A Neurofuzzy Learning and its Application to Control system

Seema Chopra, R. Mitra, and Vijay Kumar

Abstract—A neurofuzzy approach for a given set of input-output training data is proposed in two phases. Firstly, the data set is partitioned automatically into a set of clusters. Then a fuzzy if-then rule is extracted from each cluster to form a fuzzy rule base. Secondly, a fuzzy neural network is constructed accordingly and parameters are tuned to increase the precision of the fuzzy rule base. This network is able to learn and optimize the rule base of a Sugeno like Fuzzy inference system using Hybrid learning algorithm, which combines gradient descent, and least mean square algorithm. This proposed neurofuzzy system has the advantage of determining the number of rules automatically and also reduce the number of rules, decrease computational time, learns faster and consumes less memory. The authors also investigate that how neurofuzzy techniques can be applied in the area of control theory to design a fuzzy controller for linear and nonlinear dynamic systems modelling from a set of input/output data. The simulation analysis on a wide range of processes, to identify nonlinear components on-linely in a control system and a benchmark problem involving the prediction of a chaotic time series is carried out. Furthermore, the well-known examples of linear and nonlinear systems are also simulated under the Matlab/Simulink environment. The above combination is also illustrated in modeling the relationship between automobile trips and demographic factors.

Keywords—Fuzzy control, Neuro-Fuzzy Techniques, Fuzzy Subtractive Clustering, Extraction of Rules, and Optimization of Membership Functions.

I. INTRODUCTION

IN a conventional fuzzy approach the membership functions and the consequent models are fixed by the model designer according to a priori knowledge. If this set is not available but a set of input-output data is observed from the process, the components of a fuzzy system (membership and consequent models) can be represented in a parametric form and the parameters are tuned by neural networks. In that case the fuzzy systems turn into neurofuzzy system. A fuzzy system can explain the knowledge it encodes but can't learn or adapt its knowledge from training examples, while a neural network can learn from training examples but can not explain what it has learned. Fuzzy systems and neural networks have complementary strengths and weaknesses. As a result, many researchers are trying to integrate these two schemes to generate hybrid models that can take advantage of strong points of both. This is also the motivation of our research.

The authors are with Electronics and Computer Engineering Department, Indian Institute of Technology, Roorkee 247667 Uttranchal, India. (phone: +919837309673; fax: 01332-273560/285368; e-mail: seemadec@iitr.ernet.in, rmtrafec@iitr.ernet.in, vijecfec@iitr.ernet.in).

Recently, neurofuzzy system modeling has attracted a lot of attention [6, 10, 14, 15, 23, 24, 27-32]. In general, this approach involves two major phases, structure identification and parameter identification. The structure identification amounts to 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). Fast computation speed is attained by requiring much less tunable parameters. There is a need for effective methods for tuning the membership functions so as to minimize the output error measure or maximize performance index.

For structure identification, different researchers [16-20, 25, 30-32] use different methods to extract initial fuzzy rules from given input-output data. Clustering techniques [2-4, 22, 25] have been recognized as a powerful alternative approach to develop fuzzy systems. Clustering of numerical data forms the basis of many classification and system-modeling algorithms. The purpose of clustering is to identify natural grouping of data from a large data set to produce a concise representation of a system's behavior. Clustering algorithms typically require the user to prespecify the number of cluster centers and their initial locations. The preceding discussion shows that different researchers have used different clustering algorithms and different cluster validity indices to decide on the number of rules. A clustering method called subtractive clustering forms the basis of the present work.

For parameter identification, most systems [2, 14-15, 19] use backpropagation to refine parameters of the system. However backpropagation suffers from the problem of local minima and low convergence rate. To alleviate these difficulties, different methods of least square estimation (LSE) are proposed [17]. However these methods suffer from the necessity of initializing a certain parameter. The hybrid algorithm means combination of gradient descent and least square estimate (LSE) solves the above problem as discussed in [10]. Hence in this paper backpropagation learning rule is used to tune the parameters in the hidden layer and parameters in the output layer will be identified by the least square method.

