International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 6, June - 2013

Intelligent Methods for Tuning of Different Controllers

Afshan Ilyas and Mohammad Ayyub

Department of Electrical Engineering
Zakir Hussain College of Engineering & Technology
Aligarh – Uttar Pradesh, India

Abstract: This paper presents different methods for tuning of controllers. Ziegler-Nichols tuning formula has been used for predicting the range of gain coefficients of PID controller. Recently, it has been noticed that PID controllers are often poorly tuned and some efforts have been made to systematically resolve this matter. Thus Fuzzy logic can be used in context to vary the parameters values during the transient response, in order to improve the step response performances. This paper also discusses the Fuzzy control as applied to various industrial processes. Normally the control rules and membership functions of Fuzzy control are obtained by trial and error. Genetic Algorithm technique is supposed to provide better tuning of the parameters and is applied for tuning the parameter of the PID & PI-Like Fuzzy controllers. Simulation analysis has been carried out for the different processes and the results shows that the optimized fuzzy controller gives better performance than a conventional PID and fuzzy logic controller.

1. Introduction

PID controllers are the most widely used type of controller for industrial applications. They are structurally simple and exhibit robust performance over a wide range of operating conditions. In the absence of the complete knowledge of the process these types of controllers are the most efficient of choices.

Many tuning formulae that have been devised such as the Ziegler-Nichols one, assures a good load-disturbance attenuation, but often fail to achieve satisfactory performances, and therefore the operator has to use their experience and might fail to attain the best performances [4].

In this context, the use of fuzzy logic seems to be particularly appropriate, since it allows us to make use of the operator's experience and therefore to add some sort of intelligence to the automatic control. Fuzzy logic controllers (FLC's) are used when the processes are too complex for analysis by conventional mathematical techniques.

But there are some difficulties that prevent the designing of fuzzy controllers from being systematic. First, the choice of the overall control structure is a problem faced by many designers. Second, in designing of the FLC, not only structure parameters of the FLC need to be designed, but also the gains of the conventional controller need to be tuned. Because of its complicated cross-effects, analytical tuning algorithms for these parameters are really difficult [7].

Genetic Algorithms provide a way of surmounting this shortcoming. These algorithms use some of the concepts of evolutionary theory, and provide an effective way of searching a large and complex solution space to give close to optimal solutions in much faster times than random trial-and-error. They are also generally more effective. The main objective of this work is to investigate the use of genetic algorithms in tuning of PID controller. The algorithm searches for the controller gains K_p (proportional gain), K_i (integral gain) and K_d (derivative or differential gain) so that specifications for the closed-loop step response are satisfied.

This paper is organized as follows. Section 2 reviews the ZN tuning formula in the context of tuning the PID controller. Section 3 presents the tuning procedure, based on fuzzy logic controller. Tuning of PID based on GA and GA-optimized PI-type FLC were present in section 4 and section 5 respectively. Conclusions follow in section 6.

2. TUNING OF PID CONTROLLER

2.1 Ziegler-Nichols Method

Proportional-Integral-Derivative (PID) controllers have been in existence for nearly two-thirds of a century. They remain a key component in industrial process control as over 90% of today's industrial processes are controlled by PID controllers. Due to its simplicity, versatility, speed, reliability, flexibility and robustness, many industries still rely on this stalwart controller for all types of control. Example includes temperature, engine speed and position control among many others. The PID controller is probably the most used feedback control design. PID controller has the general form

$$u(t) = K_p\, e(t) + K_i \int_0^t e(x).dx + K_d\, \frac{de(t)}{dt}$$

where e(t) = y_{sp}(t) - y(t) is the system error (difference between the reference input and the system output), u(t) the control variable, K_p the proportional gain, T_d the derivative time constant and T_i the integral time constant.

The tuning problem consists of determining the values of these three parameters with the aim of satisfying different control specifications such as set-point following, load disturbance attenuation, robustness to model uncertainties and rejection of measurement noise. The value of controller parameters like K_P, K_d and K_i are reached by mainly trial and error method. But this method is very time consuming. Therefore we switch over to different tuning techniques which give more accurate results with less time. Ziegler-Nichols method is generally used for the purpose in which the parameters like ultimate gain K_u and ultimate period T_u is first calculated by Routh array criteria [10], and then K_p, K_i and K_d are calculated as shown below.

