International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Stabilization of Non-Linear System Using Single Input Neuro-Fuzzy Logic Controller Based on Radial Basis Function Network

Ravi Kumar Soni[1], Dr. M. J. Nigam[2]

[1, 2] IIT Roorkee, Electronics and Communication, Roorkee, Uttarakhand, India

Abstract: ***This paper proposes a Single Input Neuro-fuzzy Logic Controller based on Radial Basis Function Network (SI-NFRBFN) for non-linear systems. To obtain single input from multi inputs a Distance Method is suggested. Using this method all the uncertain inputs are simplified into a single input known as distance. With the help of this variable the control unit matrix introduced in Hybrid Neuro-fuzzy Logic Controller based on Radial Basis Function Network (HNFRBFN) is reduced from its 2-dimensional state to a single dimension state, without much affecting the response of the system. By obtaining single dimensional control matrix it is seen that the computational analysis in the learning part of HNFRBFN controller reduces drastically. Due to this the tuning effort and execution time of the proposed SI-NFRBFN controller is minimized. The simulation results show that the SI-NFRBFN controller provides better performance and is more efficient in comparison to HNFRBFN controller.***

Keywords: Radial basis membership function, Neural network, Fuzzy logic controller, Gradient descent algorithm.

1. Introduction

A number of efforts have been performed to develop different controllers for non-linear systems including both (conventional) linear and modern control schemes. As all the real time systems present in any field are non linear in nature. Therefore linear controllers are unable to control these real time systems. Recently Fuzzy systems, Neural networks (NN), Radial basis function networks etc. [1]-[3] are used as universal function approximator for non-linear system because these systems or networks are not linear in nature. Hence, with the help of these, a controller for non-linear system can be designed easily. To increase the efficiency & speed of the controller many hybrid techniques also came in sought like neuro-fuzzy logic controller, fuzzy logic controller based on radial basis function network etc [4]-[5]. These hybrid techniques are developed by adding the theories of individual's mechanism to get the superior properties of each mechanism in the hybrid controller. For example, neural networks estimation power and low level training properties enhances the fuzzy system and the the effective-humanly thought process and reasoning of fuzzy system is brought into NN by neuro-fuzzy method [6]-[8].

Here a novel hybrid algorithm depend on an error correction algorithm is deliberated to design a neuro-fuzzy network. Its name is hybrid neuro-fuzzy logic controller based on radial basis function network (HNFRBFN), which implements Takagi-Sugeno form of fuzzy controller [9]. Main improvement in the suggested approach is that the controller is designed for a system without any previous information about the sub-division of input set and the no. of fuzzy rules. The HNFRBFN is derived from radial basis function (RBF) neurons network which consist a centre vector and a width vector. The input set is gets subdivided into subsets with each RBF neuron [10]. The RBF neuron is used to derive the premise part (IF part) of the fuzzy rule. And the consequence part (THEN part) of the fuzzy rule is synthesized by weighted control action matrix which incorporates an error correction algorithm. Here two stages get involved in this algorithm. Initially the RBF neurons are used to generate the basic structure without any previous knowledge about how to sub-divide the input set. The gradient descent mechanism as an error correction tool is suggested in the 2nd stage and it tries to improvise the response of the plant with the help of parameter learning. Initially, all the membership curves have constant centre & width and the information are inseminated in the network. The parameters of the control matrix are obtained by utilising the recursive error estimation [11]. Then, the parameter of the control matrix is retained fixed, while the error is circulated backwards in the system to modify the response of the plant using the gradient descent.

Later on in this paper, the proposed HNFRBFN controller is modified. In the modified controller we give only one uncertain input to controller instead of error and change in error to approximate the antecedent part of the fuzzy rule. Therefore the modified controller is termed as Single Input hybrid neuro fuzzy logic controller based on radial basis function network (SI-HNFRBFN). Generally conventional FLC's consist of a rule table in two-dimensional format which is skew-symmetric in nature [12]-[13]. But the control unit matrix introduced in the proposed HNFRBFN controller is symmetric in nature. The distance from the main diagonal element defines the absolute magnitude of the controlled input. The PID- tuned FLC's also consist of this property which uses error, change in error & sum of error as its uncertain input variables. The absolute value of the controlled input is directly proportional to the perpendicular distance from the main diagonal line. On the basis of this proportionality, a new variable is suggested, called signed distance variable [14]-[15]. This distance represents the actual state of the controlled input from the main diagonal line and it is the sole uncertain input in the modified HNFRBFN controller called SI-HNFRBFN. In this way, the number control actions in the control unit matrix of SI-

HNFRBFN controller is greatly reduced in comparison to the HNFRBFN controller. Moreover, from the simulation results we can see that the control performance of the control unit matrix of both the controller's are almost equivalent to each other.

