An Experimental System for Advanced Heating, Ventilating and Air Conditioning (HVAC) Control

Michael Anderson ^{a,1}, Michael Buehner ^{a,*}, Peter Young ^a,
Douglas Hittle ^b, Charles Anderson ^c, Jilin Tu ^c,
David Hodgson ^b

^a*Department of Electrical and Computer Engineering, Colorado State University,
Fort Collins, Colorado 80523, USA*

^b*Department of Mechanical Engineering, Colorado State University, Fort Collins,
Colorado 80523, USA*

^c*Department of Computer Science, Colorado State University, Fort Collins,
Colorado 80523, USA*

Abstract

While having the potential to significantly improve heating, ventilating and air conditioning (HVAC) system performance, advanced (e.g., optimal, robust and various forms of adaptive) controllers have yet to be incorporated into commercial systems. Controllers consisting of distributed proportional-integral (PI) control loops continue to dominate commercial HVAC systems. Investigation into advanced HVAC controllers has largely been limited to proposals and simulations, with few controllers being tested on physical systems. While simulation can be insightful, the only true means for verifying the performance provided by HVAC controllers is by

actually using them to control an HVAC system. The construction and modeling of an experimental system for testing advanced HVAC controllers, is the focus of this article.

A simple HVAC system, intended for controlling the temperature and flow rate of the discharge air, was built using standard components. While only a portion of an overall HVAC system, it is representative of a typical hot water to air heating system. In this article, a single integrated environment is created that is used for data acquisition, controller design, simulation, and closed loop controller implementation and testing. This environment provides the power and flexibility needed for rapid prototyping of various controllers and control design methodologies.

Key words: Heating ventilating and air conditioning (HVAC), Experimental system, Rapid prototyping environment, Advanced MIMO control.

* Corresponding author. Tel. +1 970 491 2800; Fax +1 970 491 2249

Email addresses: `mlanderson@hotmail.com` (Michael Anderson),
`mbuehner@engr.colostate.edu` (Michael Buehner), `pmy@engr.colostate.edu`
(Peter Young), `hittle@engr.colostate.edu` (Douglas Hittle),
`anderson@cs.colostate.edu` (Charles Anderson), `tu@cs.colostate.edu` (Jilin
Tu), `dh688526@engr.colostate.edu` (David Hodgson).

¹ M. Anderson while preparing this article, was a Graduate Student in the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, Colorado.

1 Introduction

Accurate heating, ventilation, and air conditioning (HVAC) system models are required for controller synthesis and a physical test bed is required for controller verification. Often times, the tasks of system identification, controller synthesis, and controller verification are done using various software and analysis tools that are not directly compatible with each other. This may lead to complications and errors when the data are transported between the various platforms. Furthermore, it is often necessary to custom write code to implement different controllers, which is a time-consuming and error-prone task. In order to alleviate these problems, a setup was developed that allowed for data acquisition (DAQ), modeling, simulation, and controller design, simulation, and verification within a single integrated software/hardware environment. Auto-code generation tools were employed so that controllers could be implemented directly from the high-level design, with no necessity for the designer to write their own code. The building of this integrated environment, which serves as a rapid prototyping platform for designing, testing, and implementing a wide variety of control algorithms, is the focus of this paper.

Note that while other simulation packages exist [10,18], they do not have the controller design and physical system implementation capabilities of the setup presented within. The paper concludes with a brief demonstration of the flexibility of the environment considered herein by designing, implementing, and verifying two vastly different control architectures. Among these controllers is a full MIMO robust controller. While a Linear Quadratic Gaussian (LQG) MIMO controller has been implemented on a room size air conditioner[11], the controller demonstrated here is the first known implementations of an

$\mathcal{H}_\infty$ robust controllers on a physical system using commercial style HVAC components.

2 Integrated Development Environment Setup

Fig. 1. The Experimental HVAC System

In commercial heating, ventilation, and air conditioning (HVAC) systems, a *central air supply* provides air at a controlled temperature and flow rate for use in heating (or cooling) a space. A heating coil is used in the central air supply for heating the discharged air. Regulating the rate at which hot water flows through the heating coil controls the temperature of the discharged air. The flow rate of the discharged air is regulated to maintain a predetermined static air pressure within the duct. Typically, the space within a building is divided into smaller *zones*, allowing the temperature within each zone to be maintained independently of the others. Each zone contains a reheat coil that is used to moderate the final temperature of the air discharged into the zone.

The experimental HVAC system, shown in Fig. 1, was constructed for verifying the performance of the controller designs. This system (consisting of external and return air dampers, a variable speed blower and a heating coil) is similar to the central air supply in a commercial HVAC system. A diagram representing this system is shown in Fig. 2, with the associated mnemonics defined in Table 1.

Fig. 2. Diagram of the Experimental System and Interface Signals

The temperature of the discharged air is a function of the temperature and flow rate of both the air and water flowing through the coil. The flow rate of the air is primarily a function of the speed at which the blower is operating, but it is also affected by the position of the return air and external air dampers. The dampers allow the return and external air mix to be varied, in regulating the temperature of the air flowing into the coil. A three-way mixing valve allows the flow rate of the water through the coil to be varied.

The physical system was connected to PC to form an integrated environment used for rapid prototyping. An overview of the hardware and software used are given next. For more details about the experiment setup, see [2].

