Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 11 (15): 2754-2763, 2011
ISSN 1812-5654 / DOI: 10.3923/jas.2011.2754.2763

Optimal Design of Neural Fuzzy Inference Network for Temperature Controller

[1]Nisha Jha, [2]Udaibir Singh, [3]T.K. Saxena and [1]Avinashi Kapoor
[1]Department of Electronics, University of Delhi South Campus, New Delhi 110 021, India
[2]Department of Electronics, Acharya Narendra Dev College, University of Delhi, Govindpuri, Kalkaji, New Delhi 110019, India
[3]National Physical Laboratory, Dr. K.S. Krishnan Road, New Delhi-110 012, India

Abstract: In this study, a Neural Fuzzy Inference Network (NFIN) for controlling the temperature of the system has been proposed. The NFIN is inherently a modified fuzzy rule based model possessing neural network's learning ability using hybrid learning algorithm which combines gradient descent and least mean square algorithm. In contrast to the general adaptive neural fuzzy networks where the rules should be decided in advance before parameter learning is performed, there are no rules initially in the NFIN. The rules in the NFIN are created and adapted as on-line learning proceeds via simultaneous structure and parameter identification. The NFIN has been applied to a practical water bath temperature control system, designed and developed around Atmel's 89C51 microcontroller. In the above system, four experiments were conducted on water bath each for 250 and 500 mL min^{-1} flow of water for different volume of water and power of heater. The performance of NFIN has been compared with Fuzzy Logic Controller (FLC) and conventional Proportional Integral Derivative (PID) controller. The three control schemes are compared through experimental studies with respect to set point regulation. It is found that the proposed NFIN control scheme has the best control performance of the three control schemes.

Key words: PID, adaptive control, neural fuzzy inference network, temperature control, fuzzy logic control

INTRODUCTION

The Proportional-Integral-Derivative (PID) controller (Yazdizadeh *et al*., 2009) has been commonly used in process industries, since it has many advantages such as simple designing technique, easy application and parameter design methods and so on. It is well known that appropriate values of PID parameter are the most important aspect which influences the PID controller performance and is hard to get especially for large time-delay or time-variation uncertain system. Some kinds of self-tuning PID controller have been presented to solve these problems (Wu *et al*., 2005; Wen and Liu, 2004; Huapeng and Handroos, 2004; Astrom *et al*., 1993; Astrom and Hagglund, 1995; Chu and Teng, 1999; Ho and Xu, 1998). In this study, we use the neurofuzzy based tuning formula of PID controller for developing Neural Fuzzy Inference Network (NFIN) system.

The concepts of fuzzy logic and artificial neural network for control problem have been developed into a popular research topic in recent years (Hsu, 2007; Fakhrazari and Boroushaki, 2008; Lin and Xu, 2006). The reason is that the classical control theory usually requires a mathematical model. The inaccuracy of mathematical modeling of the plants usually degrades the performance of the controller, especially for nonlinear and complex control problems (Astrom and Wittenmark, 1989). On the contrary, the advent of the Fuzzy Logic Controllers (FLC's) (Draincov *et al*., 1996; Harris *et al*., 1993; Sugeno, 1985; Tareghian and Kashefipour, 2007) and the neural network controllers (Miller *et al*., 1990; Yabuta and Yamada, 1991) based on multilayered Back Propagation Neural Networks (BPNN's) has inspired new resources for the possible realization of better and more efficient control (Kosko, 1992; Lin *et al*., 1996; Hourfar and Salahshoor, 2009) over traditional adaptive control systems (Narendra *et al*., 1991). That is, they do not require mathematical models of the plants. The traditional neural networks can learn from data and feedback but the meaning associated with each neuron and each weight in the network is not easily understood. For a BPNN, its nonlinear mapping and self-learning abilities have been the motivating factors for its use in developing intelligent control systems (Yabuta and Yamada, 1991). However, slow convergence is the major disadvantage of the BPNN. Alternatively, the fuzzy logic systems are easy to

Corresponding Author: Avinashi Kapoor, Department of Electronics, University of Delhi South Campus, Benito Juarez road, Dhaula Kuan, New Delhi-110021, India Tel: +91-9350571397 Fax: 91-011-24110440

appreciate because it uses linguistic terms and the structure of if-then rules (Miller *et al.*, 1990). The simplicity of designing these fuzzy logic systems has been the main advantage of their successful implementation in a lot of industrial process (Islam *et al.*, 2007). There has a lot of fuzzy PI, fuzzy PD and fuzzy PID control schemes were proposed in literature (Visioli, 2001; Haiguo and Zhixin, 2007; Juang and Lin, 1998). Despite the advantages of the conventional FLC over traditional approaches, there remain a number of drawbacks in the design stages. Even though rules can be developed for many control applications, they need to be set up through expert observation of the process. The complexity in developing these rules increases with the complexity of the process. FLC's also consist of a number of parameters that are needed to be selected and configured in prior, such as selection of scaling factors, configuration of the center and width of the membership functions and selection of the appropriate fuzzy control rules. In contrast to the pure neural network or fuzzy system, the neural fuzzy network (Nurnberger *et al.*, 1999; Azeem *et al.*, 2003; Chopra *et al.*, 2005; Kasabov, 1996; Caswara and Unbehauen, 2002; Munasinghe *et al.*, 2005; Ouyang *et al.*, 2005; Arbaoui *et al.*, 2006) representations have emerged as a powerful approach to the solution of many problems (Lin *et al.*, 2001).

