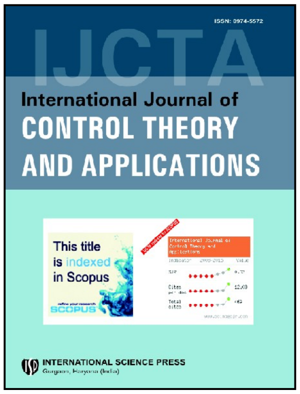

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 6 • 2017

An Evolutionary Approach involving Training of ANFIS with the help of Genetic Algorithm for PID Controller Tuning

Murali Dasari[1], A Srinivasula Reddy[2] and M Vijaya Kumar[3]

[1] *Department of Electrical and Electronics Engineering, Research scholoar , JNTU College of Engineering, Ananthpur, Andhra Pradesh, E-mail: murali_dvr@yahoo.co.in*
[2] *Department of Electrical and Electronics Engineering, Principal ,CMR Engineering College, Hyderabad, Telangana E-mail: svas_a@rediffmail.com*
[3] *Department of Electrical and Electronics Engineering, Professor, JNTU College of Engineering, Ananthpur, Andhra Pradesh E-mail: mvk_2004@rediffmail.com*

Abstract: ANFIS has good ability and performance in system identification, prediction and control and has been applied in many different systems. The ANFIS has the advantage of good applicability as it can be interpreted as local linearization modeling and conventional linear techniques for state estimation and control are directly applicable. The performance of ANFIS systems can be enhanced by improved training. In this paper a revolutionary approach involving Genetic Algorithm (GA) is employed for training the ANFIS. In order to validate the approach a simulink model having a PID controller is constructed. The results are measured in terms of the RMSE value. It can be observed from the results that GA-ANFIS has delivered a much improved performance when compared to traditional ANFIS.

Keywords: ANFIS, GA, PID, RMSE, Simulink

I. INTRODUCTION

ANFIS is a very effective modeling approach which combines the attributes of both the fuzzy inference system and neural network. The amalgamation of fuzzy logic with architectural design of neural network led to creation of neuro- fuzzy systems. These systems derive benefit from feed forward calculation of output coupled with back propagation learning capability of neural networks. While doing so, they keep the interpretability of fuzzy system unaltered [1]. The updating and training of ANFIS parameters that consist of the antecedent and conclusion parameters is one of the main issues encountered in ANFIS. The training of ANFIS, in the antecedent parameters is more difficult than the conclusion parameters. A multitude of methods have been used to optimize the fuzzy membership functions in the literature. These methods can be divided into two types including derivative based and heuristic algorithms in general [2]. Shoorehdeli *et al.* [3-7] proposed hybrid methods composed particle swarm optimization (PSO). He used recursive least square (RLS) and extended Kalman filter (EKL) for training. In different studies, they proposed factor recursive least square for training the conclusion parameters and Lyapunov stability theory to improve the performance of ANFIS. In addition to these, they used NSGA-II the training of all parameters of ANFIS structure. Similarly Zangeneh *et al.* [8] proposed a new type of training ANFIS is applying complex type (DE/current-tobest/ 1+1/bin & DE/rand/1/bin) on predicting of Mackeyglass time series.

Solving complex optimization problems have remained an active area of research and posed numerous challenges to researches. Most of the time, the complex real world optimization problems have multiple local solutions. Researchers have drawn inspiration from naturally occurring phenomena in solving these optimization problems. mimicking the behavior of natural systems (or) naturally occurring phenomena have given rise to multiple optimization approaches like Particle Swarm Optimization (PSO) [9] Ant Colony Optimization (ACO) [10] Genetic Algorithm (GA) [11] Bacterial Foraging Optimization Algorithm (BFOA) [12] Differential evolution (DE) [13] Immune Algorithm (IA), etc. These algorithms have adapted from naturally occurring process. They can be referred using different names with the names like Evolutionary Algorithms and metaheuristic approaches being commonly used.

The metaheuristic approaches typically combine heuristic algorithms which are usually problem specific in a more generalized frame work. So, metaheuristics can be considered as processes which strategies to find an optimum (or) a near optimum solution. These metaheuristic approaches are approximate and non-deterministic and they usually employ mechanisms to have a good convergence and provide near optimum solutions.

In this paper we have proposed GA-Trained ANFIS system. The performance of the GA-ANFIS system is compared with the performance of traditional ANFIS. The results of the optimization are presented and discussed. Similarly in order to validate the performance a Simulink model of a system with PID controller is built and tested for performance.

