©2010 International Journal of Computer Applications (0975 – 8887)
Volume 1 – No. 11

# GA Based IFLC Design for an Industrial Process

Dr.P. Subbaraj
Director Research
Arulmigu Kalasalingam College of Engineering ,
Tamilnadu, India

P.S. Godwin Anand
Research Scholar
Anna University,Chennai,India

## ABSTRACT
Fuzzy Logic with PID control (IFLC) has been applied for various applications which provide better performances compared to independent FLC, and PID. Although expert-system-based solutions are effective in controlling the processes. Design Fuzzy logic controller has traditionally been achieved through a process of trial and error. Such approach cannot obtain optimized FLC; more formal methods of knowledge base optimization are required. Genetic Algorithms (GAs) provide such a method to optimize the FLC parameters to globally optimum. In this paper, the FLC and the PID controller is optimally designed using the genetic algorithm. The effectiveness of the proposed approach (GAIFLC) is compared to a previous IFLC designed based on trial and error method and conventional PID controller for a three tank system. The simulation results of the proposed approach provide a satisfactory response in all means.

## Keywords
FLC, GA, PID, Optimization, IFLC.

## 1. INTRODUCTION
Proportional-integral-derivative (PID) controller has been the most popular control loop feedback mechanism since 1950s, and has been extensively used in controlling industrial process. Many industrial process systems may not be as readily described mathematically due to the complexity of the components of the plant and the interaction between them. Although PID controller is used widely, the design of the controller is based on their precise mathematical models, which are usually very difficult to achieve owing to the complexity, nonlinearity, time varying and incomplete characteristics of the existing practical systems.[1]

Emerging intelligent techniques have been developed and extensively used to improve or to replace conventional control techniques, because these techniques do not require a precise model. One of the intelligent techniques, fuzzy logic developed by Lotfi A. Zadeh [2], is applied for controller design in many applications. The variety of fuzzy control applications indicates that this technique is becoming an important tool for complex processes [3]. Fuzzy control is a promising new way to face complex process control problems and the tendency is to increase their range of applicability in industrial processes [4],[5].

IFLC has been applied for various applications which provides better performances compared to independent FLC and PID[6], [7], [8]. Although expert-system-based solutions are effective in controlling the processes, this methodology has inherent limitations since it is designed to mimic a human operator with inherent decision-making limitations . its is very difficult to design a FLC based on trial and error method for IFLC [6]. Based on the knowledge and experience gained about the process the FLC is designed for the process [7], Heuristic knowledge is applied to define fuzzy membership functions and rules [8]. All this FLC are designed by the expert who has a through knowledge about the process and the Knowledge base differ expert to expert.

In the absence of such knowledge, a common approach is to optimize these FLC parameters through a process of trial and error [9]. This approach becomes impractical for systems having significant numbers of inputs since the rule-base size grows exponentially and consequently the number of rule combinations becomes significantly large [10]. The use of Genetic Algorithms (GA) in this regard can provide such a solution [11], [12], [13].Genetic Algorithms (GAs) [14] are robust, numerical search methods that mimic the process of natural selection. Although not guaranteed to absolutely find the true global optima in a defined search space. Genetic fuzzy systems are capable of dealing with the curse of dimensionality for complex problems with high dimensionality [15].

In this paper, the integrated FLC and PID was optimally designed using the genetic algorithm (GAIFLC) The effectiveness of the proposed approach is compared to a previous IFLC designed based on Heuristic knowledge and conventional PID controller. The simulations results are presented.

## 2. GAIFLC
The scheme consists of a conventional PID control with the FLC both optimized by the Genetic Algorithm (GAIFLC). Figure 1 illustrate the control structure of the proposed work. The Fuzzy uses the command input $y_m$ and the plant output $y_p$ to generate a suprivisory command output $y^1_m$ described by the following equations.