This paper contributes a neurofuzzy system for the given set of input-output data, which is obtained in two steps. First, the data set is partitioned into a set of clusters based on the similarity of data. Then using subtractive clustering algorithm a fuzzy if-then rule is extracted from each cluster to form a fuzzy rule base. Secondly, a fuzzy neural network is designed

accordingly to optimize the parameters of the fuzzy system. To decrease the size of the search space and speed up the convergence we are using a hybrid learning algorithm which combines the gradient descent and least square estimator (LSE) method. The proposed combination has the advantage of determining the number of rules automatically, learns faster, consumes less memory and produces lower root mean square error than other methods. A benchmark problem in chaotic time series prediction and automobile trip generation based on its demographics are simulated to compare the performance of the above combination with the published results of other algorithms.

II. NEUROFUZZY 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.
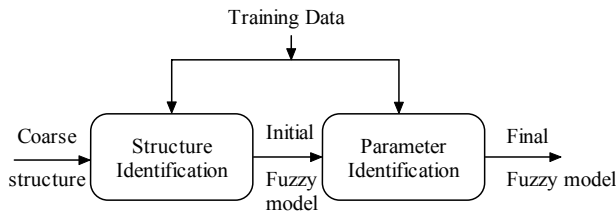

Fig. 1 Steps for Neurofuzzy Learning

To initiate the structure tuning, a training set composed of input-output data, which contains n inputs and one output must be provided. Without loss of generality, we assume that the data points have been normalized in each dimension so that they are bounded by a unit hypercube. We consider each data point as a possible cluster center and define a measure of the potential of data point as discussed in [2, 3]. 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.

One can also write this rule in the more familiar form:
Rule i: If X_1 is A_{i1} & X_2 is A_{i2} &... then class is c1.

where X_j is the j'th input feature and A_{ij} is the membership function (Gaussian type) in the i'th rule associated with the j'th input feature.

The membership function A_{ij} is given by

$$A_{ij}\left(X_j\right) = \exp\left\{-\left(\frac{X_j - m_{ij}}{\sigma_{ij}}\right)^2\right\} \qquad (1)$$

where m_{ij} is mean and σ_{ij} is deviation.

Note that 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.

In parameter identification, the neural network techniques are used to refine the parameters of the initial fuzzy rules. A neural network with three 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 described below:

1. Layer 1: Units in this layer receives the input value (X1, X2,....., Xn) 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 $O_{ij}^{(1)}$, by computing the corresponding Gaussian membership function:

$$O_{ij}^{(1)} = A_{ij}(X_j) \qquad (2)$$

2. Layer 2: 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 $O_j^{(2)}$ of this performs the AND operation by product of all its inputs from layer 1. For instance,

$$O_j^2 = \prod_{i=1}^{n} O_{ij}^1 \qquad (3)$$

3. Layer 3: This layer contains only one node whose output $O^{(3)}$ represents the result of centroid defuzzification, i.e.,

$$O^3 = \frac{\sum\limits_{j=1}^{J} O_j^2 . c_j}{\sum\limits_{j=1}^{J} O_j^2} \qquad (4)$$

Here 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. After that a hybrid learning algorithm which combines gradient descent and least square estimator method is used to refine these parameters. Each epoch of the hybrid learning procedure is composed of a forward pass and backward pass. In the forward pass, input data is supplied and functional signals go forward to calculate each node output. The consequent parameters are identified by least square estimator method. After identifying the parameters, the functional signals keep going forward till the error measure is calculated. In the backward pass, the error rates (derivative of the error measure w.r.t. each node output) propagate from the output end towards the input and the

premise parameters are updated by gradient method. The details of Hybrid learning algorithm is given by Jang in [10] and we are using the same procedure.