This paper comprises four sections. Section II discusses the gradient descent method with design of HNFRBFN controller. Section III describes the signed distance method with formation of SI-HNFRBFN. In section IV, both the controllers are applied to an unstable plant using MATLAB Simulink to obtain different result with comparisons. In the last section, conclusions are summarized by comprehending the simulation and results.

2. Design of HNFRBFN with Gradient Descent Method

A. Gradient Descent Method

Gradient descent method is a first-order up-gradation technique. The local minima of a function can be found out by this method. In this method initial guess is assumed as starting point and we perform the gradient or partial derivative of the function at this point. We continue the solution in the negative aspect of the gradient and iterate this process. This algorithm ultimately converges when the gradient becomes zero.

Let us consider we have P number of observation using for the actual training of the network. So the combined error (E) of all the P observation is given as:

$$E = \sum_{P} E^P \tag{1}$$

$$E^P = \frac{1}{2}\sum_{O}\left(t_o - y_o^P\right)^2 \tag{2}$$

Where, E^P is the error for observation point P, t_o & y_o^P are target output or desired output and actual output at P_{th} observation respectively, O is the total number of outputs and the factor ½ is used for the computational convenience because the gradient of squared error is two times which is cancelled by this factor. Now we have to find out the gradient (G) of the combined error (E).

$$G = \frac{\partial E}{\partial w_o} = \frac{\partial}{\partial w_o}\sum_{P} E^P \tag{3}$$

$$G = \sum_{P}\frac{\partial}{\partial w_o} E^P \tag{4}$$

Where, w_o is the initial synaptic interconnection weight. Now we have to simply perform the above derivation which can be carried out by chain rule of differentiation.

$$\frac{\partial E^P}{\partial w_o} = \frac{\partial E^P}{\partial y_o^P}\cdot\frac{\partial y_o}{\partial w_o} \tag{5}$$

From equation (2):

$$\frac{\partial E^P}{\partial y_o^P} = -\left(t_o - y_o^P\right) \tag{6}$$

And we know that the current or actual output (y_o^P) is given as:

$$y_o^P = \sum_{i} w_{o_i}^P \cdot x_i \tag{7}$$

Where, x_i represents the i_{th} input variable. Now, take the derivation of above equation (7) w.r.t. $w_{o_i}^P$:

$$\frac{\partial y_o^P}{\partial w_{o_i}} = \frac{\partial}{\partial w_{o_i}}\sum_{i} w_{o_i}\cdot x_i = X \text{ (say)} \tag{8}$$

On substituting the values from equation 6 & 8 in equation 5:

$$\frac{\partial E^P}{\partial w_{o_i}} = -\left(t_o - y_o^P\right)\cdot X \tag{9}$$

The above equation represents the gradient of the error for P_{th} observation w.r.t. initial weight for i_{th} input. So the correction $\left(\Delta w_{o_i}\right)$ that we have to apply to the weights is negative of the gradient. Hence the correction in weights $\left(\Delta w_{o_i}\right)$ is given as:

$$\Delta w_{o_i} = \eta\cdot\left(t_o - y_o^P\right)\cdot X \tag{10}$$

Where, η is the proportionality constant known as learning rate. It will not affect the direction of the gradient. It only affects the rate of gradient. Since the gradient is already a negative quantity therefore we have to add this quantity to the corresponding existing weight to obtain new weight (w_i) for next iteration.

