EVOLUTIONARY DESIGN OF FUZZY LOGIC CONTROLLERS WITH THE TECHNIQUES ARTIFICIAL NEURAL NETWORK AND GENETIC ALGORITHM FOR CART-POLE PROBLEM

Reena Thakur[1], Vinay Kr Singh[2], Manu Pratap Singh[3]

[1]Department of Information Technology, AEC, Agra, India

[2]Department of Information Technology, AEC, Agra, India

[3]Department of Computer Science, ICIS, Dr. B.R. Ambedkar University, Khnadari Campus, Agra, India

([1]rina151174@rediffmail.com, [2]vksingh100@rediffmail.com, [3]manu_p_singh@hotmail.com)

Abstract- **This paper focuses on the Genetic Algorithm learning paradigm applied to train the ANNs for balancing the cart-pole balancing system. The studied system is a classic control problem namely "cart-pole" problem. We will apply the unconventional techniques Artificial Neural Network, Genetic Algorithm and Fuzzy Logic to a classic control problem "cart-pole". In this paper we have tried to train the Artificial Neural Network (ANN) with using Genetic Algorithms (program is written in MATLAB) which is compared with the output obtained using the Artificial Neural Network Toolbox provided in MATLAB.**

In proposed approach we have used both ANNs and Genetic Algorithm to get more optimal solution. Here we applied the approach for the Fuzzy logic technique to design a Fuzzy Logic Controllers (FLC) using ANNs and Genetic Algorithm (GA). The Fuzzy rules which are needed to control the problem will be framed with the combination of Artificial Neural Networks and Genetic Algorithm. It has been found that such a searching technique converges intelligently and much faster than conventional learning means. Performance of the presented neural network training using the genetic algorihtms is much better and providing more accurate results.

Keywords: **Artificial Neural Network, Genetic Algorithm, Fuzzy Logic Controllers.**

I. INTRODUCTION

The control of a cart-pole system is widely used as a benchmark problem for testing the efficiency of reinforcement learning algorithms [8]. Artificial Neural Networks can be trained to simulate the execution of the rule base of the Fuzzy Logic Controllers (FLC) using Genetic Algorithms(GA) for determining the solution of the cart-pole problem (program is written in MATLAB).The genetic design approach discussed in this paper offers a convenient and complete way to design a fuzzy controller in the shortest time. We are expecting better solution in terms of number of iterations, performance of the presented neural network training using the genetic algorihtms must be much better with more accurate results. Again in terms of requirement of fewer epochs for training the ANN using GA comparing to ANN toolbox provided in MATLAB.

We will describe two earlier applications of genetic algorithms to the automatic generation of a fuzzy rule base, and compare these with the ANN toolbox output to our approach. One of the earliest applications of genetic algorithms to the design of fuzzy systems was developed by Karr (1991), again to solve the cart-pole problem [2]. The production of the fuzzy controller begins with the definition of the fuzzy sets used to describe each input variable. Each of the four input variables are characterized by three fuzzy sets—NEGATIVE, ZERO, and POSITIVE—yielding 81 possible combinations. The fuzzy system designer then assigns one of seven choices for the output to each input combination. The resulting fuzzy system represents the expert's "best guess". The membership function extrema are then encoded into a bit string, and a genetic algorithm is applied to shift the membership functions so as to find locations which improve performance. The evolved system consistently outperforms the original, being capable of recovering from initial positions that fail under the original rule base.

In second application, Genetic algorithms for automatic design of fuzzy logic controllers have been developed [9], using sophisticated membership functions that intrinsically reflect the nonlinearity encountered in many engineering applications. But the sophistication obtained by the machine based automatic design could not be reached by manual design which is exclusively based on a painstaking trial-and-error process. As the number of variables increases the length of the string encoding the system size increases exponentially, with a corresponding exponential increase in the complexity of the search space. Such a system is unlikely to scale well for more complex problems.

