DISSERTATION

INVESTIGATION OF A NONLINEAR CONTROLLER THAT COMBINES STEADY STATE PREDICTIONS WITH INTEGRAL ACTION

Submitted by

David A. Hodgson

Department of Mechanical Engineering

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2010

COLORADO STATE UNIVERSITY

November 30, 2009

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY DAVID A. HODGSON ENTITLED INVESTIGATION OF A NONLINEAR CONTROLLER THAT COMBINES STEADY STATE PREDICTIONS WITH INTEGRAL ACTION BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

David B. Olsen

Dr. Daniel B. Olsen

Charles W Anderson

Dr. Charles W. Anderson

William S. Duff

Adviser: Dr. William S. Duff

P M Young

Co-Adviser: Dr. Peter M. Young

Allan T. Kirkpatrick

Department Head: Dr. Allan T. Kirkpatrick

ABSTRACT OF DISSERTATION

INVESTIGATION OF A NONLINEAR CONTROLLER THAT COMBINES STEADY STATE PREDICTIONS WITH INTEGRAL ACTION

Cross-flow water-to-air heat exchangers are a common element in heating ventilating and air conditioning (HVAC) systems. In a typical configuration the outlet air temperature is controlled by the flow rate of water through the coil. In this configuration the heat exchanger exhibits non-linear dynamics. In particular the system has variable gain. Variable gain presents a challenge for the linear controllers that are typically used to control the outlet air temperature. To ensure stability over the entire operating range controllers need to be tuned at the highest gain state. This leads to sluggish response in lower gain states. Previous research has shown the use of steady state predictions of the flow rate needed to produce zero steady state error has improved the transient response of a heat exchanger.

In this project a nonlinear controller that provides smooth mixing between steady state predictions and integral control was introduced. Bounds for the steady state error introduced by the controller were theoretically derived and experimentally verified. The controller outperformed a properly tuned nominal PI controller for both input tracking and disturbance rejection.

David A. Hodgson
Department of Mechanical Engineering
Colorado State University
Fort Collins, Colorado 80523
Spring 2010

ACKNOWLEDGMENTS

This has been a long process and I have many people to thank. The list is too long to include everyone here, but some people deserve special recognition.

First of all I am indebted to my committee. Without their help and willingness to be flexible it would not have been possible for me to complete my dissertation. I would like to thank Dr. Young and Dr. Anderson for all of their input and for sticking with me through all of these years. I would like to thank Dr. Duff and Dr. Olsen for taking on their duties at such a late stage in the game. In a short time, both have made my work better. I would also like to thank Dr. Hittle who introduced me to this project and provided guidance for many years.

I finished my research while running the experiment remotely. This would not have been possible if it had not been for many friends that were willing to go to the lab at all hours of the day (and in all types of weather) to trouble shoot and in some cases make needed repairs. In no particular order I need to thank: John Taggart, Matt Taggart, Dann Frazier, Tim Milburn, Matt Davis, and Brook Yockey.

I need to thank my family that has been there since the beginning. As the years of graduate school turned into more than a decade, their support was unwavering (as far as I know). I appreciate their patience during this process.

Lastly, and most importantly, I need to thank my wife, Alexis, and my daughter, Talia. They have both made many sacrifices that enabled me to focus on my work. I look forward to giving them the time and attention they deserve.

DEDICATION

For My Dad

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Problem Statement

Many components of heating ventilation and air conditioning (HVAC) systems exhibit non-linear dynamics with variable gain. In addition, it is common practice to control HVAC systems with linear control architectures. In order to ensure stability for all operating conditions linear controllers need to be tuned while the system is in a high gain state. This results in sluggish response when the system encounters a lower gain state [14].

Water-to-air cross-flow heat exchangers are common in HVAC systems. In a typical application the outlet air temperature is controlled by the flow rate of water through the valve. In this configuration a heat exchanger has non-linear dynamics with variable gain [30]. When the air flow and water flow are relatively low, a small change in water flow rate will produce a relatively large change in the outlet air temperature when compared to the case when the air flow and water flow are relatively high and the same change in water flow rate is experienced.

Proportional plus integral (PI) controllers are easy to implement and conceptualize. For that reason they are the most common form of controller used with HVAC equipment [6]. Delnero et al. [8, 10, 16], developed a controller which augmented a PI controller with steady state predictions. Under limited testing the controller performed better than a nominal PI controller in both signal tracking and disturbance rejection.

The goal of the research presented in this dissertation was to expand upon the methods of Delnero et al. Section 1.2 elaborates on the goals of the research. Section 1.3 provides an outline for the rest of this document.

1.2 Goals of Research

The purpose of this project was to investigate methods for improving the transient response of a system with variable gain. The main focus was to explore the possibility of expanding and improving upon the the methods of Delnero et al.

Specifically the goals were to:

1. Make the necessary repairs and upgrades to the experimental setup in REPEAT so it could be operated remotely.

2. Develop a dynamic model of the *REPEAT HVAC* system.

3. Expand the procedure of Delnero et al. to allow for a smooth mixing of control between the steady state prediction and the integral controller, rather than a discrete choice between the two.

4. Create a control architecture which allows for on-line learning to control the mixing between the steady state prediction and the integral control.

   (a) Test the control architecture in simulation using the dynamic model developed during the project.

   (b) Test the control architecture on the *REPEAT HVAC* system.

1.3 Overview of Dissertation

A description of the experimental *REPEAT HVAC* system is given in chapter 2. Chapter 3 presents a review of the literature in the areas of heat exchanger modeling and control. The development of a new nonlinear controller is presented in chapter 4. The results of a preliminary investigation of the *REPEAT HVAC* system is given in chapter 5. In addition, chapter 5 contains the derivation of dynamic models for the system. The controller of chapter 4 is applied to the *REPEAT HVAC* system in chapter 6. The performance of the controller is compared to nominal PI control in simulation and on the physical system. Chapter 7 concludes the dissertation.

Chapter 2

The Experimental *REPEAT HVAC* System

2.1 Introduction

The experimental test bed for this project, *REPEAT HVAC*, is located in the REPEAT laboratory at the foothills campus of Colorado State University. This chapter gives a brief overview of the equipment and it's operation. Much of the initial set-up and calibration of the system was conducted primarily by Michael Anderson and Christopher Delnero [10], [4], and [22].

A picture of the system is shown in figure 2.1. A schematic of of the system is shown in figure 2.2. The main components of the system are: a water to air heat exchanger (heating coil)[1] an electric water boiler[2], an air blower driven by a three phase motor [3] which is powered by a variable frequency drive[4], an air flow meter[5], a coriolis water flow meter[6],

---

[1]21 row, four-pass, 24 inch x 24 inch cross section, 1/2 copper coils, 0.010 inch-thick aluminum fins.

[2]ARGO Industries, model 24-240.

[3]Dayton model 4C770A

[4]Allen-Bradley 160S-AA04NSF1P1

[5]Brandt model DSK9211-1-1-1-12x12.

[6]Micro Motion model DS065S239SU sensor paired with an RFT97121PNU transducer.

two air flow control dampers[7], an equal percentage three way water flow control valve[8], and seven resistance temperature detectors (RTD)[9]. The system is controlled by a computer[10] through the use of MATLAB's real time target [1]. Table 2.1 summarizes the command signals generated by the control computer during experimentation. Table 2.2 summarizes the measurement that were recorded by the control computer during experimentation.

Table 2.1: Summary of Command Signals Generated by the Control Computer.

Symbol	Description	Minimum Value	Maximum Value
C_{bs}	Commanded Blower (fan) Speed	30%	70%
C_{hwp}	Commanded Power To Boiler	0%	100%
C_{vp}	Commanded Valve Position	5%	95%
C_{dr}	Commander Return Damper Position	0%	100%

Table 2.2: Summary of Signals Recorded by the Control Computer.

Symbol	Description	Primary Control Signal	Minimum Value	Maximum Value
T_{ae}	Outdoor Air Temperature	Monitored	-12°C	9°C
T_{ar}	Indoor (Return) Air Temperature	Monitored	12°C	24°C
T_{ai}	Coil Inlet Air Temperature	C_{dr}	8°C	12°C
T_{wi}	Coil Inlet Water Temperature	C_{hwp}	47°C	52°C
f_a	Flow Rate of Air Across The Coil	C_{bs}	0.17 m^3/s	0.71 $^3/s$
f_w	Flow Rate of Water Through The Coil	C_{vp}	0.058 L/s	0.57 L/s
T_{ao}	Coil Outlet Air Temperature	C_{vp}	21°C	39°C

Section 2.2 gives a brief description of the air side sensors and actuators. Section 2.3 gives a brief description of the water side sensors and actuators. Section 2.4 describes the configuration of the system for this project. Section 2.5 concludes the chapter.

[7]Positioned with Landis and Staefa, No. 4 Damper Actuators.

[8]Landis and Staefa, No. 698-3120

[9]Landis and Staefa 1000 ohm.

[10]Dell Optiplex GX1p, 500 MHz Pentium with 512 MB RAM

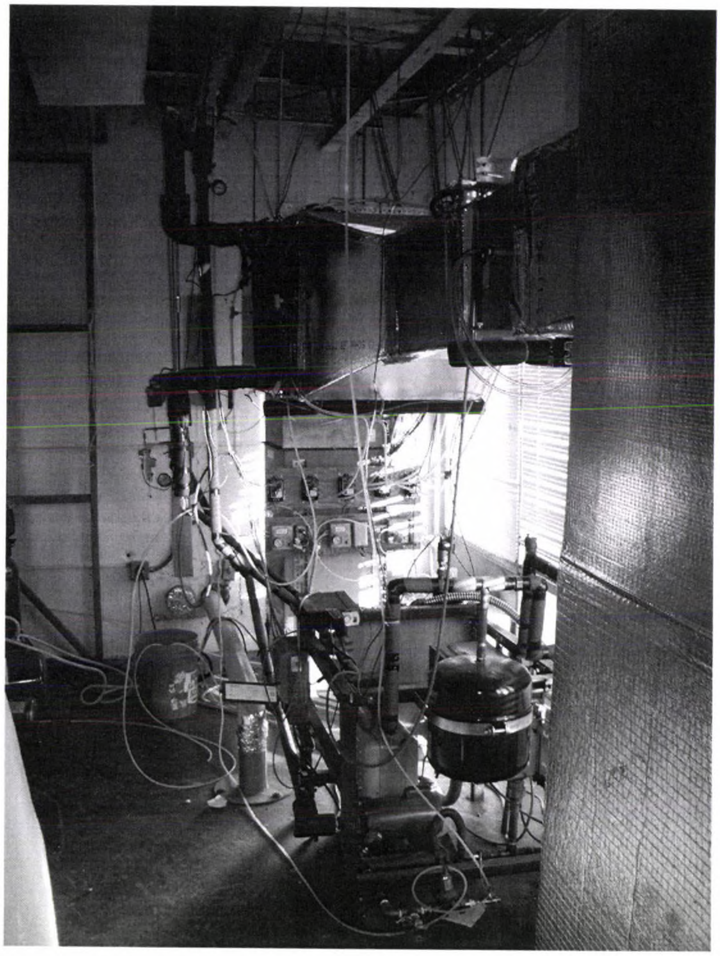

Figure 2.1: The *REPEAT HVAC* system located at the foothills campus of Colorado State University.

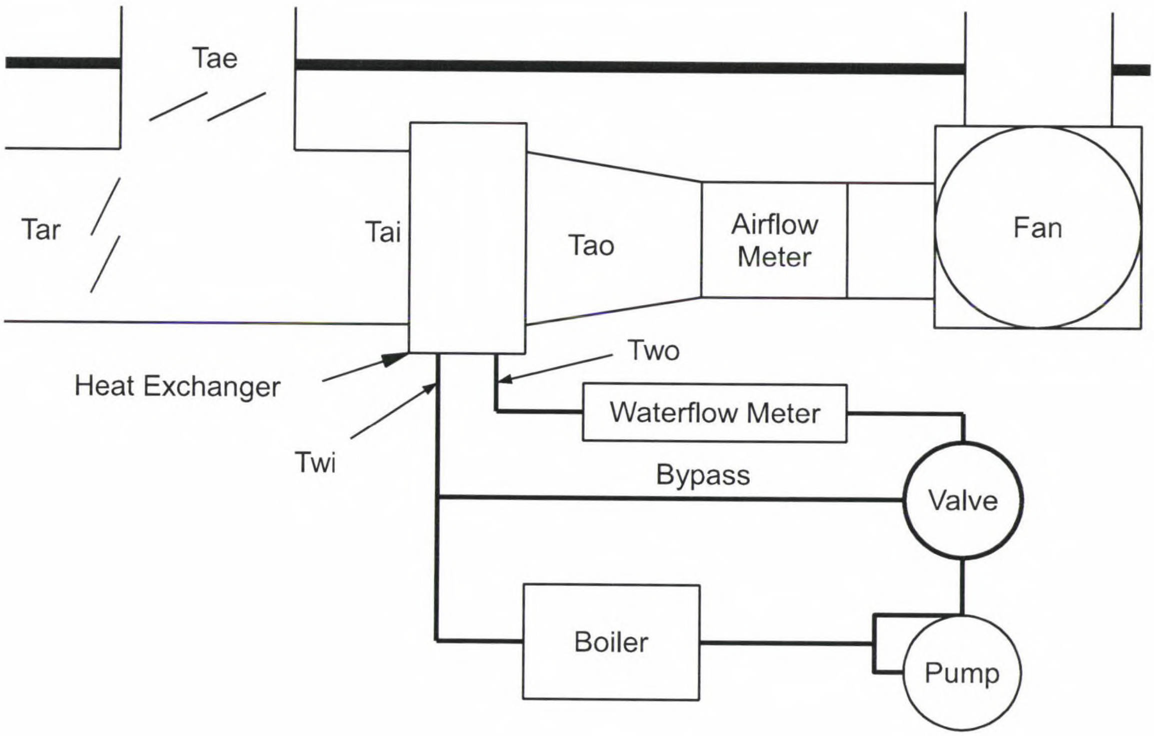

Figure 2.2: A Schematic of the *REPEAT HVAC* system. The main components of the system are shown as well as the location of temperature measurements.

2.2 Air Side

Outdoor air was drawn into the system where it was mixed with return air from inside the building. The fraction of outdoor and return air was regulated by pneumatically actuated dampers. The mixed stream was heated by a water to air heat exchanger. The air velocity was measured by an air flow meter downstream of the heat exchanger. The variable speed fan discharged the air. The majority of the air was discharged outside the building, but a fraction was discharged into the building.

2.2.1 Sensors

Air temperature was measured in four locations: where the outdoor air enters the system T_{ae}, where the return air enters the system T_{ar}, just upstream of the heating coil T_{ai}, and just downstream of the heating coil T_{ao}. The flow rate of the air was measured in one location downstream of the heating coil f_a.

2.2.2 Actuators

The air flow was controlled by three actuators: the fan, the return air damper, and the outdoor air damper.

2.2.2.1 Fan

The fan was located down stream of the heat exchanger and pulled air through the system before discharging most of the air out of the building. The fan was controlled by a variable frequency motor controller. The commanded speed of the fan (blower) supplied by the control computer is referred to as C_{bs}.

2.2.2.2 Dampers

There were air flow control dampers, the outdoor air damper and the return air damper. Figure 2.3 shows a picture of the return air damper in a partially open position. The position of each damper was controlled by a pneumatic valve. The position of each damper controls the mixing of outdoor and return air. The commanded position of the return damper supplied by the control computer is referred to as C_{dr}. The commanded position of the

Figure 2.3: The return air damper in a partially open position. The return air RTD (temperature sensor) is also visible.

outdoor air dampers is referred to as C_{de}. In this project the return air damper and outdoor air damper were electronically ganged by the control computer so that opening the return air damper resulted in closing the outdoor air damper. Only the position of the return air damper is discussed in the rest of this document.

2.3 Water Side

Water was continuously pumped through the system by a centrifugal pump, see figure 2.4. Water flowed through the boiler where it was heated. The heating elements were controlled with PWM via a solid state switch[11]. After leaving the boiler the water was split into two streams. One stream passed through the heat exchanger where it was cooled by the air. The flow rate of the water going through the heat exchanger was measured by the coriolis flow meter. The other water stream by-passed the heat exchanger. The two water streams recombined downstream of the heat exchanger where a pneumatically controlled valve was used to control the flow rate of water that flowed through the heat exchanger.

2.3.1 Sensors

For this project water temperatures were measured in two locations: just upstream of the heating coil T_{wi}, and just downstream of the heating coil T_{wo}. The flow rate of the water through the heat exchanger was measured just downstream by a coriolis flow meter. The measured flow rate of water is referred to as f_w. Figure 2.5 shows a picture of the water flow meter.

2.3.2 Actuators

The water flow rate through the heat exchanger was controlled by a pneumatic valve. The commanded position of the valve supplied by the control computer is refereed to as C_{vp}. Figure 2.6 shows a picture of the water flow control valve. The temperature of the water leaving the boiler was controlled by the electric elements in the heater. The commanded

[11]Chromalox 7750-2-090-1-01

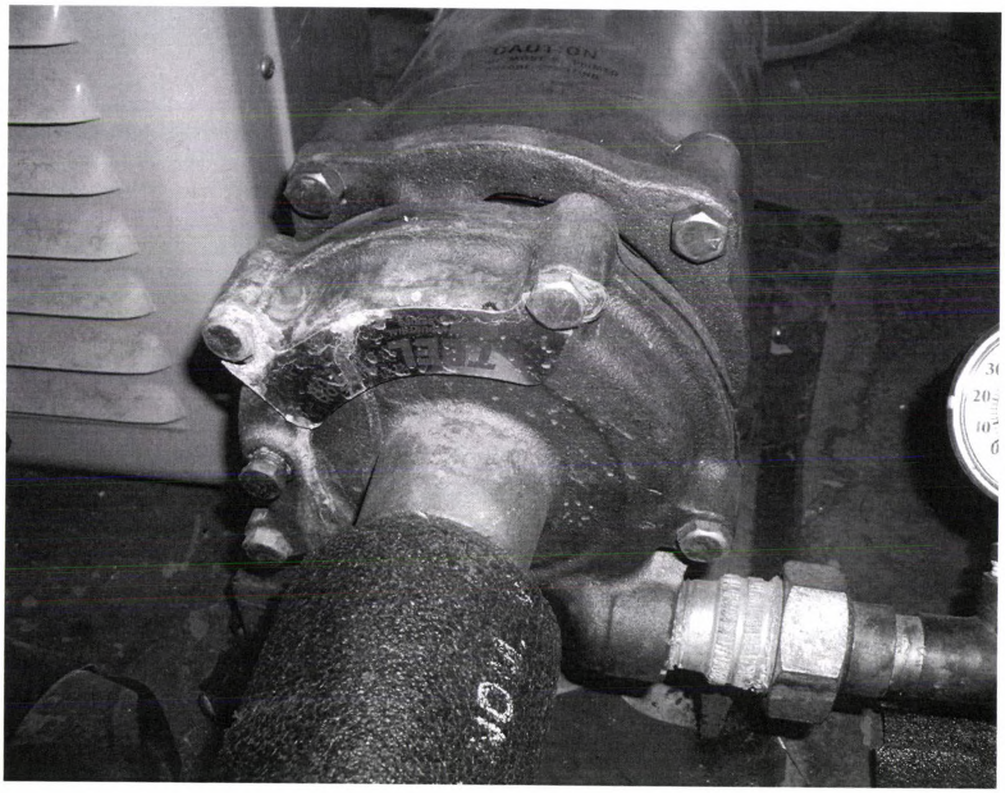

Figure 2.4: The centrifugal pump continuously circulated water through the system.

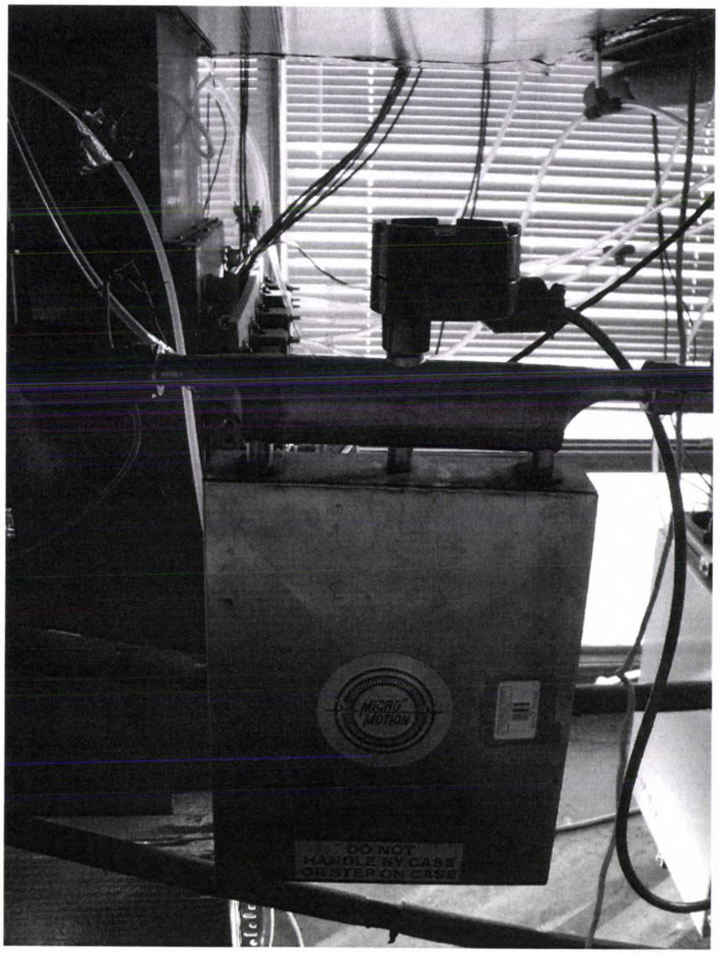

Figure 2.5: The water flow meter measured the flow rate of water through the heat exchanger.

power of the heating elements supplied by the control computer is referred to as C_{pwh}. Figure 2.7 shows a picture of the water heater.

2.4 System Configuration

For this project there were eight measurements (T_{ae}, T_{ar}, T_{ai}, T_{ao}, T_{wi}, T_{wo}, f_a, and f_w) and four control signals (C_{dr}, C_{vp}, C_{bs}, and C_{hwp}) available. Figure 2.8 shows a rough diagram of the system and indicates how the signals affect each other. This is a simplified diagram and all dependencies are not shown. For example the position of the return air damper has a minor affect of the flow rate of air. Depending on how the signals are used several different control architectures are possible.

For this project the inlet water temperature and inlet air temperature were controlled by individual proportional plus integral controllers as shown in figure 2.9. The dynamic model presented in chapter 5 was developed using this system configuration. The desired inlet water temperature is referred to as dT_{wi} and the desired inlet air temperature is referred to as dT_{ai}.

In this project multi-input-single-output (MISO) control of the water to air heat exchanger was investigated. A basic schematic of the control architecture is shown in figure 2.10. The controllers were tasked with generating a control signal, C_{vp}, such that the output air temperature of the heat exchanger (T_{ao}) tracked the desired outlet air temperature (dT_{ao}) in the presence of measured and unmeasured disturbance inputs. For this project the flow rate of air was the measured disturbance signal. Fluctuations in the water inlet temperature and air inlet temperature are considered unmeasured disturbance inputs since the controller does not have access to them.