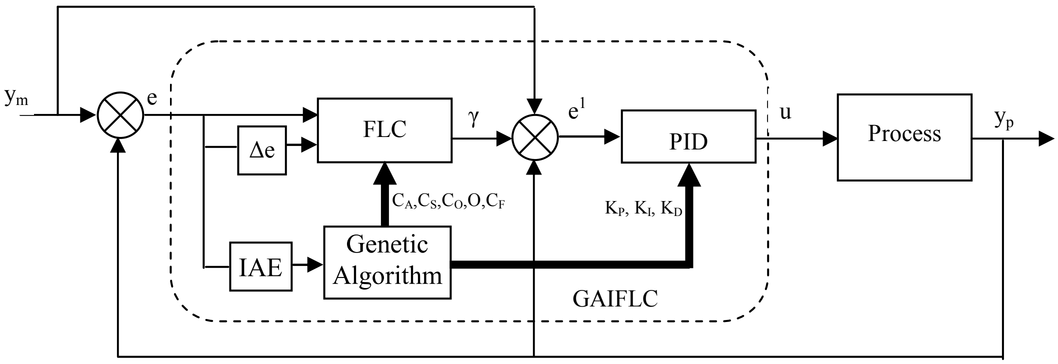


**Figure 1 Control Structureof GAIFLC**

$$e(k) = y_m(k) - y_p(k) \tag{1}$$

$$\Delta e(k) = e(k) - e(k-1) \tag{2}$$

$$\gamma(k) = F\,[e(k),\, \Delta e(k)\,] \tag{3}$$

$$y_m^1(k) = y_m(k) + \gamma(k) \tag{4}$$

In the above, e(k) is the tracking error between the command input $y_m(k)$ and the plant output $y_p$ (k), and Δe(k) is the change in the tracking error. The term F[e(k), Δe(k)] is a nonlinear mapping of e(k) and Δe(k) based on fuzzy logic described below. The term γ(k) = F[e(k), Δe(k)] represents correction term so that the compensated command signal $y^1{}_m(k)$ is simply the sum of the external command signal $y_m(k)$ and γ(k). The correction term is based on the error e(k) and Δe(k). The compensated command signal $y^1{}_m(k)$ is applied to a conventional PID controller as shown in figure 1. The equations determining the PID controller are as follows.

$$e^1(k) = y_m^1(k) - y_p(k) \tag{5}$$

$$\Delta e^1(k) = e^1(k) - e^1(k-1) \tag{7}$$

$$u(k) = u(k-1) + K_P\,\Delta e^1(k) + K_I\,\Delta e^1(k) + K_D \Delta e^1(k) \tag{9}$$

The quantity $e^1(k)$ is the supervised tracking error of the supervised command input $y^1{}_m$ (k) and the plant output $y_p(k)$, and $\Delta e^1(k)$ is the change in the supervised tracking error. The control u(k) is applied as input of the plant. The purpose of the IFLC is to modify the command signal to compensate for overshoots and undershoots present in the output response when the process has unknown nonlinearities. Such nonlinearities can result in significant overshoots and undershoots if a conventional PID control scheme is used. The FLC and PID tuning values is optimized by GA based on the minimization of Integral Square error (ISE) of the closed loop response.

# 3. FUZZY LOGIC CONTROLLER

The implementation of the fuzzy logic based term is $u(t)$ = F[$e(t)$, $\Delta e(t)$]. In the description standard terminology is used to form fuzzy set theory, for a treatment of fuzzy sets, $e(t)$, and $\Delta e(t)$ as inputs to the map F, and u($t$) as the output. Associated with the map, F is a collection of linguistic values L={ NB, NS, ZO, PS, PB} that represent the term set for the input and output variables of F. In this case seven linguistic values are used. The meaning of each linguistic value in the term set L should be clear from its mnemonic; for example, NB stands for negative big, NS for negative small, ZO for zero and likewise for the positive (P) linguistic value. Associated with the term set L is a collection of membership functions. $\mu = \{ \mu_{NB}, \mu_{NS}, \mu_{ZO}, \mu_{PS}, \mu_{PB} \}$ Each membership function (MF) is a map from the real line to the interval [-1 +1 ]. In this application the MF used is the (triangular or trapezoidal type). The height of the MF in this case is one, which occurs at the points optimized by GA. The realization of the function F[$e(t)$, $\Delta e(t)$] deals with the setting of linguistic values. This consists of scaling the inputs $e(t)$ and $\Delta e(t)$ appropriately and then converting them into fuzzy sets. The symbol $C_e$ is the scaling constant for the input $e(t)$ and the symbol $C_{de}$ is the scaling constant for the input $\Delta e(t)$. For each linguistic value $l \in$ L, assign a pair of numbers $ne(l)$ and $\Delta e(l)$ to the inputs $e(t)$ and $\Delta e(t)$ with the associated membership function {$ne(l) = \mu l$ ($C_e$ $e(t)$ ), $n\Delta e(l) = \mu l$ ($C_{de}$ $\Delta e(t)$) }. The numbers $ne(l)$ and $n\Delta e(l)$, $l \in$ L are used in the computation of F[$e(t)$, $\Delta e(t)$] [6]. As soon as fuzzy inference is applied to each rule, the activation level for all output variable (MFs) are obtained, and the defuzzification procedure takes place. In order to compute the final control action, $u$(t), the most commonly used method is the center of area [6]. The result is the center of area of the profile described by the membership functions, limited in the respective activation level. Equation (18) shows the defuzzified output