Fig. 2 Architecture of Fuzzy neural network

III. SIMULATION RESULTS

The simulations results are further divided into two parts. In the first part, Neurofuzzy learning is used to identify the Fuzzy PI and PD type controllers. In second phase, we shall use the proposed learning algorithm to illustrate some examples such as a benchmark problem of chaotic time series prediction. A comparison between our systems and the published results of other two systems proposed by Jang [10] in 1993 and Chiu [2] in 1994 is presented.

A. Identification of Fuzzy Controller

The neurofuzzy learning is used for identification of PD type FLC (FPDC) and PI type FLC (FPIC). The Block diagram of FPIC is shown in Fig. 3.

Fig. 3 Block Diagram of FPIC

The change in error is defined as

$$\Delta e(k) = e(k) - e(k-1) \qquad (5)$$

where $e(k)$ is the error at the kth sample.

All membership functions (MFs) for controller inputs (i.e., e and Δe) and incremental change in controller output (i.e., Δu) are defined on the common normalized domain [-1, 1]. The

membership functions are shown in Fig. 4.The operation of PI type FLC can be described by

$$u(k) = u(k-1) + \Delta u(k) \qquad (6)$$

In (6), k is the sampling instant and Δu is the incremental change in controller output, which is determined by the rules of the form If e is E and Δe is ΔE, then Δu is ΔU. The rule base for computing is a standard [21] one as shown in Table I.

Fig. 4 MFs for e, Δe and Δu
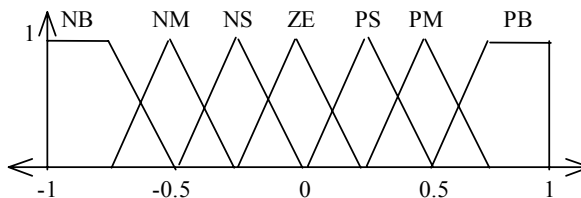
NB-Negative Big, NM-Negative Medium, NS-Negative Small, ZE-Zero Error, PS-Positive Small, PM-Positive Medium, PB-Positive Big

TABLE I
RULE BASE

Δe/e	NB	NM	NS	ZE	PS	PM	PB
NM	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

The FPIC in Fig. 3 uses 49 rules and 7 membership functions in each variable to compute output and exhibits good performance [21]. Next, we investigate the following – Given some data describing the output (Δu) as a function of Inputs (i.e., e and Δe), now main aim is to extract a smaller set of rules using neurofuzzy learning to do the same. Then, the performance of the simple controller (identified system) compare with the original one. Now the following steps are followed:

1. Data Generation

To identify the FPIC and FPDC, some data is needed, i.e., a set of two-dimensional input vectors X={X_1, X_2,….. X_n} and the associated set of one-dimensional output vectors as Y={Y_1,……. Y_n} where X={e and Δe} and Y={u} is required. Here, the training data has been generated by sampling input variables e and Δe uniformly at the step size of 0.1 from the existing FLC [21] and computing the value of {u} for each sampled point. The number of data points generated is 441.

2. Rule Extraction and Membership Functions

After generating the data, the next step is to estimate the initial rules. Then after applying Subtractive Clustering algorithm, four clusters (rules) are extracted. The unit step

