Recurrent Neural Network Based Fuzzy Inference System for Identification and Control of Dynamic Plants

Rahib Hidayat Abiyev

Abstract- This paper presents the development of recurrent neural network based fuzzy inference system for identification and control of dynamic nonlinear plant. The structure and algorithms of fuzzy system based on recurrent neural network are described. To train unknown parameters of the system the supervised learning algorithm is used. As a result of learning, the rules of neuro-fuzzy system are formed. The neuro-fuzzy system is used for the identification and control of nonlinear dynamic plant. The simulation results of identification and control systems based on recurrent neuro-fuzzy network are compared with the simulation results of other neural systems. It is found that the recurrent neuro-fuzzy based system has better performance than the others.

Keywords- Fuzzy logic, neural network, neuro-fuzzy system, control system

I. Introduction

FUZZY systems have found a number of practical applications in identification, control, prediction and diagnosing. These systems are thoroughly dealing with ill-defined, uncertain systems, can model the qualitive aspects of human knowledge and reasoning process [1-5]. Traditionally, to develop a fuzzy system, human experts often carry out the generation of IF-THEN rules by expressing their knowledge. In case of complicated processes it is difficult for human experts to test all the input-output data, to find necessary rules for fuzzy controller. To solve this problem and simplify the generating of IF-THEN rules, several approaches have been applied [2-5]. Nowadays for this purpose the use of neural networks take importance. Using neural network structure and its learning abilities the construction of fuzzy system is considered. The integration of fuzzy system and neural network allow to construct computationally efficient hardware and software products. It is connected with the capabilities that they posses. Fuzzy systems provide powerful framework for representation of expert knowledge, neural network provide learning capabilities that increase the flexibility, adaptability of the system. The combination of neural networks with fuzzy knowledge base helps to reduce the searching space and time for achieving optimal solution. In general, there are two ways about development of systems based on combination of fuzzy system and neural network.

The neural network is represented by fuzzy parameters which is known as fuzzy neural network, and the functionality of fuzzy system is realized by neural network structure, which is known as neuro-fuzzy or neural fuzzy systems. During construction of neuro-fuzzy systems the following requirements are necessary. These are finding the optimal values of neural network parameters and the necessary number of optimum rules.

Neuro-fuzzy system combines the learning capabilities of neural networks with the linguistic rule interpretation of fuzzy inference systems. The synthesis of neuro-fuzzy inference system for controller includes the generation of knowledge base rules that have IF-THEN form. Here, the problem consists in the optimal definition of the premise and consequent part of fuzzy IF-THEN rules for controller through the training capability of neural networks, evaluating the error response of the system.

There are two types of IF-THEN rules used in fuzzy systems. The first one consists of rules, whose antecedents and consequents parts utilize fuzzy values and it is called as Mamdani-type fuzzy rules.

$$\begin{aligned}&\text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and} \ldots \text{and } x_n \text{ is } A_{in}\\ &\text{THEN } y_i \text{ is } C_i\end{aligned} \tag{1}$$

Here x_j and y_i are input and output variables of system, respectively, i=1..l is the number of rules, j=1…n is number of input signals. A_{ij} and C_j are input and output fuzzy sets, respectively.

The second type fuzzy system uses the rule base that has fuzzy antecedent and crisp consequent parts.

$$\begin{aligned}&\text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and} \ldots \text{and } x_n \text{ is } A_{in}\\ &\text{THEN } y_i = b_i + \sum_{j}^{n} a_{ij} x_j\end{aligned} \tag{2}$$

This type of rule is Takagi-Sugeno-Kanag (TSK) type fuzzy IF-THEN rules. The second type of fuzzy system approximates nonlinear system with linear systems. This type of systems employs the other types of fuzzy reasoning mechanism.

Manuscript received July, 2003. This work was supported by the Near East University, Lefkosha, TRNC, Turkey.
Rahib H. Abiyev is with the Department of Computer Engineering, Near East University, Mersin-10, TRNC, Turkey (e-mail: rahib@neu.edu.tr).

Main problem in neural systems is their fast learning. To improve this characteristic several approaches has been proposed. Several investigations have been made in [6-8] by using multilayer perceptron, radial based function neural networks, self organized maps. But disadvantages of these works are not sufficient learning speed of neural networks. The combination of neural network with fuzzy system allows to increase the learning speed. For learning of parameters of such systems the supervised algorithms are widely used. It has good speed, and convergence. One of such algorithms is the back-propagation algorithm. Back-propagation algorithm allows to minimize error function very fast. In the paper the supervised algorithm is applied for training neuro-fuzzy system coefficients.

Different neuro-fuzzy structures are developed for solving identification and control problems [2,5]. In [9] using feedforward neural network the development of adaptive neuro-fuzzy inference system (ANFIS) is presented. The ANFIS structure implements TSK type fuzzy system in a five layers network structure. Using back- propagation and least square algorithms the learning of neuro-fuzzy system is carried out. In [10,11] neural fuzzy controller NEFCON based on architecture of the generic fuzzy perceptron described. For learning of the network parameters the fuzzy error back-propagation algorithm is used. In [12] a training procedure with variable system structure approach for fuzzy inference system is presented. The training dynamics and stability of the system is analyzed, the method for creating stabilizing forces on the training dynamics of neuro-fuzzy system is proposed. In [13,14] the multi-layer feedforward neural network is represented by fuzzy numbers for inputs, targets and connection weights. The learning algorithm of fuzzy neural network is described. In [15] using α- level procedure the training of fuzzy feedforward neural network is considered. In [16-17] the fuzzy neural network is applied for control of technological processes. As a model of fuzzy neuron the minimum operation of weighted input signals is used. Trapezoid fuzzy numbers that are characterized by four parameters represents the weight coefficients of network. Using α cut and interval arithmetic the training of network parameters is carried out. Some of neuro-fuzzy systems have been developed by using recurrent neural network. In [19] the concept of fuzzy system based on recurrent network is proposed. In [20] recurrent fuzzy network is used for nonlinear modeling. The operation principle of this network is similar radial based function network. Recurrent self-organized neural-fuzzy inference system is represented. For learning of networks parameters the supervised learning algorithm is used. In [21] TSK-type recurrent neuro-fuzzy neural network (TRFN) is developed. The outputs of fourth layer are used for feedback connection. The learning problems TRFN by using supervised algorithm and genetic algorithm are presented. In [23,24] using recurrent fuzzy neural network the construction of controller for control dynamic plant are considered. The interval arithmetic is used to train the parameters of network.