$$u^* = \frac{\int \mu_c(u) \cdot u\, du}{\int \mu_c(u)\, du} \tag{10}$$

Where $u^*$ is the defuzzified value,
and $\int$ denotes an algebraic integration

# 4. GENETIC ALGORITHM

Genetic algorithms (GA) are usually used as optimization techniques. It has been shown that GA also perform well with multimodal functions (i.e., functions which have multiple local optima). Genetic algorithms work with a set of artificial elements (binary strings, e.g., 0101010101), called a population. An individual (string) is referred to as a chromosome, and a single bit in the string is called a gene. A new population (called offspring) is generated by the application of genetic operators to the chromosomes in the old population (called parents). Each iteration of the genetic operation is referred to as a generation. A fitness function, specifically, the function to be minimized, is used to evaluate the fitness of an individual. One of the important purposes of the GA is to reserve the better schemata, i.e. the patterns of certain genes, so that the offspring may have better fitness than their parents. Consequently, the value of the fitness function increases from generation to generation. In most genetic algorithms, mutation is a random-work mechanism to avoid the problem of being trapped in a local optimum. Theoretically, a global optimal solution can be found [16]. The basic operations of a simple genetic algorithm, i.e. reproduction, crossover and mutation, are described below.

## 4.1 Chromosome representation

Most GAs search the global optimal solutions through the natural genetics and the evolution theory on a population. Each individual coded as a binary string in the population is called a string or chromosome. A new generation of GAs is evolved from the existing population. In applying the technique of GAs to solve the problems on hand, a string scheme is employed to encode the candidate solutions (chromosomes) in the form of symbolic strings.

## 4.2 Fitness function

The original GA and its many counterparts, collectively known as GAs, are computational procedures which mimic the natural process of evolution. The survival of the fittest principle leads to improvements in the species. Since GAs are heuristic procedures, they are not guaranteed to find the optimum, but they are able to find very good solutions for a wide range of problems. A fitness function (or objective function) is used to determine the fitness of each candidate solution. A fitness value is assigned to each individual in the population.

Integral of absolute error is a better all-round performance indicator of control system response where overshoot, settling and rise times are the main performance considerations [11]. The IAE was therefore used as a measure of performance.

$$IAE = \int_0^t |e(t)|\, dt \tag{11}$$

In Controller Design problems IAE has to minimized, hence the objective function J is set as mentioned in equation (12)

$$J = IAE_{SP=1} \times IAE_{SP=5} \tag{12}$$

## 4.3 Selection

The selection process is centered upon the specified cost function. The selection scheme is used to draw chromosomes from the evaluated population into the next generation. Tournament selection is one of many methods of selection in genetic algorithms. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected [17]. Deterministic tournament selection selects the best individual (when p=1) in any tournament. A 1-way tournament (k=1) selection is equivalent to random selection. The chosen individual can be removed from the population that the selection is made from if desired, otherwise individuals can be selected more than once for the next generation

## 4.4 Crossover

Crossover provides a mechanism for individual strings to exchange information via a probabilistic process. Once the reproduction operator is applied, the members in the mating pool are allowed to mate with one another. First, the genetic codes of the two parents are mixed by exchanging the bits of codes following the crossover point. For example, consider two parent strings where the crossover point is 5 (i.e., the fifth bit in the string)

P1 = 10101|010; P2 = 01111|100;

The separator symbol ‘‘|” indicates the crossover site. The resulting offspring have the following:

P01 = 10101|100; P02 = 10101|010:

## 4.5 Mutation

In each iteration, every gene is subject to a random change, with the probability of the pre-assigned mutation rate. In the case of binary-coding, the mutation operator changes a bit from 0 to 1, or vice versa. All in all, the mutation operation introduces new genes into the population, so as to avoid the problem of being trapped in local optima. Offspring are generated from the parents until the size of the new population is equal to that of the old population. This evolutionary procedure continues until the fitness reaches the desired specifications.