2.5 Conclusion

This chapter has described the physical system that was used during the project. The system had some important characterizes:

1. The plant was open loop stable. Any set of steady inputs (T_{ai},T_{wi},f_w,f_a) resulted in a steady output (T_{ao}). The value of the steady output was only a function of the steady

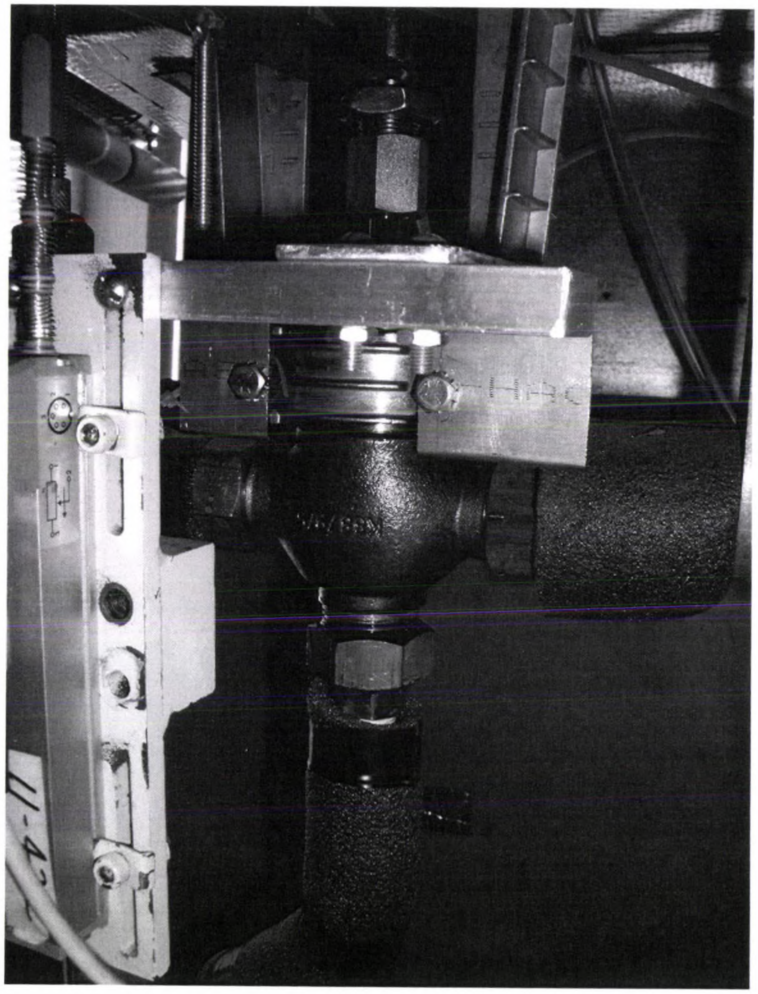

Figure 2.6: The water flow control valve. The position of the valve determines the flow rate of water through the heat exchanger. Valve was positioned by a pneumatic actuator.

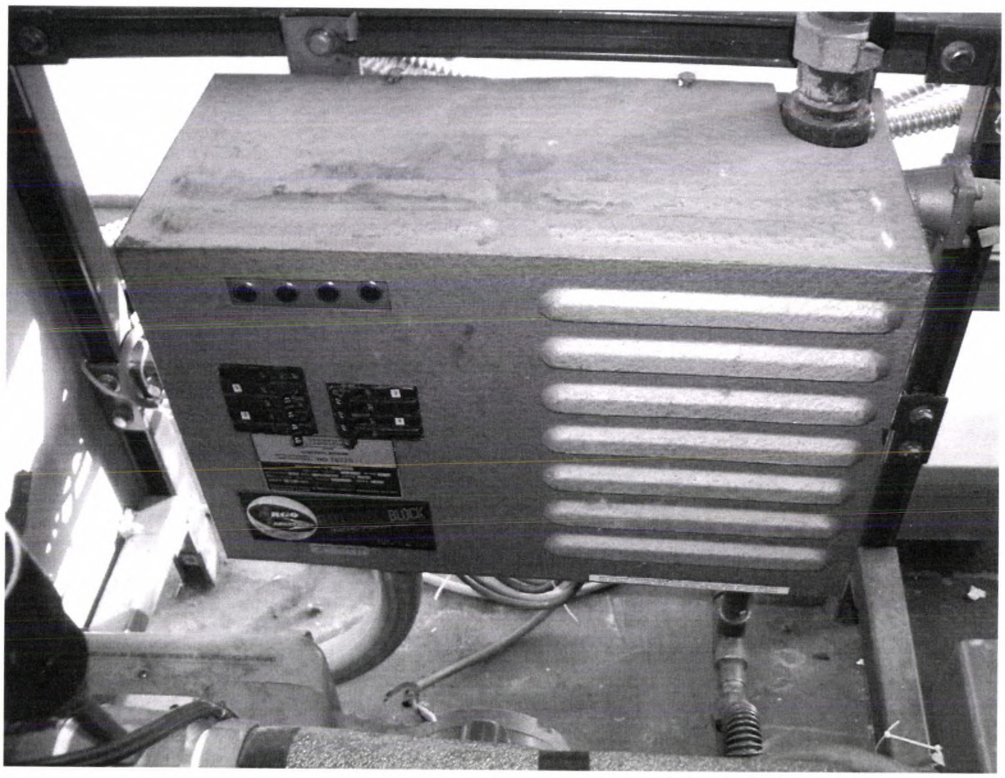

Figure 2.7: The water flowing through the system was maintained at a near constant temperature by the boiler.

inputs.

2. The plant was insensitive to initial conditions. For a given input sequence, the output converged to a single trajectory independent of the initial state of the heat exchanger.

3. There was variable gain between the control input (f_w) and the output (T_{ao}).

   a. The gain was always positive and decreased monotonically (for fixed T_{wi}, T_{ai} and f_a) as the control input increased.

   b. The gain between the control input and the output increased as f_a decreased.

   c. The gain between the control input and the output increased as T_{wi} increased.

   d. The gain between the control input and the output increased as T_{ai} decreased.

4. The plant was *overdamped*. Step changes in any input (T_{ai},T_{wi},f_w,f_a) produced an output (T_{ao}) that did not oscillate. The steady state gain from any input to the output was larger than the gain for any frequency of an oscillating input.

5. For a constant desired outlet temperature (dT_{ao}) an integral controller was capable of producing zero steady state error.

The current research project required a significant amount of data to be generated by the *REPEAT HVAC* system. To facilitate the data generation, modifications and upgrades were made to the equipment to allow for it to be controlled remotely. In addition, the system has required several repairs during the course of the project. The details of the modifications and repairs can be found in appendix A.

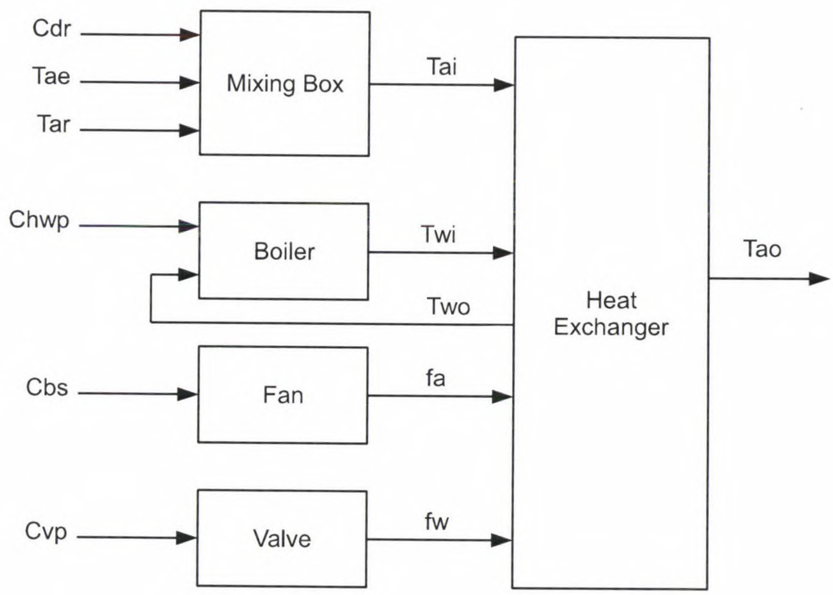

Figure 2.8: A block diagram of the *REPEAT HVAC* system which shows how the system components and signals interact.

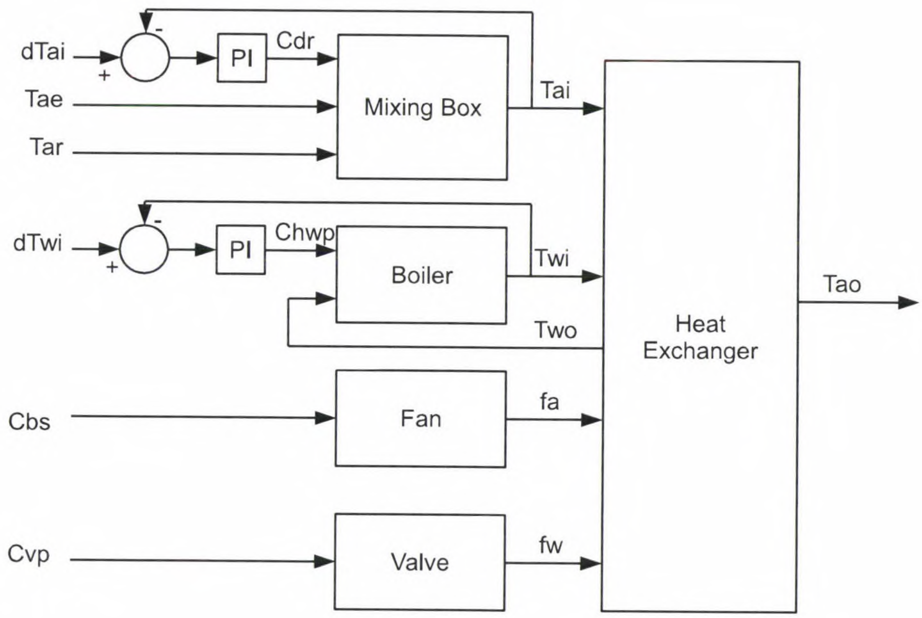

Figure 2.9: A block diagram of the *REPEAT HVAC* system. During this project the air and water inlet temperatures were controlled by independent proportional plus integral controllers.

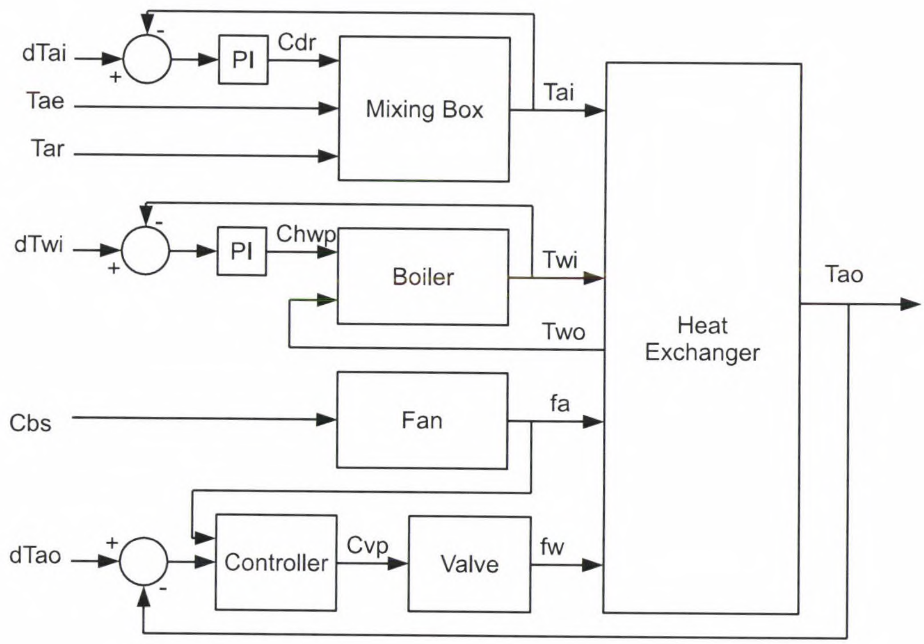

Figure 2.10: A block diagram of the *REPEAT HVAC* system. The controllers investigated in this project commanded the valve in order to control the outlet air temperature. In addition to the tracking error, the controllers had access to the flow rate of air across the coil.

Chapter 3

Review of the Dynamic Modeling and Control of Heat Exchangers

3.1 Introduction

This chapter provides a review of the research on modeling and control of heat exchangers. Section 3.2 reviews techniques for creating dynamic models of cross flow heat exchangers. Both first principle and data driven models are considered. Section 3.3 reviews the research into the control of heat exchangers. Section 3.4 is devoted to reviewing the work of Delnero et al. since that work provides a staring point for the research presented in this document.

3.2 Dynamic Modeling of Cross Flow Heat Exchangers

3.2.1 Introduction

The main component of the *REPEAT HVAC* system that will be investigated in this project is the water to air multi-pass crossflow plate-fin heat exchanger. The flow to the heat exchanger is directed such that it is in a countercurrent configuration. This section reviews the work of other researches in the study of the dynamics and control of similar heat exchangers. Section 3.2.2 reviews the work on the development of first principle models for cross-flow heat exchangers. Section 3.2.3 reviews data driven models of heat exchangers.

3.2.2 First Principle Models

The governing equations of a cross-flow plate-fin liquid-to-gas heat exchanger are comprised of the continuity, momentum, and energy equations for each fluid stream as well as the Fourier conduction equation for the core. The initial conditions are the three dimensional temperature, pressure, and flow fields in each fluid and the three dimensional temperature field in the core.[1] The boundary conditions are the pressure, temperature, and velocity of the fluids at the fluid ports, as well as the temperature of the shroud which separates the heat exchanger from it's surroundings. Typically many assumptions about the fluid flow and heat transfer in a heat exchanger are made:

1. Both fluids can be considered incompressible.

2. Viscous dissipation is negligible.

3. Conduction along the flow direction of each fluid is negligible.

4. Both fluid flows are predominantly one dimensional.

5. The liquid can be modeled as plug-flow so there is only temperature variation in the direction of flow.

6. The thermal mass of the gas in the heat exchanger can be neglected.

7. The transverse (in the fluid to fluid direction) thermal resistance of the core is negligible.

8. The conductance of the core in the direction of fluid flow is negligible.

9. The heat transfer to or from the environment surrounding the heat exchanger can be neglected.

With these assumption only three governing equations are required. The equations consist of simplified forms of the energy equation for each fluid and a simplified conduction equation for the core. The initial conditions become the one dimensional temperature distribution in liquid, and in the core. The boundary conditions become the fluid inlet flow rates and temperatures. Multipass heat exchanges, such as the one studied in this project, have

[1] If there is a change of phase in one or both of the fluids, the enthalpy of the fluid is needed instead of the temperature

increased complexity and typically they are modeled by treating each pass separately and propagating the solution to subsequent passes.

The assumptions given in this section have been the starting point for many researchers that have studied the dynamics of cross flow heat exchangers. No general solution is available and researchers have been forced to make additional assumptions and only study a limited set of boundary and initial conditions.

Gartner and Harrison developed frequency response methods for for either variations in inlet temperature [13], or variations in liquid flow rate [12]. The models were experimentally validated. The models developed are only valid around discrete operating points and are not appropriate for modeling step changes in any input variables.

Pearson et al., [28] studied the transient response of a single row tube and fin liquid to gas cross flow heat exchangers to step changes in the liquid flow rate. In the analysis the governing equations were combined to a single mixed partial differential equation. The response of the outlet gas temperature to a step change in liquid flow rate was approximated by the superposition of two first order responses. The gain and time constants can be calculated from the results of steady state tests.

Kabelac [18] incorporated variations of heat transfer coefficient with flow rates into a linear model and studied the frequency response of a multi-pass cross flow liquid to gas heat exchanger by breaking the coil into many individual lumped models. The equations developed allow for variations in flow and temperature, but the results presented only show the frequency response to variations in liquid flow rate.

Spiga and Spiga [34] used the Laplace methods to study the response of a single pass cross flow heat exchanger to step changes in inlet temperature given that the initial (temperature) conditions are all uniform. The model provides the entire two dimensional temperature profile of the heat exchanger.

Chen and Chen [7] improved the computational time of Laplace transform method of Spiga [34]. The model developed is only applicable to variations in inlet temperatures.

Romie F. E. [31] investigated the accuracy of ignoring the thermal mass of the core of a cross flow heat exchanger. Variations in inlet temperature were considered. Neglecting the core capacitance greatly increased the speed of calculations and in some cases provided accurate results.

Xuan [38] used Laplace techniques to study the effects of flow maldistributions on the transient response of multipass crossflow heat exchangers. Inlet temperatures were varied and through numerical inversion of the Laplace transforms of the solutions the temperature fields in the heat exchanger were calculated. It was concluded that maldistribution of flow can greatly reduce the performance of the heat exchanger.

As part of a guide to HVAC equipment the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) [6] reviewed the existing literature on dynamic models of coils (in 1998). The review found that all models fit two basic classes: linearized transfer function models and simplified lumped models. Furthermore, it was concluded that neither of the modeling methods was appropriate for closed loop control evaluations.

In the book [30] Roetzel and Xuan review and summarize the work in the area of heat exchanger dynamics (in 1999). They found that typically simulations of cross flow heat exchangers are achieved by either Laplace transformation or finite-difference methods. Further they found that most of the research had been focused on step changes in inlet temperature. They point out that the transient behavior of crossflow heat exchangers is more mathematically complex than other heat exchangers and thus is less often studied.

Husaunndee et al., used a lumped first order model of a heat exchanger as part of the SIMBAD [32] toolbox that they developed to simulate building energy systems in SIMULINK [1]. The coil model allows for both wet and dry conditions, which is important for cooling applications. The model was compared to experimental data for step changes in fluid flow rate.

Mishra et al., [20] developed a model of a cross flow heat exchanger in which neither stream is mixed. Finite difference methods were used to analyze how longitudinal conduction in the core and maldistribution of flow affects the transient response of the system. Step, ramp, and exponential variations in inlet temperature were studied. In a separate study [21] the same researchers studied the response of a cross flow heat exchanger with neither stream mixed to step and ramp changes in flow rates and step, ramp, and exponential changes in temperature. Both fluid streams were assumed to be fully developed and turbulent. No experimental comparisons were made.

Delnero, et al., [9, 10] presented a solution to the mixed partial differential equation given in [28] for the case of a step change in flow rate of liquid from an initial zero flow condition. The solution developed and the approximations suggested in [28] were compared to experimental data generated from the *REPEAT HVAC* system.

Syed and Idem [35] used a finite difference method to study the transient response of a cross flow heat exchanger to step changes in temperature. Optimum grid size and time steps were investigated over a range of dimensionless parameters.

3.2.3 Data Driven Models

Because of the difficulties in formulating a first principle dynamic model of a crossflow heat exchanger many researchers have developed data driven models. These models typically include feedforward or recurrent neural networks and the use of non-linear system identification techniques.

Wright et al., [37] generated data from a physical two phase shell tube heat exchanger. The flow rate of each stream was randomly changed between two values, 40% and 60% of full scale. Two sets of data were generated. In the first set of data the two flow rates were uncorrelated, in the second set of data the flow rates were correlated. Box and Jenkins techniques (combined autoregressive and moving averages) were used to create best fit transfer functions. The resulting system models are only good around the operating point.

Bittanti and Piroddi [5] created a nominal model of a saturated vapor to liquid heat exchanger. The liquid flow rate was taken as the system model and the system output was the liquid outlet temperature. The vapor temperature and the liquid inlet temperature were modeled as disturbances. The dynamics of the input actuator and output sensor were incorporated into the system model. The system was sampled at one second intervals. A non-linear ARX model was used,

$$\tilde{T}_{\text{out}}(n) = f\left(T_{\text{out}}(n-1), \ldots, T_{\text{out}}(n-4), \dot{m}(n-1), \ldots, \dot{m}(n-10)\right) \tag{3.1}$$

A two hidden layer feedforward neural network was used for $f(\cdot)$ in equation (3.1). Through trial and error a network with 14 neurons in the first hidden layer and seven neurons in the second hidden layer was used. The authors took care to ensure the training data spanned the input and output space of the system. Standard backpropagation was used to train the network. The accuracy of the predictor model was analyzed with three methods. First the one step look ahead output of the predictor was compared to the output of the nominal system model. Next the steady state outlet temperature was determined for several flow rates. Lastly the settling time of the neural network predictor and the nominal model to step changes in flow rate were compared. The neural model achieved best results when flow rates were near or above the nominal value. The authors note that the structure of (3.1) was chosen to be consistent with previous work and the amount of delay time needed depends on the operating point of the nominal model, specifically the liquid flow rate. At low flow rates the system responds slower. A longer time history is needed to produce an accurate model when the flow rate is low.

Renotte et al., [29] created a nominal model of a two tank acid cooling system. The outlet of the system was the temperature of the acid leaving the second tank. The inlet of the system was the temperature of the cooling water. The temperature of the acid entering the first tank was considered a disturbance input. Equations [3.2,3.3] show the discrete time neural state space model that was trained to predict the the output of the nominal model.

$$\tilde{x}(n) = W_{AB} \tanh\left[V_A \tilde{x}(n-1) + V_B u(n-1) + \beta_{AB}\right] \tag{3.2}$$

$$\tilde{y}(n) = W_{CD} \tanh\left[V_C \tilde{x}(n) + V_D u(n) + \beta_{CD}\right] \tag{3.3}$$

This model incorporates two single hidden layer feedforward neural networks, one for the states and one for the output. For training a model was created with two states, five hidden nodes in the state neural network, and two hidden nodes in the output neural network. The parameters of the model were trained with a simultaneous perturbation stochastic approximation (SPSA) procedure with adaptive gain sequences, step rejection, and gradient smoothing. The trained model was tested on a series of step changes to the inlet temperature. The authors report good model agreement.

Jalili-Kharaajoo et al., [17] developed a model of a counterflow water to water plate heat exchanger. The inlet temperatures of both streams and the flow rate of the cold stream were controlled to remain constant. The flow rate of the hot stream was used as the system input, and the temperature of the cold stream was the system output. A NARMAX model with a one second sampling time was developed. The inputs to the model were the three most recent hot water flow rates and the two most recent predictions of the cold stream outlet temperature,

$$\tilde{T}_{\text{out}}(n) = f\left(\dot{m}(n-1), \dot{m}(n-2), \dot{m}(n-3), \tilde{T}_{\text{out}}(n-1), \tilde{T}_{\text{out}}(n-1)\right) \tag{3.4}$$

A feedforward multilayer neural network was used for the function $f(\cdot)$ in equation (3.4). Though a feedforward neural network was used, the inputs used by the estimator include previous outputs of the estimator so the model is a recurrent neural network. The method of network training was not indicated by Jalili-Kharaajoo et al., but a properly trained network was reported to predict the outlet temperature within one Celsius degree for the test data that was used.

Tse et al., [36] constructed a physical air handling unit which contained as the central component a water to air heat exchanger. A neural network was used to create a one step predictive model of the system. The sample time used was one minute. As shown in equation (3.5) the inputs were the current inlet and outlet temperatures of the air and water, and the current and most recent flow rates of the air and water. The output of the network was the predicted outlet air and water temperatures.

$$\begin{aligned}[T_{ao}(n+1), T_{wo}(n+1)] &= f\,[T_{ai}(n), T_{ao}(n), T_{wi}(n), T_{wo}(n), \dot{m}_a(n), \dot{m}_a(n-1), \\ &\quad \dot{m}_w(n), \dot{m}_w(n-2)]\end{aligned} \tag{3.5}$$

The network had a single hidden layer of 20 nodes. The network was trained with backpropagation and a variable learning rate. The network was trained offline with data generated by varying the flow rate of water. Once trained the predictions were compared with the actual output of the system for a period of 90 minutes. The error in outlet air temperature prediction was typically under 5% for the test data presented.

3.3 Control of Heat Exchangers

3.3.1 Introduction

This section reviews the control procedures implemented by other researchers. Most, if not all, industrial applications of HVAC systems are controlled by single-input-single-output (SISO) proportional plus integral (PI) or proportional plus integral plus derivative (PID) controllers. Section 3.3.2 reviews the research into the use of PI, PID, and other linear controllers applied to heat exchangers. Researchers have begun to investigate the performance of more advanced control architectures applied to heat exchangers. The use of neural control techniques is of importance to this project. Section 3.3.3 reviews the use of neural model predictive control applied to heat exchanges. Section 3.3.4 reviews the use of neural internal model control applied to heat exchangers. Section 3.3.5 reviews other research on the advanced control of heat exchangers.

