IOSR Journal of Electronicsl and Communication Engineering (IOSR-JECE)
ISSN: 2278-2834-, ISBN: 2278-8735, PP: 01-11
www.iosrjournals.org

Intelligent Learning Of Fuzzy Logic Controllers Via Neural Network And Genetic Algorithm

Mr.suhas Yadav[1], Prof. S.S.Patil[2].
[1,2] *Department of Electronics Engineering*
Tatyasahab Kore College of Engineering and Technology, Warananagar

ABSTRACT: Design *of an efficient fuzzy logic controller involves the optimization of parameters of fuzzy setsand proper choice of rule base. There are several techniques reported in recent literature that use neural network architecture and genetic algorithms to learn and optimize a fuzzy logic controller. This paper develops methodologies to learn and optimize fuzzy logic controller parameters based on neural network and genetic algorithm. The strategies developed have been applied to control an inverted pendulum and results have been compared for three different fuzzy logic controllers developed with the help of iterative learning from operator experience, genetic algorithm and neural network. The results show that Genetic-Fuzzy and Neuro-Fuzzy approaches were able to learn rule base and identify membership function parameters accurately.*
Keywords: Fuzzy logic controller; Neural network; Genetic algorithm, Genetic-Fuzzy & Neuro- Fuzzy approaches.

I.INTRODUCTION

Control of complex and non linearsystems is an important and challenging task and various strategies have appeared in recent literature to deal with nonlinearity and strong coupling of dynamic systems. PID is a popular control method extensively used in an industrial set up. The advantages of a PID controller include its simple structure along with robust performance in a wide range of operating conditions. A lot of research has been done on PID control scheme (see for example, references [1,2]) and the available methods for tuning PID gains are advanced and accurate. This makes the PID control as one of the most favored control strategies. However, the design of a PID controller is generally based on the assumption of exact knowledge about the system. This assumption is often not valid since the development of model of any practical system may not include precise information of factors such as friction, backlash, unmodeled dynamics and uncertainty arising from any of the sources.

In recent years, there has been an increasing interest in the utilization of unconventional control strategies such as neural networks (NN), fuzzy logic, and genetic algorithm (GA) etc. These control methods derive their advantages from the fact that they do not use any mathematical model of the system. Instead they use input-output relations (Neural Network) or heuristic knowledge (Fuzzy Logic) about the system. This paper investigates the use of fuzzy logic to control a single link manipulator robot. Performance of fuzzy controllers derived from three different methods has been compared in this paper. The first (fuzzy) controller has been designed based on the operator experience and trial and error iteration. The second controller has been optimized with the help of GA, while the third controller has its parameters obtained with the help of NN.

FUZZY LOGIC / GENETIC ALGORITHM / NEURAL NETWORK

Fuzzy set theory [3] was originally proposed by Prof. Lotfi A. Zadeh of the University of Californiaat Berkeley to antitatively and effectively handle problems involving uncertainty, ambiguity and vagueness. The theory, which is now well-established, was specifically designed to mathematically represent uncertainty and vagueness and provide formalized tools for dealing with the imprecision that is intrinsic to many real world problems. The ability of Fuzzy Logic to deal with uncertainty and noise has led to its use in controls [4-5]. Fuzzy logic is inherently robust since it does not require precise, noise-free inputs. It is not limited to a few feedback inputs and one or two control outputs. Fuzzy control is most reliable if the mathematical model of the system to be Controlled is unavailable, and the system is known to be significantly nonlinear, time varying, or to have a time delay. Designing a fuzzy controller requires describing the operator's control knowledge/experience linguistically. The controller captures these traits in the form of fuzzy sets, fuzzy logic operations, and fuzzy rules. Thus, Fuzzy logic control can be used to emulate human expert knowledge and experience. The fuzzy sets and fuzzy rules can be formulated in terms of linguistic variables, which help the operator to understand the functioning of the controller.