1.1. Adaptive Neuro Fuzzy Inference Systems (ANFIS)

Fuzzy logic as an idea proposed by Zadeh was first implemented by Madani in the year 1975. Madani demonstrated the idea of implementing the fuzzy logic as a concept for use in model steam engine. Subsequently many applications evolved using the concept of fuzzy logic. Different applications of fuzzy logic for industrial and home applications can be found in the literature. Two important factors namely, selection of knowledge techniques and availability of knowledge base influence the design of Fuzzy Logic Controllers. These two factors primarily influence the applications of Fuzzy logic. This can be overcome with the use of Adaptive Neuro -Fuzzy Inference System (ANFIS). An adaptive Neuro -Fuzzy Inference System (ANFIS) is a combination of an Artificial Neural Network (ANN) and a fuzzy inference system (FIS).ANN emulates the functioning of human brain and is formulated as collection of artificial neurons. An adaptive network has multiple layers of feed forward network. In this topology each node of the multilayer network executes specific functions on incoming signals. Each node has its own specific function. In the case of adaptive network two types of nodes namely adaptive and fixed nodes are present.

The overall function network determines the function and the grouping of the neurons. It is realistic to create a complete ANFIS structure with limited mathematical representation of the system. This is possible due to the learning ability of ANFIS and its ability to identify near optimal membership function for ensuring input-output mappings. It employs a combination of back prorogation gradient methods and least squares method for training a particular FIS membership function. When the errors are within the acceptable limits during training, the system is said to have convergence.

A simple two-input ANFIS architecture is illustrated in the figure 1. It is Sugeno fuzzy inference system architecture. In the case of Sugeno FIS the output membership functions are singleton spikes where as in the case Mamadani they are a distributed fuzzy set. The singleton output membership function simplifies the defuzzification step.

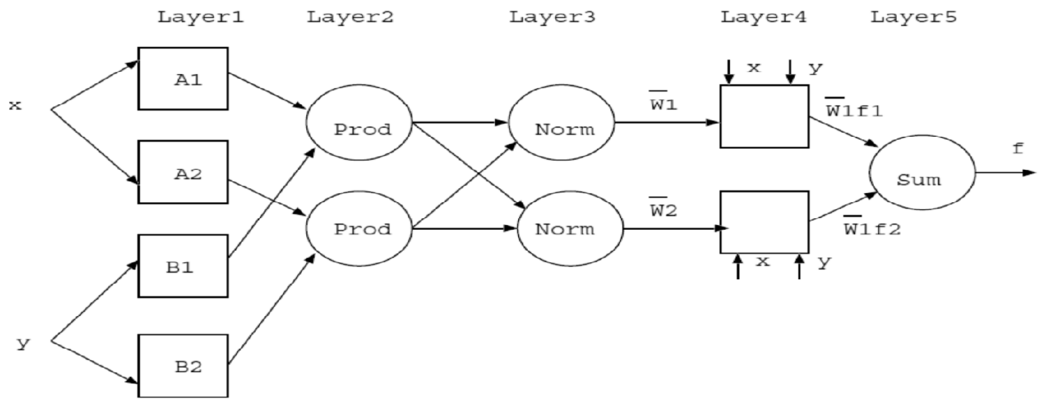

Figure 1: ANFIS Architecture

The ANIFS network comprises of five different layers with each node in the very first layer being an adaptive square node with a function. The structure comprises units and connections laid out as five connected layers.

Layer 1: This layer has the inputs variables and their representing membership functions through inputs. A triangular or a bell membership function can be used and this layer is used to supply the inputs values to the next layer.

Layer 2: This layer which forms the membership layer verifies for the weights of each MFs. It takes the input from the 1st layer and uses the MFs to depict fuzzy sets of the respective input variables. It also calculates the membership values and determines the degree to which a particular variable belongs to fuzzy set, which acts as the inputs to the next layer.

Layer 3: This is the rule layer where each node computes the activation level of each rule with the number of layers being equal to the total number of fuzzy rules in the rule base. In effect this layer performs the precondition matching of the fuzzy rules. The nodes also calculate the normalized weights for each of this layer.

Layer 4: Referred to as the defuzzification layer this layer delivers the resultant output by inferring the rules. Fuzzy singleton functions which connect the layers layer 3 and layer 4 through weights represent one set of parameters which form the basis for neuro fuzzy network.

Layer 5: The last layer is the output layer that adds all the inputs coming the preceding layer transforms the classification in to a crisp output. The structure is tuned by least square estimation and also at times using the back propagation algorithm.

1.2. Genetic Algorithm

Solving complex optimization problems have remained an active area of research and posed numerous challenges to researches. Most of the time, the complex real world optimization problems have multiple local solutions. Researchers have drawn inspiration from naturally occurring phenomena in solving these optimization problems. mimicking the behavior of natural systems (or) naturally occurring phenomena have given rise to multiple optimization approaches like Particle Swarm Optimization (PSO) , Ant Colony Optimization (ACO) , Genetic Algorithm (GA) , Bacterial Foraging Optimization Algorithm (BFOA), Differential evolution (DE) Immune Algorithm (IA), etc. These algorithms have adapted from naturally occurring process. They can be referred using different names like Evolutionary Algorithms and metaheuristic approaches. The metaheuristic approaches typically combine heuristic algorithms which are usually problem specific in a more generalized frame work. So, metaheuristics can be considered as processes which strategies to find an optimum (or) a near optimum solution. These metaheuristic approaches are approximate and non-deterministic and they usually employ mechanisms to have a good convergence and provide near optimum solutions. Metaheuristics are applications of domain specific knowledge permitting an abstract level description. These approaches typically employ three main processes; they are initialization of random population (Solution), evolution of fitness of the population and generation of new population.

