Application of Functional Link Neural Network to HVAC Thermal Dynamic System Identification

Jason Teeter and Mo-Yuen Chow, *Senior Member, IEEE*

Abstract— Recent efforts to incorporate aspects of artificial intelligence into the design and operation of automatic control systems have focused attention on techniques such as fuzzy logic, artificial neural networks, and expert systems. The use of computers for direct digital control highlights the recent trend toward more effective and efficient heating, ventilating, and air-conditioning (HVAC) control methodologies. Researchers in the HVAC field have stressed the importance of self learning in building control systems and have encouraged further studies in the integration of optimal control and other advanced techniques into the formulation of such systems. Artificial neural networks can also be used to emulate the plant dynamics, in order to estimate future plant outputs and obtain plant input/output sensitivity information for on-line neural control adaptation. This paper describes a functional link neural network approach to performing the HVAC thermal dynamic system identification. Methodologies to reduce inputs of the functional link network to reduce the complexity and speed up the training speed will be presented. Analysis and comparison between the functional link network approach and the conventional network approach for the HVAC thermal modeling will also be presented.

Index Terms—Functional link, HVAC, intelligent control, neural network, system identification.

I. INTRODUCTION

HEATING, ventilating, and air-conditioning (HVAC) systems are a permanent part of everyday life in our industrialized society. A mere 1% improvement in energy efficiency of these systems translates into annual savings of millions of dollars at the national level [1]. The evolution of microprocessor technology has significantly increased the viability of many computationally intensive control algorithms. The use of computers for *direct digital control* (DDC) highlights the recent trend toward more effective and efficient HVAC control methodologies [2], [3]. Several studies related to optimal control of HVAC systems have emerged since the early 1980's [2], [4]–[8].

The application of *intelligent* control to HVAC systems is proposed by Shoureshi and Rahmani in [7]. A fuzzy optimal controller is used for control decisions in which approximate reasoning is required, such as the estimation of human comfort. This work is incorporated into the development of an
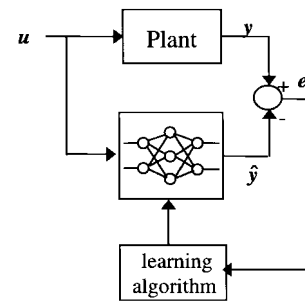
Fig. 1. Forward system identification approach.

intelligent building system (IBS) which uses a supervisor and a coordinator to set control policies for local controllers [9]. The authors stress the importance of self learning in building control systems and encourage further studies in the integration of optimal control and other advanced techniques into the formulation of such systems.

The nonlinear functional mapping properties of *neural networks* are central to their use in identification and control [10]–[14]. Although a number of key theoretical problems remain, results pertaining to the approximation capabilities of neural networks demonstrate that they have great promise in the modeling of nonlinear systems. An important question in system identification is whether a system under study can be adequately represented within a given model structure [10]. In the absence of such concrete theoretical results for neural networks, it is usually assumed that the system under consideration belongs to the class of systems that the chosen network is able to represent. Two system identification techniques are now introduced: *forward modeling* and *inverse modeling*.

The procedure of training a neural network to represent the forward dynamics of a system is often referred to as the *forward system identification* approach [10]. A schematic diagram of this process is shown in Fig. 1.

The neural network is placed in parallel with the system, and the error e between the system outputs y and network outputs $\hat{y}$ is used to train the network. This represents a classical *supervised learning* problem for which the teacher (i.e., the system) provides target values (i.e., system outputs) directly in the output coordinate system of the learner (i.e., the network model) [15].

In an *inverse system identification* approach, a network is trained in an effort to model the inverse of the plant mapping [10]. One of the simplest approaches, known as *direct inverse system identification*, is shown schematically in Fig. 2.
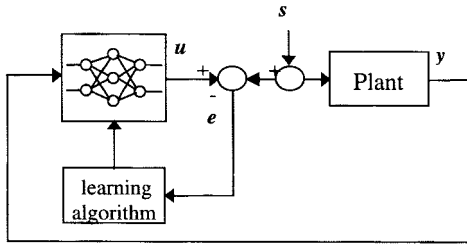
Fig. 2.   Direct inverse system identification approach.

c_p	constant pressure specific heat of air	1005 J/kg °C
f	volumetric airflow rate	m³/s
Q_{he}	heat input in the heat exchanger	W
Q_l	thermal load on the room	W
t	time	S
T_{ref}	desired thermal space temperature	°C
T_i	air temperature at location i	°C
V_{he}	effective heat exchanger volume	25.5 m³
V_{ts}	effective thermal space volume	255 m³
ρ	air density	1.19 kg/m³

A synthetic training signal s is introduced to the system, and the system output y is used as the input to the network. The network output is compared to the training signal, and this error is used to train the network.