$$w_i = w_{o_i} + \Delta w_{o_i} \tag{11}$$

$$w_i = w_{o_i} + \eta\cdot\left(t_o - y_o^P\right)\cdot X \tag{12}$$

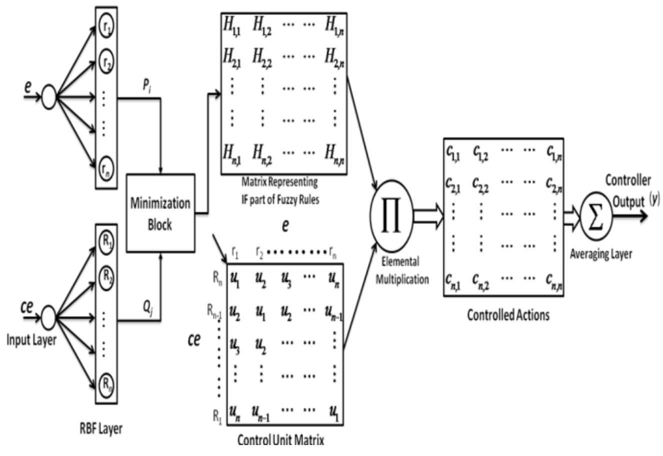

Figure 1: Architecture of HNFRBFN

B. Tuning of HNFRBFN with Gradient-Descent Method

In this subsection, formulation of proposed HNFRBFN controller with gradient descent method is described. Initially the parameters of HNFRBFN are randomized without any previous knowledge. The parameters of each control actions of control unit matrix are then optimized by gradient descent method. Here the HNFRBFN shown in Fig. 1 is configured for two inputs error (e) & change in error (ce) and one output (y), which we have to be control. At RBF layer both the uncertain inputs e and ce are approximated by their radial basis membership functions $(r_1, r_2, \ldots\ldots, r_n)$ & $(R_1, R_2, \ldots\ldots, R_n)$ respectively. Each of the radial basis membership functions has a center (o_i for e & O_i for ce) and width (d_i for e & D_i for ce) for the inputs. The approximated values at RBF layer is given by using the radial basis function:

$$P_i = \exp\left(-\left\|\frac{o_i - e}{d_i}\right\|^2\right) \tag{13}$$

$$Q_j = \exp\left(-\left\|\frac{O_j - ce}{D_j}\right\|^2\right) \tag{14}$$

Where, i and j vary from 1 to n. In the minimization block of HNFRBFN shown in Fig. 4, the intersection values (H_{ij}) are obtained by comparing the P_i and Q_j values according to the premise part of the fuzzy mechanism.

$$H_{i,j} = \min\left(P_i, Q_j\right) \quad i, j = 1,2,3,..,n. \tag{15}$$

The matrix H represents the premise part of the rule base of fuzzy mechanism. A control unit matrix (U) has been introduced here based on the degree of non-linearity or uncertainty of error and change in error. This matrix is fully synthesized by expert's knowledge and experience in control engineering. The no. of control actions in this matrix should be equivalent to the no. of radial basis membership function (n) for our convenience. Mathematically the description of control unit matrix and its individual control action is given as:

$$u_i = a_i \cdot e + b_i \cdot ce + c_i \quad i = 1,2,3,\ldots,n. \tag{16}$$

$$U = \begin{bmatrix} u_1 & u_2 & u_3 & \cdots & u_n \\ u_2 & u_1 & u_2 & \cdots & u_{n-1} \\ u_3 & u_2 & \cdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ u_n & u_{n-1} & \cdots & \cdots & u_1 \end{bmatrix} \tag{17}$$

Where, a_i and b_i are the coefficients or learning parameters of error and change in error respectively; c_i is a constant parameter. These parameters get updated automatically using gradient descent method at all iteration. The modification formula for these coefficients is given by equation (12) of gradient descent method as:

$$a_{i,new} = a_{i,old} + \eta \cdot \left(t_o - y\right) \cdot e \tag{18}$$

$$b_{i,new} = b_{i,old} + \eta \cdot \left(t_o - y\right) \cdot ce \tag{19}$$

$$c_{i,new} = c_{i,old} + \eta \cdot \left(t_o - y\right) \tag{20}$$

Further the elemental multiplication of the control unit matrix (U) and the matrix H is carried out. This multiplicative value gives the controlled action ($c_{i,j}$) for each rule in the rule base. At the averaging layer, the average of all the controlled actions with respect to the H matrix is performed to obtain the controller output (y) as:

$$c_{i,j} = H_{i,j} * U_{i,j} \quad i, j = 1,2,3,..,n \tag{21}$$

$$y = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} C_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{n} H_{i,j}} \tag{22}$$

$$y = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(H_{i,j} * U_{i,j}\right)}{\sum_{i=1}^{n}\sum_{j=1}^{n} H_{i,j}} \tag{23}$$