Table 1

Key to Mnemonics

		T_{ai}	Temp. of air input
C_{dr}	Command damper return	T_{ao}	Temp. of air output
C_{de}	Command damper external	T_{ws}	Temp. of water supply
C_{wh}	Command water heater	T_{wi}	Temp. of water input
C_{vp}	Command valve position	T_{wo}	Temp. of Water output
C_{bs}	Command blower speed	F_w	Flow rate of water
T_{ae}	Temp. of air external	F_a	Flow rate of air
T_{ar}	Temp. of air return	P_{wh}	Power (input) water heater

2.1 Control Hardware

Fig. 3. PC Based Control Hardware

Control and data acquisition (DAQ) functions for the experimental HVAC system were implemented using the *Windows98*© based PC² shown in Fig. 3. Two MATLAB supported interface cards were used in interfacing the com-

² Dell OptiPlex GX1p, 500 MHz Pentium III having 4 ISA slots

puter and the experimental system. A 12-bit, 32 channel, analog differential input card³, with two analog outputs and a user configurable, digital input or output port (8-bits) was used to interface the analog sensor signals. A 12-bit, six-channel analog output card⁴, also having 16 digital inputs and 16 digital outputs, provides the control outputs.

The external hardware was connected with the interface cards in the PC using additional hardware for signal conditioning, signal attenuation/amplification or switching. These operations were carried out using hardware contained within the *interface* and *drive* cabinets shown in Fig. 3. The interface cabinet (top) contained most of the hardware used to connect the computer to the experimental system's sensor and control signals. The drive cabinet (bottom) contained the variable frequency drive and associated hardware used to power the blower motor. It also housed the logic and power devices used in controlling the power distributed to the interface cabinet and major system components.

2.2 Control Software

All of the control application software ran under *MATLAB*®⁵. The *toolboxes* used in conjunction with *MATLAB* were: *Simulink*®, *Real-Time Workshop*® (RTW) and *Windows Target*® (WT). The *Simulink Toolbox* is an interactive graphic environment for modeling and simulating dynamic systems. *Real-Time Workshop* extends to *Simulink* the ability to interface in *real-time* to real world devices, or in the RTW vernacular, to *targets*. RTW supports both *real* and

³ National Instruments, AT-MIO-64E-1, ISA interface card

⁴ Advantec, PCL-726, 6 Channel D/A Output (ISA) card

⁵ *MATLAB* is published by *the MATH WORKS Inc.; Natick, Mass.*

virtual targets. Real targets are (I/O) devices having their own processors running real-time tasks and communicating with the PC/RTW. A virtual target is a task which runs on the PC under a real-time Windows kernel, and communicates with RTW as a virtual external process/device. For slower processes, Windows Target allows RTW to support devices not incorporating their own real-time processors. As HVAC system components are fairly slow, Windows Target was chosen for use on the experimental HVAC system.

The hardware and software tools (mentioned above) were used to create the integrated environment. Within this integrated environment, two main Simulink models were used, namely one model was used for controller simulation, and the other model was used for DAQ and controller implementation. The important features of the software package used are that data was easily passed between a command line workspace and the block diagrams (graphical models), and auto-code generation was used to implement the controllers on the physical system. The block diagrams provide a means for interfacing the physical system (DAQ and controller verification) and for controller simulation, while the workspace provides the commands required to design advanced controllers, and to analyze and plot the results. This means that we can analyze, model, simulate, implement, and test all from within the same software environment. The use of auto-code generation tools means that we do NOT write any code to implement controllers. These capabilities alleviate errors, AND since new designs may be implemented in only a few minutes, this environment provides a rapid prototyping platform for testing our controller methodologies. Note that with the above setup we can readily implement advanced, non-standard (e.g., MIMO) controllers, and furthermore our designs are implemented and tested on the real system as rapidly and easily as they are tested in simulation.

For more details about the advanced controller implementations, see [2,3].

Models for both data acquisition and control purposes were implemented in Simulink. Such a model, designed for manually controlling the experimental system while acquiring experimental data, is shown in Fig. 4. In this figure, the five blocks in the upper left corner were used in manually controlling the experiment. It should be noted that most of the blocks shown in Fig. 4 represent *subsystems*. These subsystems were used in the implementation of scaling, filtering, control and logic functions. *Slider blocks* allow the user to adjust the command levels, using a slider, to vary a scalar gain. The first slider block, “Water Heater Temp SetPoint” was used in setting the temperature ($^{\circ}\text{C}$) at which the boiler’s output water was maintained. Temperature control was accomplished using an *anti-wind-up PI* controller. The output of the PI controller was scaled to provide the proper analog output voltage using the block “Scaling1.” The second slider block “Damper Pos. Return Air,” was used to set the positions (0% to 100% of open) of the return air and external air dampers in the mixing box. They were both set using one input, since they were ganged together. This allowed the ratio of the return and external (outside) air to be varied while maintaining a “constant” combined inlet opening. The third slider block was used to adjust the water flow control valve position. The fourth slider block was used to set the blower (fan) speed as percentage of its maximum speed.

Measurements from the experimental system were read into the integrated environment using the block “AT-MIO-64e In.” The signals were then demultiplexed, filtered, scaled and connected to the scope blocks for real-time display and data logging.