Genetic algorithms are computerized search and optimization algorithms based on the process of natural selection and natural genetics. Even though the presence of genetic algorithm can be attributed to period much earlier than 1975, it was made popular with research community in the year 1975 by Holland. [11]. Genetic algorithms have adopted the concepts of natural selection and evolution. The prime idea behind genetic algorithm is to mimic the process of natural selection which is based on the concept of survival of fittest. In Genetic Algorithm, each solution is represented as a chromosome. These chromosomes are evaluated and ranked for the fitness. The flow chart of the genetic algorithm is given in the following figure.

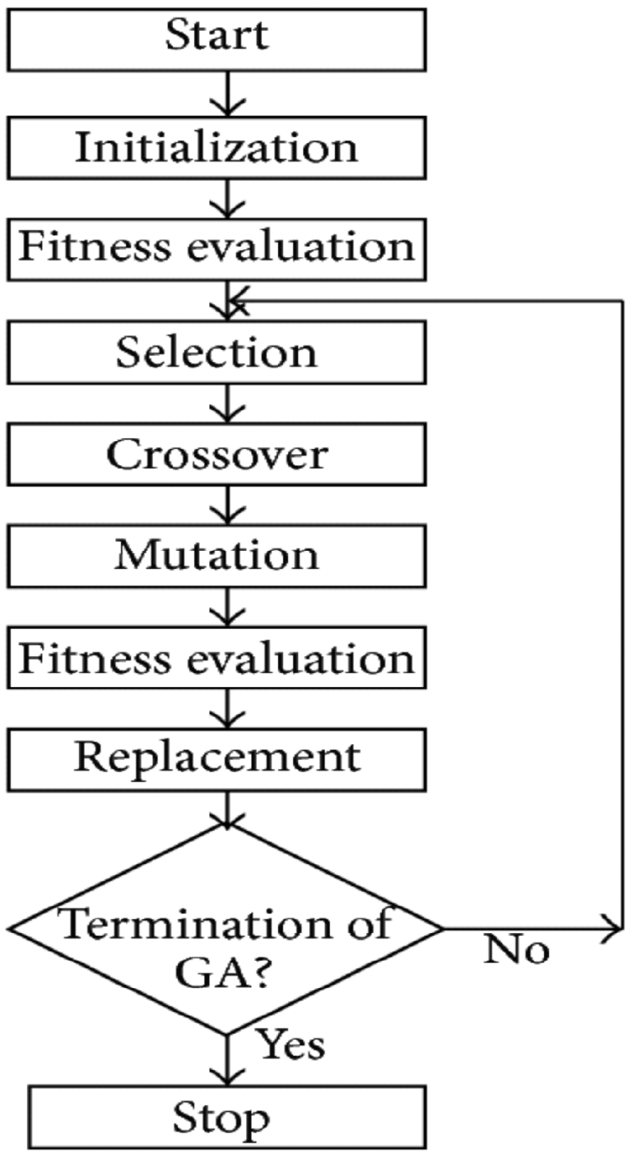

Figure 2: Flowchart for Genetic Algorithm

These initial chromosomes which are randomly populated are referred to as parent chromosomes and subsequent generations of chromosomes are referred to as child (or) offspring. The principle behind genetic algorithm is to involve better parents in the process of reproduction, so as to improve the chances of producing better offspring. Through this process of natural selection, the stronger chromosomes are carried forward to the next stage while the weaker chromosomes are eliminated.

In any optimization problem, these chromosomes represent the solution and that chromosome which is fittest is expressed to be the best result for that specific optimization problem. To aid in this process of natural selection, genetic algorithm employ three different operations, they are operators for selection, crossover and mutation. In the first stage of selection, the selection operator is used to select the better chromosomes. These chromosomes are evaluated in terms of their fitness function corresponding to that particular optimization scenario.

The individual solutions are then ranked in terms of their fitness value. Depending on the type of the optimization problem which is to either maximize (or) minimize a given objective the individual chromosomes are ranked. If the objective is to maximize then the chromosomes having the highest fitness value is ranked first. On the other hand, if the objective is to minimize a particular function then those chromosomes having the least fitness value is ranked first typically the top ranking chromosomes migrate to the next level and take part in the process of creating new generations.

By a mechanism called "Elite Count", the top ranking chromosomes usually migrate to the next stage of the process. Typically 10% of the population size is fixed to be the Elite Count. Assuming the population size is 20, the elite count is fixed at 2. There are many methods available in the literature that helps in the selection of chromosomes, these chromosomes using two methods namely roulette selection & Tournament based selection. Once the parent chromosomes are selected, a crossover operator is employed for the generation of new population.