response using these four rules is not so close to the identified system [7, 9, 25]. Hence, there is a need of optimization of these rules. Parameter optimization is used for tuning of membership functions to minimize the output error measure or maximize performance index using neural networks. Hybrid learning algorithm is used for training to modify the above parameters after obtaining the Fuzzy inference system from subtracting clustering. This algorithm iteratively learns the parameter of the premise membership functions via back propagation and optimizes the parameters of the consequent equations via linear least-squares estimation. The training is continued until the error measure becomes constant. As the value of these parameters change, the Gaussian membership function varies accordingly. The membership functions after optimization for e are mf1, mf2, mf3 and mf4, and for Δe are m1, m2, m3 and m4 shown in Fig. 5. Finally the rules are written in the form of: Rule i: If e is *mfi* & Δe is *mi* then class is ci when indices i=1 to 4.
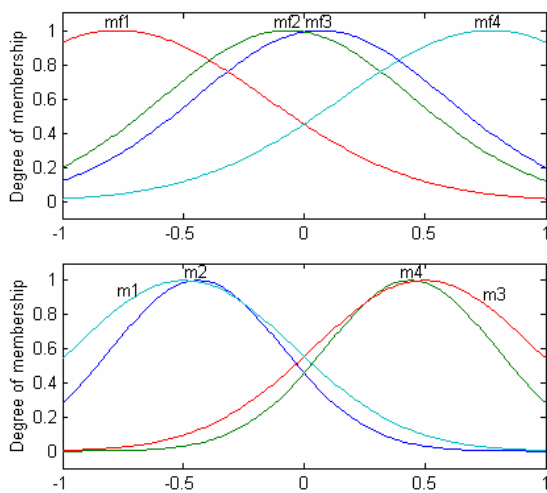
Fig. 5 MFs for e, Δe

B. Results

The neurofuzzy learning has been tested on a variety of linear and nonlinear processes. The objective here is to justify whether the Fuzzy controller with less number of rules and membership functions can provide the same level of performance as that of the original one (system with 49 rules). To demonstrate the effectiveness of the proposed combination, the results are reported for system with 49 rules and system with optimized rule base. After reducing the rules the computation become fast and it also consumes less memory. The computational time is calculated using the Process Explorer –Sysinternals software [27]. It has been shown clearly in the Table and the values of computational time and memory using same simulation time with and without clustering for each system is given in tabular forms in Table II.

In case of Fuzzy PI type and PD type controller, the system with 49 rules (original system) is denoted by FPIC and FPDC

and system with 4 (optimized rules) rules is denoted by HFPIC and HFPDC. In this paper, it is emphasize that an identified system is called satisfactory only with respect to its closeness to the target system, here FPDC and FPIC.

The proposed FLC is applied to an inverted pendulum (PT2 system) [13] which is given by the following differential equation:

$$\frac{1}{w_o^2}\ddot{x} + \frac{2D}{w_o}\dot{x} + x = Vy \qquad (7)$$

where $w_o = \sqrt{50}, D = \frac{2}{5}\sqrt{2}, V = 1$ \qquad (8)

The PT2 system models the behaviour of a two mass system, for example, spring damper combinations or revolution controls for electric motors. This system is controlled by FPIC and FPDC in both cases and results are compared in Fig. 6 and Fig. 7.
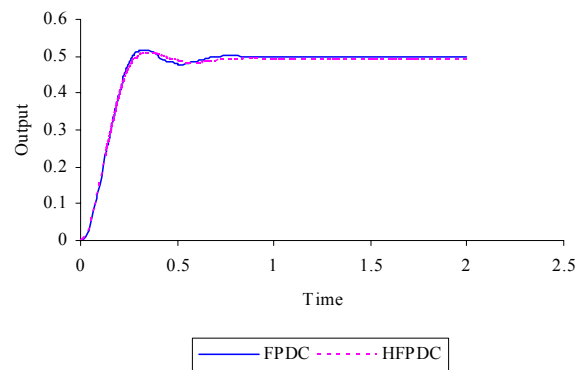

Fig. 6 Unit Step Response of inverted pendulum with FPDC and HFPDC

Secondly, the proposed fuzzy controller is applied on coupled tank for controlling the level of fluid. The control input is the pump drive voltage. The sensed output is the water depth in tank 2. The process of coupled tank [13] is given by:

$$G(s) = \frac{0.4219}{1149.893s^2 + 111.834s + 1} \qquad (9)$$

Then unit step response is observed in case of the system with 49 rules and system with reduced rule set using neurofuzzy learning. Response characteristics for the coupled tank with 49 rules and 4 rules are shown in Fig. 8.