The inverse modeling structure shown in Fig. 2 tends to force the network to represent the inverse of the plant, but there are potential drawbacks to this approach. The training signal must be chosen to sample over a wide range of system inputs, and the actual operational inputs may be hard to define *a priori* [15]. This point is strongly related to the concept of persistent excitation discussed in the adaptive control literature. A second drawback is that an incorrect inverse model can be obtained if the nonlinear system mapping is not one to one. An approach called *specialized inverse modeling* has been proposed in an effort to overcome these problems. The details of this approach can be found in [16]. The neural network identification models can be used in the adaptive control of unknown nonlinear plants.

The Exploratory and Applied Research Division of the Electric Power Research Institute (EPRI) has been supporting the authors of this paper to investigate and develop methodologies for intelligent control of an HVAC system using artificial neural network and fuzzy logic technologies. In the project, fuzzy logic is used as a tool for initializing a neural controller [12], [17], [18]. The parameters of this controller are then adjusted on line in an effort to minimize a prespecified cost index. Artificial neural networks are also used to emulate the plant dynamics, in order to estimate future plant outputs and obtain plant input/output sensitivity information. This paper focuses on the design and development of the HVAC thermal dynamic system identification using a *functional link* neural network based on the forward modeling approach. Methodologies to reduce inputs of the functional link network, minimize the complexity, and speed up the training speed will be presented. Analysis and comparison between the functional link network approach and conventional network approach for the HVAC thermal modeling will also be presented.

II. HVAC SYSTEM MODELING AND CONTROL PROBLEM DEFINITION

The single-zone thermal system model shown in Fig. 3 is chosen for our analysis [4]. The system represents a simplification of an overall building climate control problem, but retains the distinguishing characteristics of an HVAC system.

Fresh air enters the system at temperature $T_0(t)$ and volumetric flow rate $f_0(t)$ and is mixed with recirculated air at temperature $T_5(t)$ and flow rate $f_5(t)$. Air with temperature $T_1(t)$ and flow rate $f(t)$ passes through the heat exchanger, where an amount of heat given by $q_{he}(t)$ (positive for heating and negative for cooling) is exchanged with the air. The air and heat exchanger are assumed to have some capacitance, so that the resulting temperature $T_2(t)$ has a transient response. In addition, perfect mixing in the heat exchanger is assumed, so that the air temperature within and exiting the heat exchanger is $T_2(t)$.

After being conditioned in the heat exchanger, the air passes into the thermal space. The capability of applying a space thermal load is included as $q_l(t)$. The temperature $T_3(t)$ of the space has a transient response due to the capacitance of the air and the thermal space. Perfect mixing in the thermal space is assumed, so that the air temperature within and exiting the space is $T_3(t)$. Air leaving the thermal space is drawn through the fan, after which a portion may be recirculated to mix with the fresh air and the remainder may be exhausted from the system.

The conditions within the system are regulated by a controller that provides signals to control the heat input in the heat exchanger, the volumetric airflow rate, and the position of the return air damper. The controller signals are depicted as thin lines and are denoted by S_q, S_f, and S_r. The signal S_T denotes the temperature in the thermal space, while J denotes a cost associated with the performance of the system. Thermal losses between components are neglected and, thus, temperatures $T_4(t)$ and $T_5(t)$ are equal to the temperature of the air exiting the thermal space. In addition, infiltration and exfiltration effects are neglected and, thus, flow rates at locations 2–4 are equal to $f(t)$. The humidity of the air is not considered, and transient effects in the flow splitter, mixer, fan, and heat exchanger are neglected.

The system equations are derived from the conservation of energy principles and are given by

$$\rho c_p V_{he} \frac{dT_2}{dt} = f \rho c_p (T_1 - T_2) + Q_{he} \tag{1}$$

$$\rho c_p V_{ts} \frac{dT_3}{dt} = f \rho c_p (T_2 - T_3) + Q_l \tag{2}$$

where the parameters and variables are as described in Table I.