GA based search and optimization techniques have recently found increasing use in machine learning, robot motion planning, scheduling, pattern recognition, image sensing and many other engineering applications. Genetic Algorithms(GAs) are search algorithms based on mechanics of natural selection and natural genetics [6-7]. They combine survival of the fittest among the string structures with randomized, yet organized, information exchange to form a search algorithm with capabilities of natural evolution. A GA starts with a random creation of a population of strings and thereafter generates successive populations of strings that improve over generations. The processes involved in the generation of new populations mainly consist operationssuch as Reproduction, Crossover and Mutation. GAs have proven their robustness and usefulness over other search techniques because of their unique procedures that differ from other normal search and optimization techniques.

Neural Network (NN) methods have become very popular recently involving mapping of input-output vectors for cases where no theoreticalmodel works satisfactorily. An artificial neural network (ANN) [8-10] is an information-processing paradigm inspired by the manner in which the heavily interconnected, parallel structure of the human brain processes information.

They are collections of mathematical processing units that emulate some of the observed properties of biological nervous systems and draw on the analogies of adaptive biological learning. NNs are trainable systems whose learning abilities, tolerance to uncertainty and noise, and generalization capabilities are derived from their distributed network structure and knowledge representation. Learning of a NN typically implies adjustments of connection weights and biases so that the square error (between NN output and desired output) is minimized. However, NN is often called a black box, since, unlike fuzzy logic, it is difficult to interpret the knowledge stored by a NN. Knowledge in a NN is represented in the values of the weights and biases, which forms part of large and distributed network.

PROBLEM FORMULATION AND APPROACH

The control problem considered in this paper is to move an inverted pendulum from bottom down position (ϴ = -900) to a bottom up (ϴ = 900) position (see Figure 1), and control the pendulum to stabilize it in the inverted position. Fuzzy logic approach has been considered in this paper. The inputs to the fuzzy logic are error in position (the difference in desired angular position and current angular position) and error in velocity (difference between desired angular velocity and current angular velocity). The output of the controller is the torque applied at the joint. Each of the inputs and outputs has three membership functions. The fuzzy logic has been designed in three different ways: 1) From operator's expert knowledge based on iterative learning, 2) Genetic-Fuzzy
Approach, and 3) Neuro-Fuzzy Approach.

DESIGN OF FUZZY LOGIC CONTROLLER:

A Fuzzy Logic Controller is made of three components:input/output interface, knowledge or rule base, and reasoning/inference mechanism. *Input interface* consists of fuzzification unit which converts the inputs to the controller into membership grades of fuzzy sets with the help of membership functions. *Output interface* consists of defuzzification unit which converts membership grades of outputs into a crisp number. *Knowledge or rule base* comprises of a data base and fuzzy control rule base which characterizes the desired output response applied by means of a set of control rules. Fuzzy rules are linguistic type of IF-THEN statements involving fuzzy sets, fuzzy logic, and fuzzy inference. Linguistic rules describing the control system consist of two parts: an antecedent or situation block and a consequent or action block. They are usually of the form:

IF X_1 is A^i_1 AND X_2 is A^i_2 AND… X_n is A^i_n THEN Y_1 is B^i_1.

Where *X1… X2* are inputs; *Y1* is the output and *Ai n* is the input membership function and *Bi 1* is the output membership function.

The *Reasoning/Inference Mechanism* is the kernel of Fuzzy Logic Controller which has the capacity of simulating the human decision making mechanism based on fuzzy concepts and fuzzy control actions.

Design of a fuzzy logic controller is accompanied with certain problems regarding design of membership functions (type and number of membership functions, their shape and range etc.), and choosing appropriate fuzzy rules. Frequently, designing a fuzzy controller requires a number of trial and error iterations, and even then, it is very difficult to ensure that the designed controller is an optimal one. Learning how to construct an efficient fuzzy controller is, to a large extent, more of an art than a science. The rule base is an important, component of a fuzzy controller that captures the operator knowledge about the system in the form of fuzzy rules. Developing a rule base is one of the most time consuming part of designing a fuzzy logic controller. Usually it is very difficult to transform human knowledge and experience into a rule base of fuzzy logic controller. Moreover there is a need for developing efficient methods to tune membership functions i.e., to obtain optimal shapes, ranges and number of member functions etc. The following section discusses some of the approaches based on genetic algorithm and neural network that deal with these issues.