We are using ANNs and Genetic Algorithm both to get more optimal solution. Here we applied the approach for the fuzzy logic technique to design a Fuzzy Logic Controller (FLC) using ANNs and Genetic Algorithm (GA). The resultant optimal fuzzy logic controller is used in centering a cart. Artificial Neural Networks can be trained to simulate the execution of the rule base of the Fuzzy Logic Controllers (FLC) using Genetic Algorithms. Also the training of artificial neural networks using genetic algorithms is extended to include a priori control knowledge of human operators in the form of rule base table. It is shown that the system can solve more concretely a fairly difficult control learning problem. It also demonstrated the feasibility of the method when applied to a cart-pole balancing problem. The performance of the GA and ANN Optimized Fuzzy Logic controller is compared with that of the conventional ANN

978-1-4244-5967-4/10/$26.00 ©2010 IEEE

toolbox. The MATLAB software forms part of the modeling and design tools employed in this paper.

Genetic Algorithms (GAs) [3] are heuristic optimization procedures based on the principles of natural evolution. They combine the concepts of adaptation and survival of the fittest to produce suboptimal solutions for an arbitrary problem. This is achieved by means of the repeated application of mating and reproduction operators onto a population representing potential solutions.

Neural networks so as fuzzy logic [6] are dealing with important aspects of knowledge representation, reasoning and learning, but in different approaches with their advantages and weaknesses. Neural networks can learn and classify information from the examples, but it is nearly impossible to describe knowledge acquired in that way. On the other hand, fuzzy logic which enables approximate reasoning [5] has a structural knowledge representation in the form of fuzzy if-then rules but lacks the adaptability to deal with changing external environments.

II. PROBLEM STATEMENT

In this paper we are applying the unconventional techniques Artificial Neural Network, Genetic Algorithm and Fuzzy Logic to a classic control problem variously referred to as the "cart-pole", "broom balancer ","inverted pendulum", "stick balancer problem" or "pole balancing" problem. The objective is to control translational forces that position a cart at the center of a finite width track while simultaneously balancing a pole hinged on the cart's top (figure 1.1)[8]. The movement of the cart is restricted to the horizontal axis by a track, and the pole is free to move about the horizontal axis of the pivot. The state of the system is defined by four real values; the angle of the pole , the angular velocity of the pole , the position of the cart relative to the centre of the track and the velocity of the cart .

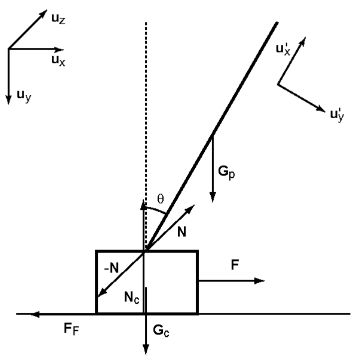

Fig 1.1: Cart-pole system

The output of the control system is a forward or backward movement for the cart as a fixed force. The cart and pole are initially placed at rest at a predetermined position. A simulation succeeds when sixty seconds of simulated time elapse without the cart reaching the end of the track or the pole falling over. For our purposes, we consider a pole angle of 0.26 radians (about 14degrees) the point at which the pole "falls over." Wieland (1991) investigates the effect of varying the angle at which a failure occurs on the process of evolving a neural network that controls the cart-pole system and concludes that increasing this parameter does not necessarily generate superior solutions, and makes training slower [7].

The cart moves either to the left or right while a pole attached to the cart is free to fall either in a clockwise or anticlockwise direction. The objective is to apply horizontal positive or negative forces to move the cart in order to maintain the pole in an upright position. The magnitude of the force F to be applied which depends on the angle θ and angular velocity ω is described by fuzzy control rules. The control of a cart-pole system is widely used as a benchmark problem for testing the efficiency of reinforcement learning algorithms.