A lumped capacitance assumption is made, implying that the capacitance of the heat exchanger and the thermal space are accounted for in the effective heat exchanger and thermal space volumes. Statements of the conservation of mass and
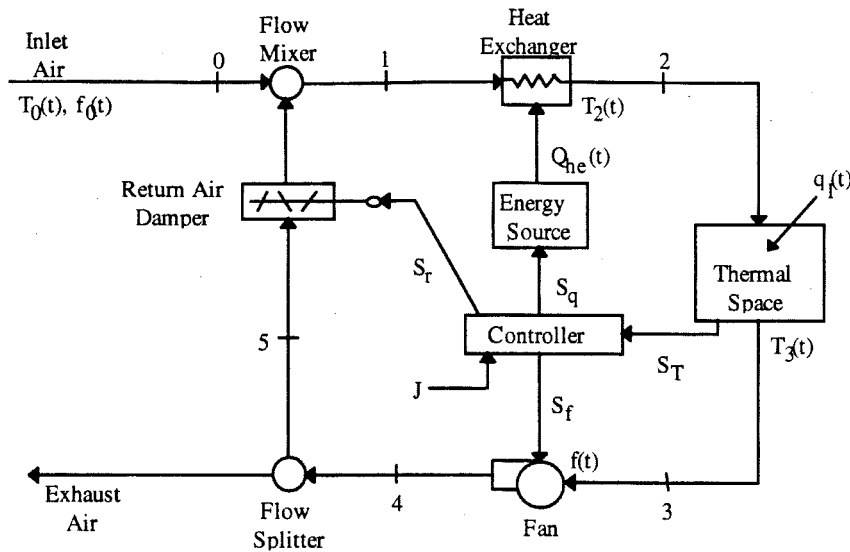
Fig. 3.   HVAC system model.

energy applied at the flow mixer yield

$$T_1 = T_3 + (T_0 - T_3)/r \qquad (3)$$

where

$$r = f/f_0 \qquad (4)$$

is the system-to-fresh-air volumetric flow-rate ratio. For $r = 1$, there is no recirculation and a once-through system is considered. The position of the return air damper as described by r is obtained from the values of f and f_0, which are computed explicitly by the controller.

Denoting $x = [T_2 \quad T_3]^T$ and making the appropriate substitutions in (1) and (2) yields

$$\dot{x}_1 = \frac{1}{V_{he}} \left[(T_0 - x_2)u_3 + (x_2 - x_1)u_2 + \frac{u_1}{\rho c_p} \right] \qquad (5)$$

$$\dot{x}_2 = \frac{1}{V_{ts}} \left[(x_1 - x_2)u_2 + \frac{Q_l}{\rho c_p} \right] \qquad (6)$$

where $u = [Q_{he}, f, f_0]^T$. For the purposes of comfort and hygiene, the minimum allowable value of u_3, the outside air volumetric flow rate, is set at 0.0354 m^3/s [19]. The equilibrium states of the system are

$$x_{1_{EQ}} = x_{2_{EQ}} = \frac{u_3 T_0 + \dfrac{u_1}{\rho c_p}}{u_3}. \qquad (7)$$

Since this is an underdetermined case, an infinite number of combinations of u_1 and u_3 provide the same steady-state output and the system does not possess a unique inverse. The ranges of the control variables are defined as

$$q_{he} \in [-4000, 4000] \text{ W} \qquad (8)$$

and

$$f, f_0 \in [0.0354, 2] \text{ m}^3/\text{s}. \qquad (9)$$

The value of u_3 can never exceed the value of u_2, due to conservation of mass principles. Thus, the controller must

command values for these inputs that satisfy the following condition:

$$u_2 \geq u_3. \qquad (10)$$

The operating ranges of the system states and outside air temperature are assumed to be

$$x_1 \in [5, 35] \,^\circ\text{C} \quad \text{or} \quad [41, 95] \,^\circ\text{F} \qquad (11)$$

$$x_2, T_0 \in [10, 30] \,^\circ\text{C} \quad \text{or} \quad [50, 86] \,^\circ\text{F} \qquad (12)$$

and the range of the reference temperature is assumed be

$$T_{\text{ref}} \in [17, 23] \,^\circ\text{C} \quad \text{or} \quad [62.6, 73.4] \,^\circ\text{F}. \qquad (13)$$

In order to most effectively use u_3, the controller must have information as to whether the outside air temperature is higher or lower than the desired room temperature. Thus, an additional input variable called T_{diff} is introduced, where

$$T_{\text{diff}} = T_{\text{ref}} - T_0. \qquad (14)$$

It is assumed that T_0 can be measured accurately with an inexpensive temperature sensor.