GENETIC-FUZZY APPROACH

The problems discussed above motivated
many researchers to devise algorithms and strategies for automatic and online rule learning along with methods to tune the membership functions. Genetic Algorithms, because of their robustness and ability to provide global solutions, have been used as a tool by a number of researchers [11-12] to identify parameters of fuzzy logic controller. Since GAs work on coding of the parameter set, and not on the derivative of a function, they are capable of solving a vast range of optimization problems including optimization of the rule set of a fuzzy logic controller. This paper uses GA to optimize parameters of domain knowledge which consists of parameters of

membership function (such as mean and variance for bell shaped membership function), and the rule base.

The first issue that arises in a GA optimization is coding of the parameter set. There are several ways to encode the parameter set for optimizing fuzzy logic. For example, both rule base as well as the membership function parameters can be encoded in one GA representation. Similarly, one could use different representations for membership functions and rule base. In this paper, one single GA chromosome represents both the parameters of membership functions for inputs/outputs as well as rule base. A bell shaped membership function is characterized by mean (μ) and variance (σ). For the control problem investigated in this paper, there are two inputs: the error and the rate of change of error, and one output: torque.

There are three membership functions (fuzzy sets) for each of these variables. Hence, there are a total of nine membership functions, and eighteen parameters (two for each membership function). For three input membership functions for each of two variables, there are a total of nine distinct rules possible. Hence, in a single representation (see Figure 2), the total number of, and parameters that a single chromosome would encode is twenty seven.

While the membership function parameters can take on real values, the parameters for rule base can take integer values of one, two or three. These values represent the consequence (one of the three output membership functions) of a given rule.

Another issue that affects the performance of a GA is the objective function or the way performance index (PI) has been defined. The direction of GA search depends on the definition of PI. Usually, for a lumped parameter system, parameters such rise time, control effort, overshoot, steady state error etc are incorporated in an objective function. In this paper, sum of squared error (between current position and final desired position) for a simulation period of 10 seconds has been taken as PI. The sample time of the simulation is 0.01 second.

$$PI = \sum_{t=0}^{10} \left(\theta_{des} - \theta(t)\right)^2 \qquad (1)$$

Minimization of this PI ensures that the system reaches its final state quickly (rise time is low) as well as steady state error is small. A simulation period of 10 seconds has been chosen based on the fact that an unoptimized fuzzy controller is able to drive the manipulator to its final position in about 4 seconds.

NEURO-FUZZY APPROACHES

Both Neural Network and Fuzzy Logic [13] are model-free estimators and share the common ability to deal with uncertainties and noise. Both of them encode the information in a parallel and distributed architecture in a numerical framework. Hence it is possible to convert fuzzy logic architecture to a neural network and vice-versa. This makes it possible to combine the advantages of neural network and fuzzy logic. A network obtained this way could use excellent training algorithms that neural networks have at their disposal to obtain the parameters that would not have been possible in fuzzy logic architecture. Moreover, the network obtained this Way would not remain a black box, since this network would have fuzzy logic capabilities to interpret in terms of linguistic variables.