Fig. 4. Simulink Model Used for Data Acquisition

3 Modeling the Experimental System

The development of a reasonably accurate model⁶ of the experimental system was necessary for the analysis, synthesis and simulation testing of HVAC controller designs. As the diagram of Fig. 2 illustrates, the system consisted of two basic parts, the air and water subsystems. These subsystems converged at the heat exchanger (heating coil) where heat energy was transferred between water and air. The water subsystem consisted of the boiler (electric water

⁶ While a perfect model is never available, the model developed here captures enough of the system dynamics that the simulated and verified controllers produce similar responses.

heater), “constant” flow rate water pump, three-way mixing valve, copper tubing, and the waterside of the heating coil. The air subsystem consisted of the external (outside) air input, return air input, ducting, blower/fan, mixing box (including external and return air dampers) and the airside of the heat exchanger. The airflow dampers and water flow control valve were pneumatically actuated, requiring the use of voltage-to-pneumatic transducers.

It was anticipated that the configuration of the experimental system will change over time, thus it was desirable to have a model that could easily be updated. Consequently, the system model was based primarily upon individual components or a logical grouping of components. Since the intent of the model was to use it for controller development and simulation, it was essential that the model accurately capture the steady state and dynamic characteristics of the system. The dynamics associated with the sensors were not separately modeled, but were incorporated into the dynamics of the overall system. Considering these objectives, the model was broken into the five subsystems identified in Table 2.

Each subsystem model was developed using models of its constituent components. Many of the components modeled exhibited nonlinear steady state behavior [5]. These nonlinear characteristics were included in all the components modeled, with the exception of the heating coil. Modeling the dynamics of heating coils is a complex problem [8,12] and was a major part of a parallel project ([6,7]. Since a nonlinear dynamic model of the heating coil was not available during the course of this project, a linear model was developed around an operating point. Within the operating range imposed by the linear coil model, the dynamic characteristics of the components were accurately

Table 2

The model's five subsystems

Subsystem	Description
Blower	variable speed centrifugal fan
Mixing Box	external and return air dampers and volume
Heating Coil	four-pass serpentine heat exchanger
Flow Control	equal percentage, pneumatically actuated valve
Boiler	electric water heater and constant speed pump

represented by first order systems with transport delays.

The overall model of the experimental system has six inputs (four commanded inputs and two disturbances from the surrounding environment), namely C_{vp} , C_{bs} , C_{dr} , C_{wh} , T_{ar} , and T_{ae} , and eight outputs, namely F_w , F_{ws} , F_a , T_{wo} , T_{ai} , T_{wi} , and T_{ws} . These mnemonics are listed in Table 1. The interconnection of the inputs, outputs and subsystems is shown in Fig. 5. Having identified the structure of the model, work proceeded in developing the subsystem models.

3.1 Data Acquisition

Prior to developing a model of the experimental system, a series of experiments designed to extract the steady state and dynamic characteristics of the components, subsystems and overall system were conducted. Specifically, the four inputs in the upper left corner of Fig. 4 (with the exception of the water heater temperature set point, which was held constant) were adjusted to various set points. For each subsystem (except the heating coil), a least squares

Fig. 5. Overall Model of Experimental HVAC System

polynomial fit was used to model the nonlinear dynamics, while first order dynamical systems were used to correct the overall subsystem dynamics. In some cases, linear interpolation was used to model components that behaved linearly. Since the purpose of this model was to design and simulate various control algorithms, some nonlinear effects (e.g. the hysteresis effects from the pneumatic actuators) were not modeled. Instead, these effects were viewed as model uncertainty and were accounted for in the advance controller designs. In the next five subsections, the subsystem models for the experimental HVAC system (shown in Fig. 5) are developed.

3.2 Blower Model

The *blower* is the main component in the variable air volume (VAV) system. A variable frequency drive allows the speed of the centrifugal fan to be changed,

varying the airflow rate through the system. The airflow rate was primarily a function of the blower speed, but it was influenced by the positions of the dampers in the mixing box. Thus the blower was modeled as a 1×2 system having the commanded blower speed (C_{bs}) and commanded return-air damper position (C_{dr}) as inputs, with airflow rate (F_a) as the output. The blower model shown in Fig. 6 contains three key blocks: “c2Fa”, “AdjFa2” and “Flow Dynamics”. These blocks modeled the commanded blower speed to airflow rate relationship, the effect of the dampers on the airflow rate and the dynamics associated with changes in the airflow rate, respectively.

Fig. 6. Overall Blower Model

Theoretically the airflow rate should have been a linear function of the fan speed. While not quite linear, the actual relationship between commanded blower speed (C_{bs}) and airflow rate (F_a) was fit using the fourth-order polynomial in eqn (1). This equation was implemented in the model using the block “c2Fa”. This relationship assumes that the return air damper was fully open (and the external air damper fully closed) and represented the maximum airflow rates attainable for any given blower speed.

$$\begin{aligned}
 F_a = & 1.23 \times 10^{-8} C_{bs}^4 - 3.93 \times 10^{-6} C_{bs}^3 + 3.77 \\
 & \times 10^{-4} C_{bs}^2 - 2.32 \times 10^{-3} C_{bs} - 1.7 \times 10^{-2}
 \end{aligned} \tag{1}$$

The positions of the return air and external (outside) air dampers impacted the

airflow rate. The dampers were “ganged” together by the controller/interface, so the positions of both are determined by the return air damper control signal (C_{dr}). In the overall blower model shown in Fig. 6, the block “AdjFa” predicted the airflow rate (as a percentage of the maximum possible airflow rate) as a function of the return air damper position. This again is a nonlinear relationship and was approximated using the third-order polynomial in eqn (2).

$$Fa_{adj} = -0.0233C_{dr}^3 - 0.0287C_{dr}^2 + 0.119C_{dr} + 0.933 \quad (2)$$

The overall blower model was formed by placing a block representing the airflow dynamics after the product of the peak airflow block (c2Fa) and the block correcting this flow rate based upon the damper positions (AdjFa). The accuracy of the blower model was verified using data from the experimental system as input to the model. The model’s airflow rates were plotted along with the measured flow rates in Fig. 7. The blower model captures enough of the blower’s dynamic and steady state characteristics for controller synthesis. Most of discrepancies are due to hysteresis affects from the pneumatically controlled dampers and sensor noise, which are sources of model uncertainty.