In this study, a Neural Fuzzy Inference Network (NFIN) is proposed to combine the advantages of fuzzy logic and neural networks. The NFIN is a fuzzy rule-based network possessing neural network's learning ability. A major characteristic of the network is that no pre-assignment and design of the rules are required. The rules are constructed automatically during the on-line operation. Two learning phases, the structure identification as well as the parameter learning phases (Lin and Lin, 1996), are adopted on-line for the construction task. The structure identification determine the proper number of rules needed i.e., finding how many rules are necessary and sufficient to properly model the available data and the number of membership functions for input and output variables. Parameter learning phase is used to tune the coefficients of each rule (like the shape and positions of membership functions). In this study, a Neural Fuzzy Inference Network (NFIN) is proposed to overcome the disadvantages of the BPNN and FLC.

Temperature control is an important factor in many process control systems (Khalid *et al.*, 1993; Khalid and Omatu, 1992; Tsai *et al.*, 2008). If the temperature is too high or too low, the final product is seriously affected. Therefore, it is necessary to reach some desired temperature points quickly and avoid large overshoot. Since the process-control systems are often nonlinear and tend to change in an unpredictable way, they are not easy to control accurately.

In the present study we conducted four sets of experiments, each for 250 and 500 mL min^{-1} continuous flow of water with different volume of water and power of heater. In these experiments, the tracking performance of the three controller's i.e., NFIN controller, FLC controller and conventional PID controller in respect of controlling a set point temperature of 50°C are studied under the same training process via a simulation of above water bath temperature control systems. This study shows that the NFIN has good control performance of the three temperature-control system and is able to cope with the disadvantages of the BPNN.

NEURAL FUZZY INFERENCE NETWORK (NFIN)

NFIN learning: The learning scheme is mainly composed of two steps. In the first step, the number of rules nodes (hence the structure of the network) and initial rule parameters (weights) are determined using structure identification; in the latter all parameters are adjusted using parameter identification as shown in Fig. 1.

To start the structure tuning, a training set composed of input-output data which contains n inputs and one output must be provided. The data points have been assumed to be normalized in each dimension and they consider as a possible cluster center which define a measure of the potential of data point (Chiu, 1994). To extract the set of initial fuzzy rules, firstly data is separated into groups according to their respective classes. Subtractive clustering is then applied to the input space of each group of data individually for identifying each class of data. Each cluster center may be translated into a fuzzy rule for identifying the class.

A fuzzy rule of the following form is adopted in our system:

- Rule 1: If X_1 is A_{i1} and X_2 is A_{i2} and... then class is c_1

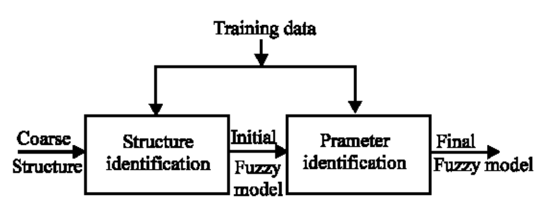

Fig. 1: Steps of learning scheme for NFIN

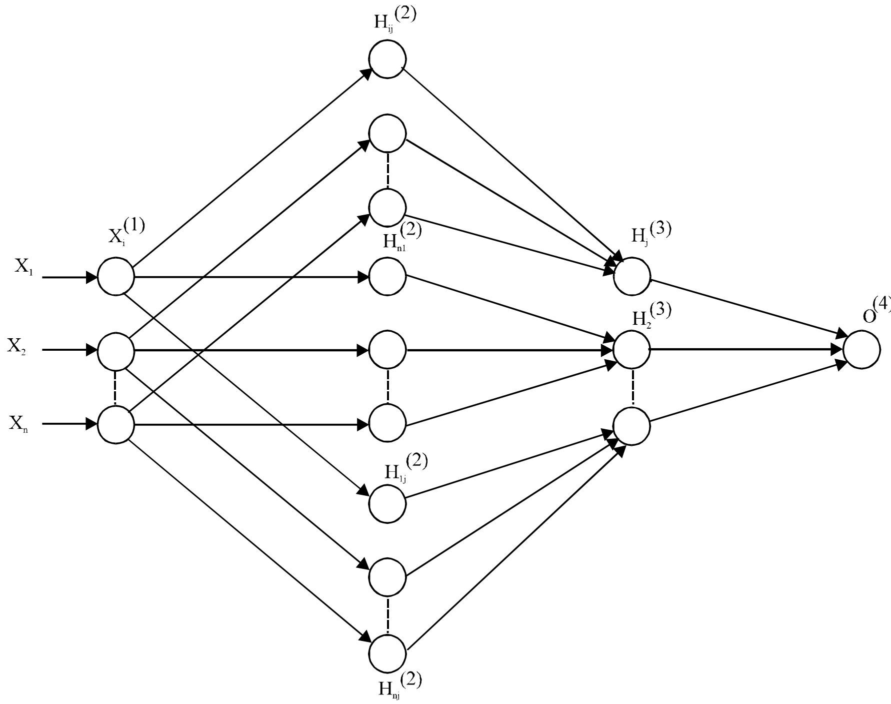

Fig. 2: Structure of Neuro-fuzzy control

where, X_i is the i'th input variable and A is the membership function (Gaussian type).

For each rule, the first antecedent corresponds to the first input, the second antecedent corresponds to the second input etc. and for output we use centroid defuzzification method.

The parameters of the initial fuzzy rules are tuned by using neural network techniques through parameter identification. A neural network with four layers is designed based on the fuzzy rules obtained in first phase. To realize the described fuzzy inference mechanism, the operation of a neural network is shown in Fig. 2 and structure of NFIN described below.