3.3.2 Linear Controllers

Hamilton et al., [15] investigated the PI control of a heating coil for different types of water flow control valves. The use of an equal percentage valve reduced the oscillations of the system at part load when compared to a system that used a linear control valve.

Anderson et al., [4, 23] developed an linear model of the *Repeat HVAC* system about a single operating point. A robust multi-input-multi-output (MIMO) controller for the system was developed that had better performance than a collection of SISO control loops that is typical of HVAC systems.

Alotaib et al., [2] created a finite difference model of a single pass water to air crossflow heat exchanger. A PI controller was designed to control the outlet air temperature by adjusting the water flow rate. The ability of the controller to reject disturbances was tested at several operating points. The system was very sensitive to disturbances at low air flow and low water flow operating points.

Haines and Hittle [14] provide guidelines for tuning PI controllers for HVAC equipment. The suggested procedure is to place the system in it's highest gain state and tune the controller to provide a critically damped response to step changes in commanded output.

The authors suggest that self tuning and/or adaptive controllers will likely by part of the next generation of HVAC control systems.

3.3.3 Neural Model Predictive Control

In model predictive control a model of the system is created and then used as a dynamic predictor for the system. The output of the controller is found by optimizing the output of the model with respect to a cost function. This section reviews the use of neural model predictive control procedures applied to heat exchangers

Lim and Ling [19] investigated the use of generalized predictive control on a shell and tube heat exchanger. Sampling times from three to fifteen seconds and prediction horizons of one to three minutes were used. In each case a third order plant model was assumed. Good control required both a small sampling interval and a large prediction horizon. This results in a large number of samples and a computationally complex optimization problem.

Bittanti and Piroddi, [5] investigated several neural controllers based on minimum variance and generalized minimum variance inverting controller methods. Steady state off-set errors occurred for both classes of controllers. To overcome this a variable gain integrator was placed in parallel to the neural controller. The gain on the integrator decreased as the magnitude of the error increased. For large errors (which would be present just after step disturbances) the integrator had a minimal affect on the control signal.

Renotte et al., [29] used a neural state space model (see section 3.2.3) as a process simulator for a model predictive controller. The model predictive optimization procedure was used to train a neural controller in an actor-critic architecture. The controller was evaluated on a simulated liquid to liquid heat exchanger.

Jalili-Kharaajoo et al., [17] compared generalized predictive control of a heat exchanger to a neural network based predictive controller. The neural network based control provided somewhat better control than the standard generalize predictive controller for the limited simulation results presented.

Pappa et al., [26] compared the performance of a standard PID controller and a neural model predictive controller in the realtime control of a counter flow heat exchanger. The input to the neural controller consisted of the ten previous control signals and the ten previous outlet temperatures. The sampling time was one second. The neural model predictive controller produced quicker response with less overshoot than the PID controller.

3.3.4 Neural Internal Model Control

Internal model control relies on the development of an model of the inverse dynamics of the system. If a perfect inverse model is available then only feedforward control is needed. In practice a feedback controller is used in parallel with the inverse model to correct for modeling errors. The section reviews the use of neural internal model control procedures applied to heat exchangers.

Diaz et al., [11] trained a multilayer neural network to invert the dynamics of a cross flow heat exchanger. The neural network was used to augment the output of PI and PID controllers. The PI and PID controllers were tuned for states with medium gain. For high gain states the PI and PID controllers resulted in large oscillations in the output signal. The augmented controller was able to reduce the amplitude of the oscillations.

3.3.5 Other Neural Control Techniques

This section reviews neural control techniques applied to heat exchangers that do not fit the model predictive control nor the internal model control architecture.

Pappa and Shanmugam [27] created a neural model of a heat exchanger. The model was used make predictions of the one step ahead outlet temperature. The actual error and the predicted error were used as inputs to two identical PID controllers whose outputs were combined to provide the control signal to the plant. The use of the predicted outlet temperature increased the controllers ability to reject disturbances.

Hepworth and Dexter [33] tested the performance of PI control on different operation points of a industrial heating coil. The controller was tuned at a high gain state and performed sluggishly in low gain states. To improve the performance a feedforward radial basis function (RBF) network was trained as a predictor of the output that would produce zero steady state errror. and then used to augment the control action of the PI controller. The RBF augmented controller was able to provide quicker response in low gain states, but some undesirable overshoot was encountered.

Anderson et al., [3] building on [33] compared PI control to two neural control systems on a numerical model of a heat exchanger. In the first system investigated a neural network was trained to predict the steady state response of a PI controller over the range of possible set point and disturbance inputs. The network was then used to augment a proportional controller. This method produced better performance than a standard PI controller. In the second investigation an actor critic reinforcement learning agent was trained on the numerical model. The reinforcement learning agent produced significantly better performance than a standard PI controller.

Delnero et al., [8, 10, 16] building on [3] augmented a PI controller with steady state predictions. The augmented controller had better performance for both step changes in the desired output and step changes in disturbance inputs. These methods are the starting point for the current project and are described in detail in section 3.4.

3.3.6 Conclusion

This section has review the research on the control of heat exchangers. The next section gives full details of the control method developed by Delnero et al. [8, 10, 16].

3.4 Augmented PI Control

3.4.1 Introduction

This section provides details of the augmented PI controller presented by Delnero et al. [8,10,16]. The control procedure combined a PI controller with a feedforward neural network. The output of the neural network was not simply added to the output of the PI controller to create the control signal. Instead, the neural network was used to adjust the integral term of the PI controller. Section 3.4.2 provides details of the controllers operation and some of the motivation for its development. Section 3.4.3 provides an overview of how the procedure was applied to the *REPEAT HVAC* system and presents a summary of the performance of the procedure.

3.4.2 The 'Stuffing' Procedure

Proportional plus integral (PI) controllers are easy to implement and are commonly used in many control applications. The output of a PI controller consists of two terms, one term that is proportional to the current error and one term term that is proportional to the time integral of the error. Equation 3.6 is a discrete time implementation of a PI controller, where $O(n)$ is the output of the controller for time step n, $e(n)$ is the error for time step n, K_P and K_I are constants and Δt is the sampling time of the controller.

$$O(n) = K_P e(n) + K_I \sum_{j=0}^{t} e(j)\Delta t \tag{3.6}$$

When a PI controller is used on a variable gain system the controller must be tuned (the values K_P and K_I are chosen) while the system is in the highest gain state it encounters. Tuning at the highest gain ensures stability over the entire operating range, but leads to slow response when the system is in a low gain region of it's state space. In low gain regions errors decay slowly while the integral term builds up.

In [8, 10, 16] Delnero et al. expanded on the work of [33] and [3]. The goal of the procedure presented by Delnero et al. was to improve the transient response of systems that have variable gain. Instead of waiting for the integral term of the controller to build up, the neural network made a prediction of what the steady-state value of the integral term would be and replaced the current value of the integral term with the predicted steady state value of the integral term. Replacing the current integral term with the predicted steady state value was referred to as 'stuffing'.

Equations 3.7 and 3.8 show the output of the controller when the integral is stuffed at time step n_0. $net(n_0)$ is the predicted steady state value of the integral for the conditions experienced at n_0.

$$\begin{aligned} O(n_0) &= K_P e(n_0) + net(n_0) && (3.7) \\ O(n > n_0) &= K_P e(n) + K_I \sum_{j=n_0+1}^{n} e(j)\Delta t + net(n_0) && (3.8) \end{aligned}$$

If the integrator is 'stuffed' again at time step n_1 the output is given by equation 3.9.

$$O(n > n_1) \quad = \quad K_P e(n) + K_I \sum_{j=n_1+1}^{n} e(j)\Delta t + net(n_1) \tag{3.9}$$

3.4.3 'Stuffing' Applied to *REPEAT HVAC*

Delnero implemented the 'stuffing' procedure on the REPEAT HVAC system. The hot water valve position was the control variable. The desired air discharge temperature was the commanded input. The measured air discharge temperature was the output. The inlet air temperature, the outdoor air temperature, the inlet water temperature, and the flow rate of the air where considered disturbance inputs.

To obtain data for training the system was controlled with a PI controller. The PI controller was tuned at the high gain state of the system (low air flow and low water flow). The system was allowed to reach steady state for 100 combinations of command and disturbance inputs. This data was used to train a neural network to predict the steady state output of the integral term of the controller.

A trained network was used to 'stuff' the integral term whenever the predicted steady state value changed by more than 3% of the value previously used to 'stuff'.

The response to step set point changes and step changes in the disturbance inputs was measured for both the 'stuffing' procedure and a PI controller. In both cases the 'stuffing' procedure produced better transient response than the conventional PI controller.

3.5 Conclusion

This chapter has provided a review of other researchers work in the area of dynamic modeling and control of heat exchangers. The work of Delnero et al. is highlighted since it is the staring point for the research presented in this dissertation.

Chapter 4

A Nonlinear First Order Controller

4.1 Motivation and Requirements For The Project

As stated in section 1.2 the goal of this project was to expand the methods of Delnero et al. [8, 10, 16]. In this section potential areas of improvement are identified and design requirements are formulated.

As stated in section 3.4 the 'stuffing' procedure achieved better performance than a standard PI controller when the *REPEAT HVAC* system was subject to step changes in set point or step changes in disturbance input. However, there were some aspects of the procedure that can lead to undesirable performance:

1. There is no way to overcome a poorly trained steady state predictor. If there is a region of the input space that produces poor steady state predictions, the integral will be 'stuffed' with an inaccurate value every time that region of the input space is encountered. This will lead to poor performance.

2. The decision about when to stuff is made solely on the output of the steady state predictor. Obviously when a poorly trained network is used this can lead to problems, but undesirable behavior can occur even when the steady state predictor is reasonably accurate. When the inputs to the steady state predictor vary continuously (rather than in steps), the output will also vary continuously. If the disturbances to the system are slowly varying, the nominal PI controller can keep the system error small even as the value of the steady state predictor moves away from the value that was last 'stuffed'. When the change in the steady state predictor value reaches the 3% threshold, the

new value is stuffed. Since the steady state prediction will never be perfect, 'stuffing' when there is low system error will likely cause a change in the control signal and lead to an increased system error.

3. The output of the steady state predictor is only used at discrete instants in time. If the value of the steady state predictor changes continuously on it's way to a value 5% from the last value 'stuffed', the integrator will be 'stuffed' once the value has changed by 3% even though the value that is a full 5% change would be more appropriate.

These potential areas of improvement lead directly to the design requirements for this project:

1. Include some form of on-line learning to overcome (or at least ignore) a poorly trained steady state predictor. (The learning problem must be formulated so that learning can be accomplished in a reasonable amount of time.)

2. Incorporate state and/or output feedback into the 'stuffing' decision process.

3. Allow for continuous input from the the steady state predictor.

4.2 Controller Development

4.2.1 Introduction

This section presents the development of a first order non-linear controller. The controller presented is a modification of the augmented PI controller developed by Delnero et al., [8, 10, 16]. Section 4.2.2 presents a continuous time formulation of the 'stuffing' procedure. Section 4.2.3 expands the procedure to allow for mixing between the steady state prediction and integral action. Section 4.2.4 introduces the error modulated partial stuffing control architecture in which the magnitude of the system error controls the mixing between the steady state prediction and integral action. Section 4.3 presents a preliminary investigation of the control architecture developed in section 4.2.4 through the control of a simple non-linear system. Section 4.4 contains an investigation of the steady state properies of the controller developed in the chapter. Section 4.6 concludes the chapter.

4.2.2 'Stuffing' in Continuous Time

This section presents a continuous time formulation of 'stuffing' procedure used by Delnero et al. Equation (4.1) is the governing equation for a classic proportional plus integral (PI) controller, where $u(t)$ is the control signal generated by the controller, $e(t)$ is the error signal that is used as the input to the controller, K_p is the proportional gain constant, and K_i is the integral gain constant.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau \tag{4.1}$$

Taken together, equation (4.2) and equation (4.3) are a state-space realization of a PI controller. In this case the state, θ, is equal to the integral of the error signal multiplied by the integral gain constant.

$$\begin{aligned} \frac{d\theta(t)}{dt} &= K_i e(t) && (4.2) \\ u(t) &= \theta(t) + K_p e(t) && (4.3) \end{aligned}$$

Delnero et al. created a control strategy that used a steady state prediction to augment a PI controller. Equation (4.4) is a continuous time version of this controller. At discrete instants in time the state of the controller (θ) is replaced by an estimate ($\tilde{\theta}_{ss}$) of the steady state value that the state will have. t_{stuff} denotes the most recent time that the steady state prediction of the controller state was used.[1] The estimate is based on the measurements of disturbance inputs (including the desired system output) at t_{stuff}.

$$u(t) = K_p e(t) + K_i \int_{t_{\text{stuff}}}^t e(\tau)d\tau + \tilde{\theta}_{\text{ss}}\left(t_{\text{stuff}}\right) \tag{4.4}$$

Examination of the terms in equation (4.4) allows for a different interpretation of the 'stuffing' procedure. The second term can be thought of as the state of the controller, θ, and the third term can be thought of as an additional input to the controller. At t_{stuff} the state of the controller is set to zero and the value of the input, $\tilde{\theta}_{\text{ss}}$, is changed.

[1] Equation (4.4) is only valid for $t > t_{\text{stuff}}$. Initially $t_{\text{stuff}} = 0$.

4.2.3 Partial Stuffing

In the original 'stuffing' procedure the value of the integrator is instantly replaced by the steady state prediction when the signal to stuff is received. This section presents the development of 'partial stuffing' procedure. Instead of having a binary 'stuff'/'do not stuff' signal, a continuous 'stuffing signal' is used. The 'stuffing signal', $\gamma(t)$, controls how quickly the state of the controller moves toward the steady state prediction.

To begin the development consider equation (4.5) which is a slight modification of equation (4.4),

$$u(t) = K_p e(t) + K_i \int_{t_{\text{stuff}}}^{t} e(\tau)d\tau + \tilde{\theta}_{\text{ss}}(t) \tag{4.5}$$

In equation (4.5) the current output of the steady state predictor, $\tilde{\theta}_{\text{ss}}(t)$, is used rather than the value at the last 'stuffing' instant. The control law of equation (4.5) is continuously 'stuffing', but resets the integrator at discrete times, thus the most current value of the steady state predictor is always used. Equation (4.5) satisfies requirement 3 of section 4.1.

To implement partial stuffing the binary decision to 'stuff' needs to be replaced with a 'stuffing signal', $\gamma(t)$, which controls how quickly the state of the controller moves toward the steady state prediction. Equations (4.6) and (4.7) are a state space realization of the control law.

$$u(t) = \theta(t) + K_p e(t) \tag{4.6}$$

$$\frac{d\theta(t)}{dt} = K_i e(t) + \gamma(t)\left[\tilde{\theta}_{\text{ss}}(t) - \theta(t)\right] \tag{4.7}$$

The state of the controller is determined by the combined effect of integrating the error and 'decaying' towards the steady state prediction. $\gamma(t)$, is the inverse of the instantaneous time constant of the decay. When $\gamma(t)$ is zero the state tracks the integral of the error, as $\gamma(t)$ approaches infinity the state tracks the value of the steady state predictor. Equations (4.8) and (4.9) show the standard matrix form of the state space representation of the control law.

$$\frac{d\theta(t)}{dt} = [-\gamma(t)]\,[\theta(t)] + \begin{bmatrix} K_i & \gamma(t) \end{bmatrix} \begin{bmatrix} e(t) \\ \tilde{\theta}_{\text{ss}}(t) \end{bmatrix} \tag{4.8}$$

$$u(t) = [1]\,[\theta(t)] + \begin{bmatrix} K_p & 0 \end{bmatrix} \begin{bmatrix} e(t) \\ \tilde{\theta}_{\text{ss}}(t) \end{bmatrix} \tag{4.9}$$

Figure 4.1 shows the block diagram of a partial stuffing controller.

If γ is assumed to be a constant, a transfer function representation of the control law can be formulated. The transfer function from the system error to the control signal and from the steady state prediction to the control signal are shown in equations (4.10) and (4.11), respectively.

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s+\gamma} = \frac{K_p s + (K_i + \gamma\, K_p)}{s+\gamma} \tag{4.10}$$

$$\frac{U(s)}{\tilde{\Theta}_{\text{ss}}(s)} = \frac{\gamma}{s+\gamma} \tag{4.11}$$

Because of the time variation in γ, the control law is not LTI and the transfer functions can not be used to model the system, however some insight into the behavior of the control law can be made by examining the transfer functions. Equation (4.11) acts like first order low pass filter with unity steady state gain between the steady state predictor and the the control signal. The second term (in the middle expression) of equation (4.10) is also a filter. Since it's steady state gain depends on it's time constant, it behaves as a leaky integrator.

4.2.4 Error Modulated Partial Stuffing

Figure 4.2 shows a block diagram of a control architecture in which the 'stuffing signal' is created by multiplying the magnitude of the system error by a gain, K_e. With this controller the steady state predictions have more influence when the system error is large and less influence when the system error is small. Equations (4.12) and (4.13) show a state space representation of this controller.

$$\frac{d\theta}{dt} = K_i e + K_e |e| (\tilde{\theta} - \theta) \tag{4.12}$$

$$u = K_p e + \theta \tag{4.13}$$

The architecture of figure 4.2 and equations (4.12) and (4.13) is referred to as error modulated partial stuffing throughout the rest of this dissertation.

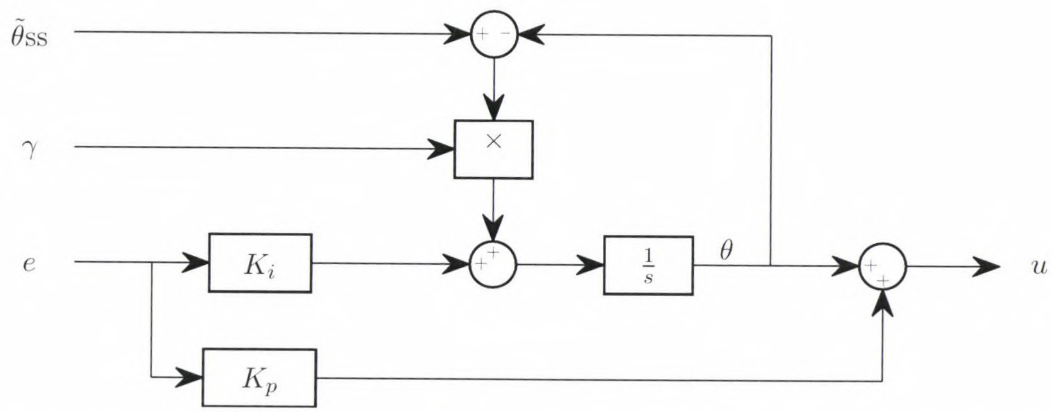

Figure 4.1: The block diagram of a 'partial stuffing' controller. e is the system error. u is the output of the controller. θ is the state of the controller. $\tilde{\theta}_{ss}$ is the predicted value of the state of the controller that will produce zero error at steady state. γ is the 'stuffing signal' which controls the degree to which the output of the controller follows the steady state predictions.

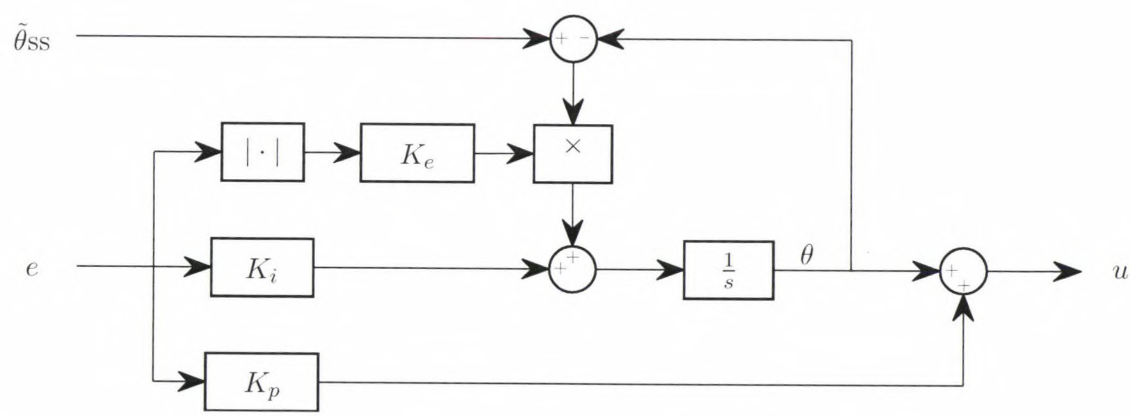

Figure 4.2: The block diagram of a error modulated 'partial stuffing' controller. e is the system error. u is the output of the controller. θ is the state of the controller. $\tilde{\theta}_{ss}$ is the predicted value of the state of the controller that will produce zero error at steady state. The magnitude of the system error and the gain K_e control the degree to which the output of the controller follows the steady state predictions.

4.3 Error Modulated Partial Stuffing Applied to a Simple Non-Linear System

4.3.1 Plant Model

In this section the tracking performance of an error modulated partial stuffing controller applied to a simple nonlinear second order plant is investigated. Equations (4.14) and (4.15) describe the nonlinear second order system that is used as the plant in this investigation.[2] $f(t)$ is the system input and $x(t)$ is the system output.

$$f(t) = \ddot{x} + 2\dot{x} + \frac{1}{2}k(x)x \tag{4.14}$$

$$k(x) = \begin{cases} x & \text{for } x \geq 0.1 \\ 0.1 & \text{for } x < 0.1 \end{cases} \tag{4.15}$$

For this investigation the plant operates such that typically $0.1 < x < 1.2$. Around a steady state operating point the system behaves as a linear second order system. The value of the output, x, at the operating point determines the local dynamics. Table 4.1 shows the second order characteristics of the system at the extremes of the typical operating range. K_{ss} is the steady state gain, ω_n is the natural frequency, and ζ is the damping ratio. For most of the operating range the system behaves locally as an overdamped system.

Table 4.1: Parameters of the linearized plant for two operating points.

x	0.1	1.2
K_{ss}	10	0.833
ω_n	0.316	1.10
ζ	3.16	0.913

4.3.2 PI Control

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau \tag{4.16}$$

As a baseline a PI controller, equation (4.16), was designed to control the non-linear plant. Selecting appropriate values of the controller gains K_p and K_i involved tradeoffs in performance around different operating points. Reasonable performance was achieved for the

[2]Physically the plant could represent a mass-damper-spring system with a hardening spring.

entire range of operating points for $K_p = 1$ and $K_i = 0.25$. Figure 4.3 shows the normalized step response of the plant model linearized around the operating points $x = 0.1$ and $x = 1.2$.

Simulink [1] was used to test the performance of the nominal PI controller. The system was required to track a series of step changes in desired output. Each step change had a duration of 100 seconds and there were a total of 100 step changes in the simulation. The value of each step was uniformly chosen from [0.1,1.2]. The RMS error for the entire simulation was 0.064 when the system was controlled by the nominal PI controller.

4.3.3 Error Modulated Partial Stuffing Control

The error modulated partial stuffing controller of section 4.2.4 was used to control the system for values of K_e ranging from one to ten. For each value of Ke the system was presented the same sequence of step changes used to test the nominal PI controller. Since the plant model is known, it was possible to calculate the control signal required to produce zero steady state error. At each time step the actual steady state control signal plus an error chosen uniformly from $\pm 20\%$ was presented to the controller as the steady state prediction signal. For each value of K_e the exact same sequence of step inputs and the exact same sequence of steady state predictions were used.

Figure 4.4 shows the trajectory of the system for two consecutive steps of the simulations for $K_e = 1$ and $K_e = 10$. For the first step shown there happened to be a significant amount of error in the steady state prediction. For $K_e = 10$ there was a balance between the affect of the steady state prediction and integral action which resulted in a large steady state error. When $K_e = 1$ the system was less affected by the steady state prediction and the controller brought the system to zero steady state error. For the second step shown in figure 4.4 the steady state prediction was more accurate and a $K_e = 10$ resulted in a fast response with zero steady state error. $K_e = 1$ also resulted in zero steady state error, but the response was slower.