Table 1: Ziegler - Nichols tuning formulas based on ultimate gain (K_u) and ultimate period (T_u)

Controller	Gain (K_P)	Integral time(T_i)	Derivative time(T_d)
P	0.5K_u	-	-
PI	0.45K_u	0.8T_u	-
PID	0.6K_u	0.5T_u	0.125T_u

2.2 Simulation Result

The performances of the different controllers have been evaluated on different plants. Here, the following transfer functions, with different values of the parameters, are considered:

$$G_1(s) = \frac{1}{s(1+s)} \quad (1)$$

$$G_2(s) = \frac{e^{-0.1s}}{(1+10s)^2} \quad (2)$$

$$G_3(s) = \frac{e^{-0.4s}}{(1+10s)^2} \quad (3)$$

$$G_4(s) = \frac{1}{(1+s)^3} \quad (4)$$

$$G_5(s) = \frac{(1-0.5s)}{(1+s)^3} \quad (5)$$

$$G_6(s) = \frac{1}{(1+s)(1+0.5s)(1+0.25s)(1+0.125s)} \quad (6)$$

$$G_7(s) = \frac{e^{-0.1s}}{(1+s)(1+0.5s)(1+0.25s)(1+0.125s)} \quad (7)$$

After the tuning phase, accomplished the unit step responses have been simulated with Matlab and Simuink. The step responses for plants described by transfer function of equation (5) and (7) are shown in Fig.1 and Fig.2.

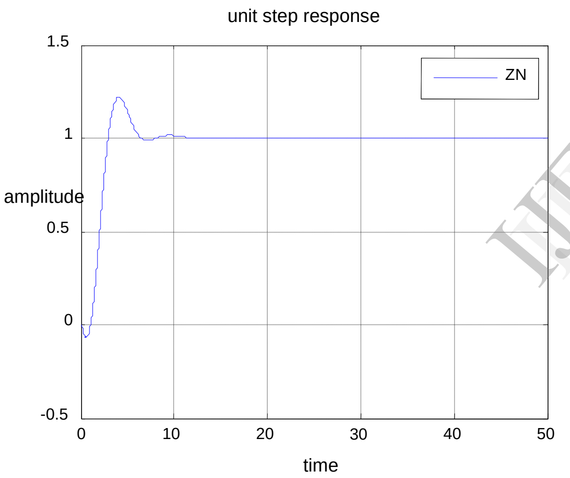

Fig.1: Step Response of $G_5(s)$ with Ziegler-Nichols Controller

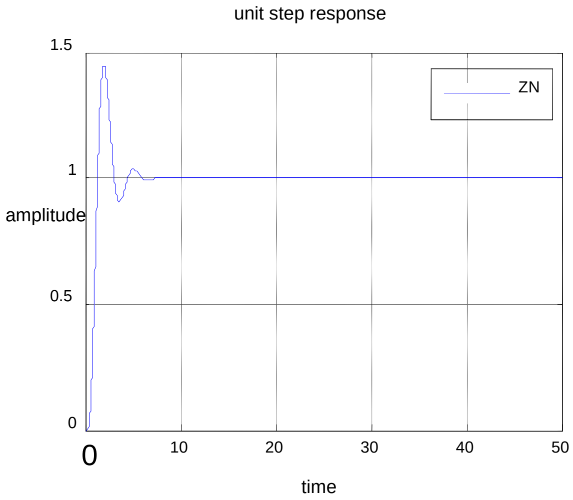

Fig. 2: Step Response of $G_7(s)$ with Ziegler-Nichols controller

3. DESIGN AND TUNING OF FUZZY LOGIC CONTROLLER

Fuzzy logic controllers (FLCs) are increasingly applied to many systems with nonlinearity and uncertainty and it is based on experience of a human operator. While controlling a plant a skilled human operator manipulates the output of the controller based on error and change in error with an aim to reduce the error with a shortest possible time.

The two types of structure of FLC have been studied so far: one is position-type fuzzy controller which generates control input (u) from error (e) and change in error, and the other is velocity-type fuzzy controller which generates incremental control input (Δu) from error and change in error. The former is called PD type FLC and the latter is called PI type FLC according to the characteristics of information that they process. In the viewpoint that the FLC is based on the knowledge of human experts, and generally FLC's applied to unknown or partially known systems, PI type FLC is known to be more practical than PD type FLC [9].