In this paper the development of fuzzy inference system based on recurrent neural network for identification and control of dynamic plants is considered.

II. Recurrent Neuro-Fuzzy Inference System

Assume that input signals applied to the network at time k are X(k). Output signals of the network are U(k). The output of neuro-fuzzy system based on feedforward neural network is determined by the following equation.

$$U(k)=F(X(k), M1(k), M2(k)) \qquad (3)$$

Here X(k) and U(k) are external input signals and network output signals correspondingly. For Mamdani type fuzzy rules M1(k) are membership functions of the parameters of premise parts - between input and hidden layers and M2(k) are membership functions of the parameters of consequent part - between hidden and output layers, respectively. For Gaussian type membership function M1(k) and M2(k) depends on two parameters: M1(k)=G(C1(k), Ω1(k)), M2(k)=G(C2(k), Ω2(k)). Here C1(k), Ω1(k) are centers and widths of membership functions between input and hidden layer, C2(k), Ω2(k) are centers and widths of membership functions between hidden and output layers, respectively. G is Gaussian function. For TSK type rules the values of M2(k) are crisp numbers and they are described only by one parameters, that is C2(k).

In this paper the structure of neuro-fuzzy system based on recurrent neural network is accepted as follows

$$U(k)=F(X(k),U(k-1),U(k-2),\dots,U(k-D) , M1(k), M2(k)) \qquad (4)$$

As shown the inputs of recurrent neural network are formed by the external input signal X(k) and one-, two-,…, D- step delayed output signals U(k-1), U(k-2),…,U(k-D) of network.

A. Architecture of Recurrent Neuro-Fuzzy System

In figure 1 the structure of neuro-fuzzy system based on recurrent neural network is shown. The input signals applied to the network at time k are $x_i(k)$ (i=1..N) and output signal of the network are u(k). N is number of neurons in the input layer. d is delay (d=1..D).

In first layer the number of nodes are equal to the sum of external inputs and one-,two-,…, D-step delayed output signals. In second layer each node corresponds to one linguistic term. For each input signal entering the system the membership degree to which input value belongs to a fuzzy set is calculated. To describe linguistic terms the Gaussian membership function is used.

$$\mu 1_j(x_i) = e^{-\frac{(x_i - c1_{ij})^2}{\sigma 1_{ij}^2}} \qquad i=1..n,\ j=1..J \qquad (5)$$

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

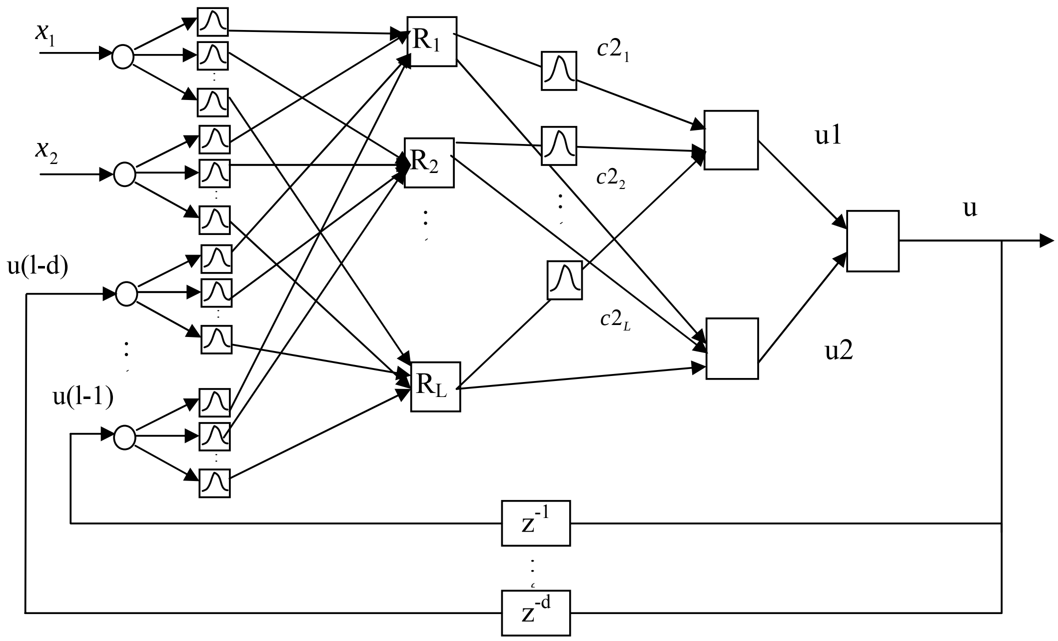

Figure 1. Structure of recurrent neuro-fuzzy inference system

$$\mu 1_j(u_i) = e^{-\frac{(u_i - c1_{i+n,j})^2}{\sigma 1^2_{i+n,j}}} \quad \text{i=1..D, j=J+1..J+P} \tag{6}$$

Here u_i=u(l-i).