3.3 *Mixing Box Model*

The mixing box was volumes of ducting prior to the heating coil including both the external air and return air ducts. Parallel blade dampers were used to vary the area of the openings, thus controlling the mix of external (outside) and return air. In the experimental system, the external and return air dampers

Fig. 7. Measured (Dotted) and Modeled (Solid) Air Flow (C_{dr} : 50%, 10%, 90%)

were “ganged” together within the controller so as to collectively maintain a constant inlet area. This configuration allowed the dampers to vary the mix of external (outside) and return air with only small variations in the airflow rate.

The mixing box was modeled by considering the temperatures and ratios of the two air streams and the dynamics of the airflow. This was done in the model shown in Fig. 8. The block “NormCdr” maps the commanded return damper signal (C_{dr}) from the voltage range to the range $[-1,0]$ with 0 corresponding to the return air damper being fully open. Since the external air damper was “ganged” to the return air damper, the normalized external damper command, namely $\tilde{C}_{de}$, was obtained (at the output of the summing node) by the simple relationship $\tilde{C}_{de} = C_{dr} + 1$. At steady state, the temperature of air exiting the mixing box was determined by using the linear interpolation in eqn (3).

$$\tilde{T}_{ai} = (C_{dr} + 1)T_{ae} - C_{dr}T_{ar} \quad (3)$$

This simplified approach worked for the experimental system, where the ratio of system pressure drop to open damper pressure drop was such that a reasonably linear relationship between blade position and air flow rate occurred. The block “flow dynamics” was used to obtain the proper output dynamics.

Fig. 8. Mixing Box Model

3.4 Boiler Model

The *boiler* subsystem consisted of an electric *water heater*, a *voltage to duty cycle converter* (for varying the average power supplied to the heating elements) and a “constant” speed water pump. The temperature of the water out of the boiler (T_{ws}) depended upon the temperature of the water returned to the boiler and the power applied to the heater (P_{ws}). For DAQ, the temperature of water out was held constant (via feedback control) at 50.5 °C. This served as the *operating point* for the boiler. The boiler model shown in Fig. 9 consisted of three blocks. The “water return” block modeled the temperature of the water returned to the boiler to be reheated. The “C2Pw” block modeled the electrical power applied to the water heater in response to the water

heater command, C_{wh} . The final block modeled the water heater, warming the returned water in response the electrical power applied. The boiler's dynamics were also included in this block. These three block subsystems are detailed in Figs. 10, 11, and 12.

Fig. 9. Boiler Model

The mean temperature of the water returned to the boiler (T_{wr}) was determined by the ratio and temperatures of the water discharged from the heating coil and that which bypassed the heating coil. Calculation of T_{wr} required four parameters, the flow rate and temperature of the water bypassing the coil (F_{ws} and T_{ws}) and the flow rate and temperature of the water discharged from the coil (F_w and T_{wo}). The *water return* block from Fig. 9, which is detailed in Fig. 10, calculated the temperature of the water returned to the boiler using the linear interpolation defined in eqn (4). Note that 0.5 °C represents the thermal losses in the bypass.

$$T_{wr} = \frac{(T_{wo}F_w) + (T_{ws} - 0.5)(F_{ws}F_w)}{F_{ws}} \quad (4)$$

The controller command (C_{wh}) was used to vary the duty cycle of the (207 volts) AC power supplied to the water heater. This relationship is defined in eqn (5) and was implemented in the “C2Pw” block shown in Fig. 11.

Fig. 10. Block Modeling the Temperature of Water Returned to Boiler

$$P_w = 4833.5C_{wh} - 2523.21 \quad (5)$$

Fig. 11. Command to Duty Cycle (Power) Converter Model

Having the outputs of the previous two blocks (T_{wr} and P_w) as its inputs, the “water heater” block shown in Fig. 12 modeled the temperature of the water output from the boiler (T_{ws}). The electric power (P_w) supplied to the water heater warmed the water returned to the boiler (T_{wr}), raising its temperature as a function of the water flow rate (F_{ws}) and the applied power. The transfer function block labelled “TF” captured both the steady-state temperature rise in response to the power applied to the water heater (P_w), as well as the dynamics of the water heater output. The transfer function coefficients were selected by fitting experimental data. Near the operating point, the water flow rate through the water heater was considered constant and a constant transport delay was adequate for modeling the transport of the water from the water heater’s input to output.

Fig. 12. Heater Model for Temperature of Water Out of Boiler

The model of the water heater was validated by using data from the experimental system as input. The model's output is plotted along with the experimental systems output in Fig. 13. This model adequately captures most of the boiler dynamics. The discrepancies arise from unmodeled dynamics and sensor noise, both of which are forms of model uncertainty.

Fig. 13. Measured (Dotted) and Modeled (Solid) Boiler Output Water Temperature

3.5 Water Flow Control Valve Model

The three-way water flow control valve, being an *equal percentage* type, exhibits a nonlinear relationship between valve position and water flow rate. The three-way valve controls the flow rate of hot water through the heating coil, diverting the excess flow around the coil and back to the boiler. This flow, in conjunction with the water exiting the heating coil, provided a “constant” water flow rate through the pump and boiler. The valve was positioned using a piston and spring type pneumatic actuator fitted with a “positive positioning relay”. An *electronic-to-pneumatic transducer* (E/P) was used to control the pneumatic pressure applied to the actuator in proportion to the applied voltage.

