Neural Networks and PI Control using Steady State Prediction Applied to a Heating Coil

Chris C. Delnero, *Department of Mechanical Engineering, Colorado State University*
Douglas C. Hittle, *Department of Mechanical Engineering, Colorado State University*
Peter M. Young, *Department of Electrical and Computer Engineering, Colorado State University*
Charles W. Anderson, *Department of Computer Science, Colorado State University*
Michael L. Anderson, *Department of Electrical and Computer Engineering, Colorado State University*

ABSTRACT

A new approach to controlling HVAC heating coils using a neural network to predict the future steady state output of the controller is investigated. The approach implements a controller that resets the value of a proportional plus integral (PI) control loop with the expected value needed to achieve the desired steady state coil temperatures. The traditional PI control method achieves the desired steady state values after the integral term accrues enough summed error to do so. Since the heating coil system has variable gain, performance of the PI control can be sluggish at low gain states. The method investigated in this report shows that by using a properly trained neural network to reset the PI control loop at the moment of a set point change or coil inlet disturbance, the rise time of this PI/net controller is reduced greatly from the rise time of the same PI controller acting alone.

1 INTRODUCTION

Heating in a central HVAC system is often done by controlling the outlet air temperature from a hot water heating coil. Control usually involves a proportional plus integral (PI) control device that monitors the air outlet temperature and modulates the water flow rate via a control valve. A schematic of the heating coil under control of a PI controller is shown in figure 1.

Typically, the PI controller is tuned to produce a quick stable response for the highest gain state of the system. If the system moves to a lower gain, then the controller will remain stable but produce a sluggish response. This happens because the outlet air temperature will not increase as much for a given change in water flow rate at a low gain state as it will for a high gain state. Thus, a longer time is needed for the integral term to accrue enough summed error to bring the system to the newly specified set point. Therefore, most HVAC system PI controllers perform poorly when a low gain state is encountered and perform reasonably well when a high gain state is encountered.

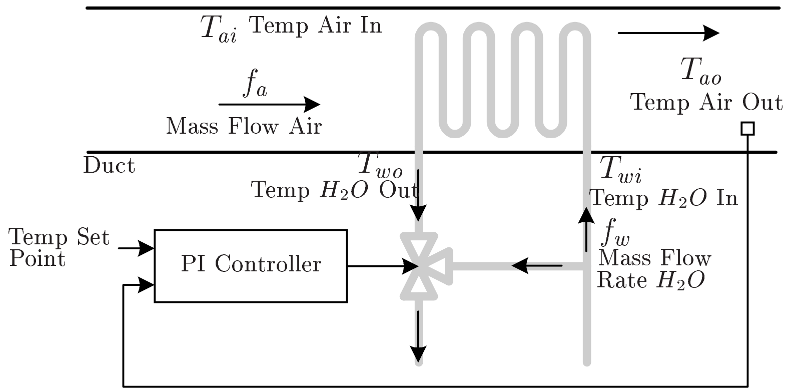

Figure 1: Schematic of heating coil with PI control

If, at the moment of a set point change or system disturbance, the controller could predict the valve position needed to bring the outlet air temperature to the specified set point, then the valve could be repositioned immediately. This would eliminate the time needed for the integral term to bring the system to the steady state and thus produce a controller that could perform well for the entire range of high and low gain states (Anderson et al., 1998).

Neural networks have been used on several HVAC applications. Massie, Curtiss and Kreider (1998), used neural networks to predict chiller equipment performance (Massie et al., 1998). Diaz et al. (1999), used neural networks to analyze heat exchanger data (Diaz et al., 1999). Anderson, Hittle et al. (1998), investigated several PI control configurations using neural networks applied to a HVAC system (Anderson et al., 1996). Curtiss et al. (1994), used neural networks to optimize energy consumption for a commercial scale HVAC system (Curtiss et al., 1994). Ang Heng Kah et al. (1995), used neural networks to control the temperature of an air conditioned bus more effectively (Kah et al., 1995). Seem (1997), while not using neural networks used a pattern recognition adaptive controller (PRAC) to adjust the proportional and integral gain time constant of a controller while under closed loop control (Seem, 1997).

The purpose of this report is to show how using a properly trained neural network to predict future valve positions combined with a PI controller performs as well or better than the same PI controller acting alone for the entire range of gain states expected by the valve and heating coil system.

The experimental apparatus used in this report is shown in figure 2. A variable speed fan near the outlet duct draws air through the system. Outside air enters through the outside air duct and return air enters through a duct open to the inside of the lab. Each air flow is controlled by a parallel blade damper near where the air flows meet in the mixing box. After the mixing box, the air passes through a filter and then through a heating coil. The air finally passes through the fan and is discharged. The hot water side consists of a pump, an electric water heater, an expansion tank, and a three way mixing valve that controls the amount of water flow through the heating coil.