Here $c1_{ij}$ and $\sigma 1_{ij}$ are the center and width of the Gaussian membership function of the j-th term of i-th input variable, respectively. n is number of external input signals. J is is number of linguistic terms for external input signal. P is number of linguistic terms for one-, two-,…, D delayed output signal of network.

In the third layer the numbers of nodes correspond to the number of rules. Each node represents one fuzzy logic rule. Here to calculate the values of output signals of the layer AND (min) operation is used

$$\mu_l = \prod_j \mu 1_j(r_i), \quad l\text{=1..L, j=1..J+P} \tag{7}$$

Here $r_i=\{x_1,..,x_n,u_1,\dots,u_D\}$, I=1,…,n+D. Π is min operation.

These μ_l signals are input signals for the next layer. This layer is a consequent layer. In this fourth layer the output signals of previous layer are multiplied to the weight coefficients of network. Weight coefficients of neuro-fuzzy system are represented by fuzzy set of output variables. They are described by Gaussian function. If as a defuzzification operation we use "local mean of maximum" then only the center of Gaussian function is used in the next layer for defuzzification. In this case during development of control system the width of Gaussian function is not used. In formula (8) the parameters $c2_1$ will represent the center of fuzzy coefficients. Outputs of fifth layer are calculated as

$$u1 = \sum_{l=1}^{L} \mu_l * c2_l, \quad l\text{=1..L} \tag{8}$$

and
$$u2 = \sum_{l=1}^{L} \mu_l$$

Using the values of calculated variables, in the last layer the output of the fuzzy system is determined.

$$u = \frac{u1}{u2} = \frac{\sum_{l=1}^{L} \mu_l * c2_l}{\sum_{l=1}^{L} \mu_l} \tag{9}$$

B. Learning of Recurrent Nneuro-Fuzzy System

The unknown parameters of the system are $c2_1$ parameters of last layer and membership functions of first layer of neuro-fuzzy system. To define the accurate values of unknown parameters supervised learning algorithm is used.

$$c2_l(t+1) = c2_l(t) + \eta \frac{\partial E}{\partial c2_l} \tag{10}$$

here γ is learning rate.

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

$$E = \frac{1}{2}\sum_{i=1}^{m}(u_i(t) - u_i^d(t))^2 \quad (11)$$

where $u_i(t)$ and u_i^d are current and desired outputs of the system, m is number of outputs. For given case m=1.

$$\frac{\partial E}{\partial c2} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial c2} = (u(t) - u^d(t))$$

The adjusting of the membership functions of input layer is carried out by correction unknown parameters $c1_{ij}$ and $\sigma 1_{ij}$. The following formulas can be used for learning these parameters.

$$c1_{ij}(t) = c1_{ij}(t) + \gamma\frac{\partial E}{\partial c1_{ij}},$$
$$\sigma 1_{ij}(t) = \sigma 1_{ij}(t) + \gamma\frac{\partial E}{\partial \sigma 1_{ij}}, \quad (12)$$

where

$$\frac{\partial E}{\partial c1_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial \mathrm{u}}\frac{\partial u}{\partial \mu_l}\frac{\partial \mu_l}{\partial c1_{ij}}$$
$$\frac{\partial E}{\partial \sigma 1_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial \mathrm{u}}\frac{\partial u}{\partial \mu_l}\frac{\partial \mu_l}{\partial \sigma 1_{ij}}$$

Here

$$\frac{\partial E}{\partial u} = u(t) - u^d(t), \quad \frac{\partial u}{\partial \mu_l} = \frac{\sum_{l=1}^{L} c2_l - u}{\sum_{l=1}^{L}\mu_l} \quad (13)$$

$$\frac{\partial \mu_l(x_j)}{\partial c_{ji}} = \begin{cases} \mu_l(x_j)\dfrac{2(x_j - c_{ji})}{\sigma_{ji}^2} & if\ j \text{ node is connected to rule node } l \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial \mu_l(x_j)}{\partial \sigma_{ji}} = \begin{cases} \mu_l(x_j)\dfrac{2(x_j - c_{ji})^2}{\sigma_{ji}^3} & if\ j \text{ node is connected to rule node } l \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

Using (10) - (14) the learning of the parameters of recurrent neuro-fuzzy system is carried out.

III. Simulations of Recurrent Neuro-Fuzzy Inference Systems

A. Identification of Non-linear Systems

The identification problem is finding relation between input and output of the system. Here the recurrent neuro-fuzzy inference system (RNFIS) is used for modelling dynamic plant. The inputs of dynamic plant are external input signals, its one-,…, d_i- step delayed values and one-, two-,…,d_o- step delayed outputs of the plant. Output of the system is determined by the following equation.

$$y(k)=f(u(k),u(k-1),\ldots,u(k-d_i), y(k-1),y(k-2),\ldots,y(k-d_o)) \quad (15)$$

In figure 2 the structure of identification scheme is shown.