Fig. 14. Blocks Forming Model for Water Flow Control Valve

The model of the water flow control valve (Fig. 14) consisted of five cascaded blocks, representing water flow rates through the heating coil (F_w) and the system’s total water flow rate (F_{ws}) in response to changes in the commanded valve position. The first block, “Cvp2Avp”, related the commanded valve position to the measured steady-state valve position. The output of this block was the electrical input to the electric-to-pneumatic transducer. The second block used a first-order system to represent the dynamics of this transducer. “Avp2Fw” relates the steady-state water flow to the (actual) valve position. This is a nonlinear relationship and was fit to experimental data with the fourth-order polynomial in eqn (6).

$$F_w = -4.9 \times 10^{-12} A_{vp}^4 + 1.3 \times 10^{-9} A_{vp}^3 - 6.9 \times 10^{-8} A_{vp}^2 + 4.5 \times 10^{-6} A_{vp} - 7.8 \times 10^{-8} \quad (6)$$

The fourth block is a first-order system that is used to represent the valve actuator dynamics. The coil offered a greater resistance to water flow than the bypass circuit. Thus, the total water flow rate (that through the coil and that diverted around it) varied as a function of valve position. The last block, “Fw2Fws”, predicted the total water flow rate through the system (F_{ws}) as a function of the water flow rate through the heating coil. The third-order polynomial in eqn (7) was used to fit this relationship between the two water flow rates.

$$F_{ws} = -240966 F_w^3 + 888 F_w^2 - 0.53915 F_w + 0.00063 \quad (7)$$

These five blocks comprise the valve subsystem. The model for the valve was verified by comparison of the model’s outputs with those of the experimental system. The valve command used to drive the experimental system for this test was captured and used as the input signal to the model. The water flow rate response of both the experimental system and the model are compared in Fig. 15. The model essentially captured the valve’s steady state and dynamic characteristics, with the discrepancies being another form of model uncertainty.

Fig. 15. Measured (Dotted) and Modeled (Solid) Water Flow

3.6 Heating Coil Model

The heating coil used in the experimental system was a four-pass, counter-flow, water-to-air heat exchanger. The transfer of heat energy from water to air depended upon the physical properties of the heat exchanger and was a function of the flow rates and temperatures of the two fluids. The relationships between the inputs and outputs were nonlinear. As mentioned previously, dynamically modeling counter-flow heat exchangers, especially the multi-pass type, is quite complex. For the system considered here, a linear model was developed around an operating point. The operating point was chosen to provide a good operating range attainable within a range of moderate temperatures, since testing occurred during the spring and summer months. Table 3 describes the operating point used for developing the linear model.

The coil was represented as a 2×4 system having the four inputs given in

Table 3

Operating point for linear coil model

T_{ai}	Temperature of air into the coil	$19.8^{\circ}C$
F_a	Flow rate of air into the coil	$0.29 \text{ m}^3/s$
T_{wi}	Temperature of water into the coil	$50^{\circ}C$
F_w	Flow rate of Water into the Coil	$1 \times 10^{-4} \text{ m}^3/s$
T_{wo}	Temperature of Water out of the Coil	$36.1^{\circ}C$
T_{ao}	Temperature of Air out of the Coil	$40.8^{\circ}C$

Table 3 and outputs: T_{ao} and T_{wo} , the temperature of the air and water out of the coil, respectively. The coil was modeled as two 1×4 subsystems, sharing the same four inputs. The “Tao” subsystem modeled the temperature of the air out of the coil and the “Two” subsystem modeled the temperature of water out of the coil. The overall model of the coil was formed from these two subsystems, as shown in Fig. 16. Since this model was linear about the operating point, the operating point “constants” were subtracted from the four inputs prior to connecting to the linear subsystems. Conversely, the operating point “constants,” were added to T_{wo} and T_{ao} outputs within the subsystems.

The “Two” subsystem of Fig. 16 models the temperature of water out of the coil as a function of the four inputs. Thus, “Two” can be thought of as representing the waterside of the heating coil. The mass of the heating coil provided heat storage capacity, which caused an exponential (first-order) output delay. In addition, the coil tubes extended over 550 inches in length and thus induced a temperature gradient, as well as another transport delay. In the model, the average temperature across the coil was used and the delays were

Fig. 16. Main Coil Model with Subsystem Blocks

represented as one transport delay. The delay times and transfer functions associated with each input were derived from experimental data obtained by forcing a *step change* in one input, while holding the others constant. This procedure was repeated several times for each successive input, to obtain a good fit between model and data.