A number of algorithms have been developed that address this problem of learning fuzzy rules and tuning membership function in a neural network architecture. ANFIS (Adaptive- Network Based Fuzzy Inference System) developed by Jang [14], is one of the pioneering works in this field. ANFIS is a fuzzy inference system developed within the framework of adaptive network (which is a superset of all kinds of feed forward neural networks with supervised learning capabilities). The learning rule proposed for this method is basically a hybrid of the gradient-descent method and the least square technique, implementable both off-line (Batch Learning) and online (Pattern Learning). This approach, based upon a gradient descent method, implements Sugeno like fuzzy system which uses differentiable functions.Subsequent to the development of ANFIS approach, a number of methods have been proposed for learning rules and for obtaining an optimal set of rules. For example, Mascioli et al [15] have proposed to merge Min-Max and ANFIS model to obtain neuro-fuzzy network and determine optimal set of fuzzy rules. Jang and Mizutani [16] have presented application of Lavenberg-Marquardt method, which is essentially a nonlinear least- squares technique, for learning in ANFIS network. In another paper, Jang [17] has presented a scheme for input selection. Jana et al [18] have presented a six-layer network, called GeNFIS (Generalized Network-based Fuzzy Inference System), based on ANFIS. The ANFIS approach has been used in a number of applications. For example, Niestroy [19] has used an ANFIS structure to approximate an optimal feedback controller for the nontrivial problem of guiding a high speed vehicle to a ground target. Few of the other neuro-fuzzy approaches include NEFGEN [20], FDIMLP [21] and NEFCON [22]

In this paper, a multilayer feed forward connectionist model [23-24] is used to learn the fuzzy logic parameters and rule base. The model learns the fuzzy logic rules and outputmembership function parameters. The input membership function parameters have been assumed to be same as unoptimized fuzzy controller in this paper. Alternatively, the input membership function parameters can be learnt from self organizing algorithms such as Kohonen's map. In the following section, a detailed description of the neural network model and its relation to fuzzy logic counterpart is presented.

The Neuro-Fuzzy model (see Figure 3)
proposed in this paper has a total of four layers. For convenience, two inputs and one output have been considered. The layer one (input layer) has three sub layers. The nodes in sub layer 1 accept the input and feed to sub layer 2, which act as input membership function. There are n1 nodes (membership functions) for input 1 and n2 nodes (membership functions) for input 2. The nodes in Sub layers 3 represent the antecedent part of the rules. These nodes (total of n1 x n2) also act as the input nodes to the neural network. Layer 2 of the network represents the output membership functions and act as consequent of the rules. Layer 1 and layer 2 are fully connected, and they together represent the rule base. The weights of the links between these layers represent the firing strength of each rule. The two nodes in layer 3 are used for defuzzification purposes. These nodes together with the weights between layer 3 and layer 4 carry out the defuzzification and convey the crisp output to layer 4, which is the output layer. The neuro-fuzzy model can learn the fuzzy rules by adjusting the weights between layer 1 and 2 and identify output membership functionparameters by adjusting weights between layer 2 and 3, and layer 3 and 4.

The following notation has been used to
describe the function of the nodes in each layer:

net_i^L : net input value to the ith node in layer L.
net_i^{1L} : input value of the ith node in sublayer L of layer 1
out_i^L : output value of the ith node in layer L.
out_i^{1L} : output value of the ith node in sublayer L of layer 1.
μ_i^2 : the mean of the ith bell shaped output membership function.
σ_i^2 : the variance of the ith bell shaped output membership function.
W_{ij}^L : weight of the link that connects jth node in layer (L-1) to ith node in layer L
I_i : ith input to the network

The functions of nodes in each of the four layers are described below:

Layer 1:
Sublayer 1: The nodes in this layer transmit the input directly.
Hence

$$net_i^{11} = I_i \text{ and } out_i^{11} = net_i^{11} \quad (2)$$

Sublayer 2: The nodes in this layer represent membership functions and carry out the fuzzification process. n1 nodes in this layer are connected to node 1 of first sub layer and n2 nodes are connected to node 2 of first sub layer. Since bell shaped membership function is used here, the mathematical operations

Carried out in this layer are:

$$net_i^{12} = \begin{rcases} out_1^{11} \text{ for i = 1, 2, …, } n_1. \\ out_2^{11} \text{ for i = } n_1+1, \text{ …, } n_1+n_2. \end{rcases} \quad (3)$$

and

$$out_i^{12} = \exp - \left(\frac{net_i^{12} - \mu_i}{\sigma_i} \right)^2 \quad (4)$$

Where μi and σi are mean and variance of the with node (input membership function).