Structure of the NFIN: The NFIN consists of nodes, each of which has some finite fan-in of connections represented by weight values from other nodes and fan-out of connections to other nodes as shown in Fig. 3. Associated with the fan-in of a unit is an integration function f which serves to combine information, activation or evidence from other nodes. This function provides the net input for this node:

$$\text{net}-\text{input}=f\left(u_1^{(k)},u_2^{(k)},.....u_p^{(k)};w_1^{(k)},w_2^{(k)},.....w_p^{(k)}\right) \quad (1)$$

where, $u_1^{(k)}$, $u_2^{(k)}$, ... $u_p^{(k)}$ are inputs to this node and $w_1^{(k)}$, $w_2^{(k)}$, ... $w_p^{(k)}$ are the associated link weights. The superscript (k) indicates the layer number. A second action of each node is to produce an activation value as a function of its net-input:

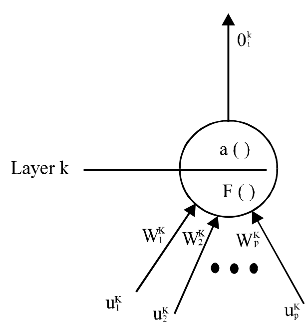

Fig. 3: Basic structure of a node in a neural network

$$\text{Output}=O_1^{(k)}=a\left(f\right) \quad (2)$$

where, a (.) denotes the activation function. In a standard form:

$$f=\sum_{i=1}^{p}w_i^{(k)}u_i^{(k)} \text{ and } \alpha=\frac{1}{1+e^{-f}} \quad (3)$$

Now the functions of the nodes in each of the layers are described below (Farivar *et al*., 2009).

Layer 1: No computation is done in this layer. Each node in this layer which corresponds to only one input variable, transmits input values to the next layer directly. That is:

$$x_i^{(1)} = u_i w_i,\ x_i^{(1)} = u_i \tag{4}$$

In the above equation, the link weight (w_i) in layer one is unity.

Layer 2: Units in this layer receives the input value (X_1, X_2... X_n) and acts as the fuzzy sets representing the corresponding input variable. Nodes in this layer are arranged into j groups; each group representing the IF-part of a fuzzy rule. Node (i, j) of this layer produces its output $H^{(2)}_{ij}$, by computing the corresponding Gaussian membership function:

$$H_{ij}^{(2)} = \exp\left[-\left\{\frac{u_i^{(2)} - m_{ij}}{\sigma_{ij}}\right\}^2\right] \tag{5}$$

where, m_{ij} is center (or mean) and σ_{ij} is width (or variance) of the membership function.

Layer 3: The number of nodes in this layer is equal to the number of fuzzy rules. A node in this layer represents a fuzzy rule; for each node, there are n fixed links from the input term nodes representing the IF-part of the fuzzy rule. Node $H_{ij}^{(3)}$ of this performs the AND operation by product of all its inputs from layer 2. For instance:

$$H_j^{(3)} = \prod_{i=1}^{n} H_{ij}^{(2)} \tag{6}$$

Layer 4: This layer contains only one node whose output $O^{(4)}$ represents the result of centroid defuzzification, i.e.,:

$$O^{(4)} = \frac{\sum_{j=1}^{J} H_j^{(3)} c_j}{\sum_{j=1}^{J} H_j^{(3)}} = \frac{\sum_{j=1}^{J} \alpha_j^{(4)}}{\sum_{j=1}^{J} H_j^{(3)}} \tag{7}$$

where

$$\alpha_j^{(4)} = H_j^{(3)} c_j$$

Where, c_j is the class of data as discussed above and it is also called the fuzzy singletons defined on output variables. Apparently, m_{ij}, σ_{ij} and c_j are the parameters that can be tuned to improve the performance of the system. The above parameters have been tuned by using parameter learning. After that a hybrid learning algorithm which combines the gradient descent method and the Least Square Estimator (LSE) method is used to refine these parameters. Since gradient descent method is generally slow and likely to become trapped in local minima when it can be apply to identify the parameters in an adaptive network.

The following parameter learning is performed on the whole network after structure learning. The idea of backpropagation (Rumelhart and McClelland, 1986) is used for this supervised learning. The goal is to minimize the error function:

$$E = \frac{1}{2}(y(t) - y^d(t))^2 \tag{8}$$

where, $y^d(t)$ is the desired output and y (t) is the current output. For each training data set, starting at the input nodes, a forward pass is used to compute the activity levels of all the nodes in the network. Then starting at the output nodes, a backward pass is used to compute $\partial E/\partial y$ for all the hidden nodes. Assuming that adjustable parameter w is m_{ij} and σ_{ij} in a node, the general learning rule used is:

$$\Delta w \,\alpha - \frac{\partial E}{\partial w} \tag{9}$$

$$w(t+1) = w(t) + \eta\left(-\frac{\partial E}{\partial w}\right) \tag{10}$$

where, η is the learning rate and:

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial(\text{activation})}\frac{\partial(\text{activation})}{\partial w} = \frac{\partial E}{\partial \alpha}\frac{\partial \alpha}{\partial w} \tag{11}$$

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial \alpha}\frac{\partial \alpha}{\partial f}\frac{\partial f}{\partial w} \tag{12}$$

To show the learning rule, we shall show the computations of $\partial E/\partial w$, layer by layer, starting at the output nodes and we will use the membership functions with centers m_i's and widths σ_i's as the adjustable parameters for these computations. These adjustable parameters are updated by the backpropagation algorithm. Using the chain rule, we have:

$$\frac{\partial E}{\partial m_{ij}^{(2)}} = \frac{\partial E}{\partial y}\sum_k \frac{\partial y}{\partial H_j^{(3)}}\frac{\partial H_j^{(3)}}{\partial m_{ij}^{(2)}} \tag{13}$$

where,

$$\frac{\partial E}{\partial y} = y(t) - y^d(t) \tag{14}$$

$$\frac{\partial y}{\partial H_j^{(3)}}=\frac{\alpha_j^{(4)}-y}{\sum_i H_i^{(3)}} \tag{15}$$

$$\frac{\partial H_j^{(3)}}{\partial m_{ij}^{(2)}}=\begin{cases} H_j^{(3)}\dfrac{2(u_i-m_{ij})}{\sigma_{ij}^2} & \text{if term node j is connected to rule node k,} \\ 0 & \text{otherwise, 0} \end{cases} \tag{16}$$