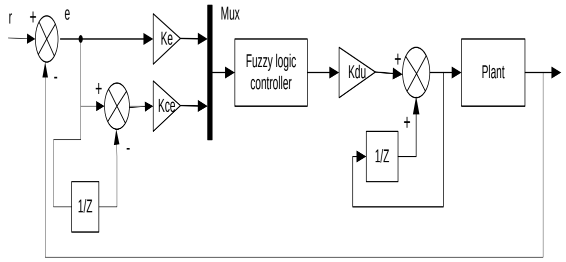

Fig 3: PI type fuzzy logic controller

One of the well accepted rule base is the linear rule base which appears in many research work and applications. As the rule base conveys a general control policy, it should be sustained and leaves most of design and tuning work to the scaling gains.

An input SF transforms a crisp input into a normalized input in order to keep its value within the universe. An output SF provides a transformation of the defuzzified crisp output from the normalized universe of the controller output into an actual physical output [9]. There is always an input limitation for FLC, so that conventional inputs i.e. e and ce and the controller output (du) are defined on the common normalized domain [-1,1]. The set of rules which define the relation between the input and output of fuzzy controller can be found using the available knowledge in the area of designing the system. These rules are defined using the linguistic variables. All the 25 rules governing the mechanism for each output are explained in Table 2.

Table 2: Basic rules table for fuzzy inference system

ce / e	NB	NS	Z	PS	PB
NB	NVB	NB	NM	NS	Z
NS	NB	NM	NS	Z	PS
Z	NM	NS	Z	PS	PM
PS	NS	Z	PS	PM	PB
PB	Z	PS	PM	PB	PVB

3.1 Simulation Results

Simulation of the example systems was being carried with the conventional controllers was being replaced by Fuzzy logic Controller (FLC). FLC was implemented with the required block available in MATLAB/SIMULINK. Mamdani type of rule-base model is used for this FIS system. This produces output in fuzzified form. It is having five input membership functions for both input variables leading to 25 rules. Input and output scaling factors are tuned manually to get the desired response. It is clear from the figures that overshoot reduces significantly with a fuzzy controller but the values of other performance parameters like IAE, rise-time and settling-time increases as compare to that of conventional controller.

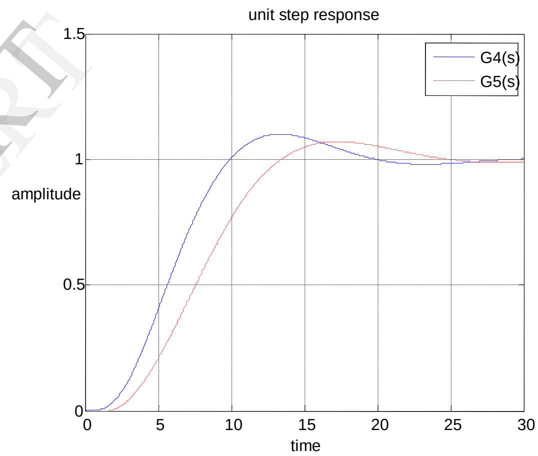

Fig 4: Step Responses of PI-type FLC of $G_4(s)$ and $G_5(s)$

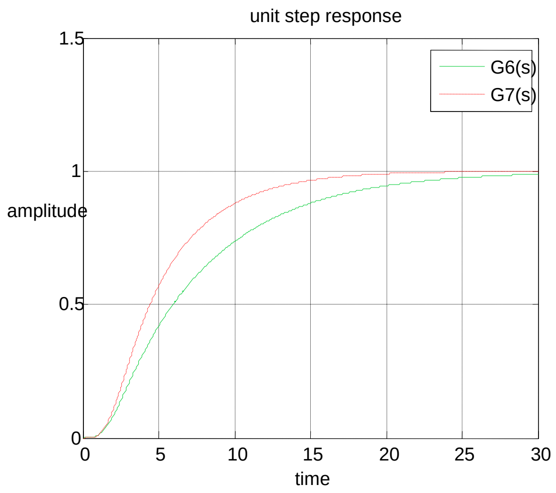

Fig 5: Step Responses of PI-type FLC of $G_6(s)$ and $G_7(s)$

4. DESIGNING OF PID CONTROLLER USING GENETIC ALGORITHM

Genetic Algorithm (GA) is a stochastic global adaptive search optimization technique based on the process of natural evolution. It is one of the methods used for optimization. John Holland formally introduced this method in the United States in the 1970 at the University of Michigan. The genetic algorithm starts with no knowledge of the correct solution and depends entirely on responses from its environment and evolution operators such as reproduction, crossover and mutation to arrive at the best solution [9].