The overall performance of the Fuzzy Controllers with 4 rules is compared with those of Fuzzy Controller with 49 rules. Response characteristics of the identified system in both cases (FPDC and FPIC) are very close to the original one. The computational time is decreased using 4 rules which are clearly shown in Table II.

TABLE II
VALUES FOR COMPUTATIONAL TIME

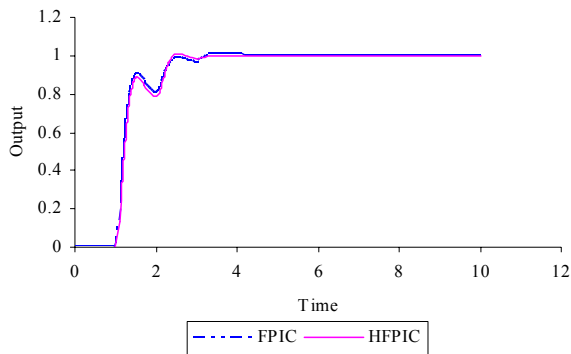| System	With 4 rules	With 49 rules
Inverted pendulum	250 ms	2 sec 31 ms
Coupled Tank	391 ms	5 sec 266ms

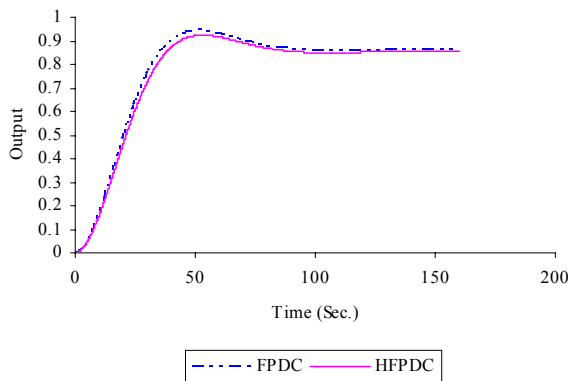Fig. 7 Unit Step Response of inverted pendulum with FPIC and HFPIC

Fig. 8 Unit Step Response of Coupled Tank with FPDC and HFPDC

C. Application Examples

Example 1: Prediction of Chaotic Time Series

In this simulation, the proposed combination is used to predict the future values of chaotic time series. The time series used in our simulation is generated by the chaotic Mackey-Glass differential delay equation [10] defined below:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \qquad (10)$$

The task is to use past values of x up to the time t to predict the value of x at some time t+Δt in the future. The standard input for this type of prediction is N points in the time series spaced s apart, i.e. the input vector is y = {x(t-(N-1) s), …, x(t-2s), x(t-s), x(t)}. For comparison with the published results of other methods, we use the same parameters τ = 17, s=6, N=4, t = 118 to 1117. The first 500 pairs (training data set) were used for training while the remaining 500 pairs (checking data set) were used for validating the identified model. The number of membership functions and rules are initially determined by clustering. Then the Hybrid learning algorithm is used for training to update the parameters. The training is continued until the error measure becomes constant. After 20 epochs error becomes constant and $RMSE_{trn}$ = 0.0015 and $RMSE_{chk}$ = 0.0014, which is same as in [2] but when compared to the published result of Jang [10] the $RMSE_{trn}$ = 0.0016 and $RMSE_{chk}$ = 0.0015 after 499.5 epochs. The predicted values of error for both training data and checking data are comparatively less. The RMSE curves are shown in Fig. 9. Curve with '*' is for training data and with '·' is for checking data.