And $m_{ij}^{(2)}$ is updated by:

$$m_{ij}^{(2)}(t+1)=m_{ij}^{(2)}(t)-\eta\frac{\partial E}{\partial m_{ij}^{(2)}} \tag{17}$$

$$=m_{ij}^{(2)}(t)-\eta\left(y(t)-y^{\alpha}(t)\right)\sum_k\frac{\partial y}{\partial H_j^{(3)}}\frac{\partial H_j^{(3)}}{\partial m_{ij}^{(2)}} \tag{18}$$

Similarly, we have:

$$\frac{\partial E}{\partial \sigma_{ij}^{(2)}}=\frac{\partial E}{\partial y}\sum_k\frac{\partial y}{\partial H_j^{(3)}}\frac{\partial H_j^{(3)}}{\partial \sigma_{ij}^{(2)}} \tag{19}$$

Where:

$$\frac{\partial H_j^{(2)}}{\partial \sigma_{ij}^{(2)}}=\begin{cases} H_j^{(3)}\dfrac{2\left(u_i-m_{ij}\right)^2}{\sigma_{ij}^3} & \text{if term node j is connected to rule node k,} \\ 0 & \text{otherwise, 0} \end{cases} \tag{20}$$

And $\sigma^{(2)}_{ij}$ is updated by:

$$\sigma_{ij}^{(2)}(t+1)=\sigma_{ij}^{(2)}(t)-\eta\frac{\partial E}{\partial \sigma_{ij}^{(2)}} \tag{21}$$

EXPERIMENTAL AND SIMULATION STUDIES

Problem statement: The continuous-time temperature control of a water bath system (Tanomaru and Omatu, 1992) is described as:

$$\frac{dy(t)}{dt}=\frac{u(t)}{C}+\frac{y_o-y(t)}{RC} \tag{22}$$

where, y (t) is system output temperature in °C, u (t) is heating flowing inward the system, Y_o is room temperature, R and C are the equivalent thermal resistance and capacity between the system borders and surroundings respectively. We assume both quantities to be constant; now rewrite the above Eq. 22 into discrete time form as:

$$y(k+1)=A(T_s)y(k)+\frac{B(T_s)}{1-e^{\frac{1}{2}y(k)-y}}u(k)+\left(1-A(T_s)\right)y_o \tag{23}$$

where,

$$A(T_s)=e^{-\alpha T_s}$$

$$B(T_s)=\frac{\beta}{\alpha}(1-e^{-\alpha T_s})$$

Equation 23 models a real water bath temperature-control system, where α and β are some constant values describing R and C. The system parameters used in this example are α = 1.00151e^{-4}; β = 8.67973e^{-3}, γ = 40.0 and y_o =25.0°C which were obtained from a real water bath. The plant input u (k) is limited between 0 volt and 5volt. The sampling period is T_s = 30 sec. The system configuration is shown in Fig. 4, where y^d is the desired temperature of the controlled plant.

Experimental setup: To see whether the proposed NFIN can achieve good performance and overcome the disadvantages of the BPNN, we compare it with the BPNN under the same aforementioned training procedure on a simulated water bath temperature-control system. The schematic diagram of the experimental setup of the water bath temperature controller is shown in Fig. 5. The hardware for controlling the temperature of the bath has been designed and fabricated around the Atmel microcontroller 89C51. The temperature of the bath is acquired with the help of PRT. When the PRT is excited with a constant current source of 1mA current, it gives the output in voltage form. The voltage is so amplified that the value of the amplified voltage is equal to the temperature. This voltage is then fed to the 4½ digit ADC. This digitized voltage is then sent to the PC by microcontroller through RS232C interface. The program in PC does the calculations using the NFIN algorithm. After doing all the calculations it generates the firing angle to control the energy in the water bath and sends the same