The basic goal of GA is to optimize functions called fitness functions. A possible solution to a specific problem is seen as an individual. A collection of a number of individuals is called a population [7]. Each solution can be represented by a binary string of ones and zeros, real number or other forms, depending on the application data. In these algorithms the fittest among a group of individuals survive and are used to form new generations of individuals with improved fitness vales. The fitness of an individual is a measure of how well the individual has performed in the problem domain [8].

The structure of the control system with GA-PID controller is shown in Fig 5. It consists of a conventional PID controller with auto-tuning its gain coefficients based on GA and a control plant.

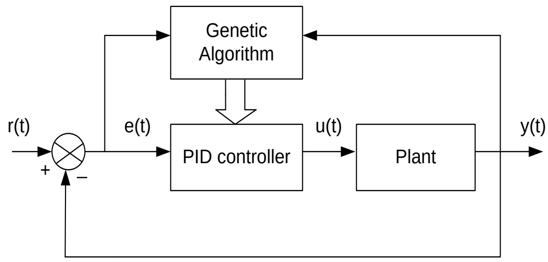

Fig 5: Block diagram of GA-PID controller

The steps involved in creating and implementing a genetic algorithm are as follows:

1. Generate an initial, random population of individuals for a fixed size.

2. Evaluate their fitness.

3. Select the fittest members of the population.

4. Reproduce using a probabilistic method (e.g., roulette wheel).

5. Implement crossover operation on the reproduced chromosome.

6. Execute mutation operation with low probability.

7. Repeat step 2 until a predefined convergence criterion is met.

Flowchart of genetic algorithm process is shown in Fig 6.

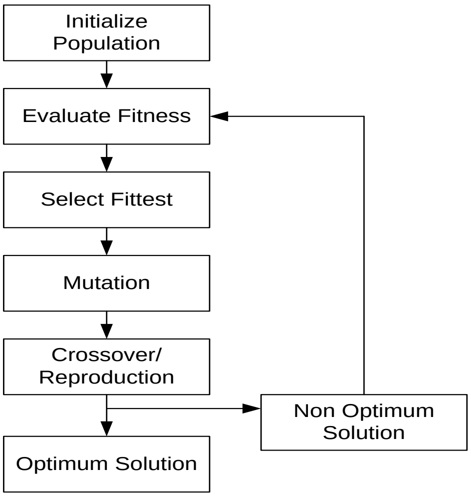

Fig .6: Flowchart of GA process

4.1 ADVANTAGES OF GENETIC ALGORITHM

- It is a simple algorithm that is easily understood and implemented.
- The algorithm is robust.
- GA is a non-linear process that could be applied to most industrial processes with good results.
- GA searches a population of points instead of a single solution. The GA is therefore not easily sidetracked to obtain a local optimal solution instead of a global optimal solution.
- GA does not need information about the system except the fitness function [7].

4.2 SHORTCOMINGS OF GENETIC

In GA based tuning method, it cannot be guaranteed that the result obtained for the process is the most optimized values although it's near optimum. As GA can different result for each new search for the same system under same conditions. In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem. Therefore, the result may not be the perfectly optimized one [10].

4.3 DESIGNING OF PID CONTROLLER USING GENETIC ALGORITHM

The main objective of this work is to investigate the use of genetic algorithms in the tuning of PID controller. The algorithm searches for the controller gains K_p (proportional gain), K_i (integral gain) and K_d (derivative or differential gain) so that specifications for the closed-loop step response are satisfied. Due to their widespread use in industry, tuning procedures for PID controllers are always a topic of interest [11].

The implementation of the tuning procedure through genetic algorithms starts with the definition of the chromosome representation. As illustrated in Fig.7, the chromosome is formed by three values that correspond to the three gains to be adjusted in order to achieve a satisfactory behavior. The gains K_p, K_i and K_d are binary strings numbers and characterize the individual to be evaluated. Each gain is represented by ten binary numbers each chromosome has thirty genes [8].

K_p	K_i	K_d

Fig 7: Chromosomes definition

A GA-PID controller can be implemented as follows:

A. Initialization

Initialization of the population size, variable bounds and string-length are required. These are the initial inputs that are required for the Genetic Algorithm process to start.