Figure 4.5 shows the calculated RMS error for each value of K_e used. A value of $K_e = 3$ produced the best results over the entire simulation. For this value of K_e there was a balance between speed of response and steady state error over the entire input sequence. The best

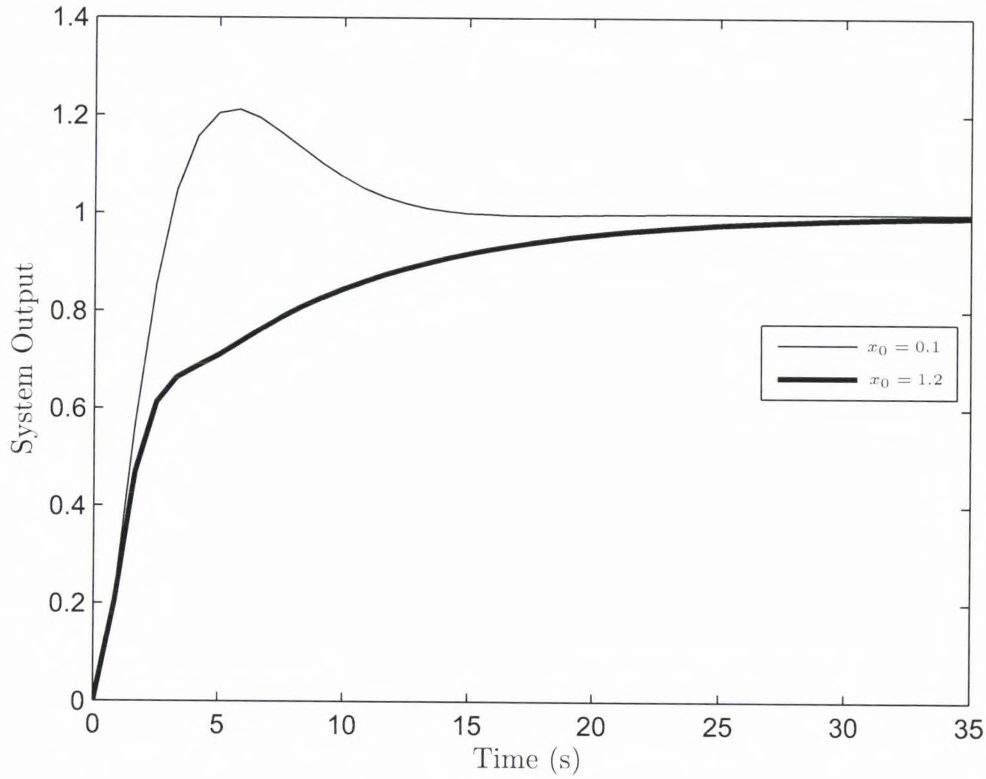

Figure 4.3: Normalized step response of the linearized system model under nominal PI control for two operating points. Around $x = 0.1$ the plant gain is high, the overshoot is just over 20%, and the settling time is about 13 seconds. Around $x = 1.2$ the plant gain is low, there is no overshoot, and the settling time is just over 25 seconds. Increasing the controller gains would increase the overshoot for the high gain case, decreasing the gains would increase the settling times for both the high gain and low gain cases.

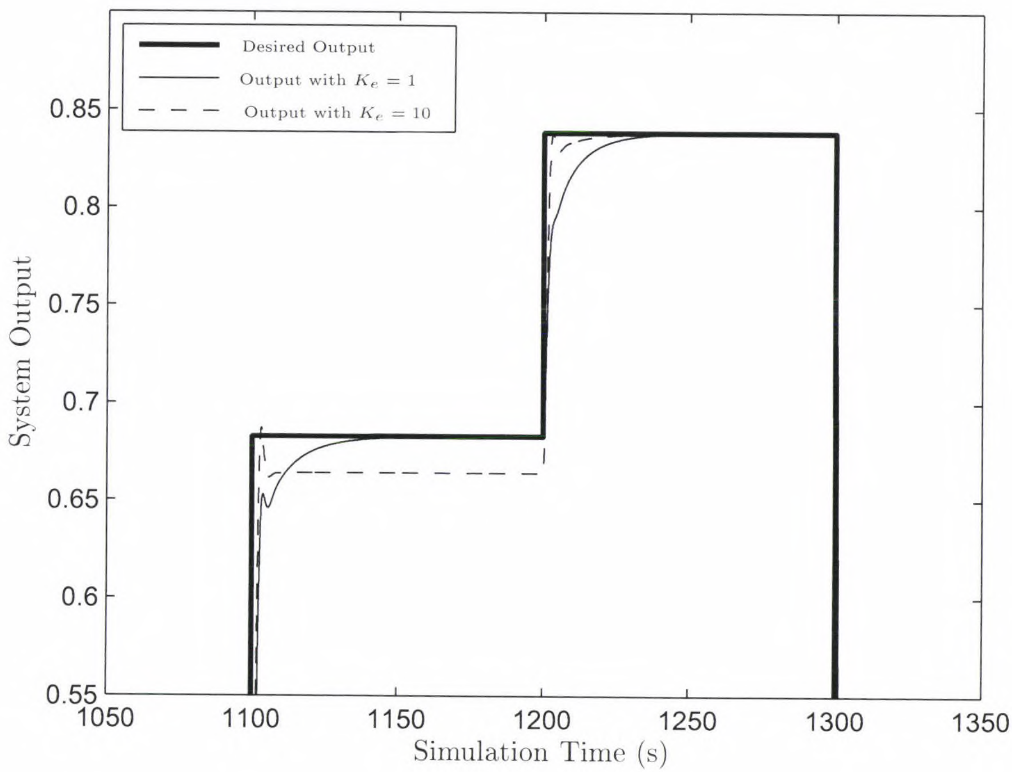

Figure 4.4: Response of the system for two different values of K_e. For the first step there was a large error in the steady state prediction and the smaller value of K_e produced less error. For the second step there was less error in the steady state prediction and the larger value of K_e produced less erro

value of K_e depends on the system and on the input sequence. If the input sequence had consisted of longer duration steps then steady state errors would have contributed more to the average error experience during each step. Smaller values of K_e which produce little or no steady state error would have performed better.

4.4 Steady State Errors with Error Modulated Partial Stuffing Control

One of the main reasons integral control has such widespread appeal is because it ensures zero steady state error for a wide class of linear and non-linear systems. The experiment of section 4.3.3 has shown that error modulated partial stuffing control can introduce steady state errors. In this section the magnitude of the steady state error is analyzed.

$$\frac{d\theta}{dt} = K_i e + K_e |e|(\tilde{\theta} - \theta) \tag{4.17}$$

$$u = K_p e + \theta \tag{4.18}$$

$$y = f(u) \tag{4.19}$$

Equation (4.17), (4.18), and (4.19) are the governing equations of a an error modulated partial stuffing controller connected to system, f. y is the output of the system and let r be the reference tracking input. Then the system error becomes $e = r - y$. If for a steady input $r = r_{ss}$ the combined system is at steady state then all the signals have steady state values and the derivative of the state of the controller is zero. The steady state equations become:

$$0 = K_i e_{ss} + K_e |e_{ss}|(\tilde{\theta} - \theta_{ss}) \tag{4.20}$$

$$u_{ss} = K_p e_{ss} + \theta_{ss} \tag{4.21}$$

$$y_{ss} = K_{ss} u_{ss} \tag{4.22}$$

Where K_{ss} is the steady state gain of the controlled system f. In equation (4.22) y_{ss} can be replaced by $r - e_{ss}$ and equation (4.20) can be used to replace u_{ss}, resulting in:

$$r_{ss} - e_{ss} = K_{ss}(K_p e_{ss} + \theta_{ss}) \tag{4.23}$$

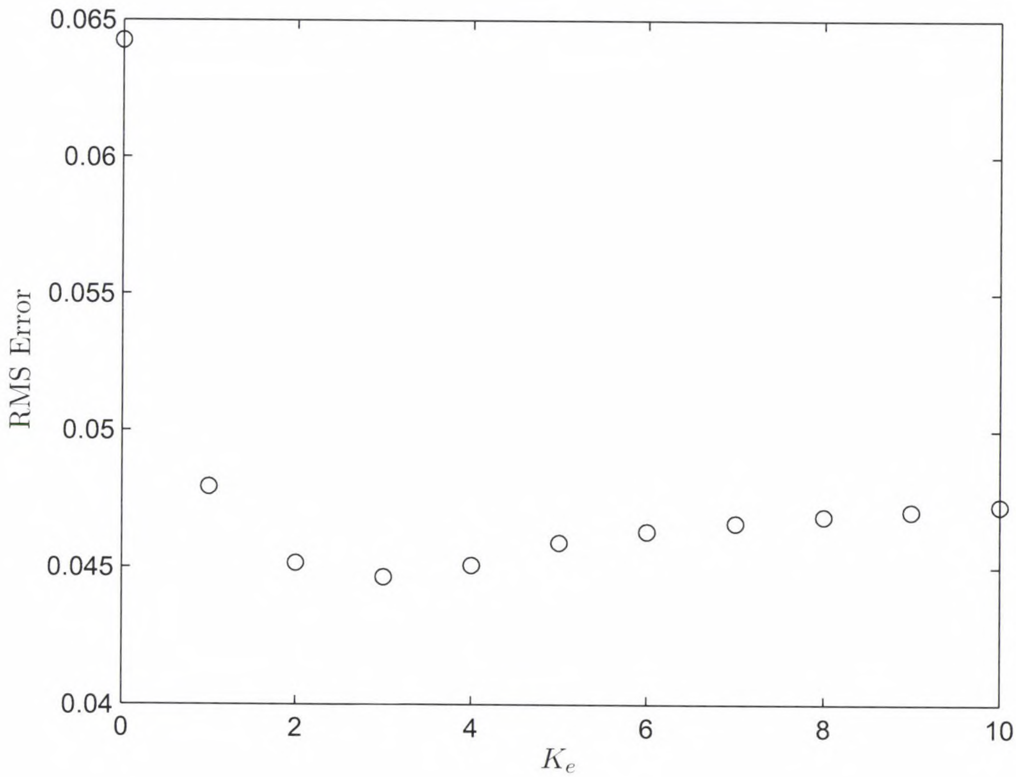

Figure 4.5: RMS error of the entire simulation for different fixed values of the gain K_e. A value of $K_e = 3$ minimized the error over the entire simulation.

Solving for θ_{ss},

$$\theta_{ss} = \frac{r_{ss} - e_{ss} - K_{ss}K_p e_{ss}}{K_{ss}} \tag{4.24}$$

Substituting equation (4.24) into equation (4.20) and rearranging terms,

$$0 = K_i e_{ss} + K_e|e_{ss}|\left(\tilde{\theta} - \frac{r_{ss}}{K_{ss}}\right) + K_e|e_{ss}|e_{ss}\left(\frac{1 + K_{ss}K_p}{K_{ss}}\right) \tag{4.25}$$

For zero steady state error $y_{ss} = r_{ss}$ so the term $\frac{r}{K_{ss}} = \frac{y_{ss}|_{e_{ss}=0}}{K_{ss}} = u_{ss}|_{e_{ss}=0}$ is the control signal that will produce zero steady state error. Since there is no proportional term when there is zero error,

$$\frac{r_{ss}}{K_{ss}} = \theta^\star \tag{4.26}$$

Where $\theta^\star$ is the value of the state of the controller that produces zero steady state error for the steady input r_{ss}. The magnitude $\left|\tilde{\theta} - \theta^\star\right|$ is the accuracy of the steady state prediction.

Substituting $\theta^\star$ and Rewriting (4.25),

$$0 = K_i e_{ss} + K_e|e_{ss}|\left(\tilde{\theta} - \theta^\star\right) + K_e|e_{ss}|e_{ss}\left(\frac{1 + K_{ss}K_p}{K_{ss}}\right) \tag{4.27}$$

Equation (4.27) is satisfied if $e_{ss} = 0$. There are two additional solutions. One for $e_{ss} > 0$, and one for $e_{ss} < 0$. For the case of positive steady state error equation (4.27) can be rewritten by removing the common factor of e_{ss} from each term,

$$0 = K_i + K_e\left(\tilde{\theta} - \theta^\star\right) + K_e e_{ss}\frac{1 + K_{ss}K_p}{K_{ss}} \tag{4.28}$$

Solving for e_{ss},

$$e_{ss} = \frac{\left(\tilde{\theta} - \theta^\star\right) - \frac{K_i}{K_e}}{\frac{1}{K_{ss}} + K_p} \tag{4.29}$$

Noting that e_{ss} is assumed to be positive in equation (4.29), the equation is inconsistent if $\left(\tilde{\theta} - \theta^\star\right) < \frac{K_i}{K_e}$. This means that if the steady state prediction is accurate enough or the gain K_e is small enough compared to K_i there will not be positive steady state error.

Analogous algebra for the case of $e_{sss} < 0$ gives,

$$e_{ss} = \frac{\left(\tilde{\theta} - \theta^\star\right) + \frac{K_i}{K_e}}{\frac{1}{K_{ss}} + K_p} \tag{4.30}$$

Again the equation is potentially inconsistent. If $\left(\tilde{\theta} - \theta^\star\right) > -\frac{K_i}{K_e}$ there will not be negative steady state error if the steady state prediction is accurate enough or if the gain K_e is small enough when compared to K_i. By taking magnitudes in equations (4.29) and (4.30), the equations can be combined. Together they put a bound on the steady state error based on the accuracy of the steady state prediction and the gains of the controller.

$$|e_{ss}| = \frac{|\tilde{\theta} - \theta^{\star}| - \frac{K_i}{K_e}}{\frac{1}{K_{ss}} + K_P} < \frac{|\tilde{\theta} - \theta^{\star}| - \frac{K_i}{K_e}}{K_P} \tag{4.31}$$

In addition, as stated in (4.32) if the magnitude prediction error is small enough there will not be positive or negative steady state error.

$$\left|\tilde{\theta} - \theta^{\star}\right| < \frac{K_i}{K_e} \Rightarrow \text{Positive and negative steady state error not possible.} \tag{4.32}$$

It is important to note that care was taken when analyzing the system error. Though bounds were developed for the magnitude of the steady state error the analysis required an assumption that steady state was reached. This assumption may not be valid for all systems and all choices of gains. For a given system it is quite possible that the combination of the controller and the system is unstable, or exhibits cyclical or aperiodic behavior.

4.5 Stability of Error Modulated Partial Stuffing Control of a Linear First Order System

In this section the stability of a first order linear system controlled by an error modulated partial stuffing controller is analyzed. Equations (4.33) and (4.34) are the state space representation of a first order plant. y is the output of the plant, the control signal u is the input to the plant, x is the state of the plant, K is the steady state gain of the plant, and τ is the time constant of the plant.

$$\frac{dx}{dt} = \frac{1}{\tau}(u - x) \tag{4.33}$$

$$y = Kx \tag{4.34}$$

Since the output is proportional to the state it is convenient to combine equations (4.33) and (4.34) into a single equation as shown in equation (4.35).

$$\frac{dy}{dt} = \frac{1}{\tau}(Ku - y) \tag{4.35}$$

The governing equations of the error modulated partial stuffing controller, (4.12) and (4.13), and the linear plant, (4.35), can be combined to form the coupled differential equations that describe the composite system of the controller and the plant.

$$\frac{d\theta}{dt} = K_i(r-y) + K_e|r-y|\left(\tilde{\theta}-\theta\right) \tag{4.36}$$

$$\frac{dy}{dt} = \frac{1}{\tau}\left\{K\left[K_p(r-y)+\theta\right]-y\right\} \tag{4.37}$$

At equilibrium the state variables (θ and y) and the input r will have steady-state values (θ_{SS}, y_{SS}, and r_{ss}), and the time derivatives of the state variables will be zero.

$$0 = K_i(r_{ss}-y_{\mathrm{SS}}) + K_e|r_{ss}-y_{\mathrm{SS}}|\left(\tilde{\theta}-\theta_{\mathrm{SS}}\right) \tag{4.38}$$

$$0 = \frac{1}{\tau}\left\{K\left[K_p(r_{ss}-y_{\mathrm{SS}})+\theta_{\mathrm{SS}}\right]-y_{\mathrm{SS}}\right\} \tag{4.39}$$

There are three potential equilibrium conditions. If the steady state output, y_{SS}, is equal to the input, r_{ss}, then equation (4.38) is satisfied independent of θ and equation (4.39) can be used to solve for θ_{SS}. For this equilibrium condition the integral control has overcome the disturbance caused by the steady state prediction and driven the system error to zero. A second possibility is that contribution from the steady state predictor and the integral control are equal and opposite so the system reaches a steady state with a non-zero error. Because the absolute value of the error is used in equation (4.38), it is possible that equilibrium will exist for a positive error and/or a negative error.

For convenience, a change of variables is used so the new state variables have a value of 0 at the equilibrium point.

$$\hat{\theta} = \theta-\theta_{\mathrm{SS}} \tag{4.40}$$

$$\hat{y} = y-y_{\mathrm{SS}} \tag{4.41}$$

$$f_1 = \frac{d\hat{\theta}}{dt} = K_i\left[r_{ss}-(\hat{y}+y_{\mathrm{SS}})\right] + K_e|r_{ss}-(\hat{y}+y_{\mathrm{SS}})|\left[\tilde{\theta}-\left(\hat{\theta}+\theta_{\mathrm{SS}}\right)\right] \tag{4.42}$$

$$f_2 = \frac{d\hat{y}}{dt} = \frac{1}{\tau}\left\{K\left[K_p(r_{ss}-(\hat{y}+y_{\mathrm{SS}}))+\left(\hat{\theta}+\theta_{\mathrm{SS}}\right)\right]-(\hat{y}+y_{\mathrm{SS}})\right\} \tag{4.43}$$

Equations (4.42) and (4.43) represent the system dynamics in the new variables $\hat{y}$ and $\hat{\theta}$. The eigenvalues of the Jacobian of the system equations are used to determine the stability of the potential equilibrium points. The stability criteria of the equilibrium points for the three different cases of steady state error follow. For the first case the Jacobian is not continuous so stability is not strictly proven. The result shown is for illustrative purposes.[3]

Case 1: $e = 0$

Solving equations (4.38) and (4.39) for the case when $y_{\mathrm{SS}} = r_{ss}$ results in,

$$y_{\mathrm{SS}} = r \tag{4.44}$$

$$\theta_{\mathrm{SS}} = \frac{y_{\mathrm{SS}}}{K} = \frac{r_{ss}}{K} = \theta^\star \tag{4.45}$$

Where $\theta^\star$ as defined in section 4.4 is the value of the control signal which will result in zero steady state error for the steady input r_{ss}. Substituting equations (4.44) and (4.45) into equations (4.42) and (4.43) results in

$$f_1 = \frac{d\hat{\theta}}{dt} = -K_i\left(\hat{y}\right) + K_e|\hat{y}|\left[\tilde{\theta} - \left(\hat{\theta} + \theta^\star\right)\right] \tag{4.46}$$

$$f_2 = \frac{d\hat{y}}{dt} = \frac{1}{\tau}\left\{K\left[-K_p\hat{y} + \left(\hat{\theta} + \theta^\star\right)\right] - \left(\hat{y} + r\right)\right\} \tag{4.47}$$

Forming the terms in the Jacobian,

$$\frac{\partial f_1}{\partial \hat{\theta}} = 0 \tag{4.48}$$

$$\lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_1}{\partial \hat{y}} = -K_i + \operatorname{sign}\left(\hat{y}\right) K_e\left(\tilde{\theta} - \theta^\star\right) \tag{4.49}$$

$$\frac{\partial f_2}{\partial \hat{\theta}} = \frac{K}{\tau} \tag{4.50}$$

$$\frac{\partial f_2}{\partial \hat{y}} = -\frac{KK_p + 1}{\tau} \tag{4.51}$$

---

[3]Typically the Jacobian is evaluated at the equilibrium point, but the use of the absolute value function in the governing equations introduces the possibility of a discontinuity at the equilibrium point so, where needed, the Jacobian is evaluated in the limit as the state variables approach the equilibrium point. Note that after the change of variable the equilibrium point is always the origin.

$$\begin{pmatrix} \lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_1}{\partial \hat{\theta}} & \lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_1}{\partial \hat{y}} \\ \lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_2}{\partial \hat{\theta}} & \lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_2}{\partial \hat{y}} \end{pmatrix}$$

Calculating the eigenvalues of the Jacobian,

$$0 = (0-\lambda)\left(-\frac{KK_p+1}{\tau}-\lambda\right) - \left[-K_i + \text{sign}\left(\hat{y}\right)K_e\left(\tilde{\theta}-\theta^\star\right)\right]\left[\frac{K}{\tau}\right] \quad (4.52)$$

$$= \lambda^2 + \frac{KK_p+1}{\tau}\lambda + \frac{K}{\tau}\left[K_i - \text{sign}\left(\hat{y}\right)K_e\left(\tilde{\theta}-\theta^\star\right)\right] \quad (4.53)$$

Both eigenvalues are negative (and the equilibrium point is stable) if the last term in equation (4.53) is positive.

$$0 < K_i - \text{sign}\left(\hat{y}\right)K_e\left(\tilde{\theta}-\theta^\star\right) \quad (4.54)$$

$$\text{sign}\left(\hat{y}\right)\left(\tilde{\theta}-\theta^\star\right) < \frac{K_i}{K_e} \quad (4.55)$$

$$\text{sign}\left(\hat{y}\right)\left(\tilde{\theta}-\theta^\star\right) < \left|\tilde{\theta}-\theta^\star\right| < \frac{K_i}{K_e} \quad (4.56)$$

Equation (4.56) represents the stability criteria for equilibrium with zero steady state error. The quantity $\left|\tilde{\theta}-\theta^\star\right|$ represents the error in the steady state prediction. There is a stable equilibrium point with zero steady state error provided that the prediction error is small enough or the gain K_e is small enough compared to K_i.

Case 2: $e > 0$

Solving equations (4.38) and (4.39) for the case $r - y_{SS} > 0$ results in,

$$\theta_{SS} = \tilde{\theta} + \frac{K_i}{K_e} \quad (4.57)$$

$$y_{SS} = \frac{K\left[K_p r + \theta_{SS}\right]}{KK_p+1} = \frac{K_p r + \tilde{\theta} + \frac{K_i}{K_e}}{K_p + \frac{1}{K}} \quad (4.58)$$

Substituting equations (4.57) and (4.58) into equations (4.42) and (4.43) results in,

$$f_1 = \frac{d\hat{\theta}}{dt} = \left\{K_i - K_e\left[\left(\hat{\theta}+\frac{K_i}{K_e}\right)\right]\right\}\left[-\hat{y} + \frac{\theta^\star - \left(\tilde{\theta}+\frac{K_i}{K_e}\right)}{K_p+\frac{1}{K}}\right] \quad (4.59)$$

$$f_2 = \frac{d\hat{y}}{dt} = \frac{K}{\tau}\left[-\hat{y}\left(K_p+\frac{1}{K}\right)+\hat{\theta}\right] \quad (4.60)$$

Forming the terms of the Jacobian,

$$\frac{\partial f_1}{\partial \hat{\theta}} = -\frac{K_e\left[\theta^\star - \left(\tilde{\theta}+\frac{K_i}{K_e}\right)\right]}{K_p+\frac{1}{K}} \quad (4.61)$$

$$\frac{\partial f_1}{\partial \hat{y}} = 0 \quad (4.62)$$

$$\frac{\partial f_2}{\partial \hat{\theta}} = \frac{K}{\tau} \quad (4.63)$$

$$\frac{\partial f_2}{\partial \hat{y}} = -\frac{KK_p+1}{\tau} \quad (4.64)$$

Determining the eigenvalues,

$$0 = \left(-\frac{K_e \left[\theta^\star - \left(\tilde{\theta} + \frac{K_i}{K_e} \right) \right]}{K_p + \frac{1}{K}} - \lambda \right) \left(-\frac{K K_p + 1}{\tau} - \lambda \right) \tag{4.65}$$

$$\begin{aligned} 0 &= \lambda^2 + \left\{ \frac{K_e \left[\theta^\star - \left(\tilde{\theta} + \frac{K_i}{K_e} \right) \right]}{K_p + \frac{1}{K}} + \frac{K K_p + 1}{\tau} \right\} \lambda \\ &\quad + \frac{K K_e}{\tau} \left[\theta^\star - \left(\tilde{\theta} + \frac{K_i}{K_e} \right) \right] \end{aligned} \tag{4.66}$$

Both eigenvalues will be negative provided that the second and third term of equation (4.66) are positive. Since the second term can only be negative if the third term is negative, the third term can be used as a stability check.

$$0 < \theta^\star - \left(\tilde{\theta} + \frac{K_i}{K_e} \right) \tag{4.67}$$

$$\frac{K_i}{K_e} < \theta^\star - \tilde{\theta} \tag{4.68}$$

So there is stable equilibrium point with positive steady state error (output is lower than reference) if the steady state prediction is too small.