In order to determine the ranges of the changes in the states over a sample period of $T_s = 10$ s, the state equations (5) and (6) are integrated from 0–10 s, using 1000 random initial conditions satisfying (8)–(12). Histograms for Δx_1 and Δx_2 are shown in Fig. 4(a) and (b), respectively.

Based on the extreme values of Δx_1 and Δx_2 shown in Fig. 4(a) and 4(b), the ranges for the changes in the states are assumed to be

$$\Delta x_1 \in [-13.0, 13.0] \,^\circ\text{C} \qquad (15)$$

and

$$\Delta x_2 \in [-1.5, 1.5] \,^\circ\text{C}. \qquad (16)$$

The range of control signals using assumptions (11)–(13), (15), and (16) will be used to train the neural network to learn the HVAC thermal dynamics.
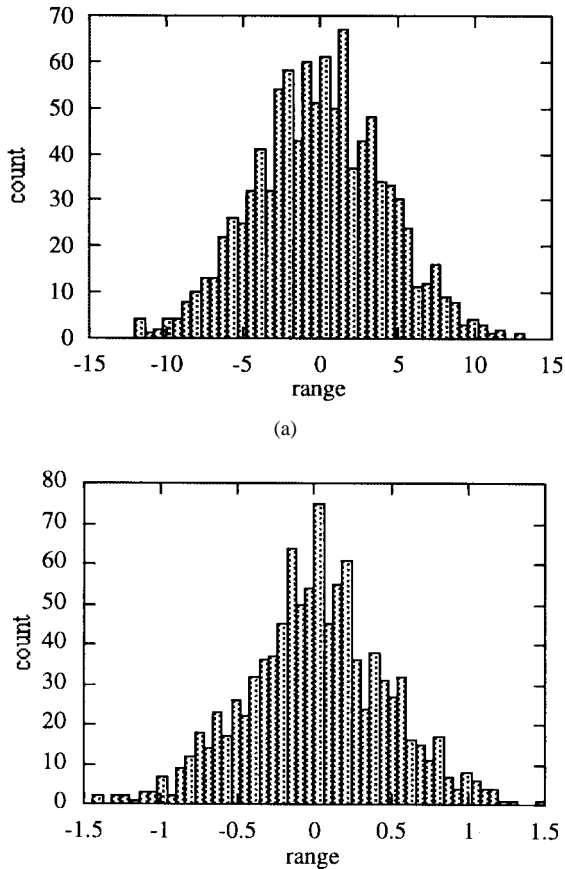
Fig. 4. (a) Histogram of Δx_1. (b) Histogram of Δx_2.

III. NEURAL IDENTIFICATION SCHEME

As mentioned previously, the objective of this paper is to design a neural controller and develop a method for adapting the controller on line, in order to minimize a prespecified cost index. Neural networks are also used for system identification, in order to provide plant sensitivity information that is used to adapt the controller.

A. The Functional Link Concept

Since neural networks are used for adaptive identification and control, the learning capabilities of the networks can have significant effects on the performance of the closed-loop system. If the information content of data input to a network can be modified in an appropriate way, the network will be able to more easily extract the salient features of the data. This is the motivation behind *functional link* mapping [20]. Functional links basically expand the original input space into higher dimensions in an attempt to reduce the burden on the training phase of the neural networks. The functional link acts on an element of an input vector or on all the input vectors by generating a set of linearly independent functions, then evaluating these functions with the pattern as the argument. In one sense, no new *ad hoc* information has been inserted into the process; nonetheless, the representation has definitely been enhanced, and separability becomes possible in the enhanced space. Thus, both the training time and training error of the network can be improved. A common example of such a

mapping is given by

$$H\colon \{x_i\} \to \{x_i, x_i x_j\}, \qquad (17)$$
$$\scriptstyle j \geq i$$

which is sometimes referred to as a *tensor* model [20].