Example 2: Automobile Trip Prediction

In this example, the neurofuzzy learning is used to estimate the number of automobile trips generated from an area based on demographics of the area. Five demographics factors as taken by S. L. Chiu in [2] were considered: population, number of dwelling units, vehicle ownership, median household income and total employment. Hence there are 5 input variables and 1 output variable. Out of the original 100 data points, we will use 75 data points as training data and 25 data points as checking data, (as well as for test data to validate the model). Using the proposed combination, the average modeling error with respect to the training data was $RMSE_{trn}$ = 0.26 and with respect to the checking data was $RMSE_{chk}$ = 0.33 as shown in Fig 9, if compared by the published results of SL Chiu with same r_a = 0.5 and 3 rules the $RMSE_{trn}$ = 0.34 and $RMSE_{chk}$ = 0.37. Comparison between the both shows that neurofuzzy learning proposed in this paper gives us better results.
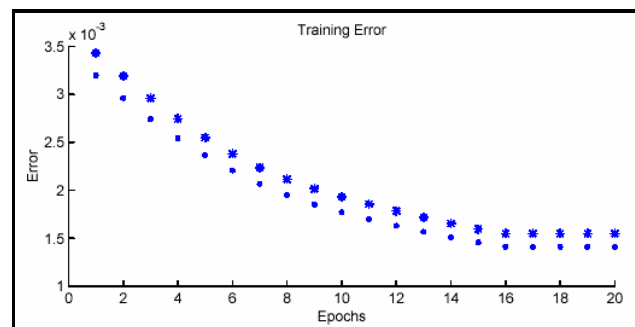

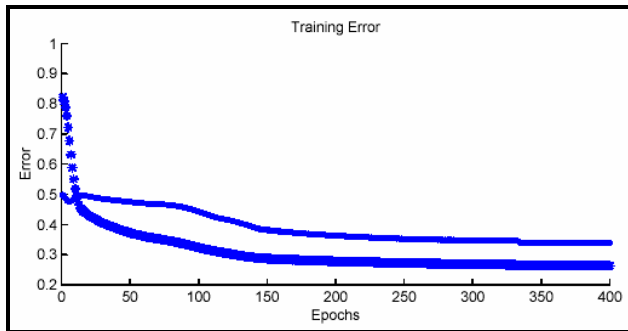Fig. 9 (a) Training and Checking RMSE curves for Example 1

Fig. 9(b) Training and Checking RMSE curves for Example 2.

IV. CONCLUSION

This paper has described a neurofuzzy learning which incorporates the architecture of Neural Network based Fuzzy inference system. A given training data set is partitioned into a set of clusters based on subtractive clustering method. This is fast and robust method to generate the suitable initial membership functions and rule base. A fuzzy IF-then rule is then extracted from each cluster to form a fuzzy rule base from which a fuzzy neural network is designed. The neural network designed in this paper is very simple and contains only three layers. Then a hybrid learning algorithm is used to refine the parameters of fuzzy rule base. The advantages of the discussed neurofuzzy learning are that it determines the number of rules automatically, reduces computational time, learns faster and produces lower RMS errors than other method. Furthermore, the proposed method is able to reduce 49 rules to 4 rules maintaining almost the same level of performance. In addition, well known examples of inverted pendulum and a coupled tank are simulated and identification of results has shown effectiveness of a proposed methodology.

REFERENCES

[1] R.R.Yager and D.P.Filev, "Generation of fuzzy rules by mountain clustering", *Journal of Intelligent and Fuzzy System*, vol.2, pp. 209-219, 1994.

[2] S.L.Chiu, "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy System*, vol.2, pp. 267-278, 1994

[3] S. Chiu, "Extracting fuzzy rules from data for function approximation and pattern classification," in Chapter 9 in *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*, ed. D. Dubois, H. Prade, and R. Yager, John Wiley, 1997.

[4] Peter Grabusts, "Clustering Methods in Neuro - Fuzzy Modelling", Available: http://www.dssg.cs.rtu.lv/download/publications/2002/Garbusts-RA-2002.pdf.

[5] Petr Pivonka, "Comparative analysis of Fuzzy PI/PD/PID controller based on classical PID controller approach". Available: http://www.feec.vutbr.cz/~pivonka/.