Case 3: $e < 0$

Solving equations (4.38) and (4.39) for the case $r - y_{\text{SS}} < 0$ results in,

$$\theta_{\text{SS}} = \tilde{\theta} - \frac{K_i}{K_e} \tag{4.69}$$

$$y_{\text{SS}} = \frac{K \left[K_p r + \theta_{\text{SS}} \right]}{K K_p + 1} = \frac{K_p r + \tilde{\theta} - \frac{K_i}{K_e}}{K_p + \frac{1}{K}} \tag{4.70}$$

Substituting equations (4.69) and (4.70) into equations (4.42) and (4.43) results in,

$$f_1 = \frac{d\hat{\theta}}{dt} = \left\{ K_i + K_e \left[\left(\hat{\theta} - \frac{K_i}{K_e} \right) \right] \right\} \left[-\hat{y} + \frac{\theta^\star - \left(\tilde{\theta} - \frac{K_i}{K_e} \right)}{K_p + \frac{1}{K}} \right] \tag{4.71}$$

$$f_2 = \frac{d\hat{y}}{dt} = \frac{K}{\tau} \left[-\hat{y} \left(K_p + \frac{1}{K} \right) + \hat{\theta} \right] \tag{4.72}$$

Forming the Jacobian,

$$\lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_1}{\partial \hat{\theta}} = \frac{K_e\left[\theta^\star - \left(\tilde{\theta} - \frac{K_i}{K_e}\right)\right]}{K_p + \frac{1}{K}} \tag{4.73}$$

$$\lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_1}{\partial \hat{y}} = 0 \tag{4.74}$$

$$\lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_2}{\partial \hat{\theta}} = \frac{K}{\tau} \tag{4.75}$$

$$\lim_{\hat{\theta},\hat{y}\to 0} \frac{\partial f_2}{\partial \hat{y}} = -\frac{KK_p + 1}{\tau} \tag{4.76}$$

Determining the eigenvalues,

$$0 = \left(\frac{K_e\left[\theta^\star - \left(\tilde{\theta} - \frac{K_i}{K_e}\right)\right]}{K_p + \frac{1}{K}} - \lambda\right)\left(-\frac{KK_p + 1}{\tau} - \lambda\right) \tag{4.77}$$

$$\begin{aligned} 0 &= \lambda^2 + \left\{\frac{K_e\left[\left(\tilde{\theta} - \frac{K_i}{K_e}\right) - \theta^\star\right]}{K_p + \frac{1}{K}} + \frac{KK_p + 1}{\tau}\right\}\lambda \\ &\quad + \frac{KK_e}{\tau}\left[\left(\tilde{\theta} - \frac{K_i}{K_e}\right) - \theta^\star\right] \end{aligned} \tag{4.78}$$

Again the third term can be used as the stability criterion.

$$0 < \left(\tilde{\theta} - \frac{K_i}{K_e}\right) - \theta^\star \tag{4.79}$$

$$\frac{K_i}{K_e} < \tilde{\theta} - \theta^\star \tag{4.80}$$

So there is stable equilibrium point with negative steady state error (output is more than reference) if the steady state prediction is too large.

Table 4.2: Summery of Equilibrium Point Stability Criteria

Case	Stability Requirement
$e = 0$	$K_i/K_e > \lvert\tilde{\theta} - \theta^\star\rvert$
$e > 0$	$K_i/K_e < \tilde{\theta} - \theta^\star$
$e < 0$	$K_i/K_e < \theta^\star - \tilde{\theta}$

Table 4.2 summarizes the stability requirements for the three potential equilibrium points. The stability conditions are mutually exclusive and cover all possible parameter

values and system inputs, except when

$$|\tilde{\theta} - \theta^{\star}| = \frac{K_i}{K_e} \tag{4.81}$$

In practice equation (4.81) can not be exactly satisfied. Therefore, when an error modulated partial stuffing controller is applied to a linear first order system *there is always a single stable equilibrium point.* The existence of a single stable equilibrium point does not guarantee the system is stable. It possible for the combined system to exhibit periodic, aperiodic, or unstable behavior.

4.6 Conclusion

The error modulated partial stuffing controller presented in this chapter allows for smooth mixing between the use of steady state predictions and integral action. In addition the controller incorporates state information, in the form of the error signal. The use of the signal error reduces the effect of the steady state predictions when the nominal PI control is able to keep the system error low. The controller satisfies project goal three from section 1.2 and design requirements one and two from this chapter. Sections 4.4 and 4.5 provide a starting point for the stability analysis of error modulated partial stuffing control, but do not provide a proof of stability.

Chapter 5

Preliminary Experimental Investigation and Dynamic Model Development of the *REPEAT HVAC* System

5.1 Introduction

The *REPEAT HVAC* system investigated in this projects consists of two major components: the hot water control valve and the water to air heat exchanger. This chapter describes the development of models for both components. Before models could be developed it was necessary to perform a preliminary experimental investigation of the system. Section 5.2 presents the results of the preliminary investigation including the development of a steady state predictive model. Section 5.3 presents the development of the dynamic models that were used to simulate the system.

5.2 Preliminary Investigation

Before the development of a dynamic model a preliminary investigation of the *REPEAT HVAC* system was carried out. The preliminary investigation aided the creation of the dynamic model presented in section 5.3 and the controllers presented in subsequent chapters. During the preliminary investigation the temperature of the inlet air was controlled such that $T_{ai} \approx 10°\text{C}$, and the temperature of the inlet water was was controlled to be $T_{wi} \approx 50°\text{C}$.

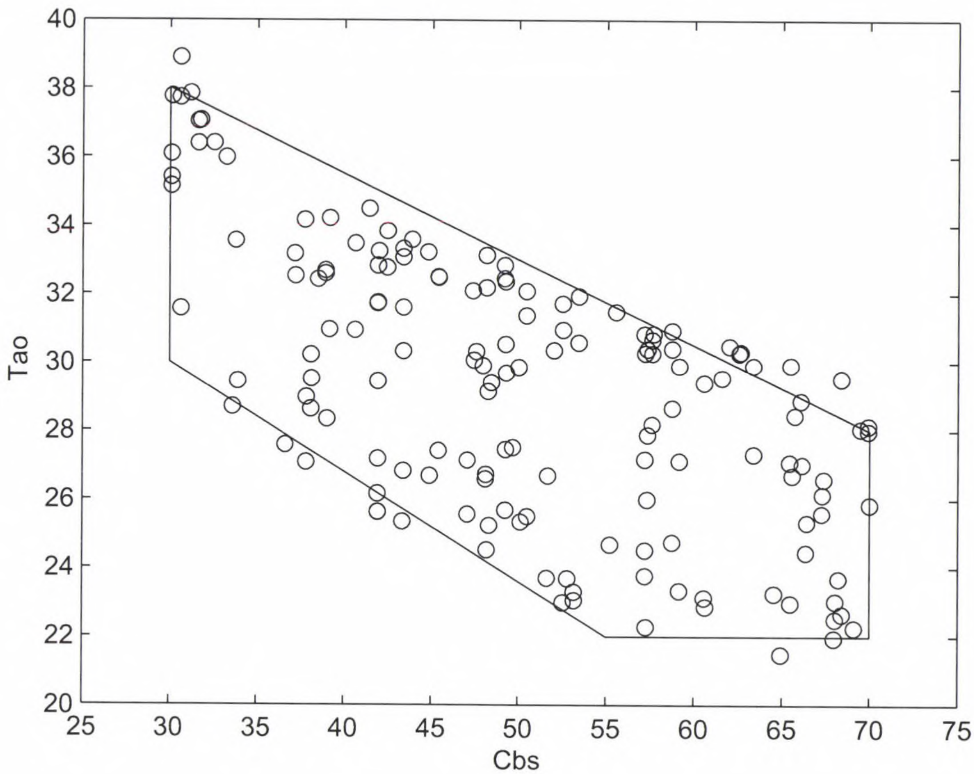

Figure 5.1: The reachable space of the *REPEAT HVAC* System. Each ∘ represents steady state outlet temperature of the coil for different values of water and air flow. In all cases the inlet air and water temperatures were controlled to be approximately 10°C and 50°C, respectively.

5.2.1 Experimental Determination of The Reachable Space

In order to evaluate controllers it is important that the input they are commanded to track is reachable. In order to determine the reachable space of the *REPEAT HVAC* system for the configuration of this project, the system was brought to steady state for 151 random values of commanded valve position and commanded blower speed (see tables B.1 and B.2 in appendix 6). Figure 5.1 shows the steady state outlet air temperature for each of the steady state test runs as a function of the commanded blower speed. The outline on the figure shows the rough piecewise linear approximation that was used to define the reachable space when the system was brought into closed loop control.

5.2.2 Experimental Determination The System Gain

Figure 5.2 demonstrates the significance of the variable gain of the system. The open loop outlet air temperature was tracked during step changes of the commanded valve position. In the high gain case the commanded fan speed was set to 30% of full flow and the valve was stepped between 5% and 15% open. In the low gain case the fan speed was set to 70% of full flow and the valve was stepped between 85% and 95% open. The high gain was measured to be 0.24C° per one percent change in commanded valve position. The low gain was measured to be 0.028C° per one percent change in commanded valve position. The ratio of the high gain to low gain was approximately 8.7.

5.2.3 Experimental Design of a Nominal PI Controller

The data generated during the gain testing was used to design a nominal PI controller. The Ziegler-Nichols method as presented in [25] was used for each step in the high gain test. Figure 5.3 shows the procedure for one step change. The tangent line to the inflection point is shown. The intersections of the tangent line with the x-axis and the level of the steady state output are used to determine the controller gains K_p and K_i. The method was applied to the ten high gain step changes shown in figure 5.2. Averages were taken over all step changes resulting in the controller gains shown in table 5.1

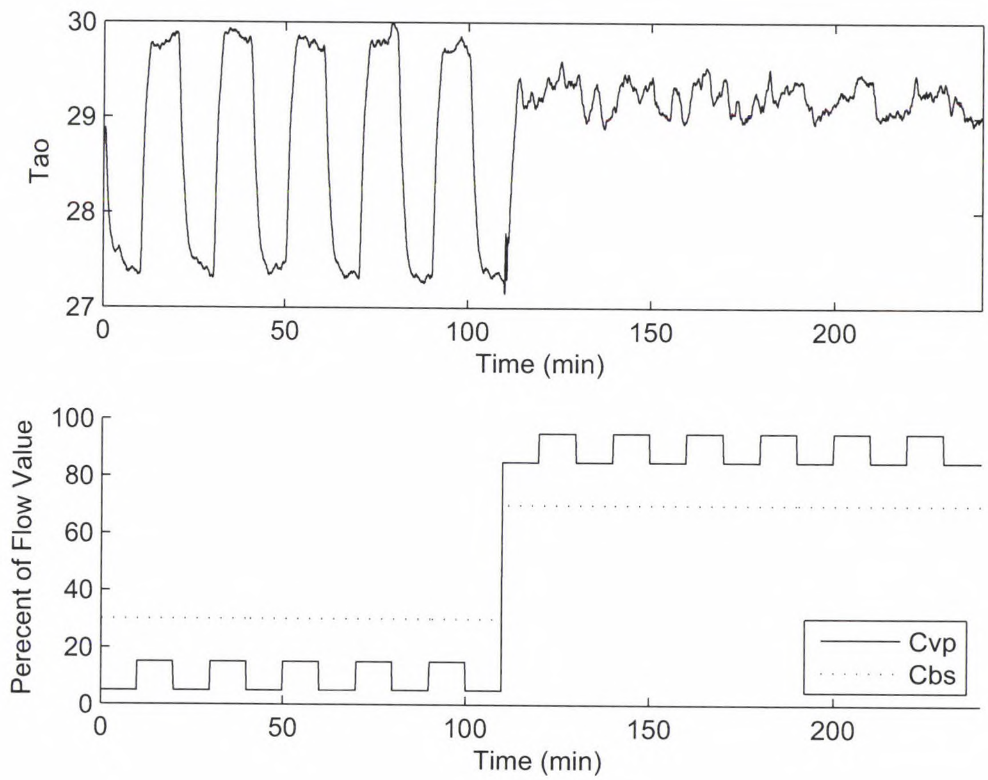

Figure 5.2: The variation in outlet air temperature for step changes in valve position. The gain from the water control valve position to the outlet air temperature is variable. When the flow rate of water and air are relatively low the system has a high gain. When the flow rate of water and air are relatively high the system has a low gain.

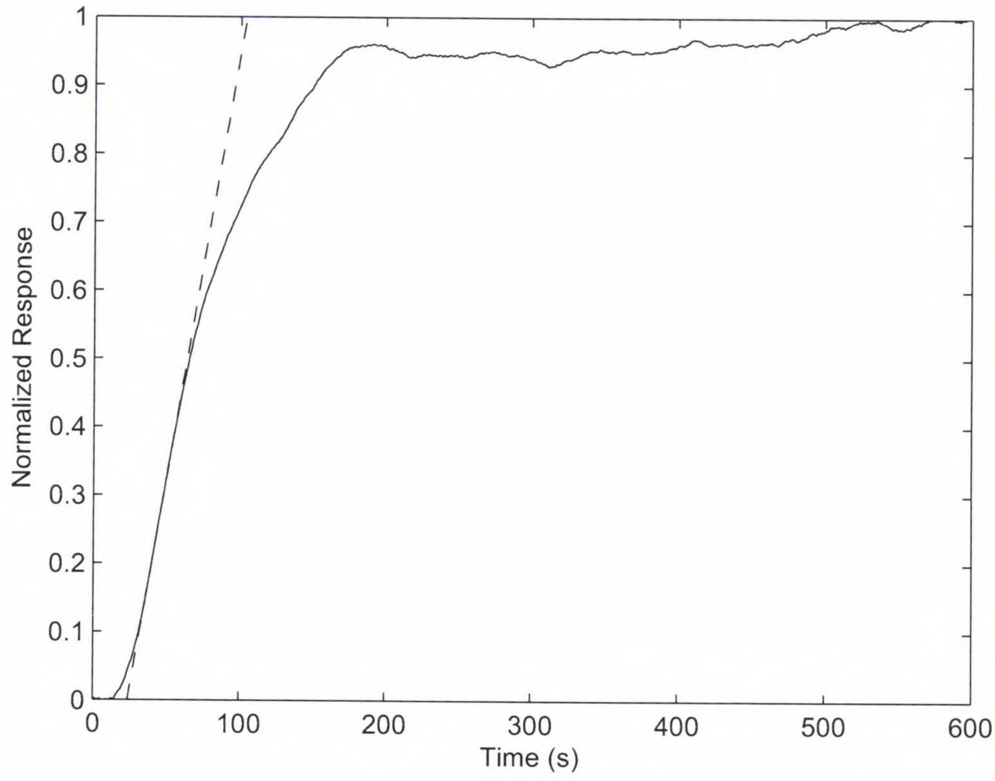

Figure 5.3: Application of the Ziegler-Nichols tuning method to a single step change in valve position. The tangent line to the inflection point is shown.

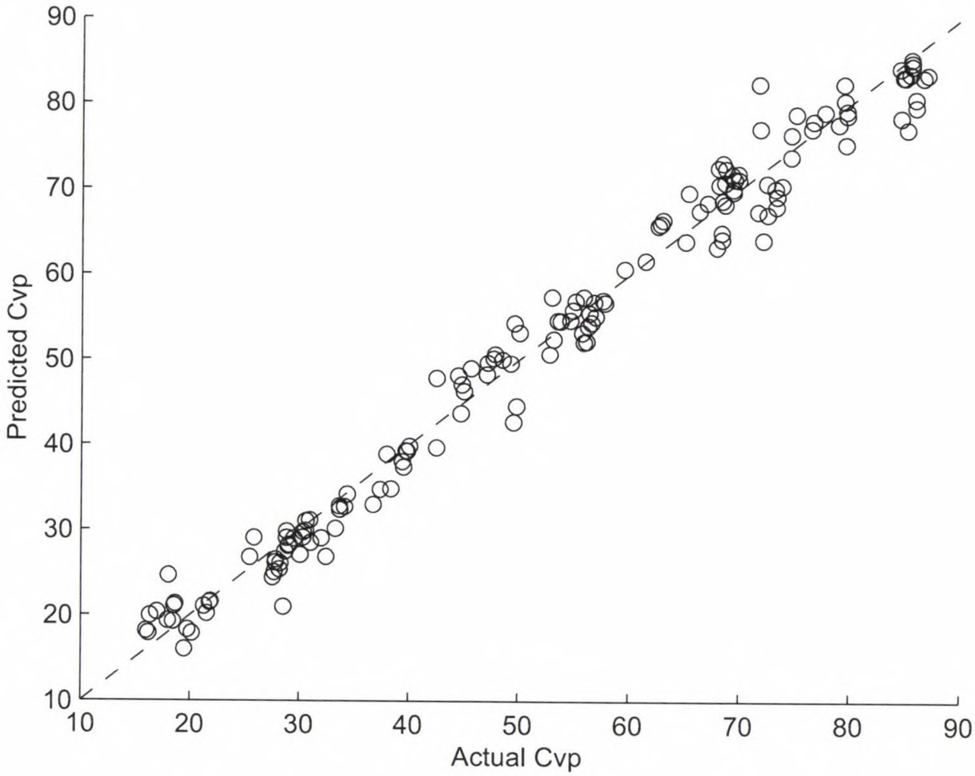

Figure 5.4: Scatter plot of predicted and actual steady state valve position. Predictions were made by a third order two dimensional polynomial. In input to the predictor were the outlet air temperature and the flow rate of air across the coil.

Table 5.1: Proportional and Integral Gains used to Control the Water to Air Heat Exchanger.

Gain	Value
K_p	2.2
K_i	0.075

5.2.4 Formulation of a Steady State Predictive Model

The data generated during the reachable space experiment was used to make a predictive model for the system. The output of the predictor is the commanded valve position needed to produce the given outlet air temperature for the given flow rate of air. To produce the model the steady state values of the commanded valve position, the flow rate of air, and the outlet air temperature were normalized to the range of [-1,1] by using the maximum and minimum values in the data set. A two dimensional third order polynomial was fit to the data using the linear least squares method. Figure 5.4 shows the results of the fit. The root mean squared error for the fit was 4.2%. The model was used to provide the 'stuffing signal' for controllers developed during the project.

5.3 Dynamic Model

This section describes the development of dynamic models for both the water flow control valve and the heat exchanger. Both systems have nonlinear dynamics so linear system identification is not an option. Both systems were modeled with nonlinear finite input response models to avoid the potential numeric instability of other nonlinear modeling methods [24].

5.3.1 Model Creation

To construct a dynamic model the valve was considered a single-input-single-output system with the commanded valve position (C_{vp}) the input and the flow rate of the water through the valve (f_w) the output. The heat exchanger is a multi-input-single output system. The inputs are the flow rate of the air (f_a), the flow rate of the water (f_w), the temperature of the incoming air (T_{ai}), and the temperature of the incoming water (T_{wi}). The output of the heat exchanger is the temperature of the outlet air (T_{ao}). Both the valve and the heat exchanger are insensitive to initial conditions so only a finite history of the inputs is needed to make prediction of the output (for more precise predictions longer input sequences are needed). It was observed from the data generated during the steady state tests that the output of the valve responds within 5 seconds to changes input while the output of the heat exchanger can take up to twenty minuets to respond to changes in the input. Based on a sampling time of 0.1 seconds[1] the output of the valve depends on the most recent 50 values of its input and the output of the heat exchanger depends on the most recent 12,000 values of each of its inputs.

$$\begin{aligned} f_w(n) &= f\left[C_{vp}(n-1), C_{vp}(n-1), \ldots, C_{vp}(n-50)\right] \\ T_{ao}(n) &= f\left[T_{ai}(n-1), T_{wi}(n-1), f_a(n-1), f_w(n-1), \ldots\right. \\ &\qquad \left.\ldots, T_{ai}(n-12000), T_{wi}(n-12000), f_a(n-12000), f_w(n-12000)\right] \end{aligned}$$

It may not be necessary to know the value of the input signal(s) at every time step to make accurate predictions. To simplify the input for the models, averages over time steps were formed. Table 5.3.1 shows how each signal was processed into inputs for the models. The size of the input space for the valve model was reduced to seven (approximately six seconds of data). The size of the input space for the heat exchanger model was reduced to 60 (15 for each input signal, just under thirty minutes of data).

Training data for the models was generated in a similar fashion to that of the steady state models. The *Repeat HVAC* system was run with the water supply temperature and air inlet temperature regulated by separate PI controllers. Data was generated by varying the inputs in random steps. The duration of the steps was varied from two to thirty minutes. The desired hot water supply temperature was held constant at 50°C. The desired air inlet temperature was varied between 5°C and 15°C. The commanded blower speed was varied between 30% and 70% of the full speed value. The commanded valve position was varied between 10% and 90% percent of the fully open position. Training data was generated over several runs.

For the dynamic valve model the time series of commanded valve position (C_{vp}) was

---

[1] During the entire course of the project the *HVAC REPEAT* system sampled data at a rate of 10Hz.

Table 5.2: Processing of Signals Into Inputs for the Dynamic Models.