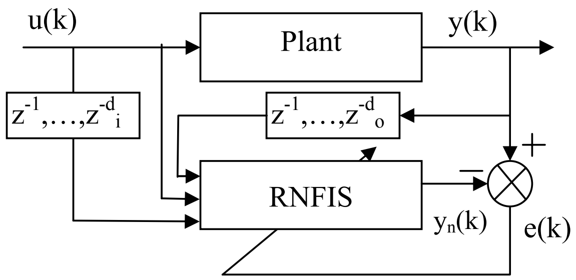

Figure.2. Identification scheme

The problem is to find such values of parameters of RNFIS by using them in the system for all input values of u(k) the difference between y(k) and $y_n(k)$ will be minimum. Here y(k) is plant output, $y_n(k)$ is output of neuro-fuzzy system. As an example a second order nonlinear plant that has been discussed in [25] is considered. The process is described by the following difference equation.

$$y(k) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{(1+y(k-1)^2 y(k-2)^2)} + u(k) \quad (16)$$

Here y(k-1), y(k-2) are one- and two- step delayed output of the dynamic plant, u(k) is input signal. As a input signal for the plant the sinusoidal signal u(k)=sin(πk/20) is given. Input signals at the same time are applied to the plant and RNFIS inputs. On the output of the plant the difference between plant output and RNFIS output is determined $e(k)=y(k)-y_n(k)$. If the value of error is not acceptable minimum value the training of RNFIS start. The training is continued until the value of error becomes less than small acceptable value. After next input signal is given to the system input and learning process is continued. The training of RNFIS is carried out for 6000 and 12000 data points. During simulation the initial values of parameters of premise part c1 and σ1 are generated randomly in the interval [–10, 10], parameters of consequent part c2 and σ2 - in the interval [–1, 1]. The training is carried out for sixteen and forty-eight rules. For this reason the learning of neuro-fuzzy system is performed for two cases. In the first case, number of neurons in hidden layer is sixteen, in the second case- forty-eight. RNFIS input includes external sinusoidal input signal and, current and one- step delayed output of the plant. As a result of learning the membership functions of premise and consequent parts of RNFIS have been found. In figure 3 the curves that describe identification results are shown. Here solid line describes the output of the plant, dashed line describes the output of the RNFIS. This figure demonstrates on-line training processes. After training obtained RNFIS model was tested by different input signal. Figure 4 demonstrate the test results of RNFIS when input signal is u(k)=sin(π*k/20)+cos(k/10).

In table I the results of simulations of identification using RNFIS and neuro-fuzzy inference system based on feedforward network (NFIS) are given. Table describes on-line learning and test results for different amount of rules and different number of iterations. To estimate efficiency of RNFIS the sum of square of errors and CPU- times are taken. Sum of square errors are calculated as

$$J = \sum_{k=1}^{K} e^2(k) \tag{17}$$

Here K is number of samples. The experiments are carried out with 16 and 48 rules. The 6000- and 12000- learning results are fixed. In on-line learning mode, because of feedback connection the CPU time in RNFIS is greater than in NFIS. The estimated sum of square errors of systems with 48 rules for both networks 1,5 – 2 times less than sum of square errors with 16 rules. The test results demonstrate the efficiency of the RNFIS for solving identification of dynamic plant. The test is carried out for 300 iterations. Test results are taken for 6000 and 12000 learning iterations. Results of comparisons of RNFIS and NFIS identifications show that in the same condition the value of sum of square errors for RNFIS system two times is less than NFIS system. The increase of a number of rules from 16 to 48 allows to decrease sum of square errors for two – three times.

B. Neuro-Fuzzy Control

The RNFIS structure and learning algorithms described above is used for development of controller to control parameters of dynamic plant. In figure 5 the structure of RNFIS based control system is given. The inputs for neuro-fuzzy controller are error and change of error. The coefficient k_u is used for scaling output signal of controller. Using the values of error and change of error it is needed to determine such values of control signal by using them in control system the target characteristic of the system would be provided.

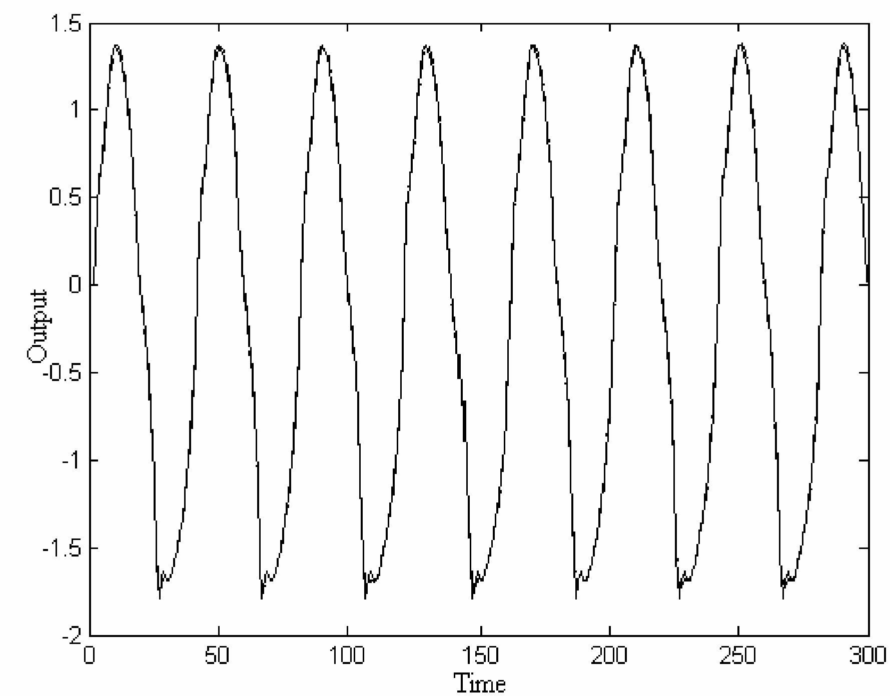

Figure 3. Learning results of identification by using recurrent neuro-fuzzy system. Solid line is output of the plant, dashed line output of the recurrent neuro-fuzzy system.