B. Encoding

Encoding techniques in genetic algorithms (GAs) are problem specific, which transforms the problem solution into chromosomes. One binary string consisting of the three PID gain coefficients: K_p, K_i

and K_d. The length of string depends on defining domain of the variables and the precision of calculation. There are many chromosome encode methods in the GA. In this paper binary number encoding method was adopted as it is easy for genetic algorithms operation.

C. Fitness Function

A fitness function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time. However in order to combine all of these objectives it was decided to design a fitness function that will minimize the error of the controlled system. The fitness of a chromosome is calculated from the integral of the absolute error (IAE).

$$J = \int_0^{\infty} |e(t)|.dt$$

Where e(t) the error between the reference signal and the system output.

In this paper the fitness function is chosen as

$$f = 1/J$$

F. Selection

The standard roulette wheel selection is applied for selecting the chromosomes. In this method parents are selected according to the fitness values. Chromosomes with higher fitness have higher chances to be selected.

D. Crossover

It is the process in which genes are selected from the parent chromosomes and new offspring is created. But not all individuals are necessarily used for crossover. In this paper uniform crossover is chosen.

E. Mutation

Mutation changes the structure of the string by changing the value of a bit chosen at random. This operator can prevent individuals falling into a local optimum. A random string with the length L is generated. If the value in a position of this random string is less than or equal to the mutation rate pm, the gene of the child in the same position will be inverse of the original [7]. In this paper, the mutation rate is chosen as pm = 0.032.

4.4 Simulation Results

- Population size: 30
- Generations: 20
- Mutation: 0.032
- Reproduction method: roulette wheel
- Crossover: Single Point Crossover [7].

The parameter range of GA-PID Controller for the second order and second order delay systems K_p ε [0,10], K_i ε [0,1], K_d ε [0,1], for the third order system K_p ε [0,20], K_i ε [0,1], K_d ε [0,5] and for the fourth order and fourth order delay systems K_p ε [0,3], K_i ε [0,1], K_d ε [0,1].

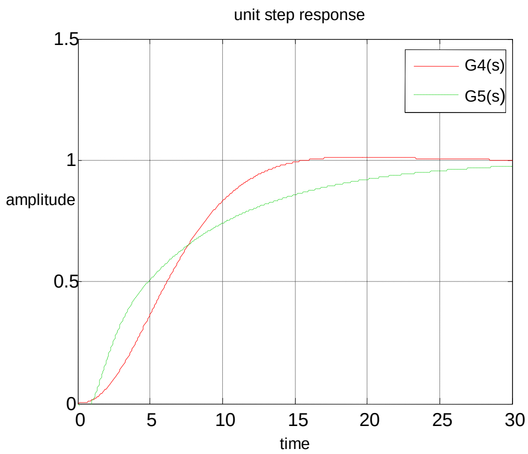

Fig 8: Responses of GA-PID controller of $G_4(s)$ and $G_5(s)$

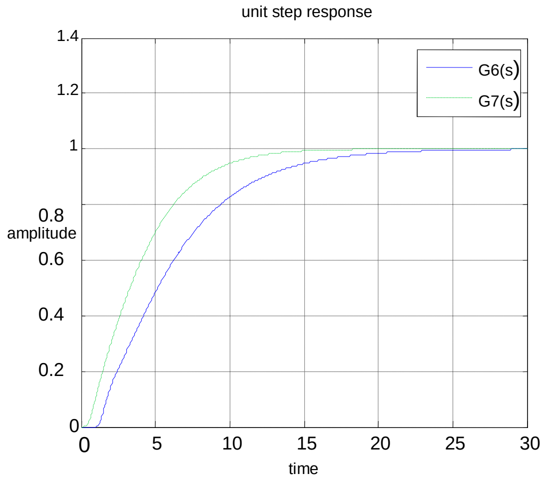

Fig 9: Responses of GA-PID controller of $G_6(s)$ and $G_7(s)$

5. OPTIMIZATION OF PI –TYPE FLC USING GENETIC ALGORITHM

The combination of genetic algorithm and fuzzy logic controllers is normally shortened as GA-FLC and this intelligent hybrid controller has found application in many scenarios like motor speed control, temperature control, robotics and in many other control systems.

This method employs the fuzzy logic technique to design a Fuzzy Proportional-Integral (PI) Controller and optimizes scaling gains of this controller using Genetic Algorithm (GA).

The inference system has three linguistic variables which are the two inputs (error signal and change in error signal) and the output (control signal), each input having five membership function sets. This results in 25-rule fuzzy inference system with inputs as the error and the rate of change in error. The output of the fuzzy logic inference system is the control action of the controller and the universe of discourse of all the variables are set within the range (-1, 1).