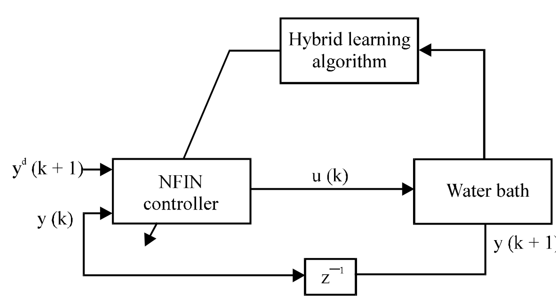

Fig. 4: Block diagram of NFIN controller

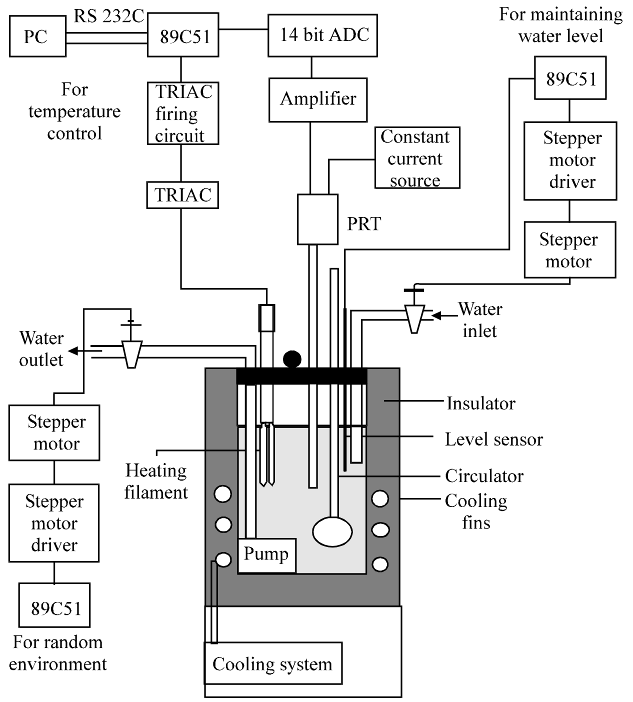

Fig. 5: Schematic diagram of the experimental setup

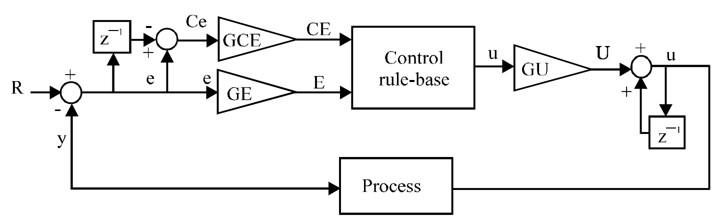

Fig. 6: Block diagram of FLC

to the microcontroller. Thereafter, microcontroller triggers the triac accordingly. The NFIN program in PC continuously monitors the temperature and accordingly controls the same in the bath. In case it senses any change in the temperature, it automatically modifies the parameters of the temperature controller. The NFIN program in PC has been written in Visual BASIC-5.0 language. The program stores the data in the user defined file as well as plots the online data in the form of graph on the screen. A specially designed varying environment is created by continuous flow of fresh water in such a way that the level of the water inside the bath remains constant even if the hot water is removed at random outflow rates. Uniform heat distribution is maintained using the circulator and the isolated system is used to minimize external disturbance. The cooling is achieved at a constant rate using the refrigeration system of the bath.

Experimental results: In this study, we compare the NFIN controller to the FLC and PID controller. Each of the three controllers is applied to the water bath temperature control system. The comparison performance measures include set-points regulation and parameter variation as change in volume of water and change in power of heater in the system.

The discrete form of PID controller can be described by well known expression (Lin *et al.*, 2006; Anderson, 1987). In this control system K_p and K_i are set as 2.5 and 100, respectively and K_d is kept at constant value of 10.

For the Fuzzy Logic Controller (FLC) as shown in Fig. 6, the input variables are chosen as e(t) and ce(t), where e (t) is the performance error indicating the error between the desired water temperature and the actual measured temperature and ce(t) is the rate of change in

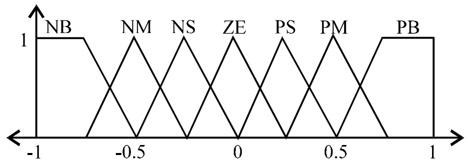

Fig. 7: Membership functions for e (t), ce (t) and u (t)

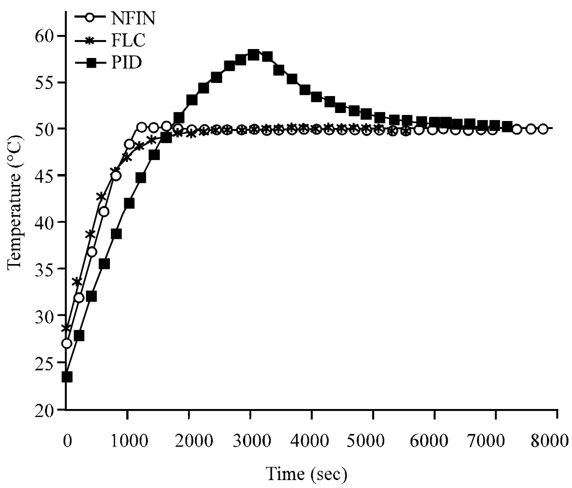

Fig. 8: Temperature response of a water bath having 5 L volume at 0.5 KW for 250 mL min^{-1} flow using NFIN, FLC and PID controller

the performance error e(t). The output or the controlled linguistic variable is the voltage signal u (k) to the heater. Seven fuzzy terms are defined for each linguistic variable. These fuzzy terms consist of Negative Large (NL), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM) and Positive Large (PL). Each fuzzy term is specified by a Gaussian membership function as shown in Fig. 7. According to common sense and engineering judgment, 49 fuzzy rules are specified in Table 1. Like other controllers, a fuzzy controller has some scaling parameters to be specified. They are GE, GCE and GU, corresponding to the process error, the change in error and the controller's output, respectively.

In the water bath, four sets of experiments were conducted, each for 250 and 500 mL min^{-1} continuous flow of water. In these experiments, the tracking performance of the three controllers i.e., NFIN controller, FLC controller and conventional PID controller in respect of controlling a setpoint temperature of 50°C are studied. The four systems of these two flows of water are categorized in terms of volume of water and power of heater as shown in Table 2. These are: (1) 5 L with

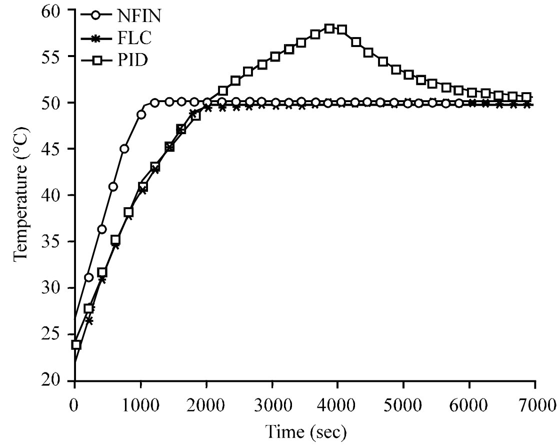

Fig. 9: Temperature response of a water bath having 10 L volume at 1.0 KW for 250 mL min^{-1} flow using NFIN, FLC and PID controller

Table 1: Fuzzy rule base

Δe/e	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NS	NS	ZE
NM	NB	NM	NM	NM	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PM	PB
PB	ZE	PS	PS	PM	PB	PB	PB

Table 2: Different values of system parameters

Parameters	Values
Kp	2.5
Ki	100
Kd	10
Power of heater	0.5,1.0 and 1.5 KW
Volume of water	5, 10 and 15L
Voltage	5 volts
Set point temperature	50°C
Rate of flow of water	250, 500 mL

0.5 KW, (2) 10 L with 1.0 KW, (3) 10 L with 1.5 KW and (4) 15 L with 1.5 KW. In this way overall eight experiments were conducted in the water bath.