The crossover operator typically combines two different parents to produce new chromosomes (solutions). Different types of crossover operators are available in the literature. The crossover probability defines the tendency to produce stronger individuals. As only stronger individuals are allowed to take part the process of new population generation, it may affect the diversity of the population and intern the diversity of solution itself.

In order to inject diversity into the solution and avoid getting trapped inside a local minima (or) maxima, the mutation operator is used. The mutation operator mutates individual chromosomes to produce new offspring. As these individual chromosomes are mutated to a specific mutation factor, it injects diversity into solution and at the same time provide good convergence, Different types of mutation operators are available and the choice of choosing a particular operator depends upon the type and need of the optimization problem.

These three operations like selection, mutation and crossover form the back bone for genetic algorithm. Apart from these three factors, other factors like population size, total number of iterations, maximum fitness value, plays a very crucial role in defining the results of optimization.

II. GA-ANFIS TRAINING

The parameters in this layer are called the antecedent parameters. The ANFIS has two types of parameters which need training, the antecedent part parameters and the conclusion part parameters. The membership functions are assumed Gaussian as in equation (1)

$$\mu_{A_I}(X) = \frac{1}{1+[\left(\frac{X-C_i}{a_i}\right)^2]b_i} \tag{1}$$

Where $\{a_i, b_i, c_i\}$ are the parameters of MFS which are affected in shape of MFs. a_i is the variance of the membership function, c_i the center of membership function and b_i is usually equal to 1.

There are 3 sets of trainable parameters in antecedent part; each of these parameters has N genes. Where, N represents the number of MFs. The conclusion parts parameters also are trained during optimization algorithm. Each chromosome in conclusion part has (I +1) ×R genes that R is equal to Number of rules and I denotes dimension of data inputs. The fitness is defined as Root Mean Square Error (RMSE). Parameters are initialized randomly in first step and then are being updated using GA algorithm. In each iteration, one of the parameters set are being updated. i.e. in first iteration for example a_i are updated then in second iteration b_i are updated and then after updating all parameters again the first parameter update is considered.

III. RESULTS AND DISCUSSION

The results of the simulation of ANFIS and a GA Trained ANFIS are discussed in this section. The results are compared in terms of the error and means square errors. It can be observed from the figure (3) and figure (4) there is a much closer correlation between the target and the output in the case of GA-ANFIS.