Control hardware consists of 1000 ohm platinum temperature probes connected to data acquisition and control hardware in a personal computer. Control signals from the computer pass to electronic to pneumatic converters whose output modulates valve and damper actuators. In addition, the air flow through the unit is monitored via a calibrated pitot rack. Data acquisition and control was accomplished using Matlab and Simulink

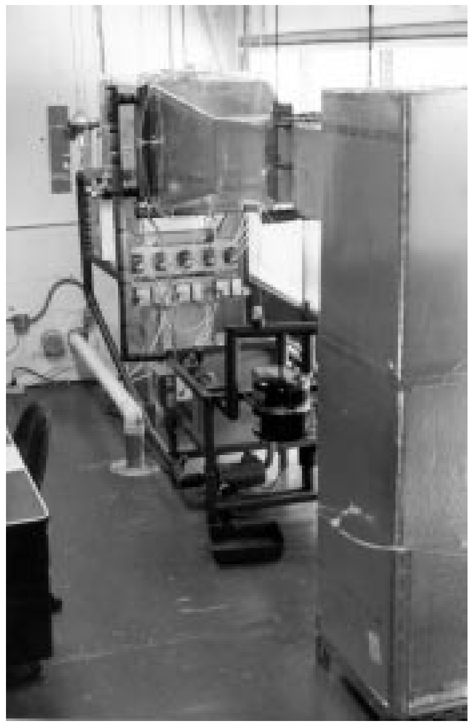

Figure 2: Over View of the HVAC Setup

2 STEADY STATE MODEL

The formulation of the neural network controller relies on the development of a steady state model. The model for the steady state prediction used the effectiveness-NTU method. This method employs the use of heat transfer coefficients and coil geometric data to express the heat transfer from the water to air side. Using equations 1 through 6, the steady state outlet water and air temperatures can be calculated knowing geometric data and heat transfer coefficients that can be found in a coil manufacturers catalog. See Kays and London (Kays an London, 1984) and Incropera and Dewitt (Incropera and Dewitt, 1996) for details.

$$\frac{1}{UA} = \frac{1}{\eta h_a A_a} + \frac{\ln(\frac{D_o}{D_i})}{2\pi k L} + \frac{1}{h_w A_w} \tag{1}$$

$$NTU \equiv \frac{UA}{C_{min}} \tag{2}$$

$$\varepsilon_p = 1 - \exp\left[\frac{1}{C_r}\left(\frac{NTU}{n}\right)^{0.22}\exp\left[-C_r\left(\frac{NTU}{n}\right)^{0.78}\right] - 1\right] \tag{3}$$

$$\varepsilon = \frac{\left[\frac{1-\varepsilon_p C_r}{1-\varepsilon_P}\right]^n - 1}{\left[\frac{1-\varepsilon_p C_r}{1-\varepsilon_p}\right]^n - C_r} \tag{4}$$

$$T_{aoss} = T_{ai} + \frac{\varepsilon C_{min}(T_{wi} - T_{ai})}{C_a} \tag{5}$$

$$T_{woss} = T_{wi} - \frac{\varepsilon C_{min}(T_{wi} - T_{ai})}{C_w} \tag{6}$$

Where:
A = area a-air side w-water side $[m^2]$
c_p = specific heat $[\frac{J}{kg*^oC}]$
C_{max} = maximum heat capacitance rate of a fluid
$[\frac{J}{s*K}] = \dot{m}(c_p)_{max}$
C_{min} = minimum heat capacitance rate of a fluid
$\frac{J}{s*K} = \dot{m}(c_p)_{min}$
$C_r \equiv \frac{C_{min}}{C_{max}}$
D = diameter i-inside o-outside [m]
ε = total heat exchanger efficiency
ε_p = heat exchanger efficiency of one pass
h = heat transfer coefficient a-air side w-water side $[\frac{W}{m^{2o}C}]$
k = thermal conductivity of coil material, $[\frac{W}{m*^oC}]$
L = length of heat exchanger [m]
$\dot{m}$ = mass flow rate $[\frac{kg}{s}]$
n = number of coil passes
NTU = Number of Transfer Units
T_{ai} = air inlet temperature $[^oC]$
T_{aoss} = steady state air outlet temperature $[^oC]$
T_{wi} = water inlet temperature $[^oC]$
T_{woss} = steady state water outlet temperature $[^oC]$
UA = overall heat transfer coefficient $[\frac{W}{^oC}]$

The coil hot water to air heat exchanger with this report is 4 pass, counter flow arranged, cross flow, finned tube, and both fluids unmixed. Most of the geometrical data can be found by simple measurements and calculations if not listed in a coil manufacturers catalog. The heat transfer coefficients are usually taken from empirical relationships printed in the literature. The air side and the water side heat transfer correlations for this heat exchanger are taken from Kays and London (Kays an London, 1984). Because the correlations in Kays and London did not exactly match the actual geometry of the heat exchanger, a few adjustments had to be made. The correlations were adjusted slightly to fit experimental steady state results while still retaining the same general form as the Kays and London correlations.