The Mamdani fuzzy inference method was considered to develop the model. Such method attempts to solve control problems by a set of linguistic rules obtained from experienced human operators [12].

5.1 Simulation Results

The GA-optimization algorithm was run for 20 generations with each generation having a population size of 30. MATLAB M-files where utilized for the encoding, testing and decoding of the tuned FLC parameters. This includes the scaling gains of the controller. The output scaling gain of the controller was adjusted over a range of 1-9 while the input scaling gain and the derivative input scaling gains were adjusted over a range of 0.0003 - 0.0009 and 0.01 - 1 respectively. The suitability of the ranges of the scaling gains was determined from the prior hand tuning of the controller.

The step responses of the optimized fuzzy PI-controller are shown below for the various systems.

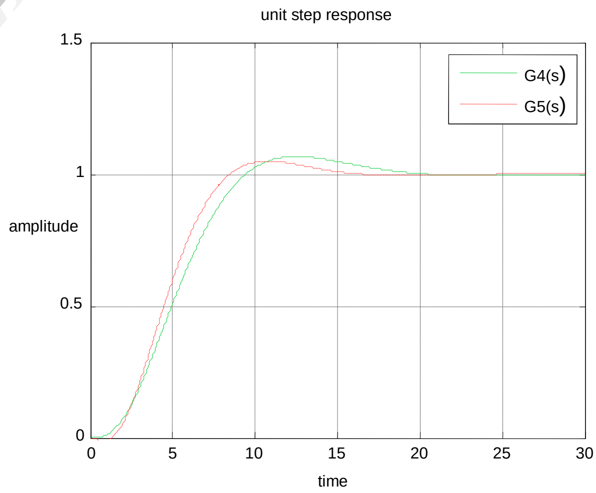

Fig 10: Responses of GA tuned PI-type FLC of $G_4(s)$ and $G_5(s)$

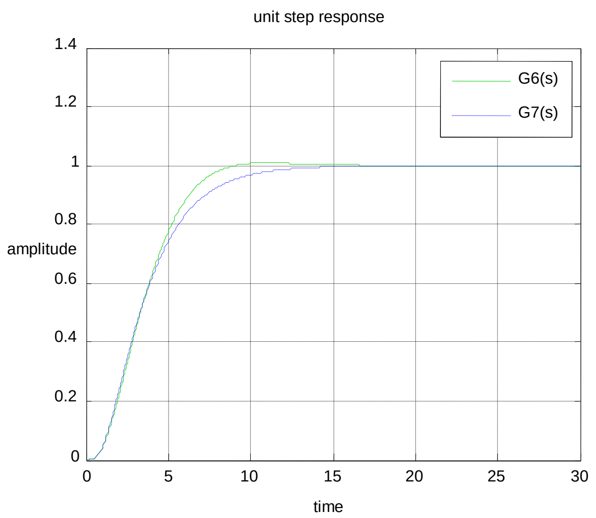

Fig 11: Responses of GA tuned PI-type FLC of $G_6(s)$ and $G_7(s)$

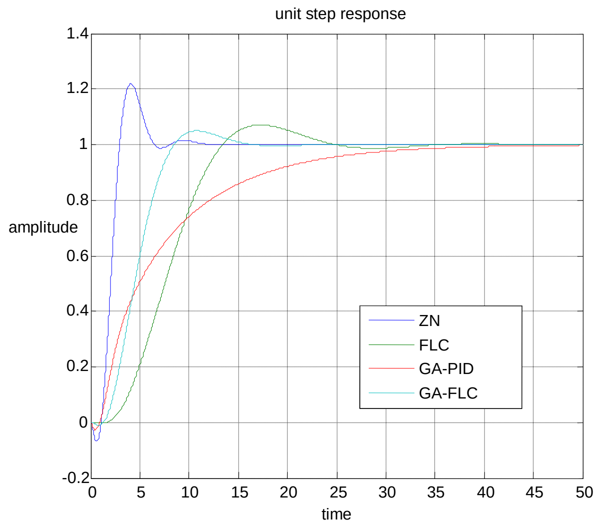

Fig .12 Step responses of $G_5(s)$ with different Controllers

RESULTS AND DISCUSSION

The performance metrics of the conventional PID controller, PI-type fuzzy controller, GA-PID and GA-optimized fuzzy logic PI controller obtained from the simulation are tabulated below and the comparison of step responses of the different controllers are also shown in Fig. 12 and Fig. 13.