Sublayer 3: The nodes in this layer are antecedents of the rule. Each node has two input values from layer two (one from one of n1 sets and n2 sets). The output of this node is determined by fuzzy AND operation:

$$net_i^{13} = \min(out_j^{12}, out_k^{12}) \text{ j=1, .., } n_1 \text{ and k=} n_1+1, \text{ …, } n_1+n_2. \quad (5)$$

$$out_i^{13} = net_i^{13} \qquad \text{i=1,…, } n_1 \times n_2 \quad (6)$$

Output of the first layer, which acts as input layer of the neural network are:

$$out_i^1 = out_i^{13} \text{ for i=1, 2, …., } n_1 \times n_2. \quad (7)$$

Layer 2:

The layers of this node act as output membership function and perform fuzzy OR operation. Each of the link between this layer and layer 1 acts as a rule and the weight associated with it acts as strength of that rule. The functions of this layer can be written as:

$$net_i^2 = \sum_{j=1}^{n_1 \times n_2} W_{ij}^2 out_j^1 \quad (8)$$

$$out_i^2 = \min(1, net_i^2) \text{ for i=1, 2, … } n_3. \quad (9)$$

The weights in this layer can attain any value between 0 and 1.

Layer 3 and Layer 4:

The nodes in layer 3 along with the node in layer
4 and the links between them constitute the defuzzification by centroid method.
The mathematical operations done in nodes of
this layer are:

Node 1 of Layer 3:

$$net_1^3 = \sum_{i=1}^{n_3} W_{1i}^3 out_i^2 \quad (10)$$

$$out_1^3 = net_1^3 \quad (11)$$

Node 2 of Layer 3:

$$net_2^3 = \sum_{i=1}^{n_3} W_{2i}^3 out_i^2 \quad (12)$$

$$out_2^3 = \frac{1}{net_2^3} \quad (13)$$

Layer 4:

$$net_1^4 = \sum_{i=1}^{2} W_{1i}^4 out_i^3 \quad (14)$$

$$out_1^4 = net_1^4 \quad (15)$$

All of these mathematical operations simulate defuzzification by centroid method given by the following equation:

$$out_1^4 = \frac{\sum_{i=1}^{n_3} \mu_i^2 \sigma_i^2 out_i^2}{\sum_i^{n_3} \sigma_i^2 out_i^2} \qquad (16)$$

where μ_i^2 and σ_i^2 can be calculated from the link weights as:

$$\mu_i^2 \sigma_i^2 = W_{1i}^3 W_{11}^4 \qquad (17)$$

$$\sigma_i^2 = \frac{W_{2i}^3}{W_{12}^4} \qquad (18)$$

6The learning algorithm used is Back Propagation method which uses gradient descent method to minimize the error function:

$$E = \frac{1}{2}\left[output(t) - \hat{output}(t)\right]^2 \qquad (19)$$

Where *output (t)* is the output at current iteration step and (t) *output is* the desired output. The gradient descent method uses the following equation to update the weights and parameters:

$$w(t+1) = w(t) + \eta\left(-\frac{\partial E}{\partial w}\right) \qquad (20)$$

where η is the learning rate.

RESULTS AND DISCUSSIONS

The theory developed above is applied to a simple plant which consists of a single link pendulum. The objective of the controller is to stabilize the pendulum in an inverted upright position. The parameters of the system are assumed as follows:

Mass (m) = 1 kg; Length (l) = 0.5 m

Approach 1*: Fuzzy Logic Derived from Trial and Error Iteration*

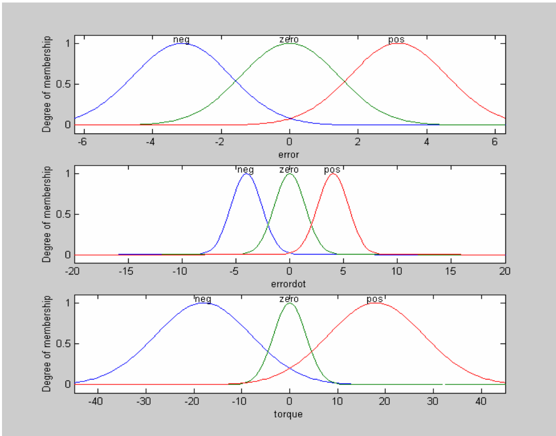

Figure 4 shows the input and output membership functions.