The simulation results for 250 mL min^{-1} continuous flow rate of water for 5 L with 0.5 KW, 10 L with 1.0 KW, 10 L with 1.5 KW and 15 L with 1.5 KW are shown in Fig. 8-11, respectively. In these graphs the temperature response of three controllers are shown simultaneously for comparison. It is clear from these figures that there is always large overshoot and settling time for conventional PID controller and also for all the systems. The temperature control performance of FLC controller is also not satisfactory as it takes large settling time. These problems with both the controllers happen because on implementing the FLC, the numbers of rules and membership functions have to be decided and tuned by hand and PID controller needs proper tuning of K_p, K_i and

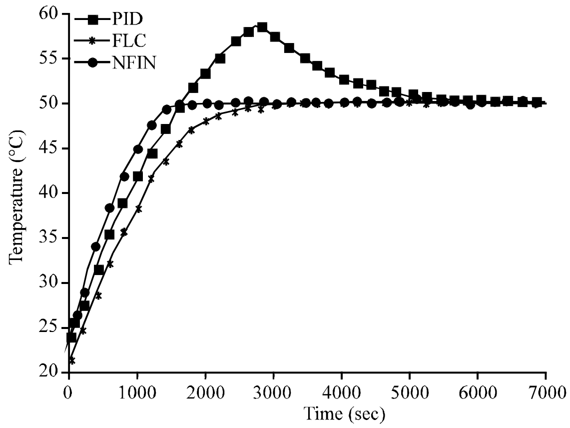

Fig. 10: Temperature response of a water bath having 10 L volume at 1.5 KW for 250 mL min^{-1} flow using NFIN, FLC and PID controller

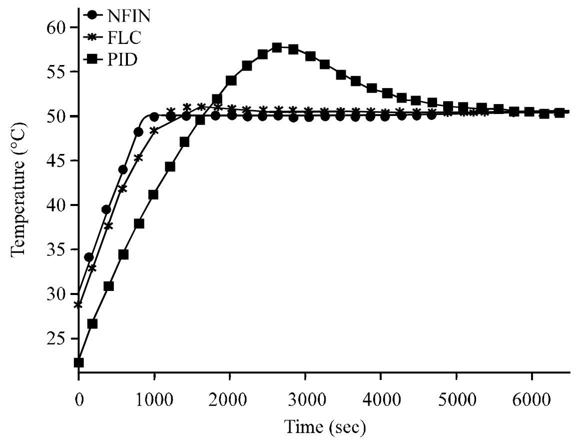

Fig. 12: Temperature response of a water bath having 5 L volume at 0.5 KW for 500 mL min^{-1} flow using NFIN, FLC and PID controller

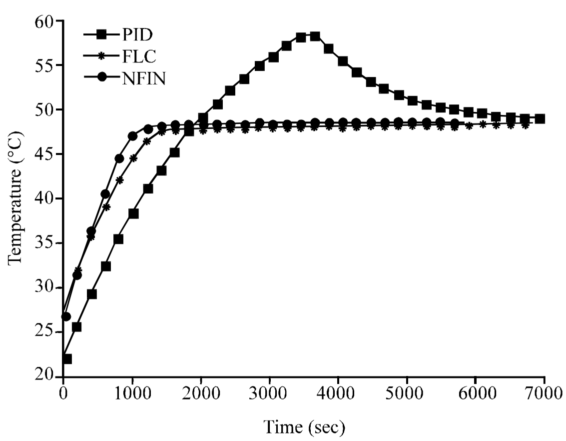

Fig. 11: Temperature response of a water bath having 15 L volume at 1.5 KW for 250 mL min^{-1} flow using NFIN, FLC and PID controller

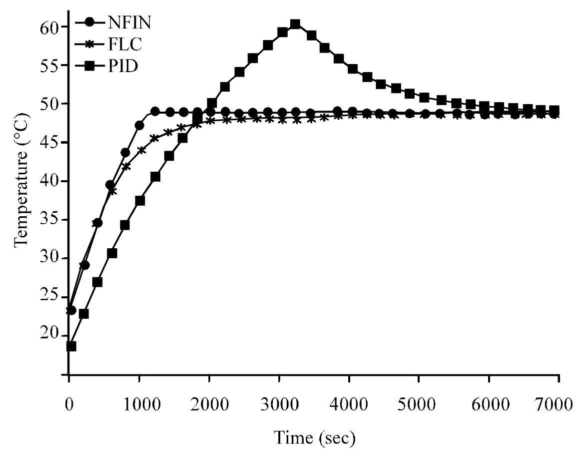

Fig. 13: Temperature response of a water bath having 10 L volume at 1.0 KW for 500 mL min^{-1} flow using NFIN, FLC and PID controller

K_d parameters. Altogether we say that both controllers require a long time in design for achieving good performance. On the other hand NFIN controller takes much less settling time and overshoot as compare to FLC and PID controller, to achieve desired temperature of 50°C. This occurs because on implementing the NFIN controller, no controller parameters have to be decided in advance. We only need to choose proper training patterns and the input vector of the NFIN controller. When we compare the results of Fig. 9 with 10 having same volume of water as 10 L but different power of heater than it is observed that 10 L with 1.0 KW system gives best result for controlling desired temperature. It means, for good tracking control of the system using NFIN, the volume of water should be increased through proportion of 0.5 L with 0.5 KW power of heater.