Each input to the coil had a corresponding transfer function relating it to the output. For the water side of the heating coil, the four transfer functions in eqns (8–11) and four transport delays, were interconnected to form the subsystem as shown in Fig. 17.

$$TF_1 = T_{wo}(F_a) = \frac{-25.8}{30s + 1} \quad (8)$$

$$TF_2 = T_{wo}(F_w) = \frac{101 \times 10^3}{30s + 1} \quad (9)$$

$$TF_3 = T_{wo}(T_{ai}) = \frac{0.4279}{s + 1} \quad (10)$$

$$TF_4 = T_{wo}(T_{wi}) = \frac{0.49}{25s + 1} \quad (11)$$

Fig. 17. Water-Side Coil Subsystem

The “Tao” subsystem of Fig. 17 models the temperature of air out of the coil as a function of the four inputs. Thus, it can be thought of as representing the airside of the heating coil. Similar to the “Two” subsystem, the delay times and transfer functions associated with each input were derived from experimental data obtained by forcing a *step change* in one input, while holding the others constant. The resulting four transfer functions in eqns. (12 – 15) and the four transport delays were interconnected in the “Tao” subsystem model, which is shown in Fig. 18.

$$TF_1 = T_{ao}(F_a) = \frac{-30}{65s + 1} \quad (12)$$

$$TF_2 = T_{ao}(F_w) = \frac{50 \times 10^3}{55s + 1} \quad (13)$$

$$TF_3 = T_{ao}(T_{ai}) = \frac{0.21}{4s + 1} \quad (14)$$

$$TF_4 = T_{ao}(T_{wi}) = \frac{0.79}{50s + 1} \quad (15)$$

Fig. 18. Air-Side Coil Subsystem

The complete coil model, containing the two coil subsystems “Two” and “Tao”, was verified using experimental data as the inputs into the model. In Fig. 19, the simulation results are compared with the experimental data as part of validating the model.

3.7 Overall HVAC System Model

Having completed the five subsystem models in Simulink, the overall system model was assembled as the graphical part of the integrated environment (as shown in Fig. 5) and configured for validation using experimental data as inputs. Actual data obtained from the experimental system was loaded into the integrated environment workspace and was seamlessly transferred to the graphical model as inputs. The model’s outputs were saved back to the workspace using scope blocks. After the simulation was run, the model’s

Fig. 19. Measured (Red) and Modeled (Blue) Step Response of Individual Coil Transfer Functions

outputs, in response to the experimental data (inputs), were plotted along with the experimental systems outputs as shown in Fig. 20. With the setup developed here, the tasks of simulation, DAQ, and plotting were all achieved using the same software tool.

In Fig. 20, the bottom plot shows the six (four command and two disturbance) inputs applied to both the experimental system and the simulation model. The top and middle plots compare the experimental systems outputs (dotted lines) with the modeled outputs (solid lines). The top plot shows the air and water temperatures, while the middle plot show the air and water flow rates as percentages of their maximum values. From examining the top plot, the temperatures of air into the heating coil (T_{ai}) and the air and water out of the coil (T_{ao} and T_{wo}) were adequately replicated by the simulation model.

Fig. 20. Measured (Dotted) and Modeled (Solid) System Outputs

The temperature of water into the coil (T_{wi}) and out of the boiler (T_{ws}) was maintained in the experimental system using a PI controller implemented in the DAQ model. While the simulation model operated “open-loop,” from the experimental systems water heater control signal (C_{wh}), the (boiler) model provided virtually an identical water temperature into the coil, (T_{wi}).

In the middle plot, the model produced a reasonable replica of the experimental system’s air and water flow rates (F_a and F_w). The steady-state error in the water flow rate was due to positioning uncertainty associated with the pneumatic actuator. A comparison of these plots confirms that the simulation model was a reasonable representation of the experimental system (at least over a range appropriate for the linear coil model).

4 Implementing Various Controller Architectures

The main thrust for developing the model was to create a single integrated environment that could be used for controller synthesis and experimental verification (i.e., an environment for rapid prototyping). Since this model was split into subsystems with measurable output signals, a wide variety of controller structures were available. Specifically, if single-input single-output (SISO) control were to be employed, then certain individual outputs in Fig. 4 would be connected in feedback to their respective inputs. An example of this was shown in the DAQ phase where a PI controller regulated the water heater temperature. If a multiple-input multiple-output controller (MIMO) were to be employed, then a group of system outputs would be connected to a MIMO controller as inputs, and the controller outputs would replace the (manually entered) commanded system inputs. In this section, two examples of vastly different control structures, namely a set of distributed SISO PI controllers and a “full” MIMO robust controller, are implemented to demonstrate the power and versatility of the systems illustrated in Figs. 4 and 5. These results are from the first known implementation of a MIMO robust controller on a physical HVAC system using commercial style components.

Fig. 21. HVAC Controller Based Upon Three SISO PI Controllers

4.1 Industry Standard PI Controller Implementation

For comparison, the HVAC system was controlled using standard HVAC techniques (i.e. individual PI controllers for each subsystem). These controllers were tuned using well-known design techniques in [9]. From here on, this reference PI controller is labelled K_{PI} . The controller architecture is given in Fig. 21.

In this setup, the PI controller $K_{PI}^{T_{ws}}$ is the same PI controller that was used to regulate the water heater for DAQ in Fig. 4. Since the deployment of the three SISO PI controllers only required access to measurable signals, simulation and implementation of K_{PI} was accomplished by rewiring Figs. 4 and 5. Since the fan had it's own built-in controller (variable frequency drive), it was controlled directly by varying the commanded blower speed (C_{bs}). The response of controller K_{PI} to step changes in F_a and T_{ao} on the physical system is shown in Fig. 22.

Fig. 22. Controller K_{PI} Experimental Test Results

Controller K_{PI} was designed to provide the best response on the physical system (while maintaining stability over the entire operating range) using the industry standard techniques given in [9]. For more details on the design, see [2]. Observe that the controller is able to track step changes in the output air temperature (T_{ao}) and is able to regulate the output air temperature in the presence step changes of airflow rate (F_a) changes (e.g., the step change

at 1800 sec.). This means that the controller is able to provide some performance in terms of tracking and disturbance rejection. However, the amount of performance is limited by the SISO control. Note the sluggish reaction of T_{ao} to a step change in its reference input around 250 sec. Note also the interaction of T_{ao} when F_a is stepped around 1800 sec, and again the sluggish recovery from that disturbance. In the next section, a MIMO robust controller is implemented to illustrate the type of performance increase that is possible.