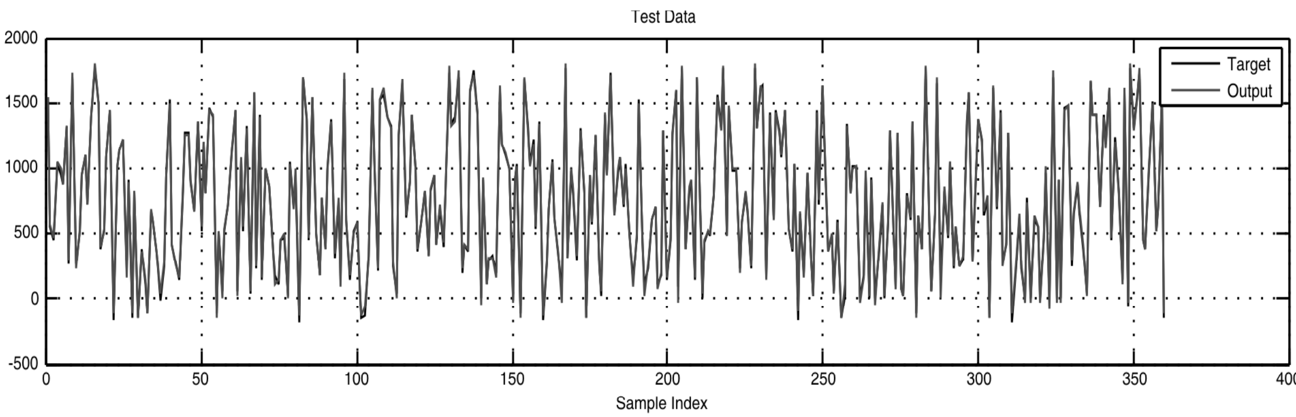

Figure 3: Plot between training and actual output for ANFIS

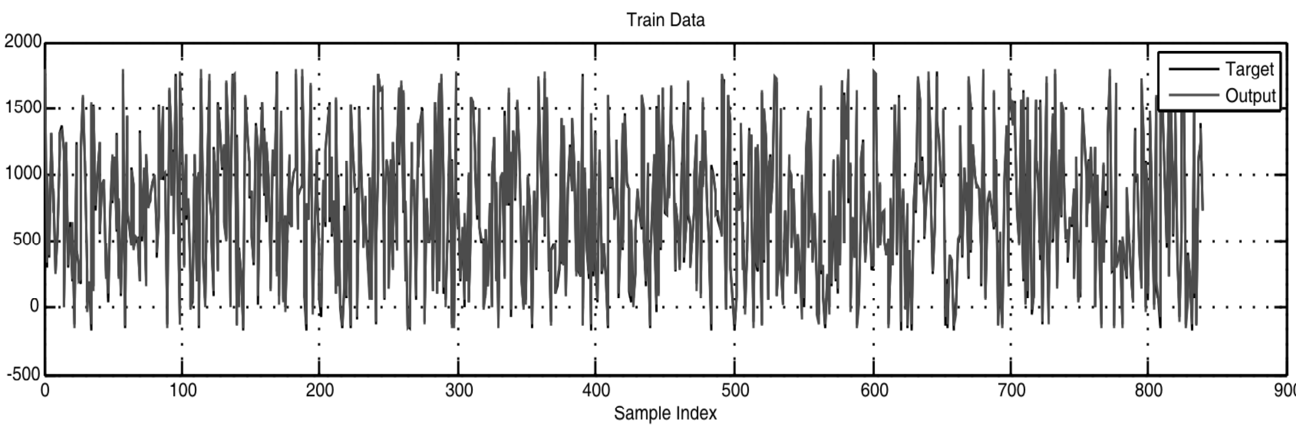

Figure 4: Plot between training and actual output for GA-ANFIS

This closer correlation points to increase accuracy of prediction and can lead to a much improved performance when predicting variables that vary very fast in real time or transient variables. Similarly