III. IMPLEMENTATION AND SIMULATED DESIGN

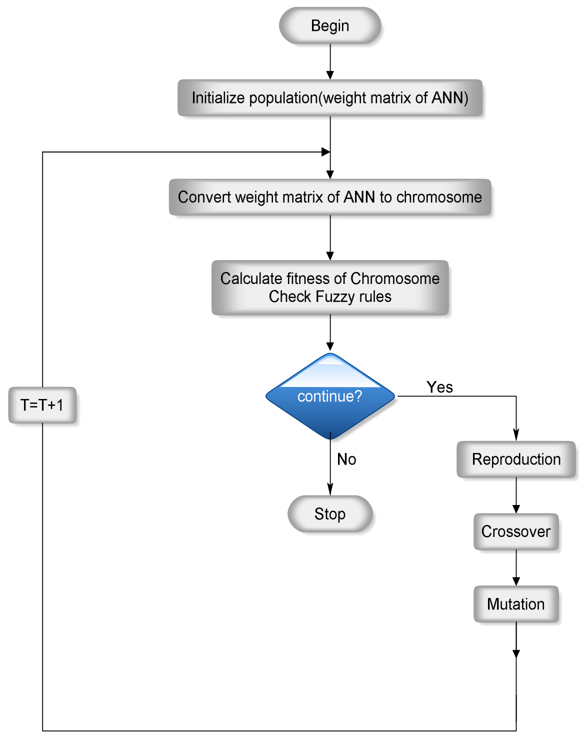

Fig 1.2: Flowchart of Implementation

Firstly the populations (weight matrix of ANN) are initialized then we have to convert the weight matrix of ANN to chromosome (string which consists of 49 rules). After converting, check the Fuzzy rules and calculate the fitness value of every individual in the solution if the best fitness value is found then reproduction is done by selecting two parents from the population .Then crossover is done between the two selected parents to generate the children. A two point crossover operator is employed in this study. The two point Crossover is a genetic operator that selects sub-section from two parent chromosomes and creates a new offspring chromosome. Then the contents of the sub-section from the two parent chromosome are exchanged. By recombining portions of good individuals, this process is likely to create even better individuals. Then the mutation is done, the best individual (the individual with minimum objective value) out of the children generated from the parent is found. The best individual then competes with its parent to survive in the next generation [1]. If the best individual is better than its parent, it is accepted as a parent in the next generation. If the best individual is worse than it's parent then the selection process is inspired and allows a bad move to be selected in the optimization process with a probability. Then the chromosome is converted to the weight matrix of ANN and the new population is used in the next iterations till the required output found.

A. *Fuzzy rule representation*

To produce fuzzy rules for a FLC with two inputs angle θ & angular velocity ω and single output F. Extension to a higher number of input and output variables is straightforward. As a first step, divide the domain regions of θ, ω and F into different regions. The number of the regions is application dependent. Let us assume that we divide the domain regions for θ, ω and F into 7, 7 and 7 regions respectively. For each region, we assign a fuzzy membership function. The shape of the membership functions selected is triangular.

The linguistic variables for θ, ω and F are NB, NM, NS, ZO, PS, PM and PH respectively.

NB – Negative Big
NM – Negative Medium
NS – Negative Small
ZO – Zero
PS – Positive Small
PM – Positive Medium
PB-Positive Big

There are 49 fuzzy rules and the fuzzy rule base can be formed as a 7x7 table with cells to hold the corresponding actions (outputs). A fuzzy rule is required for every possible condition that could exist in the physical system. The rules are commonly called production rules, and are of the form:

If Error is NB and Change-in-error is NB Then output is NB
If Error is NM and Change-in-error is NB Then output is NM
If Error is NS and Change-in-error is NB Then output is PS
If Error is ZO and Change-in-error is NB Then output is NB
If Error is PS and Change-in-error is NB Then output is ZO.

Although all possible conditions in the physical system seem imposing at first, the incorporation of fuzzy terms into the rules makes their development much easier.

B. *Chromosome representation*

Now encode the input and output regions into strings consisting of numbers from 0 to 6. The numbers 0 to 6 denote each of the possible value (NB, NM, NS, ZO, PS, PM and PB) that a variable can take.