4.2 MIMO Robust Controller Implementation

Robust control theory addresses the effects that discrepancies between the model and the physical system (model uncertainty) may have on the design and performance of linear feedback systems. Robust control provides a unified design approach under which the concepts of gain margin, phase margin, tracking, disturbance rejection and noise rejection are generalized into a single framework. Typically, the uncertainties considered in robust control theory are bounded using norms. The $\mathcal{H}_\infty$ norm is frequently applied in the robust controller design process, as it may be used to bound signal energy. The $\mathcal{H}_\infty$ robust controller design presented next, was based upon the structured singular value (μ). For information regarding the structured singular value in robust control theory see [15,17,19,20].

For the robust controller design and synthesis, a linear version of the system model was needed. Rather than forming one linear model of the entire system, it was advantageous (for the controller design task) to obtain separate linear models for each of the five subsystems. The linear models (about an operating point) were easily extracted from the individual subsystem models using a

function built-in to the integrated environment. Since a linear model for the heating coil already existed, the same operating point was used in extracting the linear models for the other four subsystems.

A full MIMO $\mathcal{H}_\infty$ robust controller, referred to herein as K_{R3} , was developed for the linear model using a software package that was compatible with the integrated environment[4]. The controller and plant interconnections are shown in Fig. 23. The 4×7 robust controller (four controller outputs / seven controller inputs) regulated the input air temperature (T_{ai}), airflow rate (F_a) and output air temperature (T_{ao}) to track reference levels, namely rT_{ai} , rF_a , and rT_{ao} , respectively. However, within this controller, the water heater control output (C_{wh}) was left as a free control variable, allowing the water supply temperature to be varied. For the specific details of the controller K_{R3} , see [2].

The controller in Fig. 23 only requires access to signals that are available in Figs. 4 and 5. Therefore, simulating the controller was accomplished by rewiring Fig. 5 and implementation was accomplished by rewiring Fig. 4. Since this single integrated environment was equipped with all the required tools, design, simulation, and implementation were performed seamlessly.

All controller designs were tested using the simulation model prior to testing on the experimental system. Step inputs were used to excite the model. Data resulting from a simulation test of the controller is plotted in Fig. 24. The simulation test indicates that the MIMO controller should be able to track step changes in the output air temperature and flow rate of air better than the controller K_{PI} on the experimental system.

After confirming the function of the controller design using the simulation model, it was tested on the experimental system. The response of the closed

Fig. 23. System Using MIMO Robust Controller, K_{R3}

loop system to step changes in the discharge air temperature and airflow rate reference inputs (rT_{ao} and rF_a) is plotted in Fig. 25. In this experiment, the reference input air temperature (rT_{ai}) was held constant at $20^\circ C$. In the top two panels of Fig. 25, the dotted lines are the reference inputs and the dashed and solid lines are the measured system outputs (i.e. the DAQ inputs). The bottom panel of Fig. 25 shows the controller outputs (DAQ outputs) and the disturbances from the surrounding environment (i.e. the system inputs).

To begin, the system was brought to steady state with a discharge air temperature (T_{ao}) of $39.5^\circ C$. Once the system reached steady state, various step changes were applied to the flow rate of air (F_a) and output air temperature (T_{ao}). The controller was designed to “tightly” control the input air temperature to track the constant input air temperature reference (rT_{ai}), which was

Fig. 24. Controller K_{R3} Simulation Test Results

held constant throughout the test. The flow rate of air was designed to track its reference level in steady state, but was allowed to vary when tracking a step change in the output air temperature. This allowed for a smaller settling time for tracking output air temperature changes. Specifically, observe the MIMO controller was able to track a 5°C step change (occurring around 700 sec) in about 200 seconds, whereas the PI based controller took roughly 900 seconds

Fig. 25. Controller K_{R3} Experimental Test Results

(see Fig. 22). This translates to roughly a 400% increase in performance (or a settling time of that is 25% of the industry standard PI). Similarly, in response to the step change in airflow rate at 2300 seconds (i.e., a disturbance to the output air temperature), the controller was able to recover the output air temperature in roughly 300 seconds, whereas the PI controller took roughly 1000 seconds to reach steady state. These results illustrate some of the power

of MIMO controllers. Another facet of this power can be seen if one looks at the action the MIMO controller takes in response to the step change in the reference input for F_a around 1100 sec. In addition to the obvious required response of dropping C_{bs} to reduce airflow, the controller simultaneously reduces C_{wh} and C_{vp} , so that there is not too much hot water flowing into the coil. As a result the temperature T_{ao} is kicked much less severely than we saw for airflow changes with the industry standard SISO PI controller approach. The MIMO controller models and accounts for multivariable interactions, instead of just reacting to them as disturbances. As a result, although the plant contains many dynamic interactions, the controller is able to make a coordinated change in several actuators to achieve essentially independent control over the reference variables.

5 Conclusions

The experimental system provided a means to develop a model of a real HVAC system, confirm the validity of the model, design MIMO robust controllers and to evaluate their performance on the physical system. One integrated environment provided a seamless tool for controller design, simulation, implementation, and validation. This greatly simplified the task of creating and maintaining the data acquisition, simulation and control models and eliminated the need for data translation/conversion between different application environments (with the potential for errors).