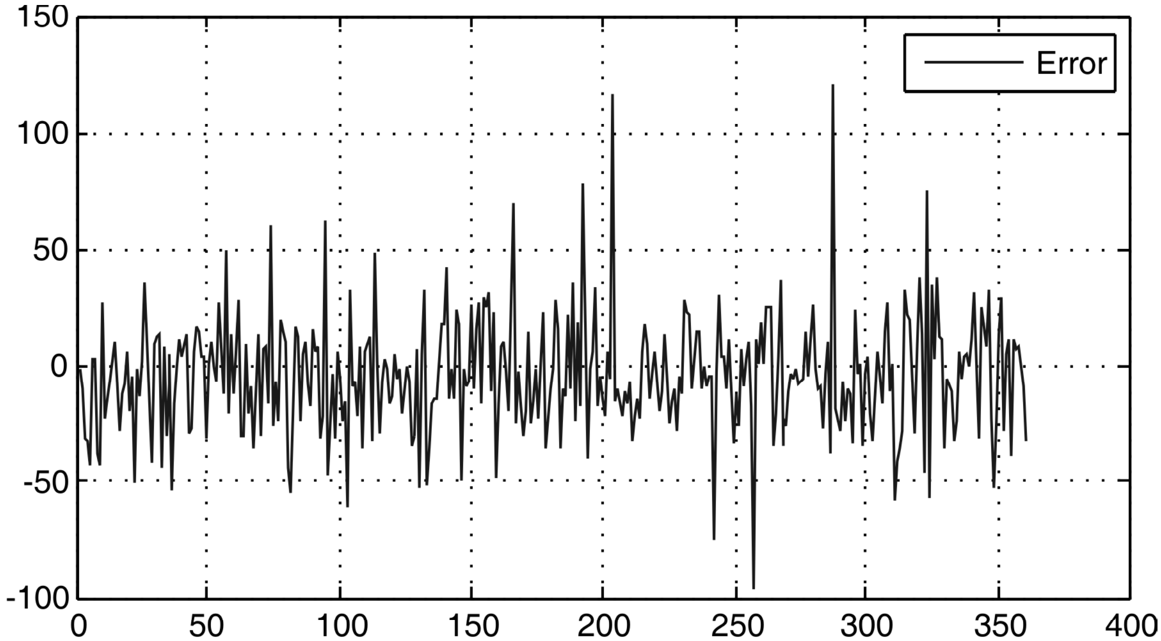

Figure 5: Plot of Error for different iterations for ANFIS

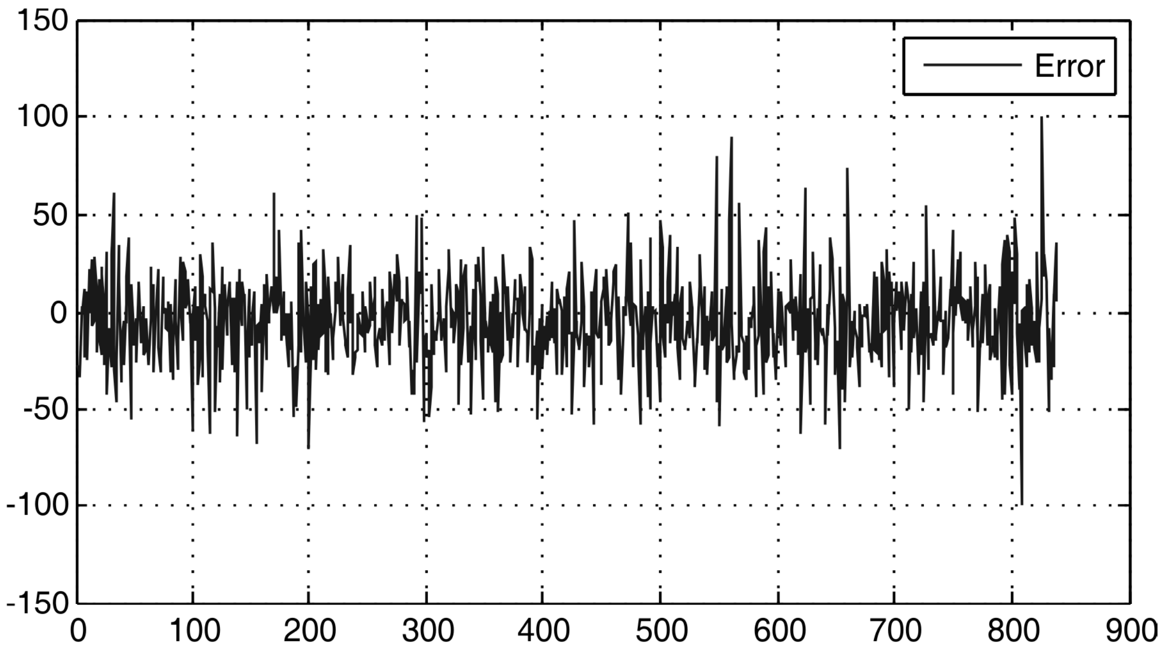

Figure 6: Plot of Error for different iterations for GA-ANFIS

Figure 5 and figure 6 provide the plot of error calculated during the course of iterations, it can be inferred from the figure the error produced by GA-ANFIS is less when compared to the error produced by original ANFIS. The MSE delivered by ANFIS stood at 655.316 where as the MSE produced by GA-ANFIS is 555.05. Similarly the better performance of GA-ANFIS is clearly visible in the case of another important performance parameter RMSE the better performance of GA-ANFIS is clearly visible. ANFIS returned a RMS error value of 25.59 while GA-ANFIS produced a reduced RMS error value of 23.45.