3 CONTROLLER DESIGN

In order to analyze the performance of the neural network controller, a standard SISO (single input single output) PI controller was implemented on the heating coil. The outlet air temperature was the measured value of the controlled variable while the hot water valve position was the controlled variable. The PI control equation is written as follows

$$O = K_p e + K_i \int e dt \tag{7}$$

Where:
O is the controller output or valve position
e is the error defined as the difference between the set point and the measured value of the controlled variable $T_{set} - T_{ao}$ [^{o}C]
K_p is the proportional gain constant
K_i is the integral gain constant, [1/s].
Written in discrete time this equation becomes

$$O_t = K_p e_t + K_i \sum_{j=0}^{t} e_j \Delta t \tag{8}$$

Here Δt is the sampling rate [s]. Using the discrete time equation at time t and $t-1$ we can write:

$$O_t = K_p e_t + K_i \Delta t \sum_{i=0}^{t} e_i \tag{9}$$

$$O_{t-1} = K_p e_{t-1} + K_i \Delta t \sum_{i=0}^{t-1} e_i \tag{10}$$

On subtracting Equation 10 from Equation 9 we have:

$$O_t = O_{t-1} + K_p(e_t - e_{t-1}) + K_i e_t \tag{11}$$

In order to obtain the proportional and integral constants, the open loop response was used with the system operating at the highest gain expected (Bekker et al., 1991). The high gain state is determined to be at low air flow and low water flow conditions. The PI controller was tuned at this state so that it would remain stable at all other gain states expected by the coil. At low gain states such as those at higher air and water flow rates, the PI constants chosen are expected to produce sluggish control of the outlet air temperature. This sluggish response is expected because the controller will have to actuate a wider range of the valve position to obtain the same outlet air temperature set point change. The Simulink (The Math Works, 1999) implementation of the PI controller is shown in figure 3.

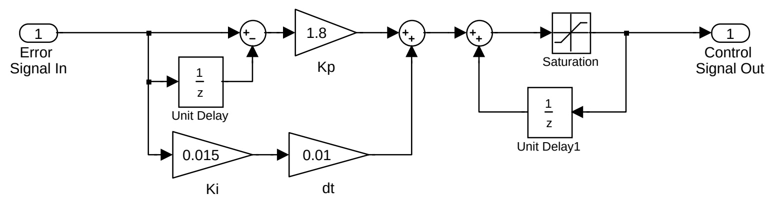

Figure 3: MATLAB implementation of the PI controller

If, at the moment of a set point change or system disturbance, the controller could be set to the value that the integral term would eventually sum to in order to bring the outlet air temperature to the specified set point, then the valve could be repositioned immediately. By positioning the valve instantaneously to the final position, the time that the regular PI controller takes to wind up to this position is eliminated. Moving immediately to the correct steady state output can be accomplished by using a well trained neural network. The output of the neural network will basically “stuff” the control loop with the correct steady state valve position the moment of a set point change instead of reaching that same valve position after the length of time needed for the integral loop to accrue the same value. See figure 4 for the implementation of the stuffing term with the original PI controller discussed previously.

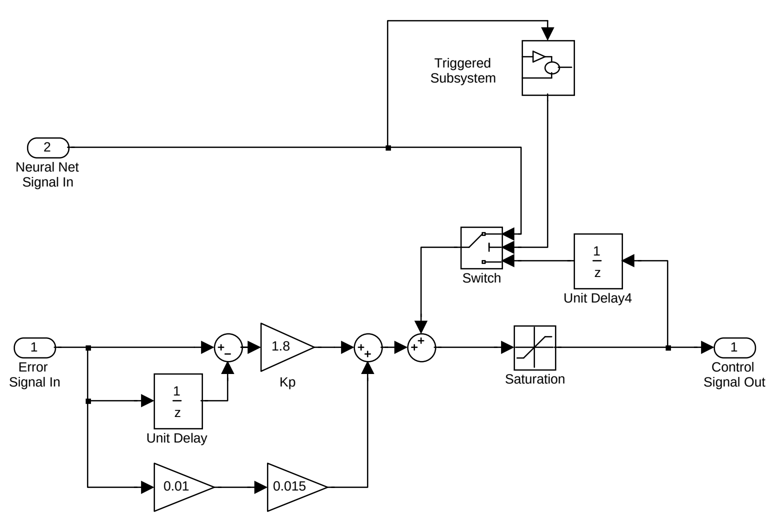

Figure 4: Schematic of PI/net controller

To predict the final steady state valve position, a neural network was created using the Neural Network Toolbox in MATLAB. A schematic of the neural net is shown in figure 5. The network was trained to produce the correct steady state valve position command that corresponds to the state of four coil parameters: inlet air temperature, water inlet temperatures, air flow rate, and outlet air temperature set point. Basically, for three coil inlet conditions, a single valve position exists that will bring the coil outlet air temperature to the desired set point. Knowing the four parameters ensures that the valve position is known also.