		Error in Velocity		
		neg	*zero*	*pos*
Error in Position	*neg*	neg	neg	zero
	zero	neg	zero	pos
	pos	zero	pos	pos

Table 1: The Fuzzy Control Rule Base (Approach 1)

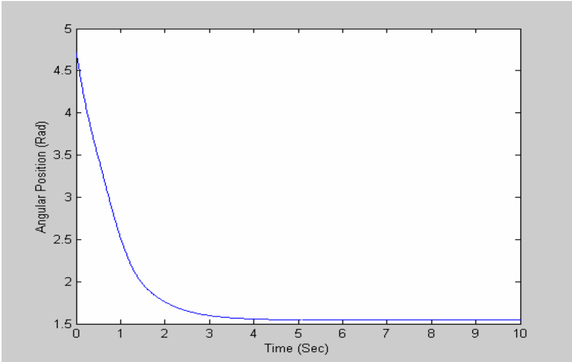

Figure 5 shows the angular position of the pendulum plotted against time. It can be seen that there is no overshoot and the system reaches the final position in approximately 4 seconds. Figure 6 shows the torque applied to the joint (controller output) plotted against time. The value of Performance Index (PI) is 444.10.

***Approach 2**: Genetic-Fuzzy Approach*

Figure 7 shows the input and output membership functions obtained from the GA optimization. The rules obtained by the algorithm are shown in Table 2.

		Error in Velocity		
Error in Position		*neg*	*zero*	*pos*
	neg	neg	neg	neg
	zero	neg	zero	pos
	pos	neg	pos	neg

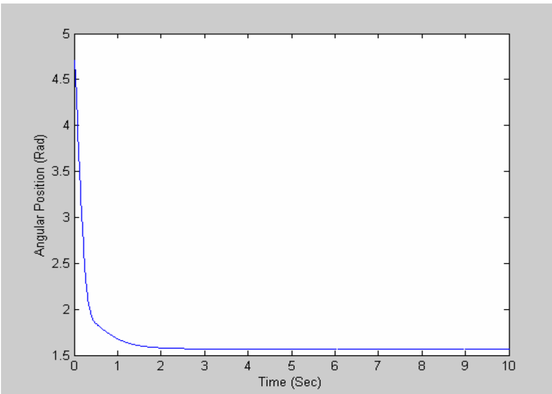

Figure 8 shows the angular position of the system plotted against time. There is no overshoot again, and system reaches the final position in approximately 2 seconds. The fast approach to final state is also associated with a very low PI value of 124.50. Figure 9 shows the controller output (torque) plotted against time. Oscillations in motor torque over the initial few seconds arise from a need to stabilize the pendulum.

***Approach 3**: Neuro-Fuzzy Approach*

The neural network has been trained from
2000 samples of input-output data obtained from a PD controller [25]. Figure 10 shows the input and output membership functions obtained from the neuro-fuzzy approach. The rules obtained by the algorithm are shown below. The values in parentheses after each rule represent the firing strength of the corresponding rule.

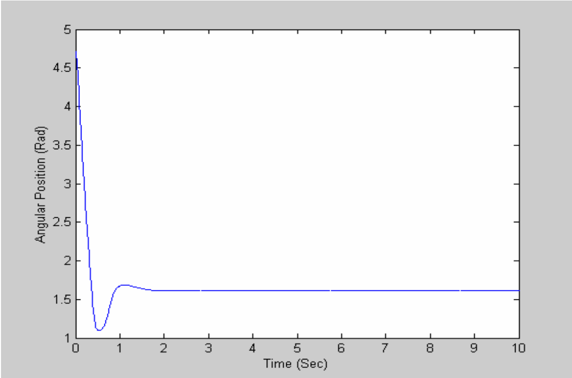

Figure 11 shows angular position of the pendulum plotted against time. The pendulum reaches the final position a bit earlier than 2 seconds, but has an overshoot. The value of PI obtained in this case is 137.98.