[6] Chia-Feng Juang and Chin-Teng Lin, An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications, *IEEE Trans. Fuzzy System* ,vol. 6, no.2, pp.13-31, 1999.

[7] Seema Chopra, R. Mitra and Vijay Kumar, "Identification of Rules using Subtractive Clustering with Application to Fuzzy Controllers," *Proc. of the Third International Conference on Machine Learning and Cybernetics*, 2004, pp.4125-4131.

[8] D. Driankov, H. Hellendorn, and M. Reinfrank, *An Introduction to Fuzzy Control*. New York: Springer-Verlag, 1993.

[9] Seema Chopra, R. Mitra and Vijay Kumar, "Identification of Self-Tuning Fuzzy PI type controllers with reduced rule set", *Proc. of the IEEE International Conference on Networking, Sensing and Control*, Mar., 2005, pp. 537-542.

[10] J. R. Yang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. on systems, man, and cybernetic*, vol. 23, no. 3, pp. 665-685, May 1993.

[11] K Tanaka, M Sano and H. Watanabe, "Modelling and Control of Carbon Monoxide Concentration using a Neurofuzzy technique," *IEEE Trans. Fuzzy System.*, vol 3, no 3, pp. 271-279, Aug. 1999.

[12] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Trans. on Systems, Man, and Cybernetics,* vol. 15, pp. 116-132, 1985.

[13] Kevin M.Passino and Stephen Yurkovich, "*Fuzzy Control*", Addison Wesley Longman, Inc., California, 1998.

[14] A. Nurnberger, D. Nauck and R Kruse, "Neuro-fuzzy control based on the Nefcon-model: recent developments", Soft Computing, vol. 2, 1999, pp. 168-182.

[15] Mohammad Fazle Azeem, Madasu Hanmandlu, Nesar Ahmed, "Structure identification of Generalized Adaptive Neuro-Fuzzy Inference Systems," *IEEE Trans. Fuzzy System.*, vol 11, no. 5, Oct. 2003, pp. 666-681.

[16] Shie-Jue Lee and Chen-Sen Ouyang, "A Neuro-fuzzy system modeling with Self-Constructing Rule Generation and Hybrid SVD-Based Learning," *IEEE Ttrans. Fuzzy System.*, vol. 11, no. 3, pp. 341-353, Jun. 2003.

[17] Chin-Teng Lin and C. S. George Lee, "Reinforcement Structure/ Parameter Learning for Neural Network based Fuzzy Logic Control Systems," *IEEE Trans. Fuzzy System.*, vol 2, no 1, pp.46-73, Feb. 1994.

[18] G. Castellano and A.M. Fanelli, "Fuzzy inference and rule extraction and using neural network". URL

[19] Chin-Teng Lin and Ya-Ching Lu, "A Neural Fuzzy System with fuzzy supervised learning", *IEEE Trans. on systems, man, and cybernetic-Part B*, vol. 26, no. 5, pp. 744-763, Oct. 1996.

[20] R.K.Mudi and N.R.Pal, A robust self-tuning scheme for PI and PD type fuzzy controllers, *IEEE Trans. Fuzzy System,* vol. 7, no.1, pp. 2-16, 1999.

[21] Kuhu Pal, Rajani K. Mudi, and Nikhil R. Pal, "A New Scheme for Fuzzy Rule-Based System Identification and Its Application to Self-Tuning Fuzzy Controllers", *IEEE Trans .on systems, man, and cybernetics—part b: cybernetics*, vol. 32, no. 4, Aug. 2002.

[22] Seema Chopra, R. Mitra and Vijay Kumar, "Fuzzy controller – Choosing an appropriate and smallest rule set", *International Journal on Computational Cognition*, vol 3, no. 4, pp. 73-79, Dec 2005.

[23] Seema Chopra, R. Mitra and Vijay Kumar, "Analysis and Design of Fuzzy Controller using Neurofuzzy Techniques" Proc*. of the International Conference on Computer Applications in Electrical Engineering Recent Advances, CERA 2005*, Sep, 2005, pp 39-43.