A coded string of a fuzzy rule may look like:
0 0 2 0 0 0 0 1 2 0 1 5 4 5 6 1 3 2 2 4 6 0 1 2 3 4 5 6 3 4 6 4 4 2 4 5 3 1 5 6 5 2 6 6 6 6 4 4 6

This string is called a chromosome. Each complete string consists of 49 fuzzy rules and has the same input conditions but different output control signal assigned to it. Therefore, we need only to encode the output signal of the fuzzy rule strings into a complete string. A complete string may then be reduced to matrix form:

$$\begin{bmatrix} 0\,0\,2\,0\,0\,0\,0 \\ 1\,2\,0\,1\,5\,4\,5 \\ 6\,1\,3\,2\,2\,4\,6 \\ 0\,1\,2\,3\,4\,5\,6 \\ 3\,4\,6\,4\,4\,2\,4 \\ 5\,3\,1\,5\,6\,5\,2 \\ 6\,6\,6\,6\,4\,4\,6 \end{bmatrix}$$

GA initializes randomly a population of complete strings. Each of these strings is then decoded into fuzzy rules and evaluated by a FLC. Each string's fitness is defined as the error between the state of the system and the target set. GA procedures, selection and crossover then proceed according to the fitness values produced by the FLC for each string.

C. *Neural Network Architecture*

Depending upon the dimensions of the input and the target vectors, suitable feed forward neural network architecture is selected. Appropriate transfer functions for the hidden and output neuron layers are selected according to the problem at hand. The input training vector, target vector, error goal and maximum generations are also set according to the problem at hand.

The standard network architecture for learning the problem contain 2 input neurons, 49 neurons in the hidden layer and one output neurons shown in the figure below.

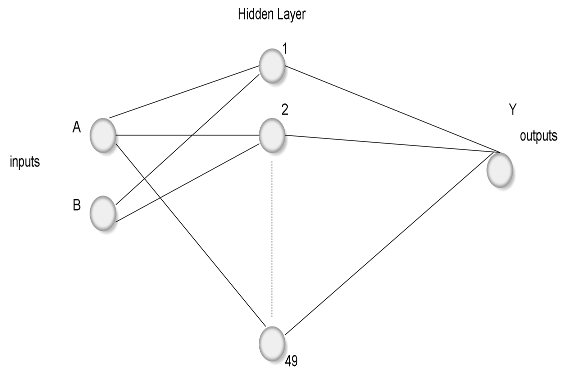

Fig 1.3: Architecture

Genetic Algorithm is a general heuristic that can be used to train an Artificial Neural Network (ANN) i.e. evolve the weights of an ANN.

IV. RESULTS

The fuzzy rule-set focuses 7x7 possible control actions corresponding to values in input error and change_in_error and therefore 49 bits are used in a string to form the look-up table, where a single bit represents each control action. This is illustrated in Figure 1.4 look-up table for control actions.

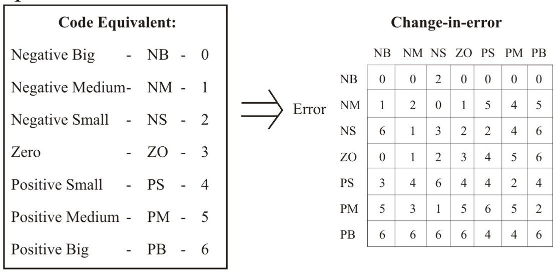

Fig 1.4: Fuzzy rule-set

The figure 1.4[4] maps two inputs to a single output as shown in the above figure, for Error =0 and Change_in_error=0 the output=0, similarly for Error =0 and Change_in_error=2 the output=2

The ANN is trained using Evolutionary Algorithm for FLC problem. The architecture used is 2-49-1 i.e. two inputs, 49 hidden neurons and one output neuron.