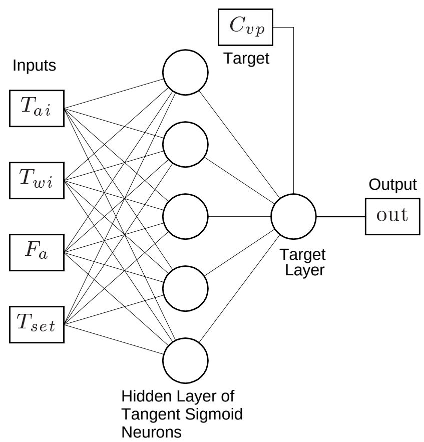

Figure 5: Schematic of the neural network

The neural network was first trained using data sets produced using the steady state model discussed in the previous section. All the input and set point values were chosen randomly within the ranges expected during coil operation. The inlet water temperature ranged between 45 ^{o}C and 60 ^{o}C, the air inlet temperature between 10 ^{o}C and 35 ^{o}C, the air flow rate between 0.2 m^3/s and 0.9 m^3/s, and the outlet air set point between 30 ^{o}C and 55 ^{o}C. Using this model, a data set of 2200 input and output pairs was generated. Of this data, 2000 was used for training, 100 for validation, and 100 for testing. A network with five units in the hidden layer was used.

Another neural network was trained using data obtained directly from steady state experiments. To obtain real steady state data, several open loop tests were performed for varying coil inlet conditions. Due to fluctuating measurements, especially the varying air flow rate (the Rocky Mountain foot hills where our laboratory is located are usually quite windy), steady state was only roughly achieved. Steady state was determined to exist when the fluctuating signals were centered around an obvious specific value for more than 100 seconds. To record data at the relative steady state, the mean value of all the signals needed for training were taken over a 50 second period within the region determined to be centered on a specific value. This averaging method was used throughout this study wherever values were extracted from real data. For the neural network trained with real data, 100 data sets were used for training, 30 for validation, and another 30 for performance measure. These 160 data sets contain the same variables used in training that the neural network trained with model data used and they also cover most of the range of the coil inputs. A network with six units in the hidden layer was used.

The next step was to incorporate a neural network in a real time controller. Using the "gensim" command in the MATLAB Neural Net Toolbox, a Simulink model of the neural network was created. The sampling rate was 100 Hz. For every time step, the neural network produced the expected valve position command or C_{vp} that corresponds to the valve position that will bring the coil to the desired set point. This C_{vp} was scaled to the actual valve position, 0 to 100% of the valve stem travel where 0% is a fully closed valve and 100% is a fully open valve. Within the original PI controller, Simulink code was developed that would observe when the network produced a valve position that differed by more 3% from the last time the network intervened in the PI control loop. When the network predicted a valve position that differed by more than 3% from its last prediction, the controller replaces O_{t-1} in equation 11 with the network prediction. For example, if the PI controller of equation 11 has been at steady state for a while and the neural network has been consistently predicting a valve position value of, say 20% open during this time, the neural network does not intervene. If one of the coil inlet conditions changes or the set point changes so as to make the neural net predict a valve position less than 17% open or greater than 23% open, then the neural network intervenes in the PI control loop. At that time step, the neural network replaces the previous control output O_{t-1} with its predicted value so the control equation for this time step becomes

$$O_t = \text{Net Prediction} + K_p(e_t - e_{t-1}) + K_i e_t \tag{12}$$

where

$$\text{Net Prediction} \approx \text{the PI controller's final value of } K_i \Delta t \sum_{j=0}^{t} e_j \tag{13}$$

The controller output is thus set immediately to what the neural net predicts. As shown by equation 13, the neural net stuffs the control loop with the value that the integral term would eventually obtain in order to reach the specified set point. Thus, the time needed for the effect of the integral term to accrue enough error to bring the controller to reach this steady state value is eliminated. For the next time step, the controller reverts back to the original PI control loop of equation 11. The neural net will not intervene again until it's output value is ±3% more or less than the value it just intervened with.

The deviation of 3% of the valve position was experimentally determined to be the optimum. 5% or greater showed that the neural network did intervene frequently enough and less than 3% showed that the neural net would never give the integral loop a chance to correct steady state error that might be inherited from an inaccurate neural net prediction.

4 CONTROLLER COMPARISON

Sample dynamic comparisons of the PI controller, PI/net controller trained with model data, and the PI/net controller trained with real data for set point changes and disturbance rejections are shown in figures 6 through 9. Rise times for a range of gain states and disturbance rejections are shown in figures 10 through 13.