# 5. ENCODING (GAIFLC)

Although fuzzy logic allows the creation of simple control algorithms, the tuning of the fuzzy controller for a particular application is a difficult task and one needs a more sophisticated procedure than that used for a conventional controller. This is due to the large number of parameters that are used to de fine the MFs and the inference mechanisms. Several methods have been developed for tuning fuzzy controllers. These involve adjustment of the MF [18] and/or scaling factors [19] and dynamically changing the defuzzification Procedure. Therefore, the approach needs as many variables as there are rules to get an optimal rule base. The advantage of the approach reported in the present paper is that it takes only three variables to optimize the rule base geometry, two variables to optimize the membership function and three scaling variables.

## 5.1 Encoding Rule Base

To design an optimal rule base A simple geometric approach is followed to modify the rule base as mentioned in [20]. For this the initial assumptions made were as follows;

- The magnitude of the output control action is consistent with the magnitude of the input values. (i.e. in general, extreme input values (premise) result in extreme output values (consequent), mid-range input values in mid-range output values and small/zero input values in small/zero output values.

- If a large negative (positive) input generates a large negative (positive) response, then it is likely that slightly smaller, negative (positive) inputs will necessitate a response of like polarity, but smaller magnitude, and so forth until a zero-crossover point is reached at which point the polarity of the response changes.

Using these generalizations, in conjunction with the concept of system symmetry, a different approach can be used which reduces the number of bits required for the rule -base dramatically. The approach is a variation of the method which involves a fixed coordinate system defined by the possible premise combinations. The consequent space is then 'overlayed' upon the premise coordinate system and is in effect partitioned into 5 regions shown in fig.2,

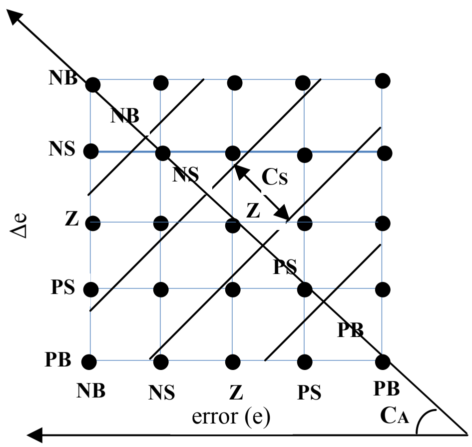


**Figure 2. GAFLC Rule Assigning**

Consequent-line angle, $\mathbf{C_A}$ (16 angles between 0-168$^o$ (i.e. 4 bits))
Consequent-region spacing, $\mathbf{C_S}$ (4-bits) Consequent-line order, **CO** (1-bit) (Defines order of consequent space partitions (i.e. NB-NS-Z-PS-PB or PB-PS-Z-NSNB)
A total of 9-bits are used to extract rule -bases consistent with the above assumptions

## 5.2 Encoding membership function

In the attempt to encode the FLC membership functions associated with the 2 inputs and 1 output, a number of assumptions are made in respect of the distribution of fuzzy sets across the universe of discourse (UOD) for each fuzzy variable. These assumptions are;

- The UOD is symmetrical about the central, zero region for each variable.
- The extreme membership functions (MF) for input variables should be unbounded in the respective positive and negative going directions.
- The inner and central UOD-range MFs could assume either triangular (trimf) or trapezoidal (trapmf) shapes only, for input and output variables. Outer UOD-range MFs for input variables were unbounded z-shaped (zmf), while output variable extreme MFs could assume the same shape as inner and central range MFs (trimf or trapmf).
- The number of fuzzy sets for the controller was fixed at 5 (NB, NS, Z, PS PB).

The MF properties altered by the GA are as follows;
1. MF shape (triangular or trapezoidal).
2.Degree of MF-centre shift to effect MF compression or expansion.

## 5.3 MF Offset Field

The optimization begins by loading a *.fis (Matlab Fuzzy file) into the FLC block in the MATLAB Simulink model. Each evaluation subsequently uses a 'genetically-altered' version of the original FLC which is defined by a MATLAB, fuzzy structure. For each evaluated FLC, the UOD -distributed MFs are initially assumed to be trapezoidal in type, thus 4 parameters are required by the FIS to define the position in the UOD of each of the 5 MFs. The significance of these parameters is illustrated below in Figure3 & 4.