The ANN was trained for the input and target values as shown in the Table 1.1

TABLE 1.1
INPUT FOR TRAINING ANN FOR FLC PROBLEM

Inputs		Target	Output Obtained using Evolutionary Algorithm	Output Obtained using ANN Toolbox
0	0	0	0.002	-0.0474
0	1	0	0.0412	0.0772
0	2	2	1.9483	1.9609
0	3	0	0.0105	0.0127
0	4	0	0.0031	-0.0103
0	5	0	-0.0029	-0.0002
0	6	0	0.002	0.0026
1	0	1	1.0095	1.0066
1	1	2	1.9675	2.0043
1	2	0	0.0136	-0.0225
1	3	1	1.041	1.0215
1	4	5	4.9671	4.999
1	5	4	3.9906	3.9938
1	6	5	5.0045	4.9994
2	0	6	6.001	5.9904
2	1	1	1.0001	1.0298
2	2	3	2.9905	2.9718
2	3	2	1.9561	2.0002
2	4	2	2.0562	2.008
2	5	4	4.0065	4.0052
2	6	6	5.9823	5.9966
3	0	0	0.0109	-0.0023
3	1	1	0.9803	1.009
3	2	2	2.0294	2.0006
3	3	3	3.0116	2.9929
3	4	4	3.986	3.9864
3	5	5	4.9783	5.0078
3	6	6	6.0301	6.0119
4	0	3	2.9826	2.9899
4	1	4	4.0084	4.022
4	2	6	5.995	5.9736
4	3	4	3.9889	4.0272
4	4	4	4.0044	3.984
4	5	2	2.0191	2.0163
4	6	4	3.9743	3.9782
5	0	5	5.0175	5.0174
5	1	3	2.9889	2.9786
5	2	1	0.0006	1.0127
5	3	5	5.0085	4.9819
5	4	6	5.9881	6.0145
5	5	5	5.0057	4.9789
5	6	2	2.0059	2.0137
6	0	6	5.9923	6.0012
6	1	6	6.0016	6.0031
6	2	6	6.0041	5.9941
6	3	6	5.9996	6.0046
6	4	4	3.9949	3.9931
6	5	4	4.0104	4.0119
6	6	6	5.9925	5.9903

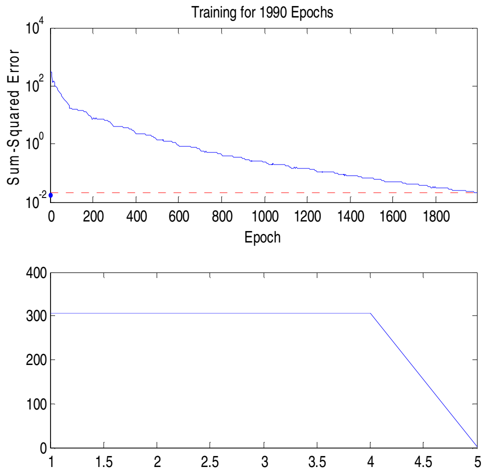

Fig 1.5: Training errors of ANNs for FLC Problem using GA

V. DISCUSSION OF RESULTS

GAs have been proven to be an extremely efficient and robust searching tool for complicated and poorly understood processes. It is also shown in the literature that such a searching technique converges intelligently and much faster than conventional learning means.

Performance of the presented neural network training using the genetic algorihtms is better. The proposed method in this implementation has the following advantages:

1. The artificial neural network is trained with fuzzy rules rather than standard real features values.
2. The weights of the neural network has been allocated as same as in artificial neural network system i.e. they are real in nature.

In this paper, we are trying to train the Artificial Neural network with using Genetic Algorithms (program is written in MATLAB) which is compared with the output obtained using the Artificial Neural Network Toolbox provided in MATLAB. Table 1.1 shows the comparision of the two outputs. First two colums show the inputs to the ANNs, 3rd column shows the output obtained from the ANNs trained using GAs and the 4th column shows the output obtained from the ANN toolbox provided in MATLAB. The two outputs are comparable and training of ANNs usung GA is much faster as compared to that using the toolbox. The parameters used for traing the ANNs using the GA are max_epoch=10000; err_goal=0.019; **epochs =1990** generations=5 Mutation probability = 0.01.