3. Design of Single Input NFRBFN (SI-NFRBFN) Controller

a) Signed Distance Method

Linguistic variable based fuzzy logic controllers (FLC's) can mimic the human thoughts to solve a particular problem with the help of some rule inferences. Generally two uncertain inputs, namely error and change in error are fed to the FLC's. The rule table with these inputs is created in two dimensional space of phase-plane as shown in the fig 2. Typically the output membership in the diagonal lines is same in the rule table and it is a common property also for all fuzzy rule tables.

ce \ e	PB	PS	Z	NS	NB
NB	Z	NS	NB	NB	NB
NS	PS	Z	NS	NB	NB
Z	PB	PS	Z	NS	NB
PS	PB	PB	PS	Z	NS
PB	PB	PB	PB	PS	Z

Saturation Region

L_{NB} L_{NS} L_Z L_{PB} L_{PS}

Saturation Region

Figure 2: Rule table of Conventional FLC with Toeplitz structure

From fig 2 it can be easily seen that each particular diagonal line consist control actions of same magnitude proportional to the distance from main diagonal line L_Z. Such type of structure is recognized as Toeplitz structure. The control unit matrix introduced in the HNFRBFN controller has similar structure to the table shown in fig 2. Only difference is that the table shown in fig 2 is skew-symmetric in nature where as the control unit matrix is symmetric nature. Therefore the concept of distance as applied in Toeplitz structure can be applied to the control unit matrix in the proposed HNFRBFN controller.

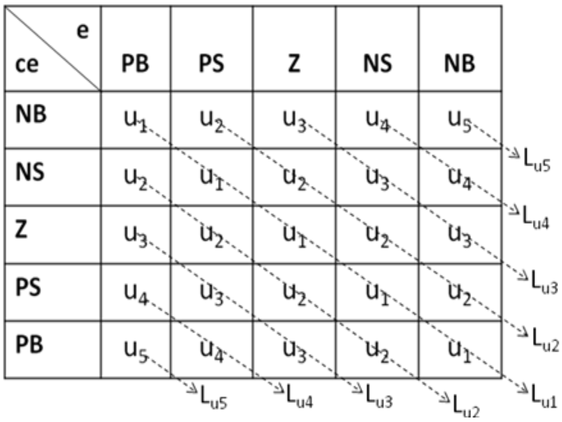

Figure 3: Control Unit Matrix with diagonal lines

According to the pattern of output values in the tables of fig 2 & fig 3, it is cleared that these tables can be simplified considerably. It is possible that the two variable input set of error and change in error can be replaced by a single variable input to obtain the corresponding output. This method of reduction was first proposed by Choi et al. and is known as the Signed distance method [16]-[18]. Using this method all the uncertain inputs are simplified into a single input known as distance (d). The magnitude of distance represents the absolute value of the space between the main diagonal line (L_{u1}) and the lines parallel to it as shown in fig 3. The distance variable d, can be derived by assuming a intersection point $A(e_0,\dot{e}_0)$ on main diagonal line by drawing a perpendicular line from a known operating point $B(e_1, \dot{e}_1)$ as shown in fig 4.

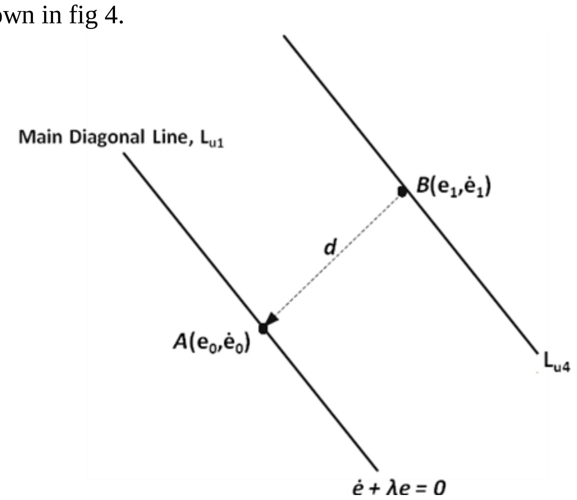

Figure 4: Derivation of distance variable, d

It can be noted that the main diagonal line can be represented as a straight line function, i.e.:

$$L_{U1}: \quad \dot{e} + \lambda e = 0 \tag{24}$$

The values of controlled actions above and below the switching line mentioned in the fig. 2 are of opposite signs where as the controlled actions in control unit matrix shown fig. 3 are of same sign. Therefore, the suggested signed distance method can be termed as only distance method in the proposed controller. In fig. 4, the point $A(e,\dot{e})$ is the intersection point on the main switching line by the line perpendicular from the operating point $B(e_l,\dot{e}_1)$ lying on the distant line L_{U4}. So the perpendicular distance, d between point A & B is given by:

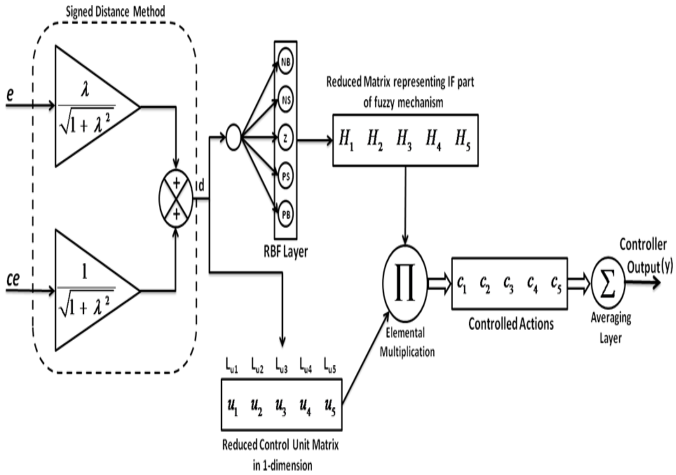

Figure 5: Architecture of SI-NFRBFN

$$d = \left[\left(e - e_1\right)^2 + \left(\dot{e} - \dot{e}_1\right)^2 \right]^{1/2} \tag{25}$$

$$d = \frac{|\dot{e}_1 + \lambda e_1|}{\sqrt{1+\lambda^2}} \tag{26}$$

In the generalised way the above equation can be written as:

$$d = \frac{|\dot{e} + \lambda e|}{\sqrt{1+\lambda^2}} \tag{27}$$

In fig. 3, in both the direction of main switching line the distance is of same sign due to symmetric property of control unit matrix of the HNFRBFN controller. This matrix is uniformed into a single dimensional space of distance d, in place of two dimensional space of e & ė as shown in the following table.

Table 1: Rule table of control action for SI-NFRBFN

d	L_{U1}	L_{u2}	L_{U3}	L_{U4}	L_{U5}
u	u_1	u_2	u_3	u_4	u_5

From the above table it can be seen that the total number of rules are drastically decreases in SI-NFRBFN comparison to the HNFRBFN controller. Hence, tuning of this controller can be easily achieved by modifying the rules.

b) Formation of SI-NFRBFN

In this subsection, formulation of SI-NFRBFN controller with gradient descent method is described. Initially the parameters of this controller are also randomized without any previous knowledge. The SI-NFRBFN shown in Fig. 4 is configured for two inputs error (e) & change in error (ė) and one output, which we have to be control. In this controller, before the RBF layer the above discussed signed distance method is used. With the help of this method the uncertain inputs e and ė are transformed into a single uncertain variable distance (d). At RBF layer input variable, d is approximated by five radial basis membership functions termed by NB, NS, Z, PS and PB. Each of the radial basis membership functions has a center and width (O_i & D_i respectively) for this input.

The approximated values at RBF layer is given by using the radial basis function:

$$H_i = \exp\left(- \left\| \frac{O_i - d}{D_i} \right\|^2 \right) \tag{28}$$

Where, *i* vary from 1 to 5. In this controller, the matrix H is representing the premise part of the rule base of fuzzy mechanism but in a reduced manner. According to the value of this distance variable the control unit matrix is modified which consist of only five numbers of control actions. This matrix is fully synthesized by expert's knowledge and experience in control engineering. The no. of control actions in this matrix should be equivalent to the no. of radial basis membership function (*n*) for our convenience. Mathematically the description of modified reduced control unit matrix in one dimensional space of distance variable and its individual control action is given as:

$$u_i = p_i \cdot d + q_i \quad i = 1, 2, 3, \ldots, 5. \tag{29}$$

$$U = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 \end{bmatrix} \tag{30}$$