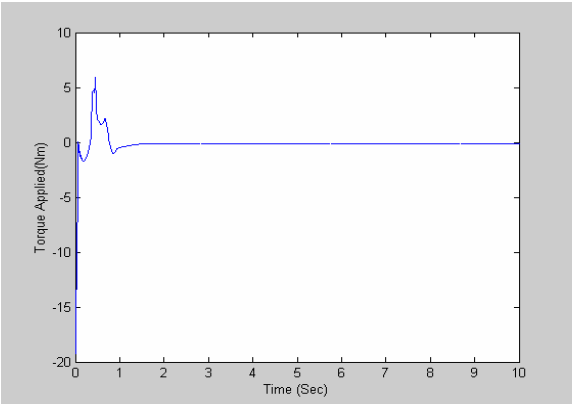

Figure 12 shows the controller output (torque) plotted against time. The rule base obtained from Genetic-Fuzzy approach is very much similar to that of first approach and can be easily interpreted. The only rule that seems to be erroneous is the last rule when error is pos and error dot is pos. This can be explained from the fact that during current simulation the system does not go in the region where this rule gets fired. Hence, GA has been unable to identify a correct consequent of this rule. The rule base obtained from Neuro-Fuzzy approach can also be interpreted in terms of linguistic relevance. The firing strengths of the rules show a trend that matches the rule base from first approach and from the Genetic-Fuzzy approach. From the results, it can be seen that the controller having the smallest PI value is the one obtained from the GA approach.

It can be expected that Genetic-Fuzzy approach has the smallest PI value, since this approach specifically carries out the minimization of PI. However, the system reaches the final position most quickly for the Neuro-Fuzzy case. The PI value of 137.98 obtained in this case is a drastic improvement over the first approach (PI=444.10) and is comparable to that of the Genetic-Fuzzy case. The graph of torque for Genetic-Fuzzy case shows undesirable oscillations, which is not present in other controllers. This oscillation can possibly be eliminated if control output term (torque) is also introduced in the objective function that GA minimizes. Another drawback of Genetic- Fuzzy approach over Neuro-Fuzzy approach is that the Genetic-Fuzzy approach takes a lot of time (a couple of hours) to converge, while the Neuro- Fuzzy approach takes only a few minutes to converge.

CONCLUSION

This paper presents a variety of methods to automatically learn the fuzzy logic controller parameters (e.g. rule base and membership functions). The methods presented are based on Genetic Algorithm and Neural Network. A study has been carried out to compare the performance of controllers developed via three separate methods. The controllers developed have been used to control an inverted pendulum. The results show that Genetic-Fuzzy and Neuro-Fuzzy controllers perform well. Genetic-Fuzzy controller was able to minimize the Performance Index (PI) and there was no overshoot. The

Neuro-Fuzzy approach brought the system to the final position in least amount of time, though it had slightly larger PI. The advantage of the Neuro-Fuzzy controller could be seen in its fast convergence as compared to the Genetic- Fuzzy approach. Both approaches were able to learn the rule base fairly accurately. Whereas the plant used for illustrating the relative performance of the three controllers is a simple one, the approach is equally applicable to more complex systems.

REFERENCES

1) Rocco, P., "Stability of PID Control for Industrial Robot Arms", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, Aug
1996, pp. 606 –614.

2) Luo, G, and Saridis, G., "L-Q Design of PID Controllers for Robot Arms", *IEEE Journal of Robotics and Automation*, Vol. 1, No. 3, Sep
1985, pp. 152 –159.

3) Zadeh, L. A, "Fuzzy Sets", *Information andControl*, Vol. 8, 1965, pp. 338-353.