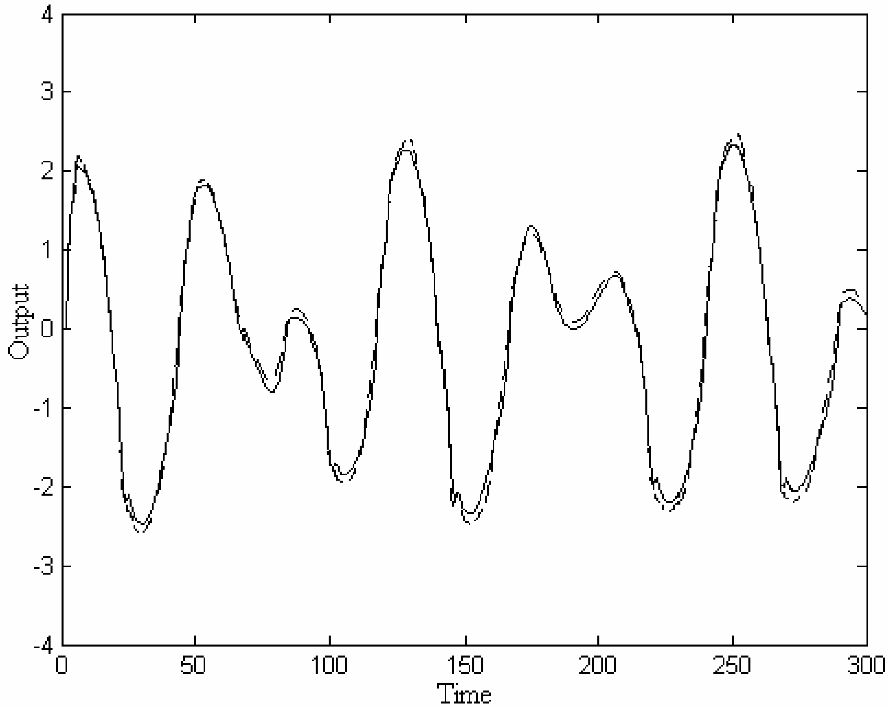

Figure 4. Test result

TABLE I
RESULTS OF COMPARISON BETWEEN TWO IDENTIFIERS

	Number of rules	Number of iteration		NFIS		RNFIS	
				Error $\sum_{k=1}^{K} e^2(k)$	CPU - time	Error $\sum_{k=1}^{K} e^2(k)$	CPU - time
On line learning	16	6000		259.19	187.9	404.5	377.06
		12000		335.64	289.68	497.6	546.89
	48	6000		243.46	474.23	245.16	576.34
		12000		288.97	630.33	290.37	793.57
Test results	16	300	6000 learning result	13.65	0.77	6.83	0.99
			1200 learning result	7.23	0.87	3.49	0.99
	48	300	6000 learning result	7.92	2.2	2.14	2.74
			12000 learning result	1.96	2.2	1.43	2.74

At first stage the fuzzy parameters of RNFIS that has Gaussian form are generated. In RNFIS structure the first layer represents input signals error, change of error and one step delayed output of neuro-fuzzy system, second layer is used to represent membership functions of fuzzy parameters of premise part. The third layer represents the number of rules. Forth layer represents membership functions of parameters of consequent part. Fifth and sixth layers realize defuzzification mechanism. Using input variables error and change of error and RNFIS structure the generation of IF-THEN rules for controller in closed loop control system is performed. The consequent part of rules includes control signal given to the object. To find association u(k)=f(e(k),e'(k),u(k-1)) between input and output variables of the controller, the learning of unknown parameters of neuro-fuzzy controller in closed loop control system is performed. For learning of the unknown coefficients of neuro-fuzzy controller the error between target characteristic of control system and current output value of implemented system (output of control object) $e(k) = g(k) - y(k)$ is used. For learning controller coefficients the above described supervised learning algorithm is used. Using learning algorithm the values of weight coefficients of neuro-fuzzy controller are determined.

Example 1. The development of RNFIS based control system is carried out for controlling dynamic plant that is described by (16).

Computer simulation of RNFIS based control system for dynamic plant (16) is carried out. The initial values of the parameters of membership functions of second (c1 and σ1) and fourth (c1 and σ1) layers are generated randomly, in the interval [–10, 10] and [–1, 1] correspondingly. During learning the values of parameters of premise (second layer) and consequent parts (fourth layer) are adjusted. In figure 6 the curve that describes the learning processes of control system for different values of set-point signals is given.