From the tables 3, 4, 5 & 6, it can be seen that the fuzzy logic controller can produce the better performance with the use of only the proportional and Integral Component (PI). When compared to the conventional PID controller, the PI-type fuzzy logic controller shows a better performance in terms of overshoot while it exhibits a slightly lesser performance in terms of rise time and settling time. Further Genetic Algorithm technique is used to optimize the conventional PID and PI-type fuzzy logic controller. By comparing the results, it can be noted that the GA-optimized PI-type fuzzy controller produces better performance in terms of achieving the desired value with for the IAE, percentage overshoot, rise time and settling time, as compared to the PI-type FLC and GA-PID controllers.

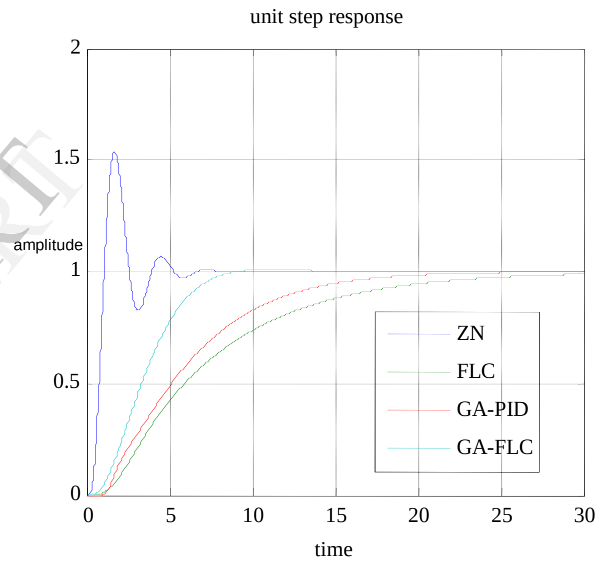

Fig .13 Step responses of $G_6(s)$ with different controllers

Table 3: Value of IAE achieved by the examined controllers tuned by different methods

	ZN	FLC	GA-PID	GA-FLC
$G_1(s)$	1.5185	3.7559	3.4404	3.4192
$G_2(s)$	3.7827	9.0040	8.4988	8.0983
$G_3(s)$	4.8676	9.0212	8.6487	7.9154
$G_4(s)$	1.9767	6.4774	6.1500	4.9485
$G_5(s)$	2.4464	7.1983	5.5370	4.5543
$G_6(s)$	1.3811	6.1054	4.5794	3.5108
$G_7(s)$	2.1240	5.0163	3.8429	3.7472

Table 4: Value of Percentage Overshoot OS (%) achieved by the examined controllers tuned by different methods

	ZN	FLC	GA-PID	GA-FLC
$G_1(s)$	49.2961	0	0	0
$G_2(s)$	45.8958	0	0	0
$G_3(s)$	62.5152	0	0	0
$G_4(s)$	48.8019	0	0	0
$G_5(s)$	20.2102	0	0	0
$G_6(s)$	53.5438	0	0	0
$G_7(s)$	68.5653	0	0	0

Table 5: Value of Rise-time (t_r) achieved by the examined controllers tuned by different methods

	ZN	FLC	GA-PID	GA-FLC
$G_1(s)$	0.7076	6.6089	6.4750	5.1251
$G_2(s)$	1.1530	5.3924	6.7234	6.5440
$G_3(s)$	2.1775	5.2174	6.4293	6.3521
$G_4(s)$	0.9534	6.0942	6.4617	6.0029
$G_5(s)$	1.4142	5.7092	6.5493	5.2133
$G_6(s)$	0.5626	6.5305	6.6646	4.9769
$G_7(s)$	0.6069	5.9776	5.9292	5.3606

Table 6: Value of Settling-time (t_s) achieved by the examined controllers tuned by different methods

	ZN	FLC	GA-PID	GA-FLC
$G_1(s)$	8.7372	9.2893	9.1558	8.2792
$G_2(s)$	9.8992	9.9106	9.8623	9.8597
$G_3(s)$	9.9100	9.9131	9.8677	9.8583
$G_4(s)$	8.2922	9.7648	9.7287	9.5387
$G_5(s)$	7.6717	9.8411	9.5328	9.0687
$G_6(s)$	6.2269	9.6567	9.6083	8.2824
$G_7(s)$	9.4797	9.4678	9.4032	8.8841