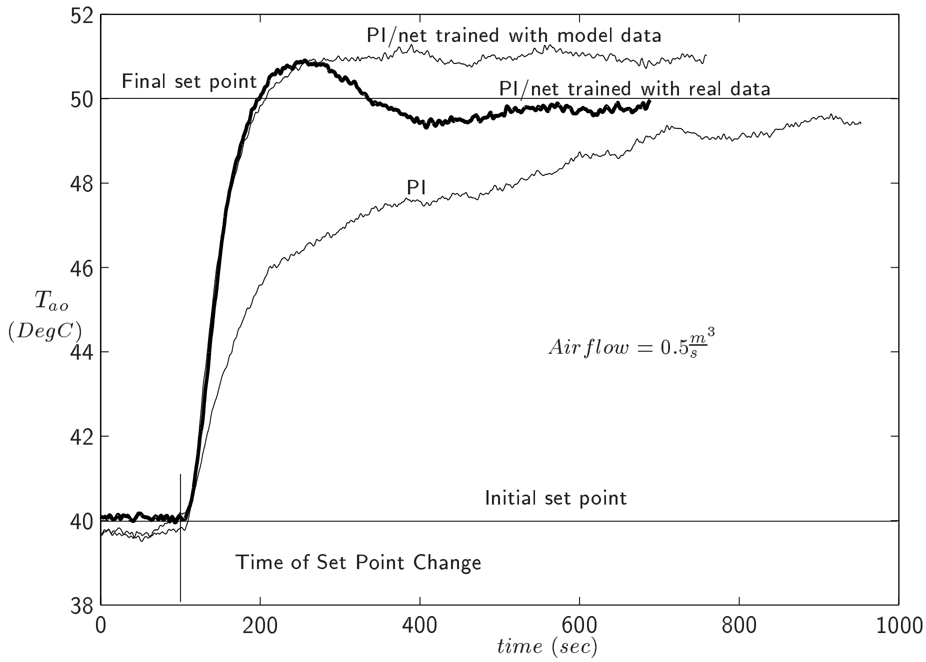

Figure 6: Dynamic Comparison of PI and PI/net Controllers for a Mid to Low Gain Increased Set Point Change

As predicted, the PI controller acting alone shows an increasingly sluggish response as the gain of the system lowers as shown in figures 10 through 13. The rise time of the PI controller goes from 100 seconds at the high gain state to 550 seconds at the low gain state. rise times for the PI/net controller trained with model data are around 100 seconds for all gain states. Rise time is defined here as the time it takes for the outlet air temperature to go from $\pm 0.5^oC$ from the initial steady state temperature to $\pm 0.5^oC$ of the final steady state temperature. Only, at the high gain state, the state where the PI controller was tuned, does the PI controller perform about as well as the PI/net controllers.

The PI controller also shows a slow response for disturbance rejection as compared with the response of the PI/net controllers as shown in figures 12 and 13. Figure 7 shows that both the PI/net controllers regain the temperature set point much faster and also with less temperature change than the PI controller. The outlet air temperature for both PI/net controllers falls almost 1.5 ^oC below the set point before regaining it within 100 seconds. The PI controller has a rise time of almost 700 seconds and lets the outlet air temperature fall 3.5 ^oC below the set point before regaining it.

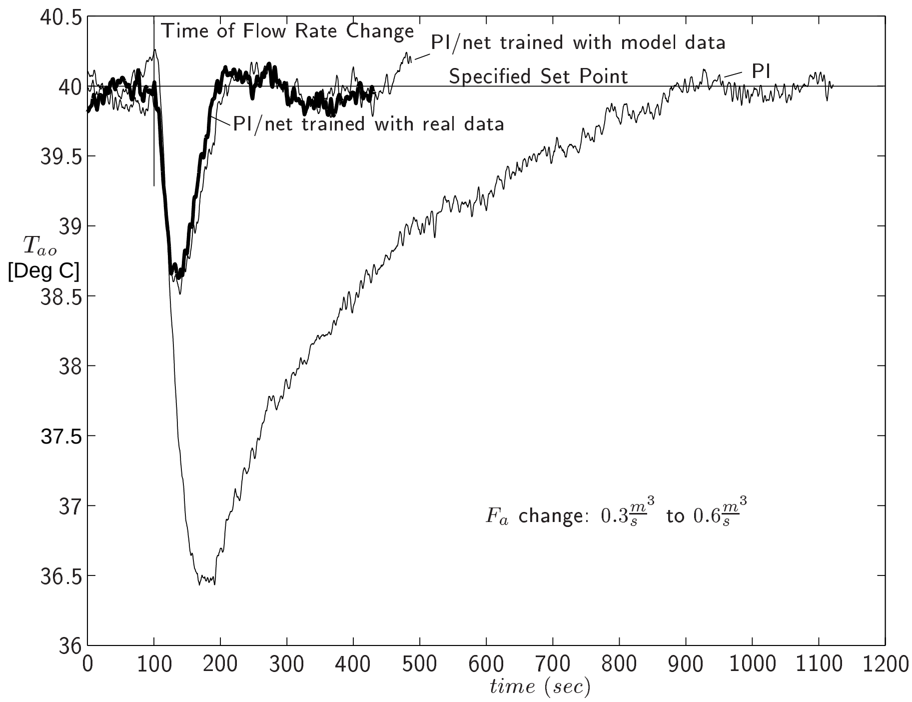

Figure 7: Dynamic Comparison of PI and PI/net Controllers for an Increased Air Flow Rate Change while Retaining Set Point