Where, p_i is the coefficient of distance variable and q_i is a constant parameter. These parameters get updated automatically using gradient descent method at all iteration. The modification formula for these coefficients is given by equation (12) of gradient descent method as:

$$p_{i,new} = p_{i,old} + \eta \cdot (t_o - y) \cdot d \tag{31}$$

$$q_{i,new} = q_{i,old} + \eta \cdot (t_o - y) \tag{32}$$

Further the elemental multiplication of the reduced control unit matrix (*U*) and the matrix *H* is carried out. This multiplicative value gives the controlled action (c_i) for each rule in the rule base. At the averaging layer, the average of all the controlled actions with respect to the *H* matrix is performed to obtain the controller output (*y*) as:

$$c_i = H_i * U_i \quad i, j = 1,2,..,5 \tag{33}$$

$$y = \frac{\sum_{i=1}^{5} C_i}{\sum_{i=1}^{5} H_i} \tag{34}$$

$$y = \frac{\sum_{i=1}^{5} (H_i * U_i)}{\sum_{i=1}^{5} H_i} \tag{35}$$

4. Simulation Results

For the simulation purpose we consider an unstable plant whose open loop transfer function is:

$$G(s) = \frac{77.8421}{1.0311s^2 - 30.5250} \tag{36}$$

The open loop transfer function of this plant is having two poles; one is stable and another is unstable. The unity feedback response of this transfer function is completely oscillatory in nature. To control the response of this plant we use our proposed controllers HNFRBFN & the modified controller SI-NFRBFN. In this simulation the HNFRBFN controller accepts five inputs which are error, change in error, actual output, gain constant and target output where as the SI-NFRBFN controller takes only four inputs namely distance, actual output, gain constant & target output. Five radial basis membership functions are used to approximate the inputs in both the controllers. The introduced control unit matrix in HNFRBFN controller consists of five control actions. These are arranged in a matrix *U* according to the uncertainty of the error and change in error using expert's knowledge and experience as shown in table 2. And in SI-NFRBFN controller this matrix is reduced in single dimension as shown in table 1. By setting the different target outputs the corresponding results have been obtained for the same unit step input which means that these controllers are able to provide different desired targets.

Fig. 6, fig.7, and fig. 8 show the comparative responses of the system when the desired output is set for unit step, sinusoidal wave, and square-wave respectively. The final values of control actions using HNFRBFN controller & SI-NFRBFN controller are tabularized in their respective table just after the figure.

Table 2: Arrangement of control actions

e / *Δe*	*NB*	*NS*	*Z*	*PS*	*PB*
PB	u_1	u_2	u_3	u_4	u_5
PS	u_2	u_1	u_2	u_3	u_4
Z	u_3	u_2	u_1	u_2	u_3
NS	u_4	u_3	u_2	u_1	u_2
NB	u_5	u_4	u_3	u_2	u_1

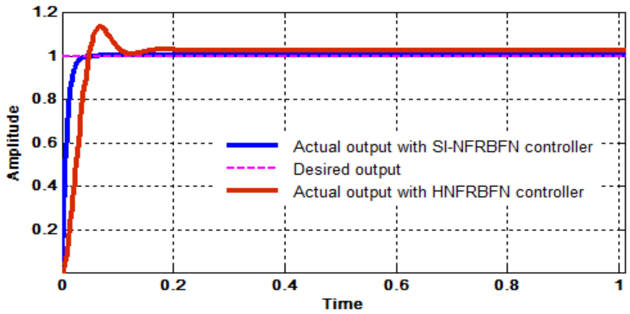

Figure 6: Unit step response of the system

Table 3: Values of control actions for unit step response with HNFRBFN controller

e / *Δe*	*NB*	*NS*	*Z*	*PS*	*PB*
PB	-0.5509	-0.0910	-0.2800	-0.8218	-1.0927
PS	-0.0910	-0.5509	-0.0910	-0.2800	-0.8218
Z	-0.2800	-0.0910	-0.5509	-0.0910	-0.2800
NS	-0.8218	-0.2800	-0.0910	-0.5509	-0.0910
NB	-1.0927	-0.8218	-0.2800	-0.0910	-0.5509