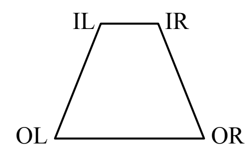


**Fig.3 Trapezoidal MF parameters**

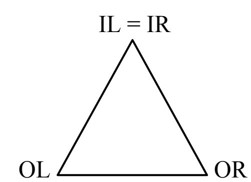


**Fig 4. Trapezoidal MF Defining triangular MF**

The Matlab Fuzzy file 'params' field has 4 UOD position parameters (outer-left(OL), inner-left(IL), inner-right(IR), outer-right(OR) ). For inner parameters (IL and IR ) equal in value, MF becomes triangular in shape. The offset field is used to effect a change of shape in the MFs. The 3-bit offset field is decoded in the range of [0, 0.1] and the application of the offset parameter modifies the shape of the MFs from triangular to trapezoidal of varying widths and positions. The MFs of each FLC fuzzy variable (e, de and u) are encoded into the GA-chromosome in this manner.

## 5.4 MF Companding Field

Application of the offset field produces MFs of different shapes (trimf or trapmf) and positions, but does not effect the distribution of the MFs, which are evenly distributed across the UOD. To enable evaluation of non-uniform distributed MFs, a further field is encoded into the GA-chromosome for each fuzzy variable, which is applied to the MFs to bring about compression and/or expansion of the associated MFs. The companding field is decoded to a value (CF) in the range [0.5 – 2], and is applied to update the MF position parameters of each MF by raising them to the power of CF (e.g. for the Z-MF, outer-left parameter; $O_{OL(new)} \Rightarrow (O_{OL(old)})^{CF}$ ) Due to the use of a normalized UOD, the position parameters are shifted to different degrees by this operation and the net effect is that;

for CF < 1 : Z-MF is compressed, NB and NS expand
for CF > 1 : Z-MF expands, NB and NS compress
for CF = 1 : uniform MF distribution
In this way, non-uniform distribution of the MFs is effected across the UOD.

## 5.5 Encoding PID Tuning Values

The GA also optimize the tuning values of PID controller. The three fields, proportional gain $K_P$, Integral gain $K_I$ and Derivative gain $K_D$ are included in the GA –chromosome each consisting of 7- bits, which are encoded to yield the tuning values of PID block of the Simulink model.

## 5.6 GA-Chromosome of FIS and PID

Three aspects of the FLC were subject to the optimization procedure; (a)Rule Base, (b)Membership Functions(MF), (c) PID tuning values. The primary assumption made was that for a symmetrical system, a corresponding FLC would also exhibit symmetry about the set point in respect of its MFs and rule -base. This assumption was exploited in order to attempt to reduce the number of bits required to define the FLC for GA optimization. Table 1 illustrates the 51-bit binary GA-chromosome used to encode GAIFLC

Table1 GA Chromosome allotment for FLC and PID Controllers

| FLC Chromosome | | | | PID Chromosome | | |
|---|---|---|---|---|---|---|
| RB 9bits 1:9 | e:MF 7 bits 10:16 | Δe:MF 7 bits 17:23 | u:MF 7 bits 24:30 | $K_P$ 7 bits 31:37 | $K_I$ 7 bits 38:44 | $K_D$ 7 bits 45:51 |

# 6. THREE TANK LEVEL SYSTEM

Liquid level control has a very large application domain in industry. Its most representative didactical equipments are the tank systems, i.e. one, three [7] or four tank systems [22]. Moreover, the three tank system (3TS) is one of the most widely used laboratory system in control theory. Figure 5 show interacting connection of a three tank system in which Tank1 noninteracting connection, Tank2 and Tank3 interacting connection.

By applying the mass balance equation we get

$$H_3(s)/F_1(s) = \frac{R_1R_2R_3}{[(A_2R_2s+1)(A_3R_3R_2s+R_2+R_3)-R_3](A_1R_1s+1)} \quad (13)$$