The same trend of results, as discussed in above section, is obtained for the remaining four systems with 500 mL min^{-1} flow rate of water as shown in Fig. 12-15, respectively. It is also noticeable that systems of 250 mL min^{-1} flow rate of water gives better result with less settling time and overshoot as compare to systems of 500 mL min^{-1} flow rate of water with same configuration. One can say that in our temperature controller the NFIN tracked well the set point temperature of 50°C by the optimal design of PID parameters using neural network in combination with fuzzy inference rules. It means among

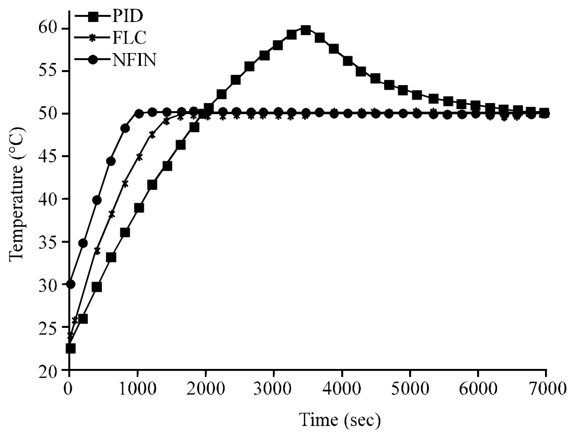

Fig. 14: Temperature response of a water bath having 10 L volume at 1.5 KW for 500 mL min^{-1} flow using NFIN, FLC and PID controller

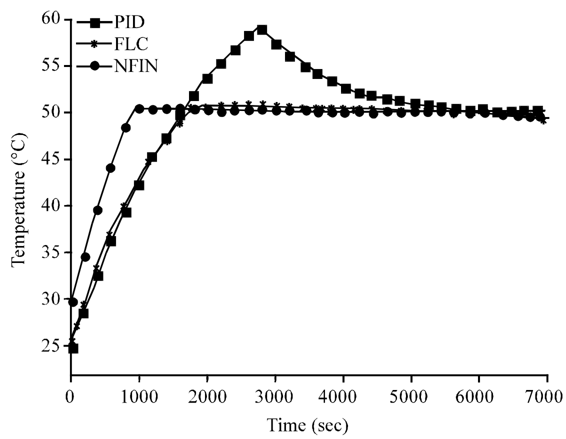

Fig. 15: Temperature response of a water bath having 15 L volume at 1.5 KW for 500 mL min^{-1} flow using NFIN, FLC and PID controller.

the three controllers, NFIN controller has the shortest rise-time and the best regulation-control performance with smallest errors in the tracking path.

CONCLUSION

In conclusion, in this study, a temperature controller based on Neural Fuzzy Inference Network (NFIN) has been proposed to control precisely the desired temperature of water bath. The NFIN is a fuzzy rule-based network possessing neural network's learning ability. The four experiments were conducted, each for 250 and 500 mL min^{-1} flow of water for different volume of water and power of heater. The experimental results of NFIN controller has been compared with FLC and conventional PID controllers, through implement on above systems. These results show that NFIN controller has better control performance in terms of less settling time with minimum overshoot and error than the other two controllers.

REFERENCES

Anderson, C.W., 1987. Strategy learning with multilayer connectionist representations. Proceedings of the 4th International Workshop on Machine Learning, June 22-25, Irvine, CA., pp: 103-114.

Arbaoui, F., M.L. Saidi, S. Kermiche and H.A. Abbassi, 2006. Identification and trajectory control of a manipulator arm using a neuro-fuzzy technique. J. Applied Sci., 6: 2275-2280.

Astrom, K.J. and B. Wittenmark, 1989. Adaptive Control. Addison-Wesley, New York.

Astrom, K.J. and T. Hagglund, 1995. PID Controllers: Theory, Design and Tuning. 2nd Edn., ISA Pub., North Carolina.

Astrom, K.J., T. Hagglund, C.C. Hang and W.K. Ho, 1993. Automatic tuning and adaptation for PID controllers-A survey. Control Eng. Practice, 1: 699-714.

Azeem, M.F., M. Hanmandlu and N. Ahmed, 2003. Structure identification of generalized adaptive neuro-fuzzy inference systems. IEEE Trans. Fuzzy Syst., 11: 666-681.

Caswara, F.M. and H. Unbehauen, 2002. A neurofuzzy approach to the control of a flexible-link manipulator. IEEE Trans. Robotics Automation, 18: 932-944.

Chiu, S.L., 1994. Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Syst., 2: 267-278.

Chopra, S., R. Mitra and V. Kumar, 2005. Analysis and design of fuzzy models using fuzzy curves. Proceedings of the 2nd Conference on Intelligent Systems and Networks, Feb. 25-26, Jagdhari, pp: 65-71.

Chu, S.Y. and C.C. Teng, 1999. Tuning of PID controllers based on gain and phase margin specifications using fuzzy neural network. Fuzzy Sets Syst., 101: 21-30.

Draincov, D., H. Hellendoorn and M.R. Frank, 1996. An introduction to Fuzzy Control. 2nd Edn., Springer-Verlag, Berlin Heidelberg, New York, ISBN: 3540606912, pp: 316.

Fakhrazari, A. and M. Boroushaki, 2008. Adaptive critic-based neurofuzzy controller for the steam generator water level. IEEE Trans. Nuclear Sci., 55: 1678-1685.