5 CONCLUSION

In controlling the outlet air temperature on an HVAC hot water-to-air heating coil, it was found that using a neural network to stuff the PI controller loop with the expected controller value needed to achieve a specified outlet air set point shows improved controller performance over the same PI controller acting alone. Not only did the PI/net controller out perform the PI controller in almost every comparison but the PI/net controller also showed a consistent response time of around 100 seconds for the entire range of gain states. The PI controller acting alone would perform well for the gain state at which it was tuned but would perform sluggishly for all other gain states encountered. At low gain states, the PI controller had a rise time of over 600 seconds while the PI/net controller had a rise time of only 100 seconds. This report makes it clear that the addition of a neural network into a PI controller substantially lowers coil response time as well as eliminating the effect of sluggish control experienced when the controller encounters a gain state different than the one it was tuned at.

In addition to the positive results shown here for an HVAC application, clearly the PI/net controller can be implemented on any system where PI control is currently used.

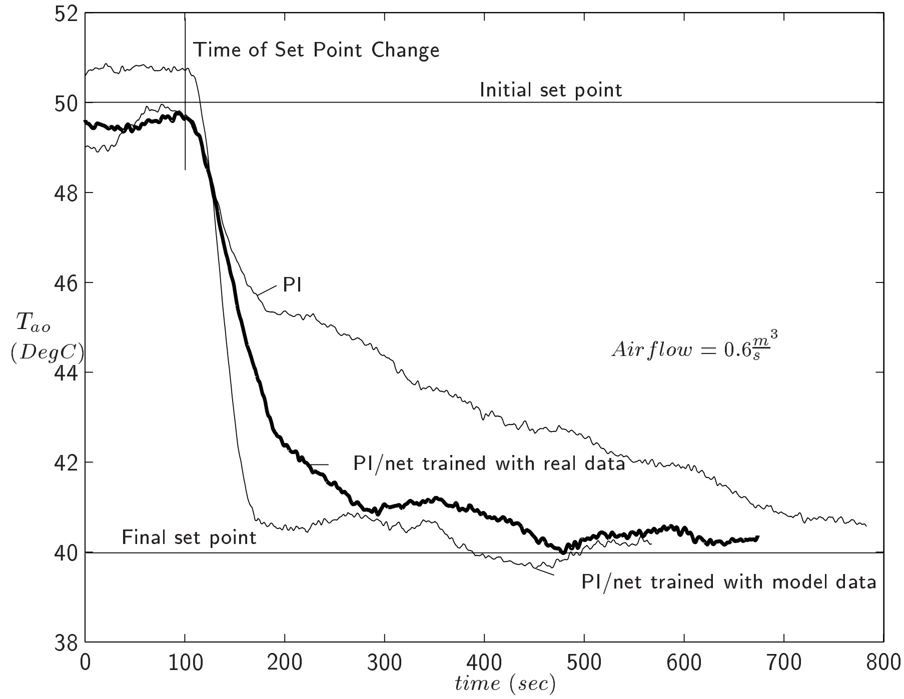

Figure 8: Dynamic Comparison of PI and PI/net Controllers for a Low Gain Decreased Set Point Change