6. CONCLUSION

In this paper, the designing and tuning of different controller using Ziegler-Nichols, fuzzy logic and Genetic algorithm has been presented. The performance metrics taken into consideration are the IAE, overshoot, rise time and settling time. The simulation is carried out on various systems and the results shows that the GA-optimized PI-type fuzzy controller gives better performance than PI-type fuzzy and GA-PID controller in terms of all the metrics. According to the profiling results, the use of soft-computing technique resulted in a better outputs. The amount of overshoot for the output response was successfully decreased using the GA tuned PI-type FLC but the values of other performance parameters like IAE, rise-time and settling-time are slightly increases as compared to conventional controller. Thus a compromise has to be made between the percentage overshoot and the other performance parameters.

REFERENCES

[1] Khaled Belarbi and Faouzi Titel, "Genetic Algorithm for the Design of a Class of Fuzzy Controllers: An Alternative Approach", IEEE Transactions on Fuzzy Systems. Vol. 8, No. 4, August 2000, pp. 398 – 405.

[2] Ogata K., "Modern Control Engineering", 4th Edition, Prentice Hall India Publication, 2002.

[3] Visioli A., "Tuning of PID controllers with fuzzy logic", IEE proc.-Control Theory Appl. Vol.148, No. 1, January 2001, pp. 1 – 8.

[4] Rao V. Dukkipati, "MATLAB An Introduction with Applications", New Age International (P) Ltd., Publishers, 2010.

[5] Drainkov D. Hellendoorn H, Reinfran M., "An Introduction to Fuzzy Control", Narosa Publication House, 2001.

[6] Chopra Seema, Mitra R. and Kumar Vijay, "Auto tuning of Fuzzy PI type Controller using Fuzzy logic", International journal of computational cognition (http://www.ijcc.us), vol. 6, no. 1, March 2008, pp. 12 – 18.

[7] Fan Liu and Joo Er Meng, "Design for Auto-tuning PID Controller Based on Genetic Algorithms", 4th IEEE conference, ICIEA, 2009, pp. 1924 – 1928.

[8] Guohan Lin and Guofan Liu., "Tuning PID Controller Using Adaptive Genetic Algorithms", 5th International Conference on Computer Science & Education Hefei, China. August 24-27, 2010, pp. 519 – 523.

[9] Saifudin Bin Mohamed Ibrahim, "The PID Controller Design Using Genetic Algorithm", dissertation submitted in the University of Southern Queensland Faculty of Engineering and Surveying, 27th October, 2005.

[10] S. K. Mallick and M. A. Khan, "Study of the Design and Tuning Methods of PID Controller Based On Fuzzy Logic and Genetic Algorithm" , thesis submitted in the Department of Electronics and Communication Engineering National Institute of Technology, Rourkela, May, 2011.

[11] Qing Wang, Pieter Spronck, Rudolf Tracht., "An Overview of Genetic Algorithms Applied To Control Engineering Problems", Proceedings of the Second International Conference on Machine Learning and Cybernetics, Wan, 2-5 November 2003, pp. 1652 – 1656.

[12] Danilo Pelusi, "Optimization of a Fuzzy Logic Controller using Genetic Algorithms", Third International Conference on Intelligent Human-Machine Systems and Cybernetics, 2011, pp. 143 – 146.

[13] 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, 2008, pp. 799 – 806.

[14] David E. Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning," Addison-Wesley, 2000.

[15] Kerry M. Mandel, "Fuzzy Logic Systems for Engineering: A Tutorial", Proceeding of IEEE Vol. 83, No.3, March 1994, pp. 345 – 377.

[16] David E. Thomas and Brain Armstrong Helonberry, "Fuzzy Logic Control – A Taxonomy of Demonstrated Benefits", proceedings of the IEEE, Vol.83, No. 3, March 1995, pp. 407 – 421.

[17] Yi-Sheng Zhou and Lin-Ying Lai, "Optimal Design for Fuzzy Controllers by Genetic Algorithms," IEEE Transactions on Industry Applications, Vol. 36, No. 1, January/February 2000, pp. 93 – 97.

[18] Han-Xiong Li, "A Comparative Design and Tuning for Conventional Fuzzy Control", IEEE Transactions on Systems, man, and Cybernetics-Part B: cybernetics, Vol. 27, No. 5, October 1997, pp. 884 – 889.