Input	Average of Samples	Approximate Time Window
1	(n-1)	0.1 Seconds
2	(n-2)	0.1 to 0.2 Seconds
3	(n-3) thru (n-4)	0.3 to 0.4 Seconds
4	(n-5) thru (n-8)	0.5 to 0.8 Seconds
5	(n-9) thru (n-16)	0.9 to 1.6 Seconds
6	(n-17) thru (n-32)	2 to 3 Seconds
7	(n-33) thru (n-64)	3 to 6 Seconds
8	(n-65) thru (n-128)	7 to 13 Seconds
9	(n-129) thru (n-256)	13 to 26 Seconds
10	(n-257) thru (n-512)	26 to 51 Seconds
11	(n-513) thru (n-1024)	1 to 2 Minutes
12	(n-1025) thru (n-2048)	2 to 3 Minutes
13	(n-2049) thru (n-4096)	3 to 7 Minutes
14	(n-4097) thru (n-8192)	7 to 13 Minutes
15	(n-8193) thru (n-16384)	14 to 27 Minutes

processed to form a vector input of delayed averages for each time step. The form of the model to predict the flow rate of the water through the valve (f_w) is shown in equation (5.1).

$$\tilde{f_w}(n) = f\left\{C_{vp}(n-1), C_{vp_2}(n-1), \ldots C_{vp_7}(n-1)\right\} \tag{5.1}$$

Where,

$$\begin{aligned}
C_{vp_2}(n-1) &= C_{vp}(n-2) \\
C_{vp_3}(n-1) &= \text{Average}\left\{C_{vp}(n-3), C_{vp}(n-4)\right\} \\
C_{vp_4}(n-1) &= \text{Average}\left\{C_{vp}(n-5), \ldots, C_{vp}(n-8)\right\} \\
C_{vp_5}(n-1) &= \text{Average}\left\{C_{vp}(n-9), \ldots, C_{vp}(n-16)\right\} \\
C_{vp_6}(n-1) &= \text{Average}\left\{C_{vp}(n-17), \ldots, C_{vp}(n-32)\right\} \\
C_{vp_7}(n-1) &= \text{Average}\left\{C_{vp}(n-33), \ldots, C_{vp}(n-64)\right\}
\end{aligned}$$

For the dynamic heat exchanger model the time series of the water flow rate (f_w), the air flow rate (f_a), the temperature of the water entering the coil (T_{wi}), and the temperature of the air entering the coil (T_{ai}) were processed to form a vector input of delayed averages for each time step. The form of the model to predict the outlet air temperature (T_{ao}) is shown in equation (5.2).

$$\begin{aligned}\tilde{T}_{ao}(n) &= f\{f_w(n-1), f_{w2}(n-1), f_{w3}(n-1), \ldots, f_{w14}(n-1), f_{w15}(n-1), \\ &\quad f_a(n-1), f_{a2}(n-1), f_{a3}(n-1), \ldots, f_{a1}4(n-1), f_{a15}(n-1), \\ &\quad T_{wi}(n-1), T_{wi2}(n-1), T_{wi3}(n-1), \ldots, T_{wi14}(n-1), T_{wi15}(n-1), \\ &\quad T_{ai}(n-1), T_{ai2}(n-1), T_{ai3}(n-1), \ldots, T_{ai14}(n-1), T_{ai15}(n-1)\} \qquad (5.2)\end{aligned}$$

Where,

$$\begin{aligned}f_{w2}(n-1) &= f_w(n-2) \\ f_{w3}(n-1) &= \text{Average}\{f_w(n-3), f_w(n-4)\} \\ f_{w4}(n-1) &= \text{Average}\{f_w(n-5), \ldots, f_w(n-8)\} \\ f_{w5}(n-1) &= \text{Average}\{f_w(n-9), \ldots, f_w(n-16)\} \\ f_{w6}(n-1) &= \text{Average}\{f_w(n-17), \ldots, f_w(n-32)\} \\ f_{w7}(n-1) &= \text{Average}\{f_w(n-33), \ldots, f_w(n-64)\} \\ f_{w8}(n-1) &= \text{Average}\{f_w(n-65), \ldots, f_w(n-128)\} \\ f_{w9}(n-1) &= \text{Average}\{f_w(n-129), \ldots, f_w(n-256)\} \\ f_{w10}(n-1) &= \text{Average}\{f_w(n-257), \ldots, f_w(n-512)\} \\ f_{w11}(n-1) &= \text{Average}\{f_w(n-513), \ldots, f_w(n-1024)\} \\ f_{w12}(n-1) &= \text{Average}\{f_w(n-1025), \ldots, f_w(n-2048)\} \\ f_{w13}(n-1) &= \text{Average}\{f_w(n-2049), \ldots, f_w(n-4096)\} \\ f_{w14}(n-1) &= \text{Average}\{f_w(n-4097), \ldots, f_w(n-8192)\} \\ f_{w15}(n-1) &= \text{Average}\{f_w(n-8193), \ldots, f_w(n-16384)\}\end{aligned}$$

Input vectors for f_a, T_{wi}, and T_{ai} were created of the same manner as those for f_w. Feed forward networks were used as function approximations for the dynamic valve model, equation (5.1) and the dynamic heat exchanger model, equation (5.2). The networks were trained with standard back propagation with validation and testing sets for early stopping [24]. Different model structures and learning parameters were tested. For each learning trial 20,000 time steps[2] were randomly selected for training with 5,000 used for validation and another 5,000 used for testing. The final version of each model consisted of a single hidden layer of ten hyperbolic tangent activation functions.

5.3.2 Model Validation In Open loop

The output of the dynamic valve and heat exchanger models were cascaded together to produce a complete system model. Figures 5.5 and 5.6 show a comparison between the model and the physical system. Data generated from the physical system was used as input to each model. For much of the input space the models were able to reproduce the output of the actual system with a reasonable degree of accuracy. In particular, high frequency variations matched quite well while there were some regions where the base outlet temperature of the model had noticeable error. To quantify the fit 5,000 samples were randomly selected from the validation run. The error of each sample was calculated. The root mean squared error for the sampled set was 0.42C°. Figure 5.7 shows the predicted outlet air temperature of the model as a function of the actual outlet air temperature of the system for each of the samples.

5.3.3 Model Validation In Closed Loop

To investigate the performance of the dynamic models in closed loop control both the model and the physical system were controlled with the PI controller presented in section 5.2.3. Measurements of f_a, T_{ai}, and T_{wi} taken during the testing of the physical system were used as inputs to the model. Both the model and the physical system were asked to track the same step changes in desired outlet air temperature in the presence in the same disturbance inputs. The closed loop behavior of the combined valve and heat exchanger model was investigated and compared to the response of the physical system. Figure 5.8 shows the actual outlet air temperature of the system and the outlet temperature of the model while they are both subject to the same inputs. The similarity, especially at high frequency, is noticeable.

---

[2]There are more than 3 million samples in the full training set.

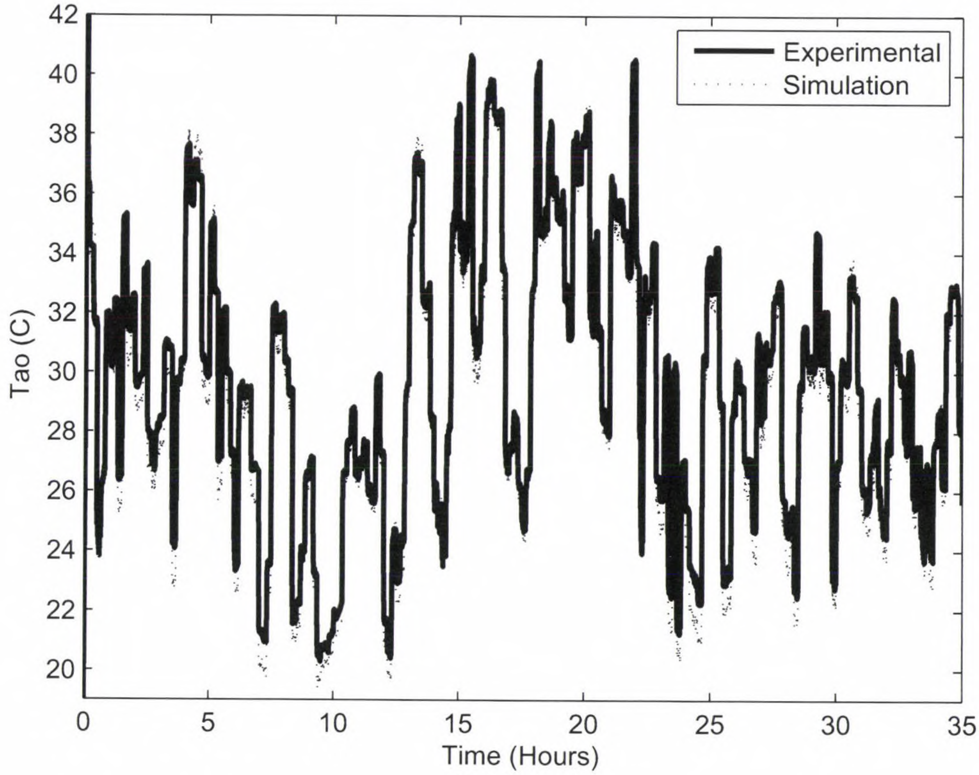

Figure 5.5: Predicted and actual outlet air temperature. The predictions were made by cascading the dynamic valve and heat exchanger models. Input data was generated from the physical system.

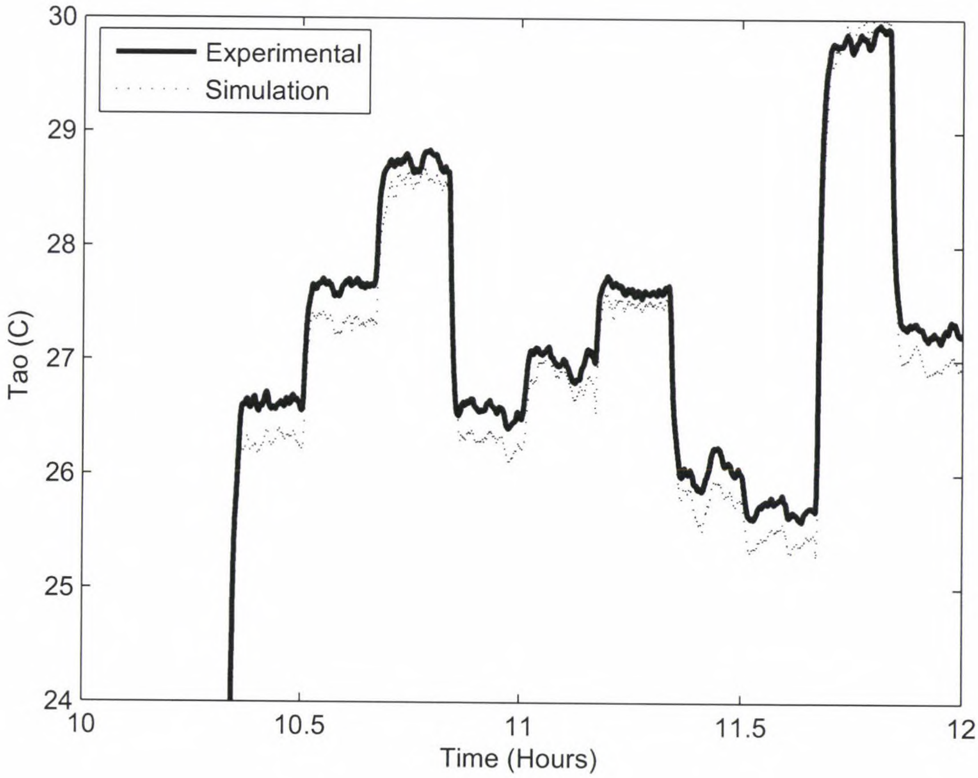

Figure 5.6: Validation of the valve and heat exchanger models with input data generated from the physical System. High frequency oscillations are tracked well while some steady state errors are present.

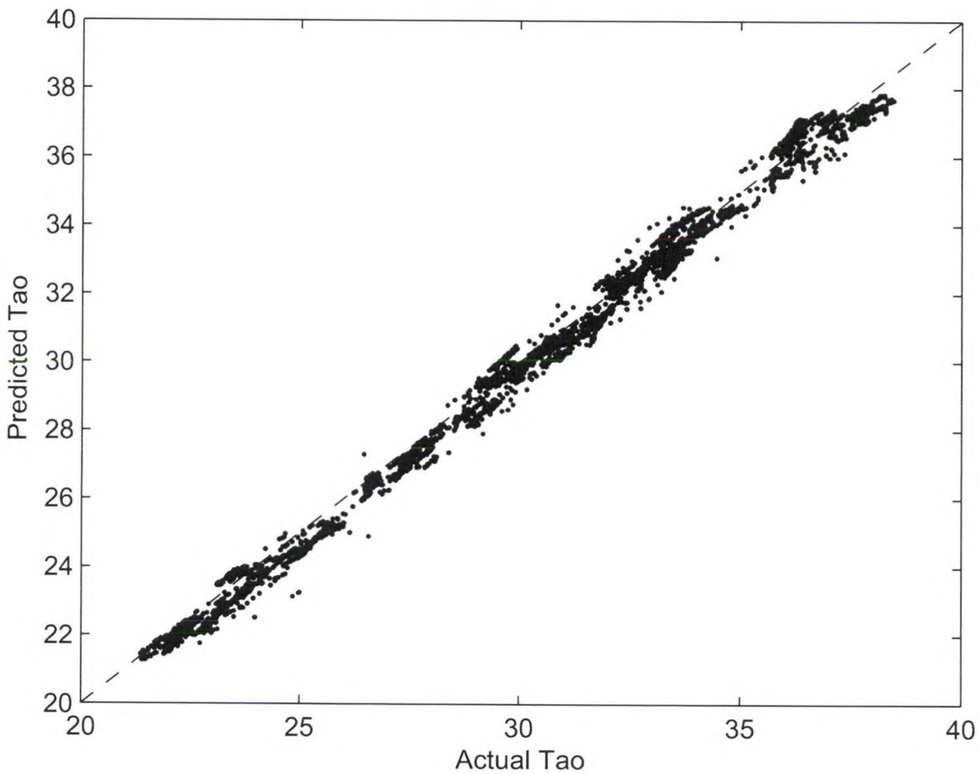

Figure 5.7: Predicted and actual outlet temperatures for 5,000 samples taken from the validation run of figure 5.5. The root mean squared error for the validation run was 0.42°C.

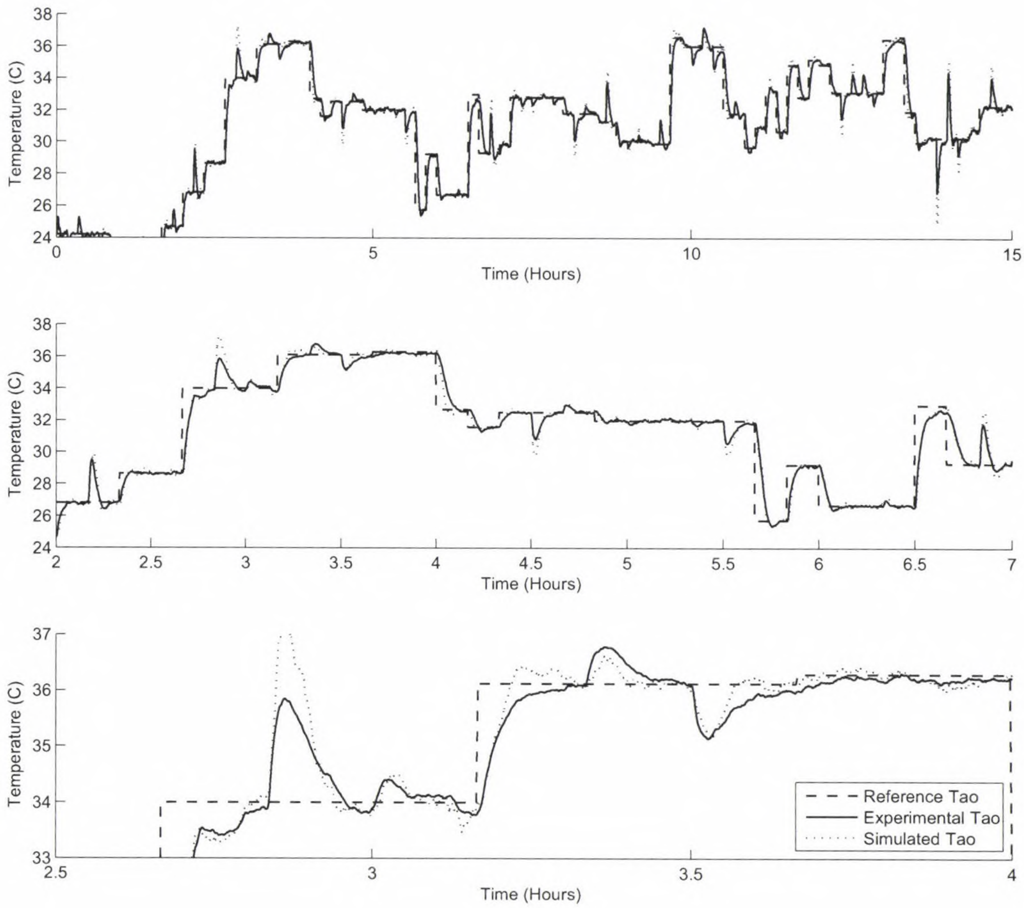

Figure 5.8: Closed Loop Validation of the Valve and Heat Exchanger with Input Data Generated from the Physical System. The response of the model and the physical system are similar.

Figure 5.9 shows a comparison between the flow rate of water in the physical system and in the model. Although model and experiment produce very similar temperature trajectories there are noticeable differences in the flow rate of water.

5.4 Conclusion

This chapter has presented the development of steady state and dynamic models. The steady state model of section 5.2.4 was used as the steady state predictor used to generate the 'stuffing signal' for controllers tested during this project. The dynamic models presented do not provide accurate results over the entire input space of the *REPEAT HVAC* system, but they do capture the important dynamics of the system. The dynamic models are accurate enough to be used in simulation as a starting point in the development of controllers for the real system.

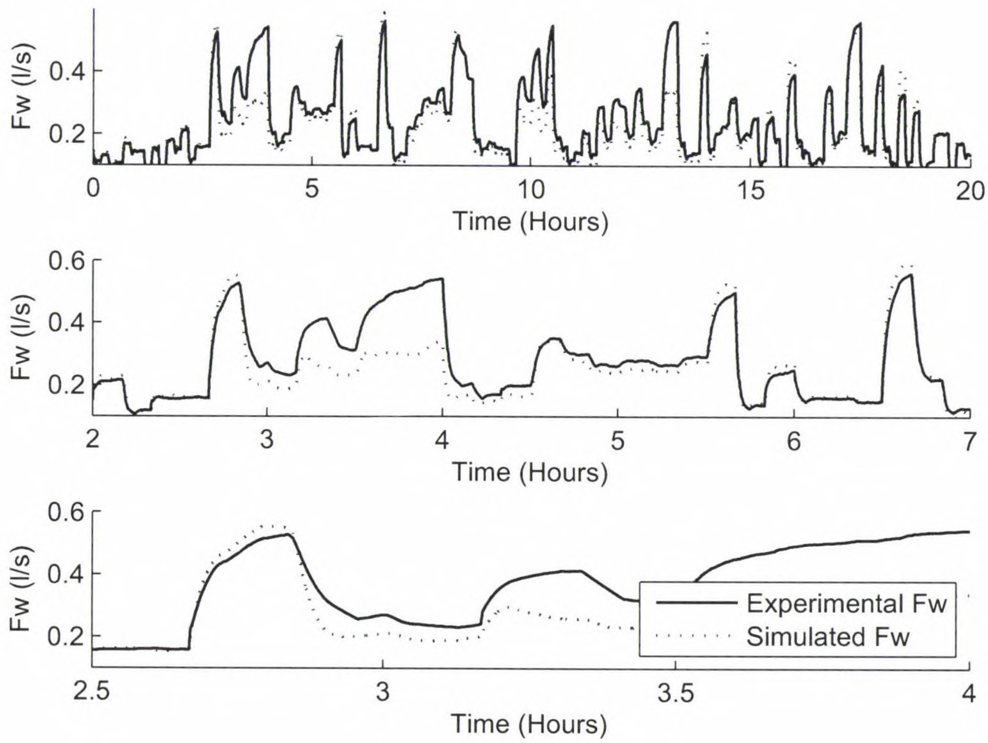

Figure 5.9: Close Loop Validation of the Valve and Heat Exchanger with Input Data Generated from the Physical System. Although the temperature trajectories are similar the flow rates diverge for part of the input space..

Chapter 6

Error Modulated Partial Stuffing Control Applied to the *REPEAT HVAC* System

6.1 Introduction

The error modulated partial stuffing controller developed in chapter 4 was applied to the *REPEAT HVAC* system and to the model of the *REPEAT HVAC* system presented in chapter 5. Section 6.2 describes the experimentation on the physical system and section 6.3 describes the experimentation on the model. Section 6.4 concludes the chapter.

6.2 Behavior of Error Modulated Partial Stuffing on the *REPEAT HVAC* System

Error modulated partial stuffing was implemented on the the *REPAT HVAC* system for two sets of experiments. In the first experiment the performance of the controller was investigated. In the second experiment the steady state errors associated with error modulated partial stuffing were investigated.

6.2.1 Performance

An experiment was conducted to investigate the performance of error modulated partial stuffing control on the *REPEAT HVAC* system. During the experiment the system experienced random step changes in either the set point of the outlet air temperature or the flow rate of air. After each step change the inputs were held constant for ten minutes. For each step change a random value of K_e was used. The value of K_e was limited to five discrete values: $K_e = \{0, 0.1K_i, 0.2K_i, 0.4K_i, 0.9K_i\}$. Figure 6.1 shows the desired outlet air temperuate, the actual outlet air temperature as well as the the value of K_e used for a portion of the experiment. The root mean squared error (between actual and desired outlet air temperature) was calculated for each ten minute step. Table B.3 of appendix B shows the RMS error and K_e used for each of 200 ten minutes steps in the experiment. Table 6.1 summarizes the results of the experiment. $K_e = 0$ corresponds to the nominal PI controller.

Table 6.1: Results of Varying K_e on the Physical System

K_e/K_i	Number of Trials	Average RMS Error (°C)	Standard Deviation of RMS Error (°C)
0	39	0.81	0.59
0.1	38	0.56	0.35
0.2	32	0.46	0.36
0.4	51	0.66	0.43
0.9	40	0.59	0.37

As can be seen in table 6.1 all values of K_e used resulted in lower mean RMS error than the nominal PI controller. To test the statistical significance of the results, single sided t-tests were performed to determine if the true mean of the error decreased when error modulated partial stuffing control was used. To conduct this test it was assumed that for each value of K_e the RMS error for each trial was normally distributed with unknown and unequal variance. Table 6.2 summarizes the results of the single sided t-tests.

Table 6.2: Reduction in RMS error for different values of K_e compared to nominal PI control.

K_e/K_i	Confidence that the true mean of the RMS error was reduced	95% Confidence interval of the % reduction in RMS error
0.1	98.8%	4.3% to 58%
0.2	99.9%	15% to 71%
0.4	91.4%	-8.5% to 46.2%
0.9	97.5%	0% to 54%

As can be seen in table 6.2 for three of the four values of K_e used there was over 95% confidence that the true mean RMS error decreased. The best performance was found for $K_e = 0.2K_i$ in which there was 95% confidence that the error was reduced by at least 15%.

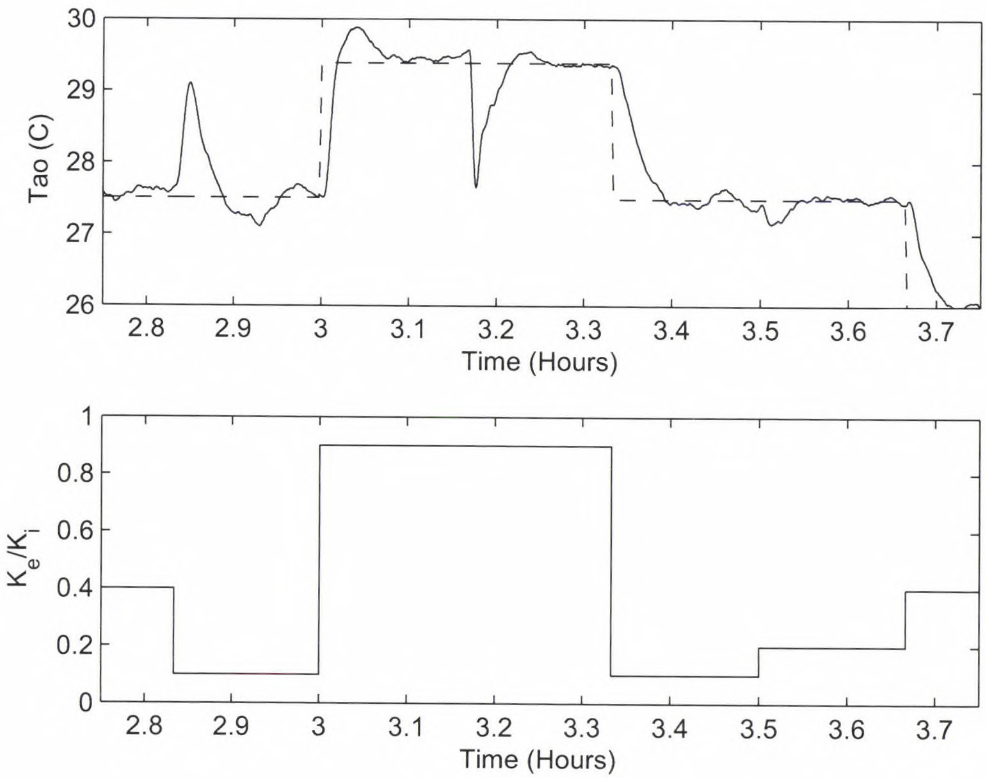
30
29
28
27
26
Tao (C)
2.8
2.9
3
3.1
3.2
3.3
3.4
3.5
3.6
3.7
Time (Hours)
1
0.8
0.6
0.4
0.2
0
K_e/K_i
Time (Hours)

Figure 6.1:

6.2.2 Steady State Errors

In section 4.4 a bound on the steady state error of a system under error modulated partial stuffing was derived. Equation 4.31 is rewritten here,

$$|e_{ss}| < \frac{|\tilde{\theta} - \theta^{\star}| - \frac{K_i}{K_e}}{K_P} \tag{6.1}$$

An experiment was run on the *REPEAT HVAC* system to test this theoretical bound. During the experiment the system experienced 61 random step changes in either the set point of the outlet air temperature or the flow rate of air. After each step change the inputs were held constant for twenty minutes. During the first ten minutes of each trial the value of K_e was set to a random value from the discrete set $\{0.1K_i, 0.2K_i, 0.4K_i, 0.9K_i\}$. During the second ten minutes of each trial the value of K_e was set to zero so the system was controlled by the nominal PI controller. For each trial, the average steady state error while the system was under error modulated partial stuffing control was calculated by averaging the outlet temperature over the last five minutes of the first half of the trial. In addition the average nominal PI steady state error was calculated using the last five minutes of each trial. Figure 6.2 shows and example for a single trial. In this case the steady state error while the system was under error modulated partial stuffing control (left half of figure) was 0.46°C. The steady state error for the nominal PI control (right half of figure) was 0.03°C.