By considering:
$A_1$=$A_2$=1m$^2$; $A_3$=0.5m$^2$ and $R_1$=$R_2$=$R_3$=2(m/(m$^3$/s)),
We get

$$G(s) = \frac{4}{4s^3+12s^2+6s+1} \quad (14)$$

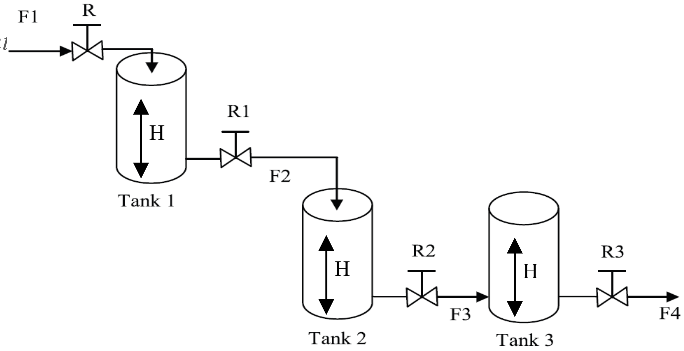


**Figure 5 Three Tank Interacting System**

# 7. SIMULATION

The effectiveness of the proposed control scheme has been assessed through simulations, The entire simulation is carried out in MATLAB & Simulink on a Core 2 Duo Processor 2.2 GHz, 2GB RAM PC Environment.

For comparing with the classical PID controller the Zeigler-Nichols (Z-N) tuning method is followed and for the human designed IFLC is taken from [7].

**Table2 PID tuning Values using ZN method**

| Parameter | Interacting Tank |
|---|---|
| $K_c$ | 1.8 |
| $T_i$ | 3.6 |
| $T_d$ | .9 |

For Fuzzy logic controller two inputs [$e$, $\Delta e$] and one output with seven membership function and 49 rules

## 7.1 Case I

For optimizing the Fuzzy logic and PID controller(GAIFLC) the GA parameters are set to

Generation = 250
Population Size = 50
Crossover rate = 0.5
Mutation Rate = 0.03

For FLC the template was set as two input one output with five membership function and 25 rules.And for the PID tuning values the search range is set as [0 to 5]

## 7.2 Case II

For testing the GAIFLC the following settings are chosen; with Initial setpoint values SP = 1 and 5 and the closed loop response is taken for 50 sec. the servo regulation of the process is tested with the following set point profile shown in (15)

$$SP(t) = \begin{cases} 0 & for\ 0 \le t \le 25 \\ 1 & for\ 25 \le t \le 100 \\ 2.5 & for\ 100 \le t \le 150 \\ 1.5 & for\ 150 \le t \le 200 \end{cases} \quad (15)$$

## 7.3 Case III

To check the Load rejection of the GAIFLC, the load disturbances $L$ is varied from 0.5 to 1 keeping set value to zero. To check the continuous load rejection the setpoint is kept as one and the load is varied for the following profile (16)

$$L(t) = \begin{cases} 0 & for\ 0 \le t \le 50 \\ 0.2 & for\ 50 \le t \le 100 \\ 0.5 & for\ 100 \le t \le 150 \\ 0.3 & for\ 150 \le t \le 200 \end{cases} \quad (16)$$

## 8. RESULTS AND DISCUSSIONS

The Table 2 shows the optimized fuzzy logic control variables. It is observed that the performance Index is minimized to 12.42 from 732.55 after 250 generations and the fig.12 shows the optimized membership functions and table 6 shows the GA optimized IFLC Rule baseafter 250 generations.

The fig.6 and 7 shows the closed loop response of three tank interacting system with GA optimized Fuzzy logic Controller and PID controller for the set point 1 and 5 the output is plotted for 50 seconds.The output response of GAIFLC shows a better performance than PID controller and Human designed IFLC in means of rise time, overshoot and settling time.

The fig.8 and 9 shows the closed loop response of three tank interacting system with GA optimized IFLC for the load variation of .5 and 1 keeping setpoint as zero. The output is plotted for 50 seconds. The response shows the GAIFLC is very effective in load rejection and bring back the system to the set point The robustness of the GAIFLC is tested with continuous setpoint and load variations. The Figure10. Depicts Closed loop response of the system for various control schemes for the setpoint profile given in section 7.2

The response shows the the GAIFLC is capable in maintaining the setpoint with minimum rise time, minimum settling time and minimum overshoot. Figure 11 shows the load regulation in which the GAIFLC shows the better performance than the other two controllers. Table 2 represents the comparison of the closed response output for set point change. The performance of the controllers are compared with respect to the risetime (Rt), Peak overshoot (Po), settling time (St) IAE, and ISE for two different setpoint. The GA IFLC seems to be better in all performances.

Table 3 represents the comparison of the closed response output for the load change. The proposed control scheme is performs better with minimum disturbance correction , IAE and ISE.