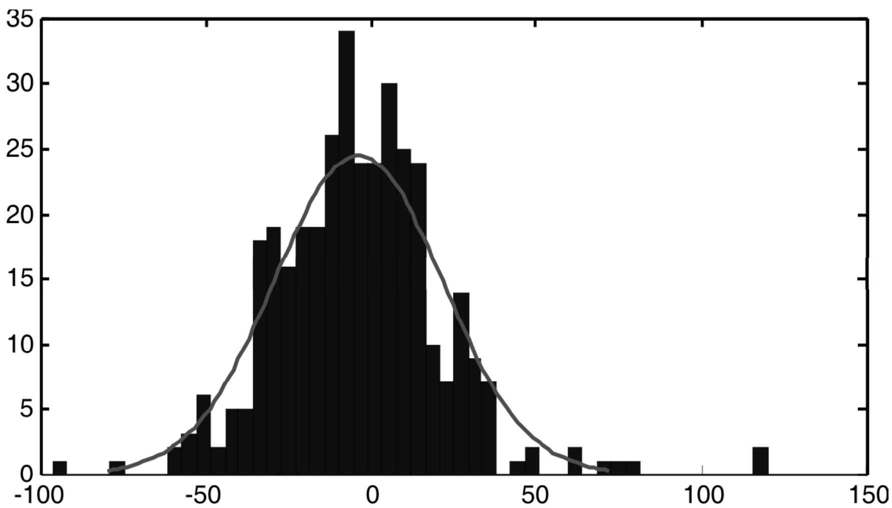

Figure 7: Distribution of Mean Error – ANFIS

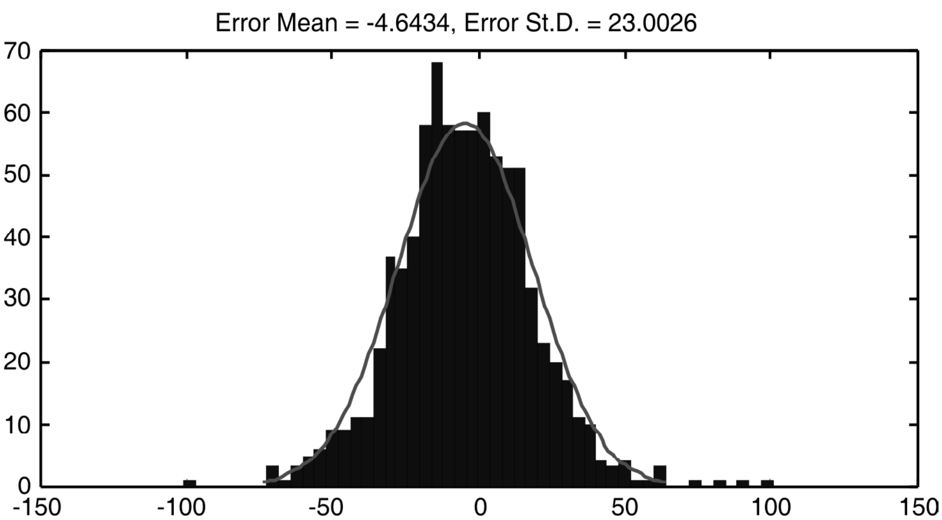

Figure 8: Distribution of Mean Error-GA-ANFIS

The distribution of mean error for the both the methods are illustrated with the help of figure 7 and figure 8. ANFIS produced a mean error of -3.8201 with a standard deviation of 25.34 while GA-ANFIS produced a mean error of 4.6434 and a standard deviation of 23.0026.

In order to test the performance of the proposed approach a test Simulink system is constructed. The system has a PID controller which is tuned by the ANFIS and GA-ANFIS approaches. The Simulink model of the system is given in the figure 9 and the performance plot of the controller settling time is given in the figure 10

It can be clearly observed from the figure 10, that the settling time and the peak overshoot of the GA-ANFIS tuned PID is much lesser when compared to the settling time and the overshoot experienced by the ANFIS tuned PID controller.

IV. CONCLUSIONS

ANFIS systems have traditionally delivered improved performances when compared to fuzzy based systems or neural system. This paper has proposed a method for improving the performance of the fuzzy system by exploiting

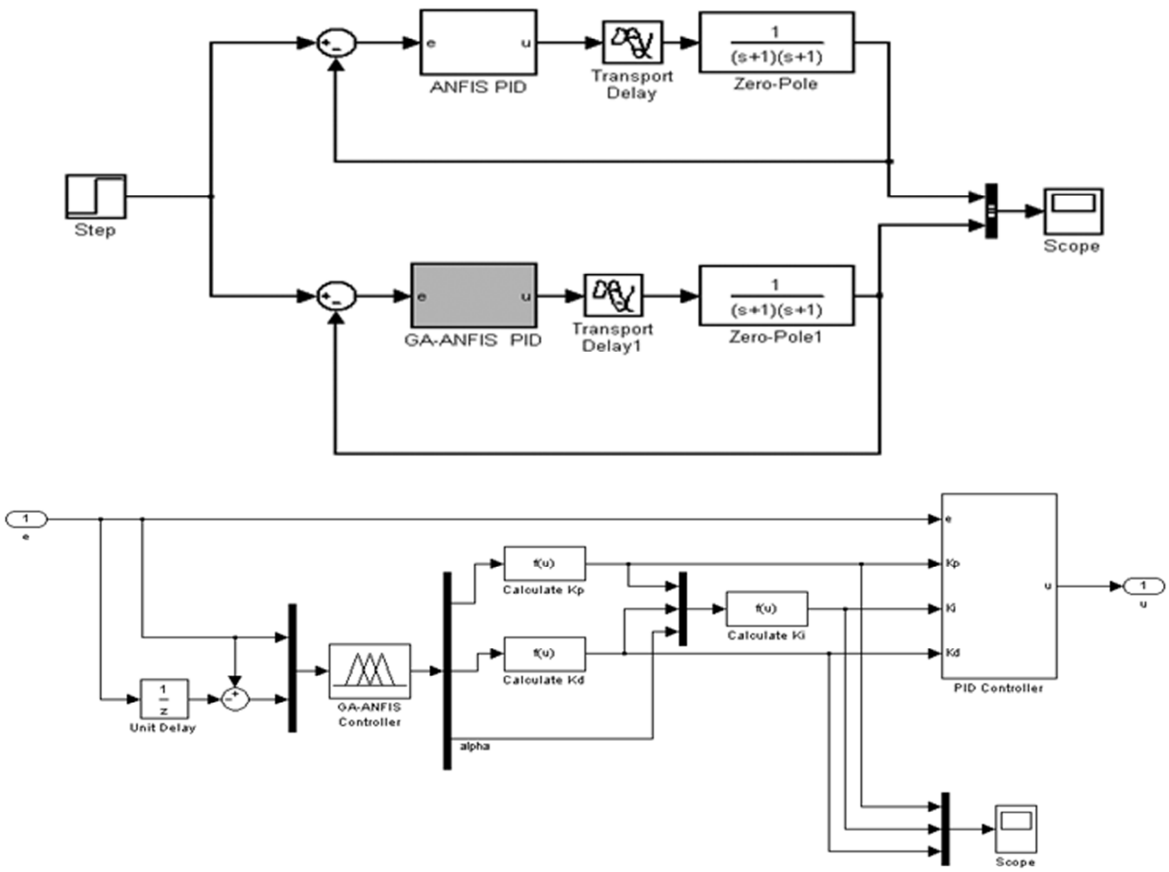

Figure 9: Simulink Block Diagram of a test system for ANFIS and GA-ANFIS PID Control