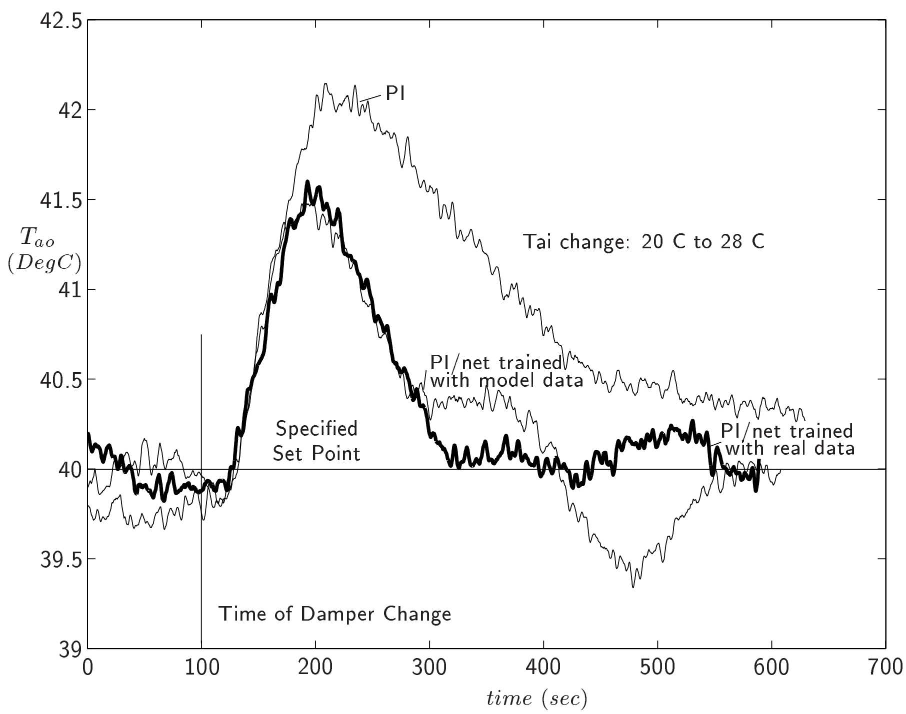

Figure 9: Dynamic Comparison of PI and PI/net Controllers for an Increased Coil Inlet Air Temperature while Retaining Set Point

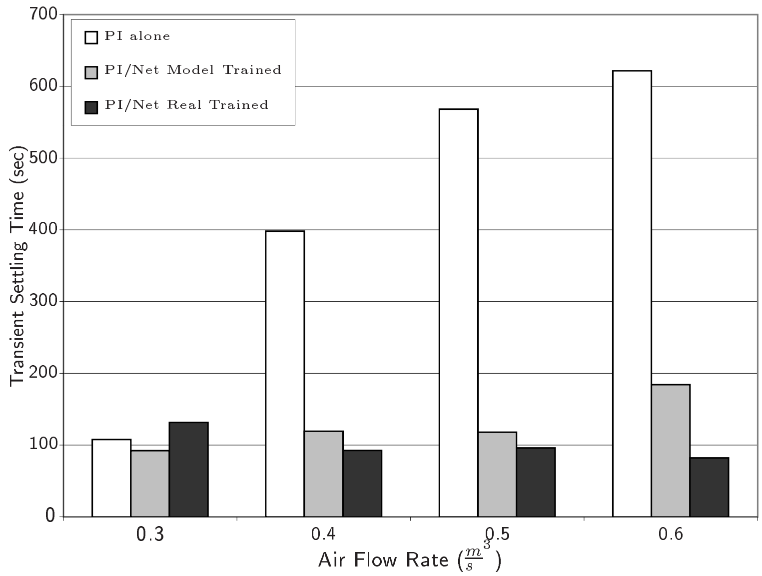

Figure 10: Rise Time Comparison of PI and PI/net Controllers for Varied Gain States with Increased Set Point Changes

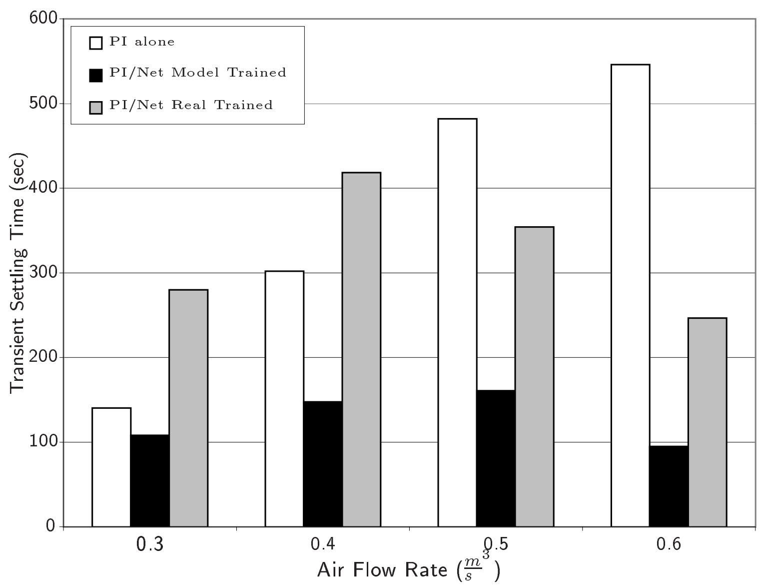

Figure 11: Rise Time Comparison of PI and PI/net Controllers for Varied Gain States with Decreased Set Point Changes