**Table 3 GA Optimized FLC and PID control variables**

| | Rule base parameters | | | Membership Parameters | | | | | | PID Tunning Values | | | J |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ca | Cs | Co | Ofset1 | Ofset2 | Ofset3 | Cf1 | Cf2 | Cf3 | $K_P$ | $K_I$ | $K_D$ | |
| 1st Gen | 1.55 | 0.8 | 0 | 0.01 | 0.07 | 0.05 | 0.60 | 0.50 | 0.66 | 3.07 | 2.5 | 3.62 | 957.3 |
| 250thGen | 0.67 | .77 | 1 | 0.08 | 0.5 | 0.07 | 1.04 | 1.11 | 0.70 | 1.65 | 1.12 | 2.65 | 12.42 |

**Table 4. Performance comparison of controlled variable for setpoint change**

| Interacting Tank (Set Point Change) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Figure No | Set Point | Control Scheme | % Peak Overshoot | Rise Time | Settling Time | IAE | ISE |
| | 1 | PID | 1.32 | 2.62 | 22.5 | 2.77 | 1.564 |
| | | IFLC | 1.13 | 2.41 | 19.38 | 1.713 | 1.156 |
| | | GAIFLC | 1.07 | 1.89 | 6.428 | 1.455 | 1.08 |
| | 5 | PID | 6.61 | 2.56 | 21 | 13.85 | 39.71 |
| | | IFLC | 5.63 | 2.38 | 17.5 | 9.552 | 33.21 |
| | | GAIFLC | 5.20 | 2.31 | 7.8 | 8.54 | 31.65 |

**Table 5. Performance comparison of controlled variable for load change**

| Interacting Tank (Load Change process outlet) | | | | | |
|---|---|---|---|---|---|
| Fig. e No | load | Control Scheme | disturbance correction | IAE | ISE |
| | 1 | PID | <15.97 | 2.75 | 1.55 |
| | | IFLC | <11.9 | 1.90 | 1.23 |
| | | GAIFLC | <6.5 | 1.33 | 1.01 |
| | 0.5 | PID | <14.77 | 1.37 | .38 |
| | | IFLC | <11.35 | .95 | .30 |
| | | GAIFLC | <6.87 | .63 | .22 |

**Table 6. Optimized rule-base of GAIFLC**

| Δe \ e | NB | NS | Z | PS | PB |
|---|---|---|---|---|---|
| **NB** | NB | NB | NS | Z | PB |
| **NS** | NB | NB | NS | PS | PB |
| **Z** | NB | NS | Z | PS | PB |
| **PS** | NB | NS | PS | PB | PB |
| **PB** | NB | Z | PS | PB | PB |

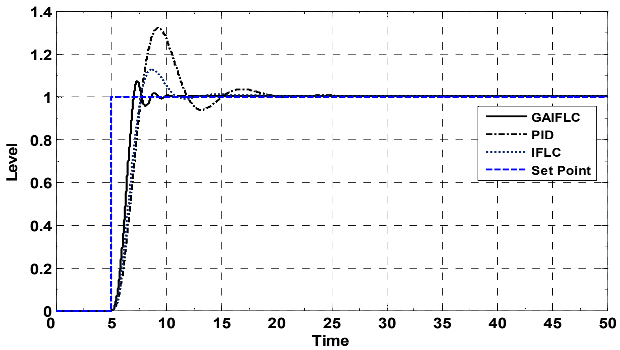


Figure.6 Closed loop response of the system for various control schemes with setpoint =1

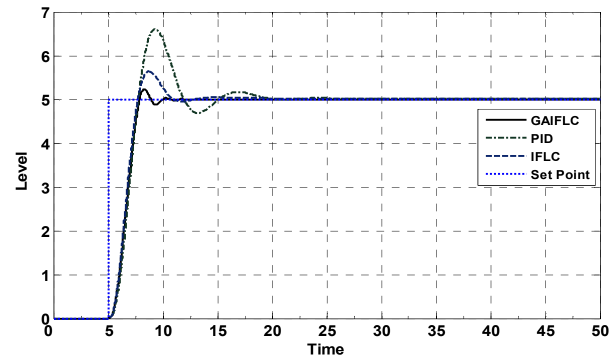


Figure 7. Closed loop response of the system for various control schemes with setpoint =5

