURE WORK:
%
%%%%%%%%%%%%%%%%%%% START OF Michael Keir's Awesome CODE %%%%%%%%%%%%

function [out] = EvapFrostMassTransRec(U)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1. Record Model Outputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global Output
global Physical_Parameter
persistent index

flag = U(end-1);
HX_num = U(end);

%Check if any information has entered program, if not return value of 0
if norm(U,inf) == 0,
   out = 0;
```

```
%If information has entered program record desired data
else

   %Initialization of program on first recorded time step
   if flag == 1,
      index = 1;
   else
      %Check to see if another time step has been taken by the simulation
      if U(end-2) > eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.time(index)'))
         index = index + 1;
      end
   end

   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Le_1(index)  =
U(1)/Physical_Parameter.EvapFrostMassTrans',num2str(HX_num),'.Le_total;'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Pe(index)    =   U(2);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.He_ro(index) =   U(3);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Te_w1(index) =   U(4);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Te_w2(index) =   U(5);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Pa(index)    =   U(6);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Te_ao(index) =   U(7);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Te_ro(index) =   U(8);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Te_sh(index) =   U(9);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.minv_e(index)=   U(10);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Phie_ao(index)=  U(11);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.mw(index)    =   U(12);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.ma_e(index)  =   U(13);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.delta_fr(index)= U(14);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.rho_fr(index)=   U(15);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.ma_fan(index)=   U(16);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.Aff(index)   =   U(17);'))
   eval(strcat('Output.EvapFrostMassTrans',num2str(HX_num),'.time(index)  =   U(end-2);'))
   out = 1;
end
```

B.1.4 Simulation Dimension File

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  FILENAME:                EvapFrostMassTransDim.m
%  WRITTEN BY:              Michael Keir
%                           University of Illinois at Urbana-Champaign
%
%  COMMENTS ON CODE STRUCTURE:  This code sets up a nonlinear evaporator with frost growth %model in
simulink
%                   Inputs: 0
%                   Outputs: 0
%
%  ORIGINAL DATE WRITTEN:       11 June 2006
%  DATE OF LAST MODIFICATION:   11 June 2006
%  MODIFICATION HISTORY:
%  DATE:                    COMMENT:
%  11 June 2006              - Original write of the stellar program...
%
%  PROBLEMS/MODIFICATIONS FOR FUTURE WORK:
%
%%%%%%%%%%%%%%%%%% START OF Michael Keir's Awesome CODE %%%%%%%%%%%%
```

```
function [varargout] = EvapFrostMassTransDim(varargin)

%Set output dimension block for the Model function
set_param(strcat(gcb,'/Model'),'OutputDimensions','26')
%Set output dimension block for the Initial Condition function
set_param(strcat(gcb,'/Initial Conditions/Enabled Subsystem/X0'),'OutputDimensions','6')
%Set output dimension block for the Record function
set_param(strcat(gcb,'/Recorded Outputs/Record Function'),'OutputDimensions','1')
%Set Dimensions of Demux at output of Model function
set_param(strcat(gcb,'/Demux'),'Outputs','[6 3 17]')

return
```

B.2 LQG Control Design Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   This m-file generates a LQG controller for a system with inputs of valve command and compressor %speed and
outputs of superheat and the pressure difference across the system. The design procedure is the %similar to that used
by He and Asada (1998) and the system in augmented with  integrators to eliminate %steady state tracking error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
clc
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Load the data set to ID the model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load IDData_1500rpm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Organize Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Input/Output Labels
InputLabels  = {'Valve Opening', 'Scaled Compressor Speed'};
OutputLabels = {'Evap Superheat', 'DeltaP/6'};
InputUnits   = {'[%]', '[rpm/100]'};
OutputUnits = {'[C]', '[kPa/6]'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Identify a Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Inputs_all  = [Data.EEV_2_ref Data.RPM/100];
Outputs_all = [Data.Tesh_2 Data.Pdelta_2/6];
IDData_all  = detrend(iddata(Outputs_all,Inputs_all,Data.SampleTime, 'InputName', InputLabels, 'OutputName',
OutputLabels, 'InputUnit', InputUnits, 'OutputUnit', OutputUnits),1);

%Perform model ID - using ident a 2nd order model was found to be appropriate
Sys         = pem(IDData_all,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Scale the model back to initial units
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Convert to continuous time
G_sc = minreal(d2c(ss(Sys.A,Sys.B,Sys.C,Sys.D,1)));

%Add states to output of plant for simulation purposes
G_s.A = G_sc.A;
G_s.B = G_sc.B;
G_s.C = [eye(2); G_sc.C];
G_s.D = zeros(4,2);

%Scale system back to match physical system
G = ss(G_sc.A,G_sc.B*[1 0; 0 0.01],[1 0; 0 6]*G_sc.C,[1 0; 0 6]*G_sc.D*[1 0; 0 0.01]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Augment the Output of the Plant for LQR set point regulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = [G_sc.A zeros(2,2); G_sc.C zeros(2,2)];
B = [G_sc.B ; zeros(2,2)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Kalman Filter - Optimal Observer
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

V = [(0.12)^2 0; 0 (8.268/6)^2]
W = Sys.K'*V*Sys.K

L = LQE(G_sc.A,ones(size(G_sc.A)),G_sc.C,W,V)
[V_obs lambda_obs] = eig(G_sc.A - L*G_sc.C)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Untuned LQR Weights - Only Integral Gain
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   %   LQR - Optimal state feedback (without noise)
   R = diag([1 1]);
   Q = [zeros(2,4); zeros(2,2) eye(2)];
   Kx = LQR(A,B,Q,R)
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Closed Loop Scaled System with reference as the input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   A_cl = [G_sc.A -G_sc.B*Kx;...
         L*G_sc.C (G_sc.A-L*G_sc.C-G_sc.B*Kx(:,1:2)) -G_sc.B*Kx(:,3:4);...
         G_sc.C zeros(2,4)];
   B_cl = [zeros(4,2); -eye(2)];
   C_cl = [G_sc.C zeros(2,4)];
   D_cl = zeros(2,2);