Where as the paremeters for training ANNs using toolbox are max_epoch=10000; err_goal=0.0199; **epochs =6243**. Fig 1.5 shows the decrease in error with epochs. The error becomes equal to the error goal in 1990 epochs.

VI. CONCLUSION

Genetic algorithms for training of Artificial Neural Networks for fuzzy logic controllers have been developed in this work, using sophisticated membership functions that intrinsically reflect the nonlinearity encountered in many engineering applications. Utilising these sophisticated membership functions, the control laws can be implemented in a simple scheme. The sophistication obtained by the machine based design could not be reached by manual design which is exclusively based on a painstaking trial-and-error process. The genetic design approach discussed in this dissertation offers a convenient and complete way to design a fuzzy controller in the shortest time.

Fuzzy systems are used to automated design of fuzzy logic controllers; by strengthening fuzzy logic controllers with genetic algorithms the implementation of optimal fuzzy logic rules and high-performance membership functions will be easier and faster. The resultant optimal fuzzy logic controller is used in centering a cart.

In this paper we have applied the unconventional techniques Artificial Neural Network, Genetic Algorithm and Fuzzy Logic to a classic control problem variously referred to as the "cart-pole", "inverted pendulum", or "pole balancing" problem, we found the following superior observations in terms of number of iterations, performance of the presented neural network training using the genetic algorihtms is much better and providing more accurate results. Again we require fewer epochs for training the ANN using GA as compared to ANN toolbox provided in MATLAB.

REFERENCES

[1] A.P. Wieland, “Evolving Controls for Unstable Systems," In S. Touretzky et. al. (ed.), Connectionist Models: Proceedings of the 1990 Summer School. San Mateo, CA: Morgan Kaufmann Publishers, Inc., pp. 91-102, 1991.

[2] C.L.Karr, "Design of a Cart-Pole Balancing Fuzzy Logic Controller Using a Genetic Algorithm," Proceedings of the SPIE Conference on the Applications of Artificial Intelligence .Bellingham, WA: SPIE—The International Society for Optical Engineering, pp. 26-36, 1991.

[3] J.H. Holland, “Adaptation in Natural and Artificial Systems,” University of Michigan Press, Ann Arbor, Michigan 1975.

[4] Kim Chwee Ng and Yun Li ,“Design of Sophisticated Fuzzy Logic Controllers Using Genetic Algorithms,” IEEE World Congress on Computational Intelligence(In Proc. 3rd IEEE Int. Conf. On Fuzzy Systems, Orlando, FL, June, 1994, vol.3, pp. 1708-1712.)

[5] LA Zadeh, “The concept of a linguistic variable and its application to approximate reasoning,” Parts 1, 2 & 3, Information Sciences, 1975.

[6] L. A. Zadeh,, "Fuzzy sets," Information and Control, vol. 8, pp. 338-353, 1965.

[7] Mark G. Cooper & Jacques J. Vidal, “Genetic Design of Fuzzy Controllers,” University of California, Los Angeles 3531 Boelter Hall Los Angeles, California 90024.

[8] Razvan V. Florian, “Correct equations for the dynamics of the cart-pole system,” Center for Cognitive and Neural Studies (Coneural) Str. Saturn 24, 400504 Cluj-Napoca, Romania. July 11, 2005; updated February 10, 2007

[9] Sheroz Khan, Salami Femi Abdulazeez, Lawal Wahab Adetunji, AHM Zahirul Alam, Momoh Jimoh E. Salami, Shihab Ahmed Hameed, Aisha Hasan Abdalla and Mohd Rafiqul Islam, “Design and Implementation of an Optimal Fuzzy Logic Controller Using Genetic Algorithm,” Journal of Computer Science 4 (10), pp. 799-806, 2008.