Farivar, F., M.A. Shoorehdeli, M. Teshnehlab and M.A. Nekoui, 2009. Hybrid control of flexible manipulator. J. Applied Sci., 9: 639-650.

Haiguo, P. and W. Zhixin, 2007. Simulation research of fuzzy-PID synthesis yaw vector control system of wind turbine. WSEAS Trans. Syst. Control, 2: 469-476.

Harris, C.J., C.G. Moore and M. Brown, 1993. Intelligent Control, Aspects of Fuzzy Logic and Neural Nets. World Scientific Press, London, UK.

Ho, W.K. and W. Xu, 1998. PID tuning for unstable processes based on gain and phase-margin specifications. IEE Proc. Control Theory Appl., 145: 392-396.

Hourfar, F. and K. Salahshoor, 2009. Adaptive feedback linearization control of nonlinear processes using neural network based approaches. J. Applied Sci., 9: 3219-3235.

Hsu, C.F., 2007. Self-organizing adaptive fuzzy neural control for a class of nonlinear systems. IEEE Trans. Neural Networks, 18: 1232-1241.

Huapeng, W. and H. Handroos, 2004. Hybrid fuzzy self-tuning PID controller for a parallel manipulator. Fifth World Congr. Intell. Control Automation, 3: 2545-2549.

Islam, S., N. Amin, M.S. Bhuyan, M. Zaman, B. Madon and M. Othman, 2007. FPGA realization of fuzzy temperature controller for industrial application. WSEAS Trans. Syst. Control, 2: 484-490.

Juang, C.F. and C.T. Lin, 1998. An on-line self-constructing neural fuzzy inference network and its applications. IEEE Trans. Fuzzy Syst., 6: 12-32.

Kasabov, N.K., 1996. Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems. Fuzzy Sets Syst., 82: 135-149.

Khalid, M. and S. Omatu, 1992. A neural network controller for a temperature control system. IEEE Control Syst. Mag., 12: 58-64.

Khalid, M., S. Omatu and R. Yusof, 1993. MIMO furnace control with neural networks. IEEE Trans. Control Syst. Technol., 1: 238-245.

Kosko, B., 1992. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall Inc., Upper Saddle River, New Jersey, USA., ISBN: 0-13-612334-1, pp: 449.

Lin, C.J. and C.T. Lin, 1996. Reinforcement learning for an ART-based fuzzy adaptive learning control networks. IEEE Trans. Neural Networks, 7: 709-731.

Lin, C.J. and Y.J. Xu, 2006. A hybrid evolutionary learning algorithm for TSK-type fuzzy model design. Math. Comput. Modell., 43: 563-581.

Lin, C.J., C.H. Chen and C.Y. Lee, 2006. A TSK-type quantum neural fuzzy network for temperature control. Int. Math. Forum, 1: 853-866.

Lin, C.T., C.T. Lin and C.S.G. Lee, 1996. Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems. Prentice-Hall, Englewood Cliffs, NJ., USA.

Lin, F.J., C.H. Lin and P.H. Shen, 2001. Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. IEEE Trans. Fuzzy Syst., 9: 751-759.

Miller, W.T., R.S. Sutton and P.J. Werbos, 1990. Neural Networks for Control. 3rd Edn., MIT Press, Cambridge, MA., ISBN: 9780262132619, pp: 524.

Munasinghe, S.R., M.S. Kim and J.J. Lee, 2005. Adaptive neurofuzzy controller to regulate UTSG water level in nuclear power plants. IEEE Trans. Nuclear Sci., 52: 421-429.

Narendra, K.S., R. Ortega and P. Dorato, 1991. Advances in Adaptive Control. IEEE Press, New York.

Nurnberger, A., D. Nauck and R. Kruse, 1999. Neuro-fuzzy control based on the Nefcon-model: Recent developments. Soft Comput., 2: 168-182.

Ouyang, C.S., W.J. Lee and S.J. Lee, 2005. A TSK-type neurofuzzy network approach to system modeling problems. IEEE Trans. Syst. Man Cybern. Part B: Cybern., 35: 751-767.

Rumelhart, D.E. and J.L. McClelland, 1986. Parallel Distributed Processing. Vol. 1, The MIT Press, Cambridge.

Sugeno, M., 1985. An introductory survey of fuzzy control. Inform. Sci., 36: 59-83.

Tanomaru, J. and S. Omatu, 1992. Process control by on-line trained neural controllers. IEEE Trans. Ind. Electron., 39: 511-521.

Tareghian, R. and S.M. Kashefipour, 2007. Application of fuzzy systems and artificial neural networks for flood forecasting. J. Applied Sci., 7: 3451-3459.

Tsai, K.I., F.J. Wang, H.C. Lee and H.C. Chiang, 2008. A novel control scheme for temperature control of machine tool coolers. J. Applied Sci., 8: 2491-2495.

Visioli, A., 2001. Tuning of PID controllers with fuzzy logic. IEE Proc. Control Theor. Appl., 148: 1-8.

Wen, S.H. and X.H. Liu, 2004. Application of temperature fuzzy self-tuning PID controller in a resistance furnace. Proc. Int. Conf. Mach. Learning Cybernetics, 2: 817-821.

Wu, W., J. Yuan and L. Cheng, 2005. Self-tuning sub-optimal control of time-invariant systems with bounded disturbance. Proc. Am. Control Conf., 2: 876-882.

Yabuta, T. and T. Yamada, 1991. Learning control using neural networks. Proceedings of the 1991 IEEE International Conference Robotics and Automation, April 9-11, Sacramento, CA., USA., pp: 740-745.

Yazdizadeh, A., A. Mehrafrooz, J. Jouzdani and R. Barzamini, 2009. Adaptive neuro-PID controller design with application to nonlinear water level in NEKA power plant. J. Applied Sci., 9: 3513-3521.