B. Neural Identification Architecture

In order to estimate the future outputs of the thermal system and obtain gradient information for the adaptation scheme, a neural identification model is developed to mimic the dynamics of the system. The forward modeling scheme discussed previously is used. Consider the multi-input, single-output nonlinear system:

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k), \cdots)$$
$$y(k) = g(\boldsymbol{x}(k)). \qquad (18)$$

For many systems, the change in any state over one sample period in the desired operating region is usually small compared to the magnitude of that state. For these systems, the neural identifiers are more effective if they are trained to learn the changes in the states as defined by $\boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k), \cdots) - \boldsymbol{x}(k)$ rather than the scaled values of the states at time index $k+1$. Training a neural identifier to learn Δx_{k+1} is basically the same as training the network to learn the system dynamics without the need to worry about the initial state conditions, because the current state x_k will be added to Δx_{k+1} to provide x_{k+1}. This approach can significantly reduce the network size, because the network does not need to learn all possible state condition, but only the dynamics itself. In addition, training the neural identifier to learn Δx_{k+1} is more accurate than to learn x_{k+1}. For example, the operating range of x_2 is [10,30], as shown in (12), while the operating range of Δx_2 is [−1.5, 1.5] as shown in (16). With the same network structure, the operating range to be covered between these two training schemes is

$$\frac{[10, 30]}{[-1.5, 1.5]} = \frac{30 - 10}{1.5 - (-1.5)} = \frac{20}{3} \approx 7.$$

Even though the training of a neural network depends on many factors, based on the above analysis and assuming that other factors are held constant, we can expect the accuracy of training the neural identifier to provide Δx_{k+1} to be on the order of seven times compared to the neural identifier to provide x_{k+1}. A generic neural identifier depicting this approach is shown in Fig. 5.

The first-order dynamics are emulated for each state by separate networks, so that the delayed effects of the inputs on the outputs can be captured accurately. For the HVAC system, since $y = x_2$, then N_g is not needed. In the absence of a mathematical model, it would be assumed that each state could be measured, since input/output patterns for the states are needed to train the neural identifiers. For the HVAC system, the current system states and inputs are mapped to the inputs of N_1 and N_2 by (17). The input $[u_1, u_2, u_3, x_1, x_2, T_0]$ is expanded according to the tensor model in (17):

$$H\colon \{x_1\} \to \{x_i, x_i x_j\}.$$
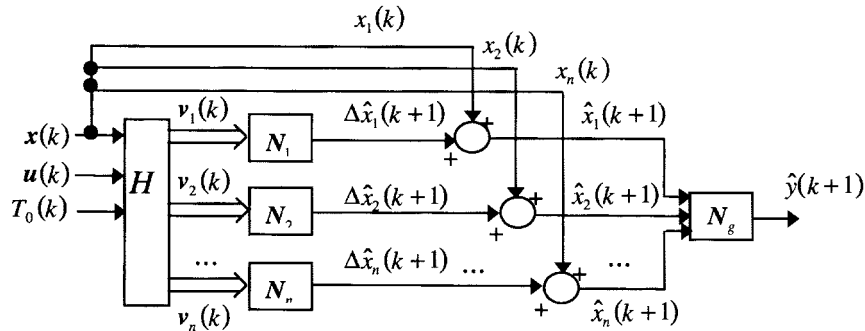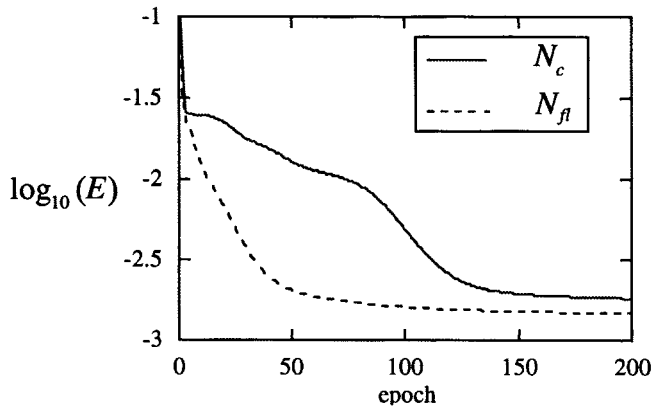$$\scriptstyle j \geq i$$

Fig. 5. A generic neural identifier.

Fig. 6. Typical training error plots for the conventional network N_c and the functional link network N_{fl}.