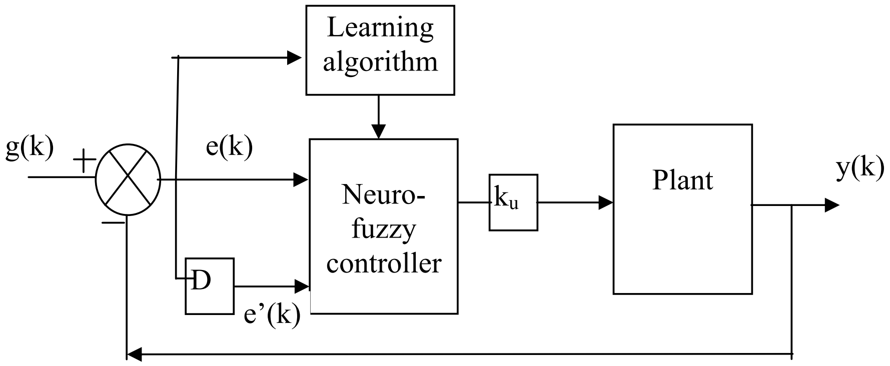

Figure 5. Structure of RNFIS based control system

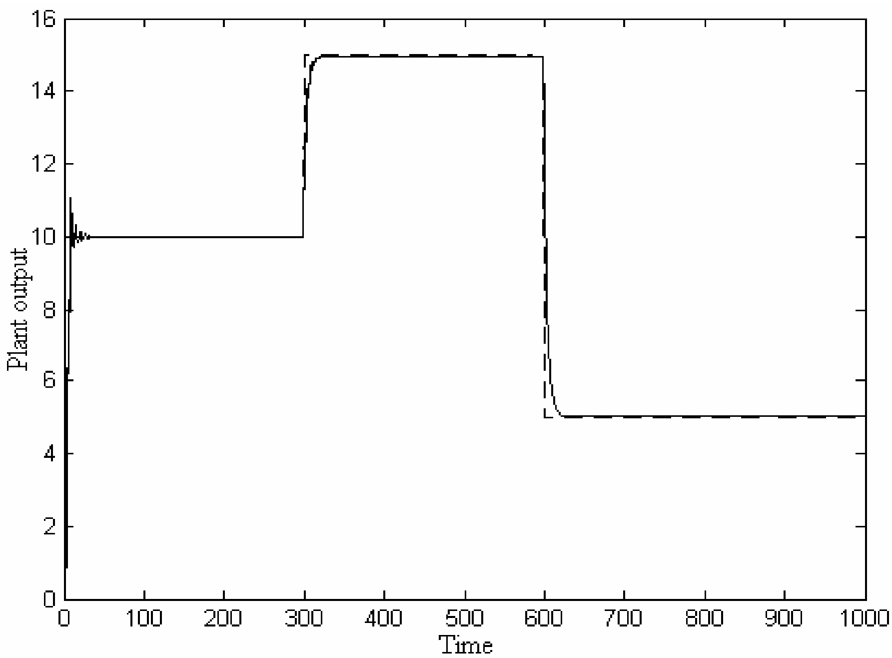

Figure 6. Learning curve of time response characteristic of control system for different values of set-point signal.

As a result of learning corresponding values of coefficients of recurrent neuro-fuzzy system are determined. Simulation results of RNFIS based control system is compared with the simulation results of NFIS and RNN (recurrent neural network) based control systems. In figure 7 results of comparative estimation of time response characteristics of control systems based on RNFIS, NFIS, and RNN are given. Number of rules are 16. The results of simulation of control system based on RNFIS shows that the value of static error of time response characteristics is absent (zero), transient overshoot is also absent. The settling time of system with RNFIS based controller is less than others. The results of simulation and experimental analysis of control system with RNFIS shows that it has better time response characteristic than others.

In table II results of comparative estimation of control systems among three controllers are given. Table also describes on-line learning and test results for different amount of rules and different number of iterations. Online learning results of control systems with three controllers show that the increasing a number of rules from 16 to 48 decreases the value of sum of square errors. The test is carried out for 100 iterations. The 1000 iteration- learning results are used for test. Results of comparisons of RNFIS, NFIS and RNN based control systems show that in the same condition the value sum of square of errors for RNFIS is less than for others. Test results demonstrate the efficiency of RNFIS for constructing controller.

Example 2. The development of control system based on RNFIS is carried out for controlling temperature of rectifier K-2 column. During simulation the model of plant is described by the following differential equation.

$$a_0 \frac{dy^2(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_2 y(t) = b_0 u(t - \tau) \qquad (18)$$

where $a_0 = 0.056\min^2$, $a_1 = 0.072\min$, $a_2 = 1$, $b_0 = 60^o C/(kgf/cm^2), \tau = 15\sec$, here y(t) is regulation parameter of plant, u(t) is controller's output, τ is delay. Sampling time for the plant is 15 sec.

Computer simulation of control system with RNFIS for plant (18) is carried out. The initial values of the parameters of membership functions c1 and σ1 are chosen in the interval [-10, 10], c2 and σ2 in the interval [–0.2, 0.2]. Number of rules are 16. During learning the parameters of RNFIS are determined. Also the development of control system with NFIS and RNN for plant (18) is carried out. The learning of controllers based on RNFIS and NFIS are carried out at the same condition. The initial values of parameters for both networks are generated in the same interval. In figure 8 the curves that describe the test results of time response characteristics of control systems with RNN,

NFIS and RNFIS are given. The results of simulation of control system based on RNFIS show that the value of static error of time response characteristics is absent (zero). Transient overshoot and settling time of system with RNFIS are less than other types of controllers. The results of simulation and experimental analysis of control system with RNFIS show that it has better time response characteristic than others.