For each trial in which the steady state error under error modulated partial stuffing was more than two times greater than the steady state error under nominal control the bound given in equation 6.1 was calculated and compared to the measured steady state error. Since the nominal PI controller should result in zero steady state error, the output of the PI controller was used as an approximation to the command signal that would produce zero steady state error ($\theta^{\star}$).

Table 6.3 summarizes the calculation of the bound for the trial shown in figure 6.2. As shown in the table the measured steady state error was within the bound. For the 10 of the 61 trials run the steady state error under error modulated partial stuffing was more than twice the steady state error of the nominal PI control. In each of the ten cases the measured steady state error was less than calculated bound.

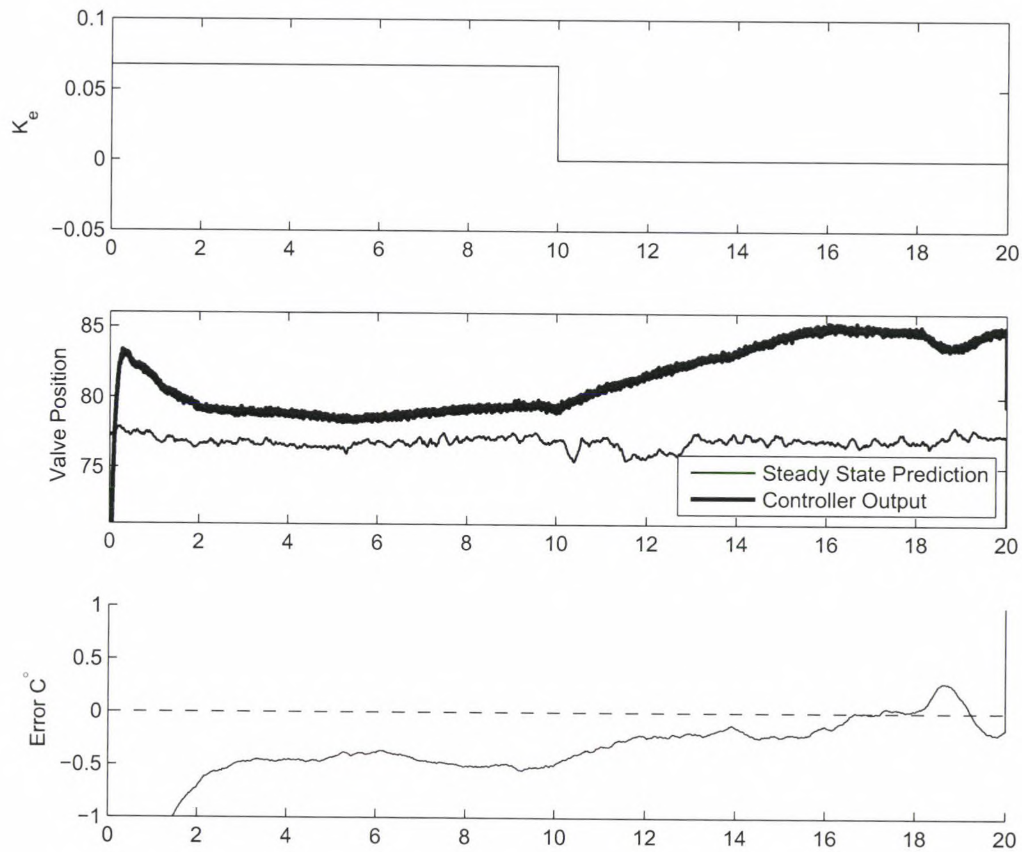

Figure 6.2: Steady state error introduced by a combination of large K_e and a large error in the steady state prediction.

Table 6.3: Calculation of the error bound for the trial shown in figure 6.2.

Term	Value
$\theta^\star$	84.6
$\tilde{\theta}$	77.0
$\lvert\tilde{\theta} - \theta^\star\rvert$	7.54
K_p	2.2
K_i	0.075
K_e	0.0675
$\frac{\lvert\tilde{\theta}-\theta^\star\rvert - \frac{K_i}{K_e}}{K_P}$	3.02°C
Steady State Error	0.46°C

6.3 Comparison of PI control to Error Modulated Partial Stuffing in Simulation

The dynamic model developed in chapter 5 was used to test the relative performance of error modulated partial stuffing control and to nominal PI control. The use of the model allowed for testing over more trials than was feasible with the physical system.

The gains (K_p and K_i) of the nominal PI controller designed for the physical system were used. The nominal PI controller was compared to an error modulated partial stuffing controller with $K_e = 0.2K_i$. For each controller 1000 trials were simulated. Each trial lasted ten minutes. For each trial a new flow rate of air and desired outlet air temperature were chosen at random. Table 6.4 summarizes the results of the simulations. A single sided t-test

Table 6.4: Comparison of nominal PI control and error modulated partial stuffing control in simulation.

K_e/K_i	Number of Trials	Average RMS Error (°C)	Standard Deviation of RMS Error (°C)
0	1000	1.63	0.84
0.2	1000	1.28	0.76

was performed on the simulation data. The confidence level that the true RMS error of the error modulated partial control case is less than the true RMS error of the nominal PI control is greater than 99.99%.

6.4 Conclusion

For experiments presented in this chapter error modulated partial stuffing outperformed nominal PI control. The steady state errors experienced during error modulated partial stuffing control were found to be within the bounds presented in chapter 4.

Chapter 7

Concluding Remarks

7.1 Introduction

This chapter give a brief overview of the project with respect to the goals stated in section 1.2.

7.2 Remote Control of *REPEAT HVAC*

The *REPEAT HVAC* system is now fully remotely controlled. Without the work done to add hardware and software to the system the project would not have been feasible. The system stands ready for future work.

7.3 Model Development

The models developed during the project were quite accurate and were useful in the development of controllers. Future controller development will be possible without the need to exclusively use the physical system. Though the model is accurate and much faster than the physical system the need to filter so much input data means the models are still a little slow and require hours of simulation to generate a meaningful amount of data. It may be possible to optimize the filtering of the input data.

7.4 Controller Development

The most significant contribution of this project is the error modulated partial stuffing controller. This controller was shown to improve the performance of the system in response to both set point changes and disturbances. The bounds on error presented are a first step in the investigation of the stability of the controller.

The use of error modulated partial stuffing control is suited to applications in which PI control provides stable control and steady state predictions are available. If the magnitude of the error in the steady state predictions is known, the steady state error bounds given in section 4.4 can be used to select the gain K_e.

In applications where the accuracy of the steady state predictions varies across the input space it may be possible to have K_e vary across the input space. For many systems it may be easy to determine the expected error in steady state predictions while it may be hard or impossible to improve the predictions due to unmeasured or uncontrollable disturbance inputs.

7.5 Learning

The one goal for the project that was not fully satisfied was the incorporation of online learning. This section is intended to provide a framework for adding learning to an error modulated partial stuffing controller. Consider example of section 4.3.3. In that section an optimal value of K_e was determined for the entire simulation. Figure 7.1 shows the value of K_e that produced the lowest error for each step. By changing the value of K_e to it's optimal value for each step, the error for the entire simulation was reduced beyond the value of it's constant K_e value as shown in figure 7.2. In this example the system was entirely repeatable so different cases could be directly compared. This is not true of a real system, but if enough data is generated it may be possible to map the input space to the best value of K_e. This mapping was attempted as part of this project, but due to the influence of disturbances and the long run times needed to reach steady state no mapping was developed.

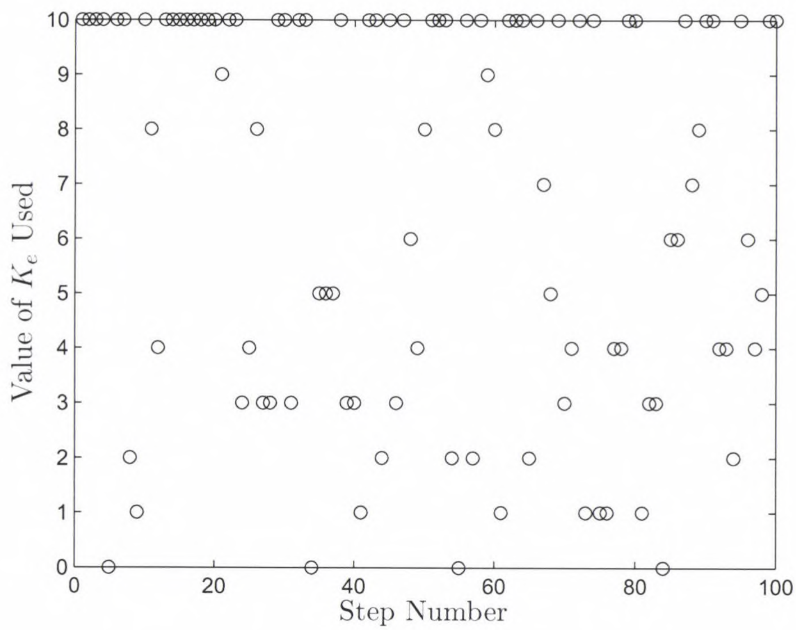

Figure 7.1: The value of K_e used for each step within the simulation is shown. By allowing K_e to change for each step the RMS error for the simulation was reduced to 0.0425. The best fixed K_e simulation resulted in an RMS error of 0.0446.

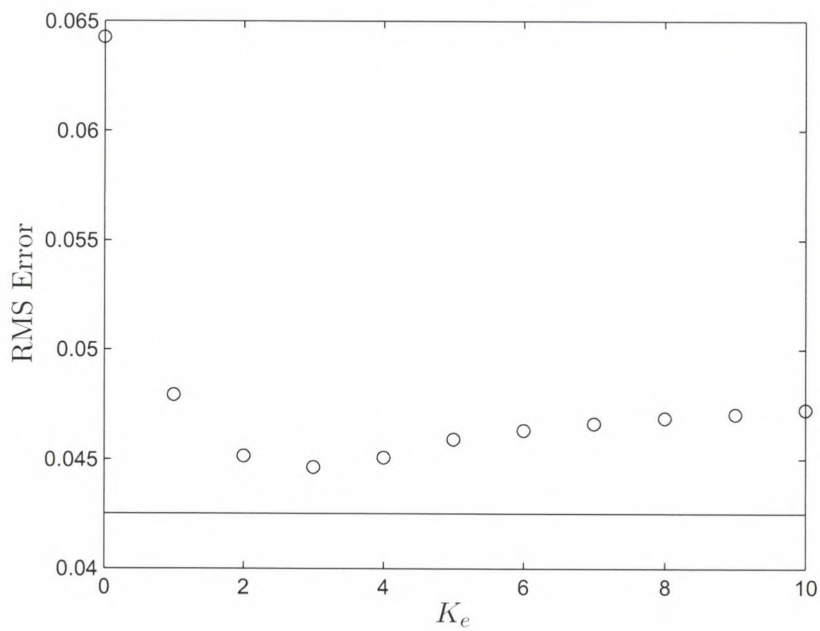

Figure 7.2: RMS error of the entire simulation for different fixed values of the gain K_e. The solid line indicates the RMS error achieved when K_e was changed for each step within the simulation

Appendix A

Changes to *REPEAT HVAC* System

This appendix describes the changes that were made to the *REPEAT HVAC* system as part of this project. Section A.1 describes the repairs that have been made to the system. Section A.2 describes what was done to establish remote communication with the control computer. Section A.3 describes the additional equipment that was installed to allow for remote operation of the system.

A.1 Repairs

During the course of the current project several repairs have been made to the system. This section describes the nature of the repairs for future reference.

A.1.1 Heat Exchanger

The most significant repair has been the replacement of the water to air heat exchanger. The original heat exchanger was ruined during a freeze event. At the start of this project a new and physically different heat exchanger was installed.

A.1.2 Boiler

The freeze event that damaged the original heat exchanger also damaged the cast iron core of the boiler. The heating elements and control circuitry were moved onto an identical replacement core. When the replacement core was installed a flexible coupling was added between the pump and the boiler to allow for compliance and reduce vibration noise.

A.1.3 Air Compressor

The original air compressor (used to provide compressed air to the pneumatic valves controlling the air mixing dampers and the water control valve) had a small supply tank. This resulted in short cycling and the eventual failure of the compressor. A new compressor was installed. Since the working pressure of the new compressor (125 PSI) is higher than the maximum operating pressure of the pneumatic valves (25 PSI) a second 10-L tank with a regulated pressure of 20 PSI was added to the system. In addition the power supply for the compressor was wired through a relay that is actuated in parallel to the circulation pump. Now the compressor is only on when the experiment is running.

A.1.4 Wire Failure

During operation the (stranded two gauge) wire connecting the solid state switch to the boiler overheated and eventually failed. After this failure the sizing of the wire was reviewed and compared to the electrical code. It was determined that the wire diameter should be increased in order to handle the maximum current draw of the heating elements in the boiler. The wire between the solid state switch and the boiler was replaced by stranded zero gauge wire.

A.1.5 Electrical Control board

During operation the solder holding a high power resistor in the electronic control for the solid state switch came loose. This caused the heating system to stop with no clear cause. Through what can only be attributed to divine intervention the loose resistor was found and re-soldered to the circuit board. The connections on the board are now periodically inspected.

A.1.6 Pneumatic Hose

The original transparent neoprene pneumatic hoses deteriorate with exposure to UV light. Because the experiment is in front of windows these hoses have periodically failed. They have all been replaced by opaque rubber hose. These hoses have a longer life, but ultimately it may be desirable to plumb the entire pneumatic system with either copper or brass tubing.

A.2 Remote Communication With Control Computer

To allow for remote control of the experimental system a communication link was established between the experiments control computer and a remote computer via the internet. This section provides details on the equipment and software used to create this link.

A.2.1 Wireless Repeaters

Unfortunately there is no internet access in REPEAT. To overcome this difficulty a pair of wireless range extenders were set up in repeater mode. One was placed in a basement window of Solar House III and the other was placed in the upper floor window of REPEAT. A 100-ft CAT5 cable connects the control computer to the repeater.

A.2.2 Installation of TightVNC and an FTP server

A TightVNC server was installed on the control computer. TightVNC is software package that allows for password protected remote control of the desktop of a computer running windows. TightVNC requires that the host computer have a static IP address, which was thankfully supplied by The Atmospheric Science Department. The control computer can now be controlled remotely by any computer that has installed the TightVNC viewer. TightVNC has an integrated file transfer process but it is a little slow so a separate FTP server, *WinFTP*, was installed on the control computer.

A.2.3 Memory Upgrade

To ensure that the control computer can handle the communication overhead the memory was expanded from 256MB to 512MB (which is the maximum that control computer motherboard can handle).

A.3 New Mechanical Equipment For Remote Operation

Once a remote communication link was established, equipment was added to the laboratory to allow for remote operation of the experimental HVAC system.

A.3.1 Pneumatic operation of Windows

Since freezing is a concern the windows that allow for airflow in and out of the the HVAC system need to be closed when the system is not in use. To allow for automatic control pneumatic cylinders were connected to the windows. The cylinders are equipped with speed controllers so they do not slam open or closed.

Cantilevered weights have been installed on the windows to ensure that the windows close when the compressor pressure drops.

A.3.2 Control Box/Time Delay Relays for equipment

Initially the valves that control the cylinders were connected in parallel to the compressor, but it was found that the valves did not work properly if the supply air was not at pressure of at least approximately 15 PSI. For lower pressures the valves move to an intermediate position and continually leak air from the supply to the ambient. To give the compressor enough time to charge the supply tank to a reasonable pressure a solid state relay with a variable delay was installed to control the pneumatic valves. Now when the experiment is started the compressor runs for approximately five minutes before the windows are opened.

A.3.3 Water Pressure regulator

The water loop of the experimental system contains pressure a pressure relief valve and an air separator, over time both of these devices and other minor leaks can cause the pressure in the water loop to drop. To ensure a constant water pressure the system was connected to the mains via a water pressure reducer. An electronic pressure transducer was added to the water loop and wired to the control computer so the water pressure can be monitored while the system is being run.

Appendix B

Experimental Data

Table B.1: Steady state data from *REPEAT HVAC* system (Part 1).

Trial	C_{bs} (%)	f_a (m^3/s)	C_{vp} (%)	f_w (l/s)	T_{ae} (°C)	T_{ar} (°C)	C_{dr} (%)	T_{ai} (°C)	T_{wi} (°C)	T_{ao} (°C)	T_{wo} (°C)
1	64.9	0.64	17.0	0.07	3.1	21.4	42.0	9.7	49.2	21.5	30.9
2	37.8	0.29	17.0	0.07	3.2	22.0	45.4	10.0	49.2	27.1	34.6
3	43.3	0.36	17.0	0.07	2.6	21.8	47.2	9.8	49.2	25.4	33.5
4	53.2	0.48	17.0	0.07	2.5	21.3	48.2	9.8	49.2	23.3	32.1
5	53.2	0.48	14.6	0.06	2.6	21.2	48.4	10.0	49.2	23.1	31.2
6	57.3	0.54	14.8	0.06	2.5	21.0	50.2	9.9	49.2	22.3	30.7
7	57.2	0.53	73.7	0.40	2.4	21.9	42.5	9.8	49.2	30.3	43.5
8	42.5	0.34	73.7	0.40	2.1	22.1	48.3	9.9	49.2	33.8	44.6
9	42.5	0.34	62.1	0.29	2.6	22.1	47.3	10.1	49.3	32.8	43.2
10	37.7	0.29	62.0	0.29	2.7	22.0	47.7	10.1	49.3	34.2	43.7
11	37.8	0.29	29.7	0.12	2.0	21.6	52.1	9.9	49.2	29.0	38.4
12	68.0	0.68	29.7	0.12	2.1	21.1	50.9	10.1	49.2	23.0	35.2
13	64.5	0.63	29.7	0.12	2.4	21.0	49.6	10.1	49.2	23.3	35.5
14	38.1	0.30	29.7	0.12	2.6	21.3	49.1	9.9	49.2	28.6	38.3
15	38.1	0.30	36.8	0.15	2.1	21.4	50.0	9.7	49.2	30.2	39.8
16	38.1	0.29	33.2	0.13	1.1	21.3	58.6	9.8	49.2	29.5	39.1
17	48.1	0.41	33.2	0.13	1.2	21.2	55.5	9.6	49.2	26.6	37.8
18	48.1	0.41	33.8	0.13	1.5	20.9	55.3	9.9	49.2	26.7	37.9
19	58.7	0.55	34.0	0.13	0.2	20.8	59.9	9.7	49.2	24.7	36.8
20	58.7	0.54	86.8	0.51	1.8	21.4	51.3	10.1	49.1	30.9	44.4
21	58.7	0.55	58.7	0.25	1.4	21.2	54.2	10.3	49.2	28.7	41.0
22	58.7	0.55	78.8	0.43	1.9	21.3	48.8	9.9	49.2	30.4	43.8
23	57.7	0.53	78.8	0.43	0.5	21.3	58.1	10.0	49.3	30.8	43.9
24	49.2	0.42	78.8	0.43	0.8	21.4	58.0	9.9	49.3	32.5	44.4
25	49.2	0.42	86.7	0.50	0.5	21.5	57.0	9.7	49.1	32.8	45.0
26	49.2	0.43	28.4	0.11	0.9	20.8	58.9	9.9	49.2	25.7	36.5
27	55.2	0.51	28.4	0.11	1.5	20.5	56.1	10.3	49.2	24.7	35.9
28	50.1	0.44	28.4	0.11	0.2	20.4	60.7	10.0	49.2	25.3	36.1
29	68.4	0.69	28.5	0.11	0.2	20.1	61.8	10.0	49.2	22.7	34.6
30	68.4	0.68	84.5	0.50	0.8	20.8	57.5	10.1	49.1	29.5	43.9
31	55.5	0.50	84.5	0.49	0.1	20.9	60.0	10.0	49.2	31.5	44.4
32	62.6	0.60	84.5	0.50	0.8	20.9	57.9	9.9	49.1	30.3	44.1
33	62.6	0.60	85.3	0.50	0.9	20.8	56.9	9.8	49.2	30.3	44.1
34	41.4	0.33	85.3	0.50	0.1	20.7	61.4	9.7	48.9	34.5	45.3
35	62.0	0.59	85.3	0.50	0.6	20.7	58.9	10.0	49.1	30.5	44.2
36	53.4	0.47	85.3	0.50	1.5	20.9	56.8	10.3	49.1	31.9	44.6
37	65.5	0.64	85.2	0.50	0.7	20.6	58.1	10.0	49.0	29.9	44.0
38	65.5	0.65	28.6	0.11	1.1	19.9	58.9	9.9	49.2	23.0	34.9
39	65.4	0.64	54.3	0.23	0.4	20.2	60.6	10.1	49.2	27.1	39.9
40	47.6	0.41	54.3	0.23	1.1	20.4	59.1	10.1	49.2	30.3	41.4
41	50.0	0.44	54.3	0.23	1.3	20.3	59.4	10.2	49.2	29.9	41.2
42	37.2	0.27	54.3	0.23	-0.1	20.1	62.8	9.7	49.3	33.2	42.5
43	37.2	0.27	47.1	0.20	-0.1	20.0	63.4	10.0	49.2	32.5	41.8
44	48.4	0.41	47.1	0.20	0.3	20.1	62.5	10.1	49.2	29.4	40.4
45	67.3	0.67	47.1	0.20	1.1	19.9	59.6	10.2	49.2	26.1	38.8
46	67.3	0.67	44.2	0.19	-0.4	19.8	64.2	9.9	49.2	25.6	38.2
47	57.2	0.53	44.2	0.19	2.3	19.9	55.7	10.4	49.2	27.2	39.2
48	48.3	0.41	44.2	0.19	2.1	20.0	59.4	10.4	49.2	29.1	40.1
49	66.4	0.66	44.2	0.19	1.1	19.7	56.7	9.4	49.2	25.3	38.2
50	66.3	0.66	38.8	0.16	1.5	19.7	56.8	9.7	49.2	24.4	37.3
51	49.3	0.43	38.9	0.16	1.4	19.7	57.0	9.7	49.2	27.5	38.8
52	49.3	0.42	79.5	0.44	2.0	20.2	55.7	10.3	49.2	32.4	44.5
53	49.3	0.43	60.6	0.28	1.6	19.9	56.3	9.8	49.3	30.5	42.2
54	49.3	0.43	52.7	0.23	4.4	19.7	45.1	10.6	49.2	29.7	41.0
55	57.3	0.53	52.8	0.23	2.4	19.8	54.3	9.7	49.2	27.9	40.3
56	57.3	0.53	80.0	0.44	1.4	19.9	54.8	9.4	49.3	30.4	43.9
57	57.3	0.53	39.0	0.16	1.4	19.5	60.2	10.1	49.2	26.0	37.9
58	49.7	0.43	39.0	0.16	1.8	19.5	61.1	10.1	49.2	27.5	38.7
59	68.2	0.68	39.0	0.16	2.1	19.5	51.9	9.3	49.2	23.7	36.8
60	51.7	0.46	39.0	0.16	1.8	19.3	57.1	9.8	49.2	26.7	38.4
61	51.6	0.47	21.0	0.09	2.5	19.1	54.3	9.7	49.2	23.7	33.8
62	36.6	0.28	21.0	0.09	2.5	19.1	56.5	9.9	49.2	27.6	36.1
63	33.6	0.24	21.0	0.09	1.3	18.9	62.2	9.8	49.2	28.7	36.6
64	41.9	0.33	21.1	0.09	0.9	18.9	63.6	10.0	49.2	26.2	35.2
65	41.9	0.34	57.2	0.25	2.1	19.5	55.3	9.7	49.2	31.7	42.3
66	38.9	0.30	57.2	0.25	2.4	19.3	53.1	9.8	49.2	32.7	42.7
67	38.9	0.30	56.7	0.25	2.9	19.3	50.3	9.7	49.2	32.6	42.6
68	39.0	0.30	29.9	0.12	2.3	19.2	59.9	10.1	49.2	28.3	38.1
69	39.2	0.29	88.6	0.51	0.5	14.1	77.2	9.7	49.0	34.2	45.2
70	39.1	0.29	50.1	0.21	0.8	14.3	76.4	9.8	49.3	31.0	41.2
71	30.1	0.18	50.1	0.21	6.1	15.0	65.0	10.5	49.2	35.2	42.8
72	30.1	0.20	52.2	0.22	6.8	15.2	51.3	10.3	49.2	35.4	43.2
73	30.1	0.19	52.9	0.22	9.6	16.1	45.0	11.2	49.2	36.1	43.5
74	30.1	0.18	73.0	0.37	5.6	16.9	59.3	10.0	49.2	37.8	45.4
75	45.4	0.38	72.9	0.37	5.9	17.3	40.9	10.3	49.3	32.5	44.0
76	45.4	0.37	37.3	0.14	3.8	17.5	54.6	9.6	49.2	27.4	38.5

Table B.2: Steady state data from *REPEAT HVAC* system (Part 2).