TABLE II
FINAL WEIGHTS OF N_{fl} CONNECTING ALL INPUTS TO ONE OF THE HIDDEN-LAYER NEURONS

input	weight	input	weight	input	weight	input	weight
1	0.0415	8	0.0136	15	0.2874	22	0.0013
2	0.0017	9	0.0009	16	0.2731	23	0.0011
3	0.0018	10	0.0028	17	0.0321	24	0.0027
4	0.0263	11	0.0044	18	0.0018	25	0.0001
5	0.0082	12	0.0034	19	0.0106	26	0.0037
6	0.0097	13	0.0055	20	0.2832	27	0.0038
7	0.0020	14	0.0010	21	0.2914		

Then, the functional link of the input becomes

$$\boldsymbol{v} = H(u_1, u_2, u_3, x_1, x_2, T_0)$$
$$= [u_1, u_2, u_3, x_1, x_2, T_0, u_1^2, u_1 u_2,$$
$$\cdots, u_1 T_0, u_2^2, u_2 u_3, \cdots, u_2 T_0, \cdots T_0^2]^T.$$

We feed the same input information to all the neural identifiers $N_1, \cdots, N_n$. Thus,

$$\boldsymbol{v}_1 = \boldsymbol{v}_2 - \cdots = \boldsymbol{v}_n = [u_1, u_2, u_3, x_1, x_2, T_0, u_1^2, u_1 u_2,$$
$$\cdots, u_1 T_0, u_2^2, u_2 u_3, \cdots, u_2 T_0, \cdots, T_0^2]^T. \quad (19)$$

The neural identifiers are trained off line to learn the system dynamics. One thousand *training patterns* are generated using random states and inputs in the assumed ranges. The number of training patterns is chosen so that the data set is statistically significant [21]. The change in each state is computed by integrating the state equations over one time step. Typical training error plots for two different networks learning the quantity Δx_1 are shown in Fig. 6. One is the *conventional network* N_c, which has inputs $[u_1, u_2, u_3, x_1, x_2, T_o]^T$, 12 hidden-layer neurons, and one output node. The other is a *functional link network* N_{fl}, which has inputs given by (19), two hidden-layer neurons, and one output node.

Fig. 6 clearly shows that N_{fl} trains faster and exhibits a slightly smaller final error. Further, N_{fl} uses a total of 59 network weights, while N_c uses a total of 97. Using (19) as the input for each identifier results in faster training with fewer network weights. Training speed is important for this

application, since network weights are adjusted on line, in order to adapt to the changes in the system dynamics.

Although the use of (19) decreases the number of weights needed for the identifiers, the increase in the number of inputs results in an increase in the computational complexity of the adaptation method. This is because the sensitivity of plant state x_i with respect to the plant inputs is estimated by

$$\frac{\partial x_i(k+1)}{\partial \boldsymbol{u}(k)} \cong \frac{\partial N_i^k}{\partial \boldsymbol{v}(k)} \frac{\partial \boldsymbol{v}(k)}{\partial \boldsymbol{u}(k)} \quad (20)$$

where N_i^k maps the inputs, states, and outside temperature at time step k to state x_i at time step $k + 1$. The dimensions of the factors on the right-hand side of (17) increase as the dimension of $\boldsymbol{v}$ increases.

There are at least two methods that can be used to justify using a reduced set of inputs. We can use *a priori* knowledge of the system dynamics based on the mathematical model and eliminate inputs that are redundant or unnecessary. For example, input $u_1(k)$ has a negligible effect on $\Delta x_2(k + 1)$. Another method involves training the networks using all inputs, then deciding which inputs to include based on the final network weight values. To illustrate this approach, magnitudes of the final weight values of N_{fl} from the previous example are listed in Table II. For clarity, only the weights that connect input neurons to one of the hidden-layer neurons are listed.