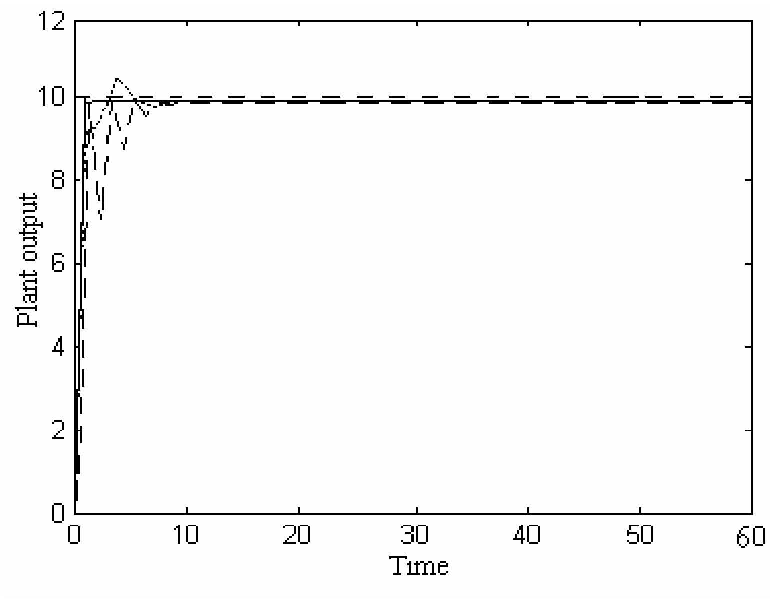

Figure 7. Test result - time response characteristic of control systems based on RNFIS (solid line), NFIS (dotted line), and RNN (dashed line).

TABLE II
RESULTS OF COMPARISON AMONG THREE CONTROLLERS

	Number of rules	Number of iteration	RNN		NFIS		RNFIS	
			$\sum_{k=1}^{K} e^2(k)$	CPU-time	$\sum_{k=1}^{K} e^2(k)$	CPU-time	$\sum_{k=1}^{K} e^2(k)$	CPU - time
On line learning	16	1000	369.68	5.66	8936	7.45	7960.8	6.21
	48	1000	158.82	8.7	10428	11.04	7911.7	14.28
Test results	16	100	112.49	1.19	109.90	1.51	100.38	1.81
	48	100	111.53	1.23	107.78	1.69	100.14	2.01

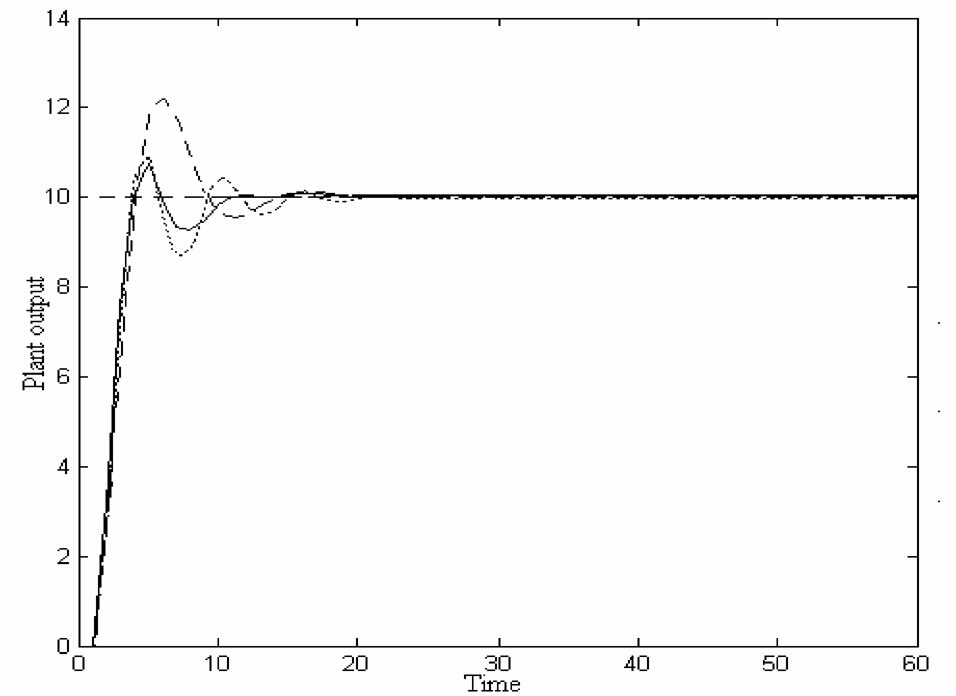

Figure 8. Test result - time response characteristic of control systems based on RNFIS (solid line), NFIS (dotted line), and RNN (dashed line).

Simulation result of control system based on RNFIS for plant (18) is compared with the simulation results of control systems based on NFIS and RNN. In table 3 the results of comparative estimation of time response characteristic of control systems based on RNFIS, NFIS and RNN controllers are given.

As shown in table III the value of sum of square errors for RNFIS based control system is less than others. The results of simulation and experimental analysis of the automatic control system with RNFIS show its efficiency.

TABLE III
RESULTS OF COMPARISON

Criteria	RNN	NFIS	RNFIS
CPU-time	0.99	0.88	0.99
$\sum_{l=1}^{M} e^2(k)$	168.35	171.70	162.62

IV. Conclusion

In this paper the development of recurrent neuro-fuzzy system for identification and control of dynamic plant is considered. The structure and learning algorithms of RNFIS are described. For learning of network the supervised algorithm is used. The learning capability of RNFIS allows automatically construct itself and to deal with non-stationary plants.

The simulation result of identification and control systems based on RNFIS are compared with other types of neural network based system. In identification problem the RNFIS test results have shown two times better performance in sum of square of errors than NFIS system. In control RNFIS based control system test results are compared with RNN and NFIS based control systems results and RNFIS based control system has shown better performance than other types of systems. Result of comparative estimation demonstrates the efficiency of presented approach.