[24] Seema Chopra, R Mitra, Vijay Kumar, "Analysis and Design of Fuzzy models using Fuzzy Curves", *Proceedings of the Second Conference on Intelligent systems & networks*, Jagdhari, Feb. 2005, pp. 65-71.

[25] Seema Chopra, R. Mitra and Vijay Kumar, "Analysis of Fuzzy PI and PD type controllers using subtractive clustering", *International Journal on Computational Cognition*, to appear in vol 4, no.1, 2006.

[26] Process Explorer – Sysinternals software. Available: www.sysinternals.com.

[27] Kasabov, N., "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems", *Fuzzy Sets and Systems* vol. 82 no. 2 pp. 2- 20, 1996.

[28] Ferry Marga Caswara and Heinz Unbehauen*,* "A Neurofuzzy Approach to the Control of a Flexible-Link Manipulator,"*IEEE Trans. On Robotics And Automation*, vol. 18, no. 6, pp. 932-944, Dec. 2002.

[29] Sudath R. Munasinghe*,*Min-Soeng Kim and Ju-Jang Lee, "Adaptive Neurofuzzy Controller to Regulate UTSG Water Level in Nuclear Power Plants," IEEE Trans. On Nuclear Science, vol. 52, vo. 1, pp. 421-429, Feb. 2005.

[30] Joseph P. Geisler, C. S. George Lee, and Gary S. May*,* "Neurofuzzy Modeling of Chemical Vapor Deposition Processes," IEEE Trans. on Semiconductor Manufacturing, vol. 13, no. 1, pp. 46-60, Feb. 2000.

[31] William W. Melek and Andrew A. Goldenberg, "Neurofuzzy Control of Modular and Reconfigurable Robots," IEEE/ASME Trans. on Mechatronics, vol. 8, no. 3, pp 381-389, Sep. 2003.

[32] Chen-Sen Ouyang, Wan-Jui Lee, and Shie-Jue Lee, "A TSK-Type Neurofuzzy Network Approach to System Modeling Problems," *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 35, no. 4, pp. 751-767, Aug. 2005.

Seema Chopra was born in Punjab, India in 1976. She received B. Tech. degree in Instrumentation and Control Engineering from Kurukshetra University in 2000, M. Tech. degree in Control Engineering from NIT Kurukshetra in 2002. Currently she is doing Ph.D. in Control and Guidance from the department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, India under MHRD scheme. She is working as Lecturer in Instrument and Control Department in Haryana Engineering College Jagadhri, India and Centre of Advance Technology, Research Centre, Jagdhri, India since 2001. Her current area of research includes Intelligent Control, Fuzzy Logic, Neural Networks, Fuzzy Clustering and SCADA System.

Ranajit Mitra was born in Allahabad, India in 1945. He received B.Sc. degree from Luckhnow University, M. Sc. (Tech.) from University of Allahabad, M.Tech. from Indian Institute of Technology, Kharagpur and Ph.D. degree in Control and Guidance from, University of Roorkee, India in 1975. Besides Post Doctoral work at DFVRL laboratory in West Germany, he had been in the faculty of department of System and Control University of Technology Baghdad and Thapar Engineering College, Patiala. For the Past 36 years Dr. Mitra is with department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, India (formerly University of Roorkee). His current area of research includes Fuzzy Control, development of new experimental systems in the area of Control and Instrumentation.

Vijay Kumar was born in Biharshariff, India in 1954. He received B.E. degree from M.M.M. College of Engineering Gorakhpur, M.E. and Ph.D. degree in Control and Guidance from, University of Roorkee, India in 1983 and 2001 respectively. He was lecturer in department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, India (formerly University of Roorkee) during 1983 to 2002, where currently he is working as Assistant Professor. His current area of research includes Fuzzy Control, Model Order Reductions.