The weights that connect all inputs to the second hidden-layer neuron are similar in magnitude to those shown in Table II. It is clear that inputs 15, 16, 20, and 21 have by far the largest effects on the output. These inputs are, in fact, the terms in (19) that contribute linearly to the system function given by (18). Although this network is trained to learn the *discrete* mapping $\Delta x_1(k + 1) = f_1(x(k), u(k), \cdots) - x1(k)$,
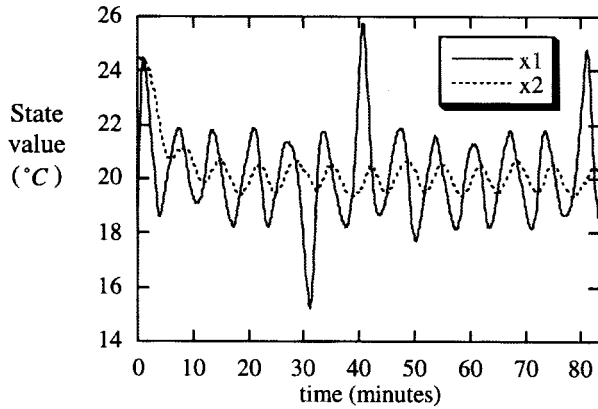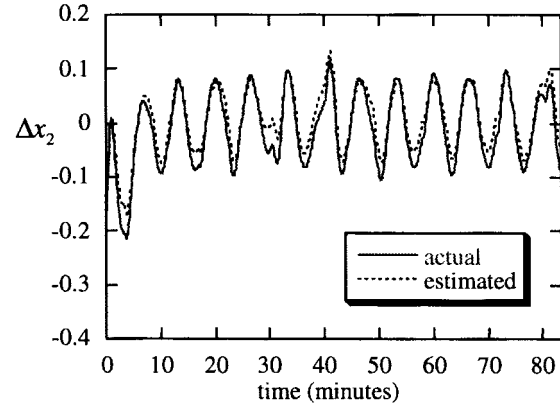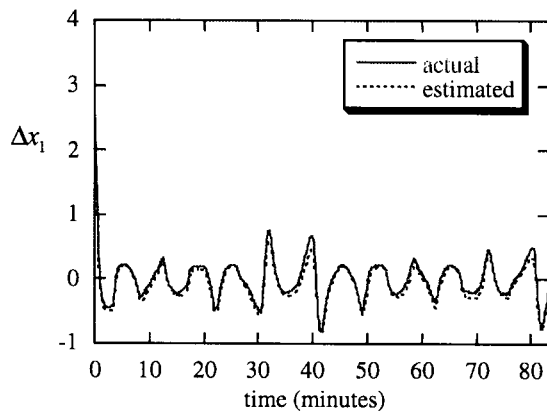
Fig. 7. State trajectories generated for testing.

Fig. 9. Actual and estimated values of Δx_2.

Fig. 8. Actual and estimated values of Δx_1.

the system varies slowly with respect to the sample period and, thus, $\Delta x_1(k+1) \approx T_s \dot{x}_1(kT_s)$.

Estimates of $\Delta x_1(k+1)$ and $\Delta x_2(k+1)$ obtained from the neural identifiers are plotted in Figs. 8 and 9, along with the actual values of the changes in states.

Since all network inputs are normalized (mapped to $[-1, 1]$), magnitudes of weights connecting hidden-layer neurons to the output neuron are on the order of 0.5 and, because only two neurons are used in the hidden layer, the effects of inputs having very small connection weights (i.e., <0.002) can be considered negligible and can be eliminated. This method may be more appealing, since information about which inputs should be used is obtained via network training and no *a priori* information is needed. After eliminating inputs with connection weights less than 0.002 in magnitude, the network can be retrained and the network weights can be evaluated again. In this case, repeating this process results in a five-input network for estimating $\Delta x_1(k+1)$ and a two-input network for estimating $\Delta x_2(k+1)$.

The trained networks are tested by simulating the system for 500 samples using input values computed by

$$u_1(k) = 3000\cos(k\pi/20) \text{ W}$$
$$u_2(k) = 1 - 0.8\cos(k\pi/30) \text{ m}^3/\text{s}$$
$$u_2(k) = \max\{u_3(k), 1 + 0.9\sin(k\pi/25) \text{ m}^3/\text{s}\}. \quad (21)$$

The input signals are chosen so that the resulting state trajectories exhibit rich dynamic behavior. The resulting state trajectories are plotted in Fig. 7.

The plots in Figs. 8 and 9 suggest that the identifiers are able to accurately predict the changes in states and, thus, provide accurate values of the states at the next time step.

IV. CONCLUSION

This paper has presented the application of conventional and reduced-ordered function link neural networks for HVAC thermal dynamic system identification. A single-zone thermal system model was chosen for our analysis. The system represents a simplification of an overall building climate control problem, but retains the distinguishing characteristics of an HVAC system. The use of neural networks for identification and control provides a means of adapting a controller on line in an effort to minimize a given cost index. The identification networks demonstrate the capacity to learn changes in the plant dynamics and to accurately predict future plan behavior.