Table 1: Values of control actions for unit step response with SI-NFRBFN controller

d	L_{U1}	L_{u2}	L_{U3}	L_{U4}	L_{U5}
u	0.2371	0.051	-0.1268	-0.3087	-0.4907

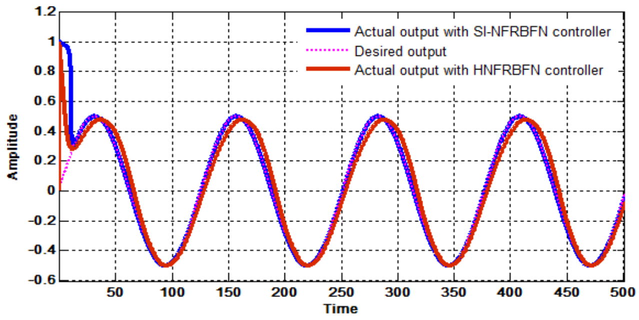

Figure 7: Sinusoidal response of the system

Table 4: Values of control actions for sinusoidal response with HNFRBFN controller

e / Δe	NB	NS	Z	PS	PB
PB	13.3284	-33.6423	-10.1569	36.8138	60.2992
PS	-33.6423	13.3284	-33.6423	-10.1569	36.8138
Z	-10.1569	-33.6423	13.3284	-33.6423	-10.1569
NS	36.8138	-10.1569	-33.6423	13.3284	-33.6423
NB	60.2992	36.8138	-10.1569	-33.6423	13.3284

Table 5: Values of control actions for sinusoidal response with SI-NFRBFN controller

d	L_{U1}	L_{u2}	L_{U3}	L_{U4}	L_{U5}
u	-2146.7	-1592.7	-1038.7	-484.7	69.3

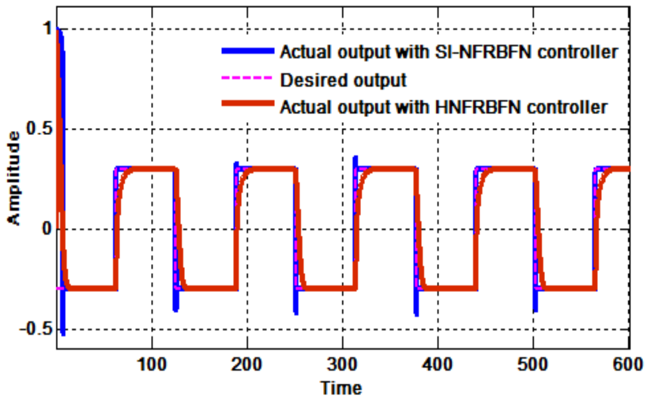

Figure 8: Square-wave response of the system

Table 6: Values of control actions for square-wave response with HNFRBFN controller

e / Δe	NB	NS	Z	PS	PB
PB	7.8812	-20.1189	-6.1188	21.8813	35.8813
PS	-20.1189	7.8812	-20.1189	-6.1188	21.8813
Z	-6.1188	-20.1189	7.8812	-20.1189	-6.1188
NS	21.8813	-6.1188	-20.1189	7.8812	-20.1189
NB	35.8813	21.8813	-6.1188	-20.1189	7.8812

Table 7: Values of control actions for sinusoidal response with SI-NFRBFN controller

d	L_{U1}	L_{u2}	L_{U3}	L_{U4}	L_{U5}
u	-1254.2	-911.30	-568.50	-225.60	117.30

In fig 6, fig 7 & fig 8, the dotted pink line represent the desired output of the system, the solid brown line represents the actual response of the system with HNFRBFN controller & the solid blue line represents the actual response of the system with SI-NFRBFN controller. The transient characteristic of unit step response shown in fig 3 with these controllers is given as:

Table 8: Transient Characteristics of unit step response

Controller / Characteristics	HNFRBFN	SI-NFRBFN
Rise Time	0.035 sec	0.017 sec
Settling Time	0.2162 sec	0.061 sec
Maximum overshoot	1.132	1.0003
Percentage Overshoot	13.2 %	0.03 %

5. Conclusion

Universal function approximations such as fuzzy system or neural network are a solution for controlling difficult non-linear system of different areas. The proposed HNFRBFN controller and SI-NFRBFN controller in this paper is an improvement over the conventional methodologies. It is evident from the results that system has no tendency to go to oscillatory state; hence there is no question of instability. Moreover, by reducing the rise time and overshoot in the transient part of the response, speed of the system is also improved. From table 9 it is clear that the SI-NFRBFN controller provides better performance and is more efficient in comparison to HNFRBFN controller. Another observation from the results is that the proposed controller is able to obtain different desired output with same input. The steady state error between the actual output and target output is minimised with time.