Trial	C_{bs} (%)	f_a (m^3/s)	C_{vp} (%)	f_w (l/s)	T_{ae} (°C)	T_{ar} (°C)	C_{dr} (%)	T_{ai} (°C)	T_{wi} (°C)	T_{ao} (°C)	T_{wo} (°C)
77	45.4	0.37	73.7	0.39	5.4	19.0	45.9	9.6	49.3	32.5	44.1
78	47.4	0.40	73.7	0.39	3.7	19.0	51.4	9.7	49.3	32.1	44.0
79	66.1	0.61	73.7	0.39	5.6	20.0	29.8	9.9	49.3	28.9	42.9
80	53.4	0.47	73.7	0.39	3.6	18.1	40.7	8.9	49.3	30.6	43.5
81	65.7	0.63	73.7	0.39	4.2	18.0	43.8	9.6	49.2	28.4	42.6
82	63.4	0.59	73.7	0.39	8.2	17.9	47.1	11.6	49.3	29.9	43.2
83	63.4	0.60	56.3	0.24	3.5	18.2	55.1	9.9	49.2	27.3	40.3
84	66.1	0.64	56.3	0.24	5.9	17.7	48.5	10.6	49.3	27.0	40.2
85	67.4	0.66	56.3	0.24	2.5	17.2	59.9	9.9	49.2	26.6	39.9
86	65.6	0.63	56.3	0.24	2.2	16.9	61.2	9.8	49.2	26.7	39.9
87	47.5	0.40	56.3	0.24	1.5	16.6	64.8	9.9	49.3	30.1	41.4
88	38.5	0.29	56.3	0.24	1.1	16.2	67.8	9.6	49.3	32.4	42.4
89	41.9	0.33	56.1	0.24	3.7	16.4	62.8	10.5	49.2	31.8	42.2
90	41.9	0.34	19.4	0.08	5.4	15.9	60.6	10.3	49.2	25.6	34.5
91	41.9	0.33	42.0	0.18	1.9	16.1	67.5	9.6	49.2	29.4	40.0
92	41.9	0.33	19.8	0.08	3.5	15.8	65.7	10.3	49.2	25.6	34.5
93	41.9	0.33	64.9	0.32	5.6	16.1	58.1	10.9	49.2	32.8	43.5
94	41.9	0.33	33.0	0.13	3.1	15.7	66.9	9.7	49.2	27.2	37.8
95	42.0	0.33	71.8	0.37	4.0	16.1	62.7	10.0	49.2	33.3	44.2
96	31.6	0.21	63.7	0.29	5.5	21.2	29.0	10.2	49.3	37.0	44.7
97	31.6	0.22	61.3	0.28	4.7	21.0	17.8	9.1	49.3	36.4	44.1
98	31.7	0.21	65.7	0.32	2.0	21.1	48.5	9.3	49.3	37.1	44.9
99	69.9	0.68	65.7	0.32	1.0	21.0	50.3	9.5	49.3	28.0	41.8
100	57.6	0.52	65.8	0.32	2.8	21.2	44.6	10.4	49.3	30.3	42.7
101	57.6	0.52	70.2	0.36	1.5	21.1	52.1	10.1	49.3	30.7	43.3
102	57.6	0.52	48.3	0.21	1.2	20.7	52.6	9.8	49.3	28.2	40.1
103	48.0	0.40	48.3	0.21	2.1	21.0	45.0	9.5	49.3	29.9	40.9
104	70.0	0.69	48.4	0.21	1.4	20.6	49.2	9.5	49.3	25.9	39.0
105	69.9	0.69	66.4	0.33	0.7	20.6	53.9	9.8	49.3	28.1	41.9
106	48.2	0.40	66.4	0.33	1.2	20.6	52.0	10.0	49.3	32.2	43.4
107	48.2	0.40	78.4	0.43	0.8	20.7	52.3	10.1	49.3	33.1	44.7
108	48.2	0.41	15.3	0.07	1.1	19.6	56.5	10.1	49.2	24.5	32.7
109	52.8	0.47	15.3	0.07	0.3	19.4	59.1	9.9	49.2	23.7	32.1
110	59.1	0.55	15.4	0.07	5.4	19.3	51.6	11.2	49.2	23.3	31.8
111	59.1	0.55	43.2	0.18	1.1	19.8	55.9	9.9	49.3	27.1	39.1
112	59.2	0.54	67.8	0.34	1.9	20.3	46.2	10.3	49.3	29.9	42.7
113	61.6	0.57	67.8	0.34	1.8	20.3	51.1	10.1	49.3	29.5	42.6
114	30.6	0.20	67.9	0.33	2.0	19.1	57.0	9.9	49.3	37.7	45.3
115	30.6	0.19	87.1	0.50	3.3	19.3	50.9	10.0	49.1	38.9	46.7
116	30.6	0.20	24.2	0.10	2.5	19.5	47.4	10.0	49.2	31.6	39.0
117	43.4	0.35	24.4	0.10	1.4	19.4	52.9	9.7	49.2	26.8	36.6
118	43.4	0.35	68.6	0.34	1.5	19.8	47.1	9.4	49.3	33.1	43.9
119	43.4	0.35	68.6	0.34	0.6	19.7	54.2	9.8	49.3	33.3	44.0
120	43.4	0.35	50.7	0.22	1.0	19.5	52.7	10.2	49.3	31.6	41.8
121	43.4	0.35	40.4	0.17	1.6	19.5	54.6	10.2	49.3	30.3	40.4
122	52.5	0.46	20.3	0.08	3.7	18.3	37.3	8.9	49.2	23.0	32.9
123	52.5	0.46	68.3	0.35	1.6	19.2	59.5	9.8	49.3	31.0	43.2
124	52.5	0.46	76.3	0.42	4.3	19.4	45.9	10.3	49.3	31.7	44.1
125	43.9	0.36	76.3	0.42	4.1	19.4	47.5	10.0	49.3	33.6	44.7
126	50.5	0.44	76.3	0.42	2.5	19.4	55.8	9.9	49.3	32.1	44.3
127	50.5	0.44	68.9	0.35	3.4	19.3	51.9	9.9	49.3	31.4	43.4
128	50.5	0.43	30.2	0.12	7.7	18.9	22.2	10.4	49.2	25.5	36.6
129	60.6	0.51	30.3	0.12	6.9	18.5	1.5	9.7	49.2	23.1	35.1
130	60.6	0.52	69.4	0.35	6.7	19.0	8.3	10.0	49.3	29.4	42.5
131	52.0	0.45	68.1	0.34	6.3	19.0	19.6	9.5	49.3	30.4	42.8
132	32.5	0.23	68.1	0.34	6.7	18.7	27.3	10.1	49.3	36.4	44.8
133	69.5	0.64	68.1	0.34	8.2	19.1	6.4	10.8	49.3	28.1	41.9
134	33.2	0.25	68.1	0.34	6.1	18.7	12.1	9.7	49.3	36.0	44.6
135	31.2	0.23	68.1	0.34	9.2	18.7	0.3	10.4	49.3	37.8	45.2
136	40.6	0.32	68.0	0.34	7.1	19.0	26.3	9.9	49.3	33.5	44.0
137	40.6	0.32	46.0	0.20	4.7	18.6	44.8	9.8	49.3	30.9	41.1
138	33.8	0.24	46.0	0.20	5.6	18.1	50.5	10.6	49.3	33.6	42.1
139	33.9	0.25	28.1	0.11	3.0	17.9	53.6	9.4	49.2	29.5	38.5
140	69.1	0.69	28.1	0.11	4.7	18.0	48.4	10.3	49.2	22.3	34.6
141	57.2	0.53	28.1	0.11	3.2	17.6	59.0	10.1	49.2	23.8	35.5
142	57.2	0.53	34.0	0.13	3.3	17.7	56.8	9.9	49.2	24.5	36.8
143	57.2	0.53	87.4	0.52	3.9	18.3	49.9	10.0	49.0	30.8	44.4
144	44.8	0.37	87.2	0.51	3.6	18.3	50.2	9.7	49.0	33.2	45.0
145	44.9	0.38	31.7	0.12	4.5	17.8	49.3	10.3	49.2	26.7	37.5
146	68.0	0.68	31.7	0.12	4.5	17.6	47.9	10.2	49.2	22.5	35.2
147	68.0	0.68	27.7	0.11	2.8	17.4	56.7	9.7	49.2	22.0	34.2
148	48.3	0.42	27.7	0.11	3.4	17.4	58.8	9.8	49.2	25.2	36.1
149	60.6	0.58	27.7	0.11	2.7	17.2	57.8	9.9	49.2	22.9	34.8
150	47.1	0.40	27.7	0.11	5.9	17.3	52.6	10.3	49.2	25.6	36.5
151	47.1	0.41	38.0	0.15	5.0	17.4	39.7	9.6	49.2	27.1	38.6

Table B.3: Results of testing Error Modulated Partial Stuffing on *REPEAT HVAC* for different values of K_e. Each step lasted ten minutes.

Step	K_e	RMS Error (°C)	Step	K_e	RMS Error (°C)	Step	K_e	RMS Error (°C)	Step	K_e	RMS Error (°C)
1	0.0300	0.69	51	0.0675	0.15	101	0.0300	0.87	151	0.0150	0.50
2	0.0300	0.85	52	0.0675	0.36	102	0.0300	0.35	152	0.0300	0.38
3	0.0675	0.54	53	0.0150	1.12	103	0.0150	0.45	153	0.0675	0.61
4	0.0000	0.85	54	0.0675	0.32	104	0.0300	0.35	154	0.0000	2.14
5	0.0075	0.73	55	0.0075	1.10	105	0.0675	0.44	155	0.0075	0.80
6	0.0675	1.52	56	0.0075	1.15	106	0.0000	2.19	156	0.0300	0.44
7	0.0150	0.33	57	0.0000	0.16	107	0.0075	0.80	157	0.0075	0.36
8	0.0150	0.18	58	0.0675	0.72	108	0.0300	0.39	158	0.0000	1.13
9	0.0075	0.05	59	0.0075	0.20	109	0.0075	0.38	159	0.0000	0.22
10	0.0300	0.15	60	0.0000	1.64	110	0.0000	1.09	160	0.0000	0.67
11	0.0150	0.21	61	0.0150	0.74	111	0.0000	0.16	161	0.0150	0.44
12	0.0000	1.01	62	0.0675	0.55	112	0.0000	0.72	162	0.0075	0.93
13	0.0150	0.35	63	0.0150	0.21	113	0.0300	0.82	163	0.0675	0.36
14	0.0075	0.11	64	0.0075	0.53	114	0.0300	0.94	164	0.0675	0.42
15	0.0300	0.63	65	0.0300	0.57	115	0.0675	0.68	165	0.0150	1.09
16	0.0075	0.58	66	0.0300	0.66	116	0.0000	0.85	166	0.0675	0.41
17	0.0675	0.54	67	0.0675	0.09	117	0.0075	0.77	167	0.0075	1.11
18	0.0675	0.45	68	0.0000	0.63	118	0.0675	1.69	168	0.0075	1.15
19	0.0075	0.65	69	0.0075	0.75	119	0.0150	0.19	169	0.0000	0.08
20	0.0150	0.17	70	0.0675	1.56	120	0.0150	0.15	170	0.0675	0.64
21	0.0300	0.49	71	0.0150	0.24	121	0.0075	0.07	171	0.0075	0.20
22	0.0075	0.13	72	0.0150	0.12	122	0.0300	0.34	172	0.0000	1.63
23	0.0300	0.64	73	0.0075	0.06	123	0.0150	0.21	173	0.0150	0.74
24	0.0675	0.25	74	0.0300	0.20	124	0.0000	1.02	174	0.0675	0.30
25	0.0300	0.13	75	0.0150	0.09	125	0.0150	0.36	175	0.0150	0.29
26	0.0000	0.48	76	0.0000	1.00	126	0.0075	0.14	176	0.0075	0.53
27	0.0150	1.13	77	0.0150	0.36	127	0.0300	0.74	177	0.0000	0.12
28	0.0300	1.53	78	0.0075	0.17	128	0.0075	0.64	178	0.0000	1.08
29	0.0000	1.04	79	0.0300	0.61	129	0.0675	0.67	179	0.0675	0.90
30	0.0300	0.44	80	0.0075	0.54	130	0.0675	0.47	180	0.0075	0.36
31	0.0300	1.68	81	0.0675	0.47	131	0.0075	0.73	181	0.0075	0.78
32	0.0675	0.92	82	0.0675	0.44	132	0.0150	0.10	182	0.0675	0.13
33	0.0000	0.28	83	0.0075	0.62	133	0.0300	0.61	183	0.0150	0.91
34	0.0000	0.35	84	0.0150	0.12	134	0.0075	0.09	184	0.0300	0.34
35	0.0675	0.75	85	0.0300	0.45	135	0.0300	0.68	185	0.0300	0.52
36	0.0300	1.11	86	0.0075	0.11	136	0.0675	0.40	186	0.0675	0.83
37	0.0300	0.91	87	0.0300	0.64	137	0.0300	0.12	187	0.0150	0.54
38	0.0300	0.44	88	0.0675	0.24	138	0.0000	0.45	188	0.0150	0.29
39	0.0150	0.45	89	0.0300	0.15	139	0.0150	1.28	189	0.0300	0.41
40	0.0300	0.40	90	0.0000	0.46	140	0.0300	1.50	190	0.0675	0.30
41	0.0675	0.48	91	0.0150	1.09	141	0.0000	1.06	191	0.0300	0.29
42	0.0000	2.26	92	0.0300	1.53	142	0.0300	0.46	192	0.0300	0.49
43	0.0075	0.78	93	0.0000	1.03	143	0.0300	1.66	193	0.0300	0.15
44	0.0300	0.41	94	0.0300	0.45	144	0.0675	1.06	194	0.0675	0.34
45	0.0075	0.35	95	0.0300	1.73	145	0.0000	0.41	195	0.0675	0.29
46	0.0000	1.16	96	0.0675	0.89	146	0.0000	0.36	196	0.0000	0.12
47	0.0000	0.14	97	0.0000	0.23	147	0.0675	0.77	197	0.0000	0.98
48	0.0000	0.71	98	0.0000	0.44	148	0.0300	1.04	198	0.0000	1.38
49	0.0150	0.32	99	0.0675	0.75	149	0.0300	0.90	199	0.0075	0.80
50	0.0075	0.76	100	0.0300	1.05	150	0.0300	0.35	200	0.0075	1.22

REFERENCES

[1] *MATLAB Version 5.3.* The Mathworks Inc., Natick, Massachusetts, 01760-1500, 1999.

[2] S. Alotaibi, M. Sen, B. Goodwine, and K.T. Yang. Numearical simulation of the thermal control of heat exchangers. *Numerical Heat Transfer*, 41:229–244, 2002.

[3] Charles W. Anderson, Douglas C. Hittle, Alon D. Katz, and R. Matt Kretchmar. Synthesis of reinforcement learning, neural networks and PI control applied to a simulated heating coil. *Artificial Intelligence in Engineering*, 11(4):421–429, 1997.

[4] Michael Anderson. MIMO robust control for heating, ventilating, and air-conditioning (HVAC) systems. M.S. Thesis, Colorado State University, Fort Collins, Colorado, 2001.

[5] S. Bittanti and L. Piroddi. Nonlinear identification and control of a heat exchanger: A neural network approach. *Journal of the Franklin Institute*, 334(1):135 – 153, 1997.

[6] Jean-Pascal Bourdouxhe, Marc Grodent, and Jean Lebrun. *Reference Guide for Dynamic Models of HVAC Equipment.* Number ISBN 1-883413-60-5. American Sodiety of Heating, Refrigeration and Air-Conditioning Engineers, Inc., Atlanta, Georga 30329, 1998. Publication compiled by ASHRAE.

[7] H.T. Chen and K.C. Chen. Simple method for transient response of gas to gas cross flow heat exchangers with neither gas mixed. *International Journal of Heat and Mass Transfer*, pages 2891–2898.

[8] C.C. Delnero, D.C. Hittle, P.M. Young, C.W. Anderson, and M.L. Anderson. Neural networks and PI control using stady state prediction applied to a heating coil. In *Proceedings of CLIMA2000*, pages 58–71, 2001.

[9] Chris C. Delnero, Dave Dreisigmeyer, Douglas C. Hittle, Peter M. Young, Charles W. Anderson, and Michael L. Anderson. Exact solution to the governing pde of a hot water-to-air finned tube cross-flow heat exchanger. *International Journal of Heating, Ventilationg, Air-Conditioning and Refrigerating Research*, 10(1):21–31, January 2004.

[10] Christopher Delnero. Neural networks and PI control using steady state prediction applied to a heating coil. M.S. Thesis, Colorado State University, Fort Collins, Colorado, 2001.

[11] G. Diaz, M. Sen, K.T. Yang, and R.L. McClain. Dynamic prediction and control of heat exchangers using artificial neural networks. *International Journal of Heat and Mass Transfer*, 44:1671–1679, 2001.

[12] J. R. Gartner and L. E. Daane. Dynamic response relations for a serpentine crossflow heat exchanger with water velocity disturbance. *ASHRAE Transactions*, 74 (Part 1):53–68, 1969.

[13] J. R. Gartner and H. L. Harrison. Dynamic characteistics of water-to-air crossflow heat exchangers. *ASHRAE Transactions*, 71 (Part 1):212–223, 1965.

[14] Roger W. Haines and Douglas C. Hittle. *Control Systems for Heating, Ventilating, and Air Conditioning*. Chapman & Hall, New York, NY, sixth edition, 2006.

[15] D.C Hamilton, R.G. Leonard, and J.T. Pearson. Dynamic response characteristics of a discharge air temperature control system at near full and part heating load. *Ashrae Transations*, 83:251–268, 1974.

[16] D. Hittle, C. Anderson, P.M. Young, C. Delnero, and M.L. Anderson. Patent: A combined proportional plus integral (PI) and neural network (NN) controller nsf# 01-035, September 2001.

[17] M. Jalili-Kharaajoo and B.N. Araabi. Neuro predictive control of a heat exchanger: comparison with generalized predictive control. In *Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits, and Systems, United Arab Emirates*, pages 675–678, 2003.

[18] S. Kabelac. The transient response of finned crossflow heat exchangers. *International Jurnal of Heat and Mass Transfer*, 32:1183–1189, 1989.

[19] K.W. Lim and K.V. Ling. Generalize predictive control of a heat exchanger. *IEEE Control Systems Magazine*, pages 9–12, 1989.

[20] M Mishra, PK Das, and S Sarangi. Transient behavior of crossflow heat exchangers with longitudinal conduction and axial dispersion. *Journal of Heat Transfer*, 126:425–433, 2004.

[21] M Mishra, PK Das, and S Sarangi. Transient behaviour of crossflow heat exchangers due to perturbations in temperature and flow. *INTERNATIONAL JOURNAL OF HEAT AND MASS TRANSFER*, 49(5-6):1083–1089, MAR 2006.

[22] Anderson M.L., Buehner M.R., Young P.M., Hittle D.C., Anderson C., Tu J., and D. Hodgson. An experimental system for advanced heating, ventilating, and air conditioning (HVAC) control. *Energy and Buildings*, 39(2):136–147, 2007.

[23] Anderson M.L., Buehner M.R., Young P.M., Hittle D.C., Anderson C., Tu J., and D. Hodgson. MIMO robust control for heating, ventilating, and air conditioning (HVAC) systems. *IEEE Transactions on Control Systems Technology*, 16:475–483, 2008.

[24] M. Nøgaard, O Ravn, N.K. Poulsen, and L.K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, London, 2000.

[25] K. Ogata. *Modern Control Engineering*. Prentice Hall, fifth edition, 2009.

[26] N. Pappa, G. Kaliraj, and J. Shanmugam. Real-time implementation of model predictive neural controller for heat exchanger. *Control and Intelligent Systems*, 23:184–189, 2005.

[27] N. Pappa and J. Shanmugam. Neural network based predictor for control of cascsded thermal process. In *Proceedings of the International Conference on Intelligent Sensing and Information, Chennai, India*, pages 289–294, 2004.

[28] J.T. Pearson, R.G. Leonard, and R.D. McCutchan. Gain and time constant for finned serpentice crossflow heat exchangers. *ASHRAE Transactions*, 80:255–267, 1974.

[29] C. Renotte, M. Van de Wouwer, and M. Remy. Neural modeling and control of a heat exchanger base on SPSA techniques. In *Proceedings of the American Control Conference*, pages 3299–3303, Chicago, Illinois, 2000.

[30] Wilfried Roetzel and Yimin Xaun. *Dynamic Behavior of Heat Exchangers*. WIT Press/Computational Mechanics Publications, Southampton, United Kingdom, 1999.

[31] F.E. Romie. Transient response of crossflow heat exchanges with zero core thermal capacitance. *Journal of Heat Transfer*, 116:775–777, 1994.

[32] Center Scientifique Et Technique Du Batiment. SIMBAD building and HVAC toolbox. Toolbox for MATLAB, published by the Math Works, Inc.; Natick, Mass 01760-1500.

[33] S.J.Hepworth and A.L.Dexter. Neural control of a non-linear hvac plant. In *Proceedings of the 3rd IEEE Conference on Control Applications*, pages 1849–1854, 1994.

[34] M. Spiga and G. Spiga. Step response of the crossflow heat exchanger with finite wall capacitance. *International Journal of Heat and Mass Transfer*, 35:559–565, 1992.

[35] F.H. Syed and S. Idern. Transient performance of a cross flow heat exchanger using finit difference methods. In *Proceedings of the 2008 ASME International Mechanical Engineering Congress and Exposition*, pages 1333–1342, Boston, MA, 2008.

[36] W.L. Tse and W.L. Chan. An automatic data acquisition system for on-line training of artificial neural network-bassed air handling unit modeling. *Measurment*, 37:39–46, 2005.

[37] J.D. Wright, A. Sen, and N.K. Sinha. On-line identification of the parameters of a dual-input heat-exchanger system. *Canadian Journal of Chemical Engineering*, 52:682–684, 1974.

[38] Y. Xuan. Transient analysis of mulitpass crossflow heat exchangers. *Heat and Mass Transfer*, 31:223–230, 1996.