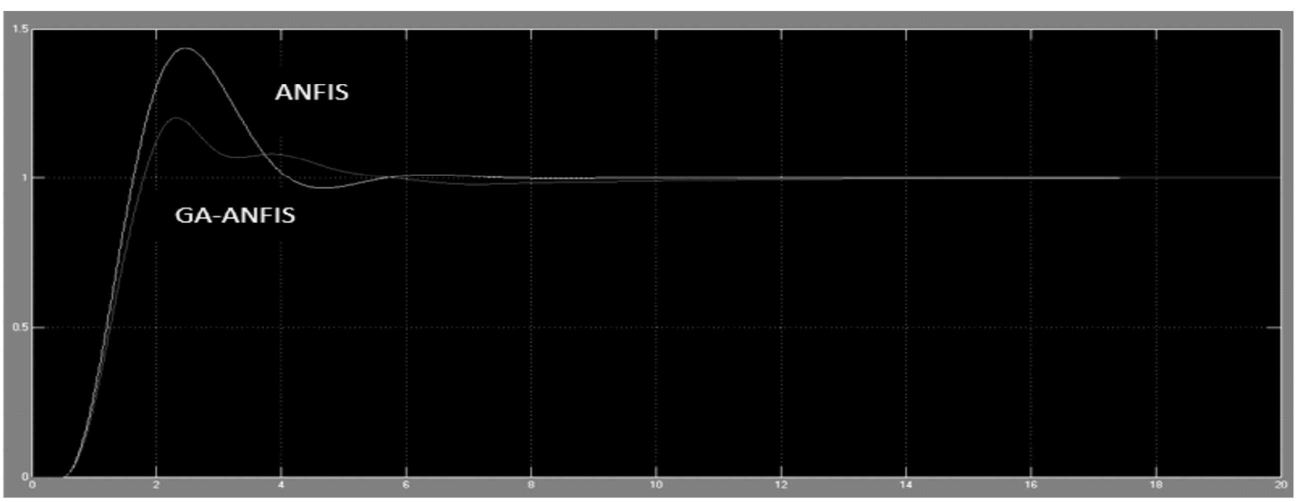

Figure 10: Performance comparison of ANFIS and GA-ANFIS

GA to train them. It can be clearly observed from the results that the results that the proposed system has delivered an improved performance when compared to traditional ANFIS systems. The validation of the proposed approach has also been successfully made with the help of a simulink model which has a PID controller in it. The results of tuning of the PID controller depict an improved performance in terms of reduced overshoots and setting time. This demonstrates the capability of the system and its ability to be employed in other real time applications like speed control of motors etc...

REFERENCES

[1] R. Alcala, J. Casillas, O. Cordon and F. Herrera, Learning TSK rule based system from approximate ones by mean of MAGUL methodology. Granada university of Spain,Oct.2000.

[2] A. Chatterjee and K. Watanabe, "An optimized Takagi- Sugeno type neuro-fuzzy system for modeling robot manipulators," Neural Computing & Applications, vol. 15(1), pp. 55-61, 2006.

[3] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter," Fuzzy Sets and Systems , vol. 160, pp. 922-948, 2009.

[4] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "A novel training algorithm in ANFIS structure," in American Control Conference, IEEE, June 2006.

[5] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "Identification using ANFIS with intelligent hybrid stable learning algorithm approaches," Neural computing & applications, vol. 18(2), pp. 157-174, 2009.

[6] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "Novel hybrid learning algorithms for tuning ANFIS parameters using adaptive weighted PSO," In Fuzzy Systems Conference, fuzzy-IEEE 2007, IEEE International , pp. 1-6, 2007.

[7] M. A. Shoorehdeli, M. Teshnehlab, A. K. Sedigh and M. A. Khanesar, "Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods," Applied Soft Computing, vol. 9(2), pp. 833-850, 2009.

[8] A. Z. Zangeneh, M. Mansouri, M. Teshnehlab and A. K. Sedigh," Training ANFIS system with DE algorithm," In Advanced Computational Intelligence (IWACI), IEEE 2011 Fourth International Workshop on, pp. 308-314, October 2011.

[9] Kennedy, J., Eberhart, R.:" Particle swarm optimization." In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995).

[10] Dorigo, M., Gambardella, L.M.: "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem" IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1999).

[11] Holland, J.H.: "Adaptation in Natural and Artificial Systems. University of Michigan Press", Ann Harbor (1975).

[12] Passino, K.M.: "Biomimicry of Bacterial Foraging for Distributed Optimization and Control". IEEE Control Systems Magazine, 52–67 (2002).

[13] Storn, R., Price, K.: "Differential evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". Journal of Global Optimization 11(4), 341–359 (1997).