REFERENCES

[1] R. Shoureshi, "Intelligent control systems: Are they for real?," *Trans. ASME*, vol. 115, pp. 392–401, 1993.
[2] D. B. Cherchas, A. Abdelmessih, and M. Townsend, "A direct digital control algorithm for control of a single environmental space," *Trans. ASME*, vol. 107, pp. 324–331, 1985.
[3] J. I. Levenhagen and D. H. Spethman, *HVAC Controls and Systems*. New York: McGraw-Hill, 1993.
[4] J. M. House, T. F. Smith, and J. S. Arora, "Optimal control of a thermal system," *ASHRAE Trans.*, vol. 97, pp. 991–1001, 1991.
[5] A. Kaya, C. S. Chen, S. Raina, and S. J. Alexander, "Optimum control policies to minimize energy use in HVAC systems," *ASHRAE Trans.*, vol. 88, pp. 235–248, 1982.
[6] J. L. Nizet, J. Lecomte, and F. X. Litt, "Optimal control applied to air conditioning in buildings," *ASHRAE Trans.*, vol. 90, pp. 587–600, 1984.
[7] R. Shoureshi and K. Rahmani, "Derivation and applications of an expert fuzzy optimal control system," *Int. J. Fuzzy Sets Syst.*, 1992.
[8] M. A. Townsend, D. B. Cherchas, and A. Abdelmessih, "Optimal control of a general environmental space," *J. Dynamic Syst., Measur., Contr.*, vol. 108, pp. 330–339, 1986.
[9] R. Shoureshi, P. Torcellini, and K. Rahmani, "Intelligent occupant comfort control system," *Trans. ASME*, vol. 45, pp. 105–110, 1992.
[10] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, "Neural networks for control systems—a survey," *Automatica*, vol. 28, pp. 1083–1112, 1992.
[11] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.

[12] M.-y. Chow and J. Teeter, "A knowledge-based approach for improved neural network control of a servomotor system with nonlinear friction characteristics," *Mechatron.*, vol. 5, pp. 949–962, 1995.

[13] M.-y. Chow and S.-O. Yee, "An adaptive backpropagation through time training algorithm for a neural controller," presented at the 1991 IEEE Int. Symp. Intelligent Control, Arlington, VA, 1991.

[14] J. T. Teeter, M.-y. Chow, and J. James J. Brickley, "A novel fuzzy friction compensation approach to improve the performance of a DC motor control system," *IEEE Trans. Ind. Electron.*, vol. 43, pp. 113–120, Feb. 1996.

[15] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," Center for Cognitive Science, Mass. Inst. Technol., Cambridge, MA, Occasional Paper 40, 1991.

[16] D. Psaltis, A. Sideris, and A. A. Yamamura, "A multilayered neural network controller," *IEEE Contr. Syst. Mag.*, vol. 8, pp. 17–21, Feb. 1988.

[17] M.-Y. Chow, "Fuzzy logic based control," in *Industrial Electronics Handbook*, D. Irwin, Ed. Boca Raton, FL: CRC Press, 1996.

[18] M.-Y. Chow and A. Menozzi, "Design methodology of an intelligent controller using artificial neural networks," presented at IEEE IECON'93, Maui, HI, 1993.

[19] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Contr., Signals, Syst.*, vol. 2, pp. 303–314, 1989.

[20] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks.* Reading, MA: Addison-Wesley, 1989.

[21] D. C. Montgomery, *Design and Analysis of Experiments.* New York: Wiley, 1991.

Mo-Yuen Chow (S'81–M'82–SM'93) received the B.S. degree from the University of Wisconsin, Madison, and the M.Eng. and Ph.D. degrees from Cornell University, Ithaca, NY, in 1982, 1983, and 1987, respectively, all in electrical engineering.

Following receipt of the Ph.D. degree, he joined the faculty of North Carolina State University, Raleigh, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. Since 1987, he has been a Pricipal Investigator in several projects in the areas of system monitoring, fault detection and control, applications of artificial neural networks, and fuzzy logic. He has published one book, several book chapters, and more than 70 journal and conference articles related to his research. He has also been a Consultant to Otis Elevator Company, Taiwan Power Company, and J. W. Harley Company and a Faculty Intern with Duke Power Company.

Dr. Chow is currently an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and served as a Guest Editor for the Special Issue on Application of Intelligent Systems to Industrial Electronics. He is currently the Chairman of the IEEE NNC Regional Interest Group Committee and President of the Triangle Area Neural Network Society. He is also listed in *Who's Who Among Asian Americans*.

Jason Teeter received the B.S.E.E. and M.S.E.E. degrees from North Carolina State University, Raleigh, in 1993 and 1996, respectively.

He currently is a Hardware Design Engineer with General Electric Fanuc Company, Charlottesville, VA, where he designs embedded firmware and hardware for use in industrial control products. His interests include digital and analog circuit design and the application of artificial intelligence to automatic control systems.