The experimental system was used to verify some MIMO controllers [2,3] with great success. Furthermore, this platform will now be used as a tool for our future research program, giving us the ability to rapidly try out an array of

different controller design approaches for HVAC systems. In the near term we plan to use this tool to verify the performance of a number of other advanced HVAC controller designs [1,14] currently under development. For instance, one such design combines robust control and reinforcement learning theories, to provide an adaptive controller, which is robustly stable even while adapting [13,14].

The power of the integrated environment developed here is that all of the aforementioned controller architectures, as well as any other controller architecture that may be desired, may be simulated and implemented using the same software tool. With the graphical interface to rewire connections and the auto-code generation capabilities, simulating and implementing the various control architectures may be done within minutes and the potential for errors is almost eliminated. The experimental system is very versatile, and has proven to be a capable rapid prototyping platform, for implementing and testing advanced HVAC controller designs.

6 Acknowledgments

The authors would like to thank the National Science Foundation for providing funding for this project under awards CMS-9804757, CMS-9732986, and ECS-0245291.

References

- [1] C.W. Anderson, D.C. Hittle, A.D. Katz and R.M. Kretchmar, *Synthesis of Reinforcement Learning, Neural Networks and PI Control Applied to a*

Simulated Heating Coil, Journal of Artificial Intelligence in Engineering, vol. 11, no 4, pp. 423–431, 1997.

- [2] Michael L. Anderson, *MIMO Robust Control for Heating, Ventilating and Air Conditioning (HVAC) Systems*, Masters Thesis, Colorado State University, Fort Collins, CO, Spring 2001.
- [3] M.L. Anderson, M.R. Buehner, P.M. Young, D.C. Hittle. C.W. Anderson, J. Tu, and D.A. Hodgson *MIMO Robust Control for Heating, Ventilating and Air Conditioning (HVAC) Systems*, Currently submitted for publication, Summer 2005.
- [4] G.J. Balas, J.C. Doyle, K. Glover, P. Packard, and R. Smith, *μ -Analysis and Synthesis TOOLBOX User's Guide*, The MathWorks, Inc., Natick, Mass. 01760-1500, 1996.
- [5] Jean-Pascal Bourdouxhe, Marc Grodent and Jean Lebrun, *Reference Guide for Dynamic Models of HVAC Equipment*, American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc. Atlanta, Georgia 30329, (ISBN 1-883413-60-5), 1998.
- [6] Christopher Delnero, *Neural Networks and PI Control Using Steady State Prediction Applied to a Heating Coil* Masters Thesis, Colorado State University, 2001.
- [7] C. Delnero, D. Dreisigmeyer, D.C. Hittle, P.M. Young, C.W. Anderson, and M.L. Anderson *Exact solution of the governing PDE of a hot water to air finned tube cross flow heat exchanger* International Journal of HVAC&R Research, vol. 10, no. 1, pp. 21-32, 2004.
- [8] J. R. Gartner and L. E. Daane, *Dynamic Response Relations for a Serpentine Crossflow Heat Exchanger with Water Velocity Disturbance* ASHRAE Transactions, vol. 74 (part 1), pp. 53-68, 1969.

- [9] R. W. Haines and D. C. Hittle, *Control Systems for Heating, Ventilating, and Air Conditioning* Sixth Edition, Kluwer Academic Publishers, Norwell, MA, 2003.
- [10] P. Haves, L.K. Norford, M. DeSimone, and L. Mei, *A standard simulation testbed for the evaluation of control algorithms and strategies, RP-825 final report* Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 1996.
- [11] X. He, S. Liu, H.H. Asada, and H. Itoh, *Multivariable Control of Vapor Compression Systems* International Journal of of HVAC&R Research, vol. 4, no. 3, pp. 205-230, 1998.
- [12] S. Kabelac, *The Transient Response of Finned Crossflow Heat Exchangers* International Journal of Heat and Mass Transfer, vol. 32, pp. 1183-1189, 1989.
- [13] R.M. Kretchmar, *A Synthesis of Reinforcement Learning and Robust Control Theory* PhD. Dissertation, Department of Computer Science, Colorado State University, 2000.
- [14] R.M. Kretchmar, P.M. Young, C.W. Anderson, D.C. Hittle, M.L. Anderson, J. Tu and C.C. Delnero *Robust Reinforcement Learning Control* Proceedings of the American Control Conference, June, 2001.
- [15] Andy Packard and John C. Doyle, *The Complex Structred Singular Value*, Automatica, vol. 29, no. 1, pp. 71-109, 1993
- [16] Sigurd Skogestad and Ian Postlethwaite, *Multivariable Feedback Control* John Wiley and Sons Ltd., West Sussex, England, 1996.
- [17] Peter M. Young, *Controller Design with Real Parameteric Uncertainty* International Journal of Control, vol. 65, pp. 469-509, 1996.
- [18] P. Peng Xu, P. Haves, and J. Deringer, *A Simulation-Based Testing and Training Environment for Building Controls*, Proceedings of SimBuild 2004,

Building Sustainability and Performance Through Simulation, August 4-6.
Boulder, CO, USA, 2004.

- [19] Kemin Zhou, John C. Doyle and Keith Glover, *Robust and Optimal Control*,
Prentice Hall, Upper Saddle River, NJ 07458, 1996.
- [20] Kemin Zhou and John C. Doyle, *Essentials of Robust Control*, Prentice Hall,
Upper Saddle River, NJ 07458, 1998.