References

[1] K. Tanaka, M. Sano, and H. Watanabe, " Modelling and control of carbon monoxide concentration using a neuro fuzzy technique," IEEE Trans. Fuzzy syst., vol. 3, pp. 271-279, Aug. 1995.
[2] M. Sugeno, and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," IEEE Trans. Fuzzy syst. Vol. 1, pp. 7-31, Feb. 1993.
[3] Y. LIn, and G. A. Cunningham, "A new approach to fuzzy-neural system modelling," IEEE Trans. Fuzzy syst., vol. 3, pp. 190-197, May 1995
[4] J. A. Leonard and M. A. Kramer, "Radial basis function networks for classifying process faults," IEEE Contr. Syst., pp. 31-38, Apr. 1991.
[5] T. L .Seng, M. B. Khalid and R. Yusof, "Tuning of neuro-fuzzy controller by genetic algorithm," IEEE Trans. Syst., Man, Cybern., vol. 29, no. 6, pp. 226-235, Apr. 1999.

[6] B. Kosko, Neural networks and fuzzy system", Englewood Cliffs, NJ: Prentice-Hall, 1992.
[7] C. T. Lin, "Neural fuzzy control system with structure and parametre learning," New York: World scientific, 1994.
[8] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decisión system," IEEE Trans. Comput., vol. 40, pp. 1320-1336, Dec. 1991.
[9] T. Sugeno and M. Sugeno, " Fuzzy identification of the system and its application to modelling and control," IEEE Trans. Syst., Man, Cybern., vol. SMC-15, no. 1, pp. 116-132, Jan. 1985.
[10] C. F. Junag and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," IEEE Trans. Fuzzy syst., vol. 6, no. 1, pp. 12-32, Feb. 1998.
[11] G. Leng, T. M. McGinnity and G. Prasad, "Design for self-organizing fuzzy neural networks based on genetic algorithms," IEEE Trans. Fuzzy syst., vol. 14, no. 6, pp. 755-766, Dec. 2006.
[12] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," IEEE Trans. Syst., Man, Cybern., vol. 22, no. 6, pp. 1414-1427, Nov./Dec. 1992.
[13] B. Liu, C. Kung and C. Y. Chen, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," IEEE Trans. Syst., Man, Cybern., vol. 31, pp. 32-53, Feb. 2001.
[14] A. A. Khan. N. Rapal, "Fuzzy PID controller: design, tuning and comparision with conventional PID controller," IEEE International Conference on Enginnering of Intelligent System, pp 1-6, 2006.
[15] Yongsheng Ding, "A non-linear PID ccontroller based on Fuzzy-tuned immune feed back law," IEEE Trans. Intelligent Control and Automation, vol. 3, pp. 1576-1580,2000.
[16] B. J. Choi, S. W. Kwak and B. K. Kim, "Design of a single-input fuzzy logic controllersand its properties," Science Direct Fuzzy Sets and Systems, vol. 106, pp. 299-308, 1999.
[17] K. Ishaque, S. S. Abdullah, S. M. Ayob and Z, Salam, "Single input fuzzy logic controller for unmaned underwater vehicles," Springer Netherlands Journal of Intelligent and Robotic System, vol. 59, pp. 87-100, Feb. 2010.
[18] S. Y. Chen, F. M. Yu and H. Y. Chung, "Decoupled fuzzy logic controller with single-input fuzzy logic," Science Direct Fuzzy Sets and Systems, vol. 129, pp. 335-342, 2002.

Author Profile

Ravi Kumar Soni received the B. Tech. degree in Electrical Engineering from ITER College, SOA University, Bhubaneswar in 2013. Then he received the M. Tech. degree in System Modeling and Control in Electronics & Communication Engineering Department from IIT Roorkee in 2015.

Dr. M.J. Nigam received the B.Tech. degree in Electronics and Communication Engineering from REC- Warangal (A.P.) in 1976. Then he received the M.E. and Ph.D. degree in Electronics and Communication Engineering from IIT Roorkee (formerly University of Roorkee) in 1978 and 1992 respectively. His areas of interest are digital image processing, guidance and control in navigational systems including high resolution rate.