References

[1] Zadeh L.A.(1975). The concept of linguistic variable and its application to approximate reasoning. Information Sciences, v.8.
[2] Kosko B. (1993). Neural networks and fuzzy systems. A dynamical system approach to machine intelligence. Prentice-Hall International Inc., Englewood Cliffs.
[3] Yager R.R., Zadeh L.A.(Eds). (1994). Fuzzy sets, neural networks and softcomputing, New York, Van Nostrand Reinhold.
[4] Witold Pedryz, editor, (1996) Fuzzy Modelling: Paradigms and Practice, Kluwer Academic Publisher, Boston.
[5] Aliev R.A., Tserkovni A.E., and Mamedova G.A. (1991). Production management at fuzzy initial information. Moscow, Energiatomizdat, (Russian)
[6] J.Reeman and D.Saad.(1997). Online learning in radial basis function networks, Neural Comput., vol.9,no.7,
[7] T.Mheskes and B. Kappen. (1993). Online learning processes in artificial neural networks. Math. Found. Neural Networks, Amsterdam, The Netherlands: Elsevier, pp.199-233.
[8] Diederich J. (1990). Artificial Neural Networks. Consept learning, Los Alamitos CA: IEEE Computer Society Press.
[9] Jyh-Shing Roger Jang. (1993). "ANFIS: Adaptive-Network-Based Fuzzy Inference System". IEEE Transactions on Systems, Man and Cybernetics, Vol.23, No.3, pp.665-683.
[10] Nauck, Detlef and Kruse, Rudolf. (1996). Designing neuro-fuzzy systems through backpropagation, In Witold Pedryz, editor, Fuzzy Modelling: Paradigms and Practice, Boston, Kluwer Academic Publisher,pp.203-228
[11] Detlef Nauck. (1994). Building neural-fuzzy controllers with NEFCON-I. In Rudolf Kruse, Jorg Gebhardt, and Rainer Palm(Eds), Fuzzy Systems in Computer Science, Artificial Intelligence, Wiesbaden, Vieweg,pp.141-151.
[12] M.Onder Efe, and Okyay Kaynak. (2000). "On stabilization of Gradient-Based Training Strategies for Computationally Intelligent Systems". IEEE Transactions on Fuzzy Systems, Vol.8, No.5, October, pp.564-575.
[13] J.J.Buckley, Y.Hayashi, and E.Czogola.(1993). Fuzzy neural networks with fuzzy signals and weights. International Journal on Intelligent Systems 8, pp.527-537.
[14] J.J.Buckley, and Y.Hayashi. (1993). Fuzzy neural networks. In L.A.Zadeh and R.R.Yager (Eds), Fuzzy Sets, Neural networks and Soft Computing, Van Nostrand Reinhold, pp.233-249.
[15] H.Ishibuchi, K. Morioka, and H.Tanaka. (1994). A fuzzy neural network with trapezoidal fuzzy weights. Proc. FUZZ-IEEE, Orlando, Florida, June 26-29, pp.228-233.
[16] R.A.Aliev, R.H.Abiyev, and R.R.Aliev. (1994). Automatic control system synthesis with the learned neural network based fuzzy controller. Moscow, News of Academy of Sciences, Tech. Cybernetics 2:pp. 192-197.
[17] R.A.Aliev, F.T.Aliev, R.H.Abiev, and R.R.Aliev. (1994). Industrial neural controllers. EUFIT'94, Promenade 9,52076, Aachen, Germany. Elita foundation
[18] R.H.Abiyev, K.W.Bonfig, and F.T.Aliev. (1996). Controller based on fuzzy neural network for control of technological process. ICAFS-96, Siegen, Germany, June 25-27,pp295-298.
[19] J.Zhang, and A.J.Morris, (1999). Recurrent neuro-fuzzy networks for nonlinear process modeling. IEEE Trans. Neural Networks, vol.10,no.2, Mart, pp.313-326,
[20] C.H.Lee, and C.C.Theng. (2000). Identification and control of dynamic systems using recurrent fuzzy neural network. IEEE Trans. Fuzzy Systems, vol. 8, pp.349-366.
[21] Chia-Feng Juang. (2002), A TSK –type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithm, IEEE Trans. Fuzzy Systems, vol.10, pp.155-170.
[22] James Keller, Ronald R.Yager, and Hossein Tahani. (1992). Neural network implementation of fuzzy logic, Fuzzy Sets and Systems, 45:pp.1-12.
[23] Rahib Abiyev. (2001). Controllers based on Softcomputing elements// Electrical, Electronics and Computer Engineering Symposium NEU-CEE2001 & Exhibition. Nicosia, TRNC, Turkey, May 23-25, pp.182-188.
[24] Rahib Abiyev. (2002). Fuzzy inference system based on neural network for technological processes control. Journal of Mathematical and Computational Applications. Turkey, pp.245-252.
[25] Jaier Nunez-Garcia and Olaf Wolkenhauer. (2002). Random Set System Identification. IEEE Transactions on Fuzzy Systems, Vol.10, No.3, October, pp.287-296.
[26] Rahib Abiyev. (2002). Neuro-Fuzzy system for technological processes control. The 6th World Multi-Conference on SYSTEMICS, cybernetics and informatics. SCI-20002, Orlando, Florida, USA. July 14-18.

Rahib Hidayat Abiyev was born in Azerbaijan, in 1966. He received Ph.D degree in Electrical and Electronic Engineering from Azerbaijan State Oil Academy (old USSR). He worked research assistant at the research laboratory "Industrial intellectual control systems" of Computer-aided control system department at Oil Academy for six years. From 1999-present he is working as associate professor at the department of Computer Engineering of Near East University, TRNC, Turkey. He is vice chairman of Computer Engineering Department. He has published more than sixty papers in related fields. He is member of IJCI, IJSP and IJIT from 2003 and member of IEEE. His research interests are Intellectual Control Systems, Fuzzy Systems, Neural Networks, Genetic Algorithms, Chaos Theory, Signal Processing, Pattern Recognition, Optimization.