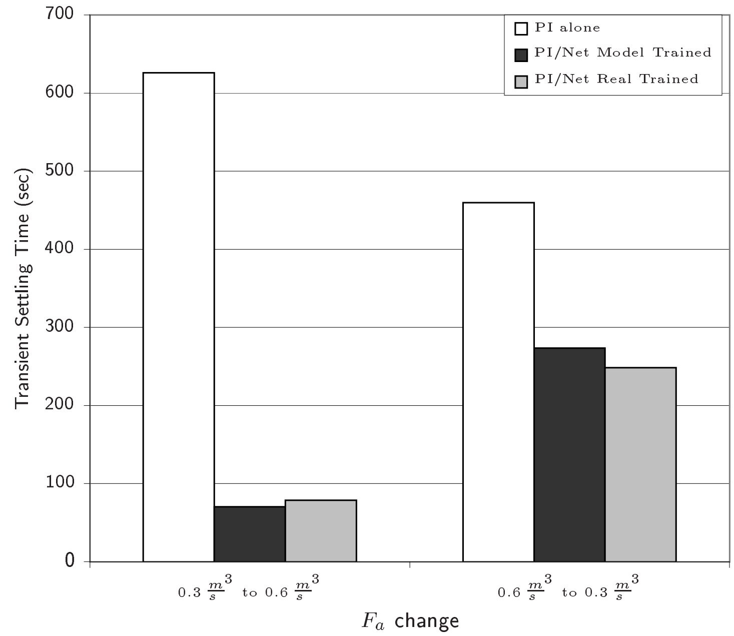

Figure 12: Rise Time Comparison of PI and PI/net Controllers for an Air Flow Rate Change while Retaining Set Point

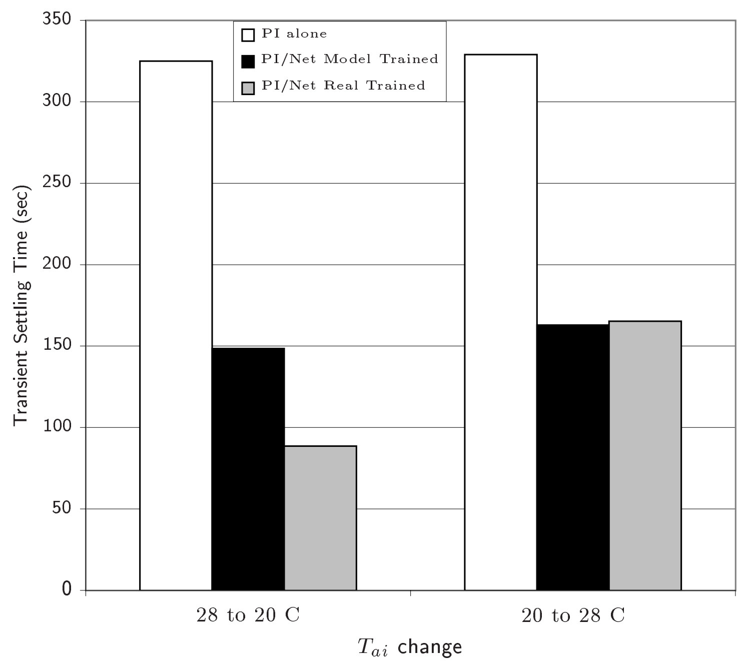

Figure 13: Rise Time Comparison of PI and PI/net Controllers for a Coil Inlet Air Temperature Change while Retaining Set Point

REFERENCES

Charles W. Anderson, Douglas C. Hittle, Alon D. Katz, and R. Matt Kretchmar. 1998. Reinforcement Learning, Neural networks and PI Control Applied to a Heating Coil.

J. Bekker and P. Meckl, D. C. Hittle. 1991. A Tuning Method for First-order Processes with PI Controller. *ASHRAE Transactions*, Vol. 92, Part II.

Darrell D. Massie, Peter S. Curtiss, Jan F. Kreider. 1998. Predicting Central Plant HVAC Equipment Performance Using Neural Networks-Laboratory System Test Results. *ASHRAE Transactions*, V. 104, Pt. 1, pp. 1-8.

Gerardo Diaz, Mihir Sen, K.T. Yang, Rodney L. McClain. 1999. Simulation of Heat Exchanger Performance by Artificial Neural Networks. *International Journal of Heating, Ventilating, Air-Conditioning and Refrigeration Research*, Vol. 5, No. 3, pp. 195-208.

Charles W. Anderson, Douglas C. Hittle, Alon D. Katz, and R. Matt Kretchmar. 1996. Synthesis of Reinforcement Learning, Neural Networks, and PI Control Applied to a Simulated Heating Coil.

P.S. Curtiss, M.J. Brandemuehl, J.F. Kreider. 1994. Energy Management in Central HVAC Plants Using Neural Networks. *ASHRAE Transactions*, V. 100, Pt. 1, pp. 476-493.

Ang Heng Kah, Quek Yeong San, Sim Chee Guan, Wan Chun Kiat, Yong Chaw Koh. 1995. Smart Air-Conditioning System Using Multilayer Perceptron Neural Network With A Modular Approach. *IEEE Proceedings*, pp. 2314-2319.

John E. Seem. 1997. Implementation of a New Pattern Recognition Adaptive Controller Developed Through Optimization. *ASHRAE Transactions*, pp. 1-13.

W. M. Kays and A. L. London. 1984. *Compact Heat Exchangers*. 3rd ed., McGraw-Hill, New York.

F. P. Incropera and D. P. DeWitt. 1996. *Fundamentals of Heat and Mass Transfer*. 4th ed., John Wiley and Sons, New York.

The Math Works Inc. 1999. *MATLAB, The Language of Technical Computing*, Version 5, Natick, Mass.