4) Mamdani, E.H. and Assilian, A., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", *International Journal of Man Machine Studies*, Vol. 7, No. 1, 1975, pp. 1-13.

5) Takagi, T. and Sugeno, M., "Fuzzy Identification of Systems and its Application to Modeling and Control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, 1985.

6) Goldberg, D., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley Longman, Inc., 1989.

7) Miachalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*,Springer Verlag, 1996.

8) Bose, N.K., and Liang, P., "*Neural Network Fundamentals with Graphs, Algorithms and Applications*", McGraw Hill Inc., 1996.

9) Rumelhart, D.E. and McClelland, J., L.,*"Explorations in Parallel Distributed Processing: a Handbook of Models, Programs, andExercises*", MIT Press, 1988.

10) Kosko, B., "*Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*", Prentice Hall, 1992.

11) de Silva, C. W., "*Intelligent Machines: Myths and Realities*", CRC Press, 2000.

12) Teo, L.S., Bin Khalid, M., and Yusof, R., "Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm**",** *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol. 29, Issue 2, 1999, pp. 226 -236.

13)Yager, R. R. and Zadeh, L. A., "*Fuzzy Sets Neural Networks, and Soft Computing*", Van Nostrand Reinhold, 1994.

14)Jang, J.-S.R., "ANFIS: Adaptive-Network- based Fuzzy Inference System",*IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 3 , 1993, pp. 665 –685.

15) Mascioli, F.M., Varazi, G.M. and Martinelli, G., "Constructive Algorithm for Neuro-Fuzzy Networks", *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*,
1997, Vol. 1, July 1997, pp. 459 –464.

16)Jang, J.-S. R., and Mizutani, E., "Levenberg- Marquardt Method for ANFIS Learning", *Biennial Conference of the North American Fuzzy Information Processing Society*, June 1996, pp. 87 –91.

17) Jang, J.-S.R., "Input Selection for ANFIS Learning", *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, Vol.
2, Sep 1996, pp. 1493 –1499.

18) Jana, A., Yang, P.H., Auslander, D.M., and Dave, R.N., "Real Time Neuro-Fuzzy Control of aNonlinear Dynamic System", *Biennial Conference of the North American Fuzzy Information Processing Society*, June 1996, pp.210 –214.

19) Niestroy, M., "Optimal Controller Approximation using Neural and Fuzzy-Neural Networks", *Proceedings of thenFifth IEEE International Conference on Fuzzy Systems*, Vol. 1, Sep 1996, pp. 486 –491.

20) Rahmoun, A., and Berrani, S., "A Genetic- Based Neuro- Fuzzy Generator: NEFGEN", *Proceedings of ACS/IEEE International Conference on Computer Systems and Applications*, 2001, pp.18 – 23.

21) Bologna, G., "FDIMLP: A New Neuro-Fuzzy Model", *Proceedings of International Joint Conference on Neural Networks*, 2001. Vol. 2,
2001, pp. 1328 –1333.

22) Nürnberger, A., Nauck, D., and Kruse, R., "Neuro-Fuzzy Control Based on the NEFCON- Model: Recent Developments", *Soft Computing
2*, Vol. 4 1999, pp. 168- 182.

23) Lin, C.T. and Lee, C.S.G, "Neural-Network- Based Fuzzy Logic Control and Decision System", *IEEE Transactions on Computers*, Vol. 40, Issue 12, 1991, pp. 1320-1336.

24) Farag, W.A., Quintana, V., H. and Lambert- Torres, G., "A Genetic-Based Neuro-Fuzzy Approach for Modeling and Control of Dynamical Systems", *IEEE Transactions on Neural Networks*, Vol. 9, No. 5, 1998, pp. 756-
767.

25) Garg, D.P., and Kumar, M., "Genetic Algorithm based PD Control and Fuzzy Logic Control of a Two Link Robot", Paper No. IMECE2002-33433, *Proceedings of ASME International Mechanical Engineering Congress and Exposition*, 2002