   %Scaled closed loop system
   CL_sys_sc_untuned = ss(A_cl,B_cl,C_cl,D_cl);
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Closed Loop Scaled System Noise Impact
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   A_n = [G_sc.A -G_sc.B*Kx;...
         L*G_sc.C (G_sc.A-L*G_sc.C-G_sc.B*Kx(:,1:2)) -G_sc.B*Kx(:,3:4);...
```

```
        G_sc.C zeros(2,4)];
    B_n = [zeros(2,2); L; -eye(2)];
    D_n = zeros(2,2);

    %Output y
    C_n_y = [G_sc.C zeros(2,4)];
    CL_n_y_untuned = ss(A_n,B_n,C_n_y,D_n);

    %Output u
    C_n_u = [zeros(2,2) -Kx];
    CL_n_u_untuned = ss(A_n,B_n,C_n_u,D_n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Tuned LQR Weights - Only Integral Gain
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    R = diag([100 10]);
    Q = [zeros(2,4); zeros(2,2) diag([0.1 0.5])];
    Kx = LQR(A,B,Q,R)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Closed Loop Scaled System with reference as the input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    A_cl = [G_sc.A -G_sc.B*Kx;...
        L*G_sc.C (G_sc.A-L*G_sc.C-G_sc.B*Kx(:,1:2)) -G_sc.B*Kx(:,3:4);...
        G_sc.C zeros(2,4)];
    B_cl = [zeros(4,2); -eye(2)];
    C_cl = [G_sc.C zeros(2,4)];
    D_cl = zeros(2,2);

    %Scaled closed loop system
    CL_sys_sc = ss(A_cl,B_cl,C_cl,D_cl);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Closed Loop Scaled System Noise Impact
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    A_n = [G_sc.A -G_sc.B*Kx;...
        L*G_sc.C (G_sc.A-L*G_sc.C-G_sc.B*Kx(:,1:2)) -G_sc.B*Kx(:,3:4);...
        G_sc.C zeros(2,4)];
    B_n = [zeros(2,2); L; -eye(2)];
    D_n = zeros(2,2);

    %Output y
    C_n_y = [G_sc.C zeros(2,4)];
    CL_n_y = ss(A_n,B_n,C_n_y,D_n);

    %Output u
    C_n_u = [zeros(2,2) -Kx];
    CL_n_u = ss(A_n,B_n,C_n_u,D_n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Tuned LQR Weights - Proportional plus Integral Gains
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Dynamic State Feedback
```

```
   R = diag([100 10]);
   Kp = diag([9.55 4]);
   Qy = (Kp*G_sc.C)'*(Kp*G_sc.C);
   Q = [Qy zeros(2,2); zeros(2,2) diag([0.019 1])];
   Kx = LQR(A,B,Q,R)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Closed Loop Scaled System with reference as the input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   A_cl = [G_sc.A -G_sc.B*Kx;...
         L*G_sc.C (G_sc.A-L*G_sc.C-G_sc.B*Kx(:,1:2)) -G_sc.B*Kx(:,3:4);...
         G_sc.C zeros(2,4)];
   B_cl = [zeros(4,2); -eye(2)];
   C_cl = [G_sc.C zeros(2,4)];
   D_cl = zeros(2,2);

   %Scaled closed loop system
   CL_sys_sc_PI = ss(A_cl,B_cl,C_cl,D_cl);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Closed Loop Scaled System Noise Impact
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   A_n = [G_sc.A -G_sc.B*Kx;...
         L*G_sc.C (G_sc.A-L*G_sc.C-G_sc.B*Kx(:,1:2)) -G_sc.B*Kx(:,3:4);...
         G_sc.C zeros(2,4)];
   B_n = [zeros(2,2); L; -eye(2)];
   D_n = zeros(2,2);

   %Output y
   C_n_y = [G_sc.C zeros(2,4)];
   CL_n_y_PI = ss(A_n,B_n,C_n_y,D_n);

   %Output u
   C_n_u = [zeros(2,2) -Kx];
   CL_n_u_PI = ss(A_n,B_n,C_n_u,D_n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Closed Loop Poles
figure(1)
subplot(1,2,1)
pzmap(G_sc,CL_sys_sc_untuned,CL_sys_sc,CL_sys_sc_PI)
ax = axis;
ax(1:2) = [-0.3 0];
axis(ax)
title('Zoomed Pole-Zero Map')
subplot(1,2,2)
pzmap(G_sc,CL_sys_sc_untuned,CL_sys_sc,CL_sys_sc_PI)
axis([-0.05 0 -0.03 0.03]);
title('Poles and Zeros that Impact Superheat Oscillation')
%legend('Plant','Untuned I','Tuned I','Tuned PI',2)
%Closed Loop Bode
figure(2)
bodemag(CL_sys_sc,CL_sys_sc_untuned,CL_sys_sc_PI,{1e-5,1})
```

```
legend('Tuned','Untuned','PI')
%Noise impact on output
figure(3)
bodemag(CL_n_y,CL_n_y_untuned,CL_n_y_PI,{1e-2,100})
legend('Tuned','Untuned','PI')
%Noise impact on input
figure(4)
bodemag(CL_n_u,CL_n_u_untuned,CL_n_u_PI,{1e-2,100})
legend('Tuned','Untuned','PI')
```