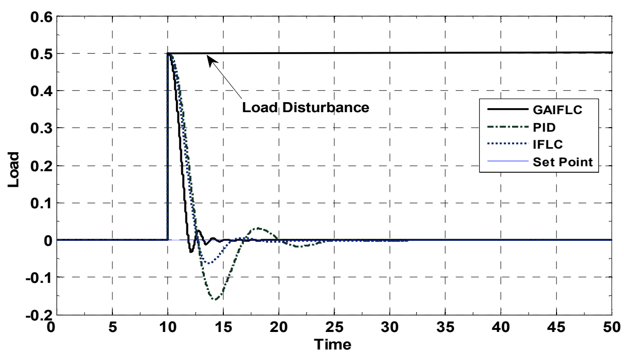


Figure 8. Closed loop response of the system for various control schemes with load =.5

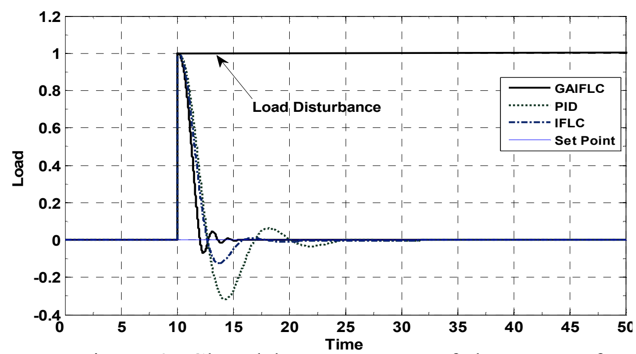


Figure 9. Closed loop response of the system for various control schemes with load =1

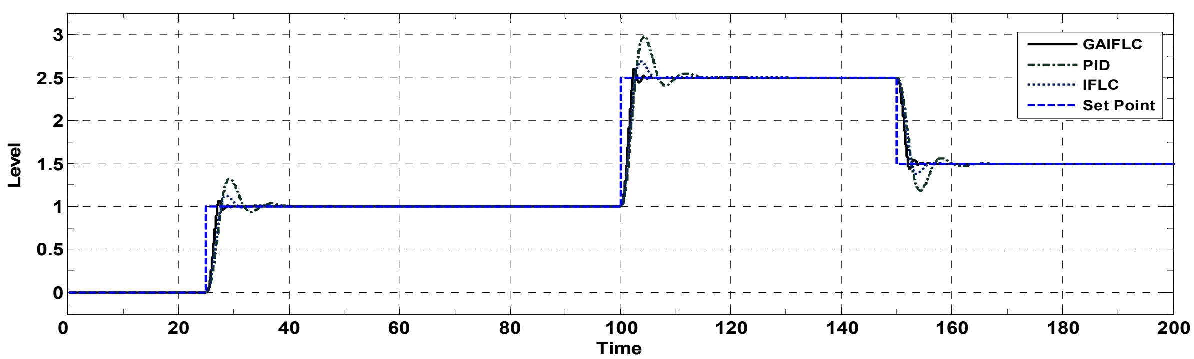


Figure10. Closed loop response of the system for various control schemes with different setpoints

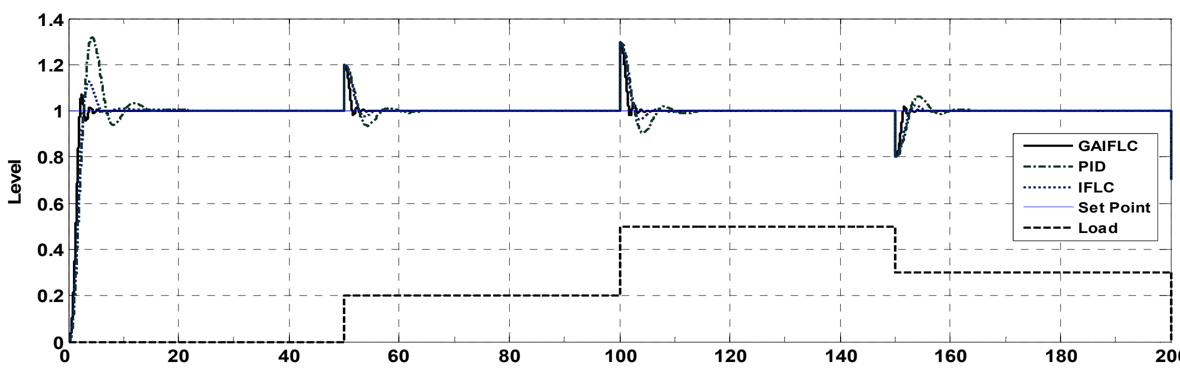


Figure11. Closed loop response of the system for various control schemes with different load

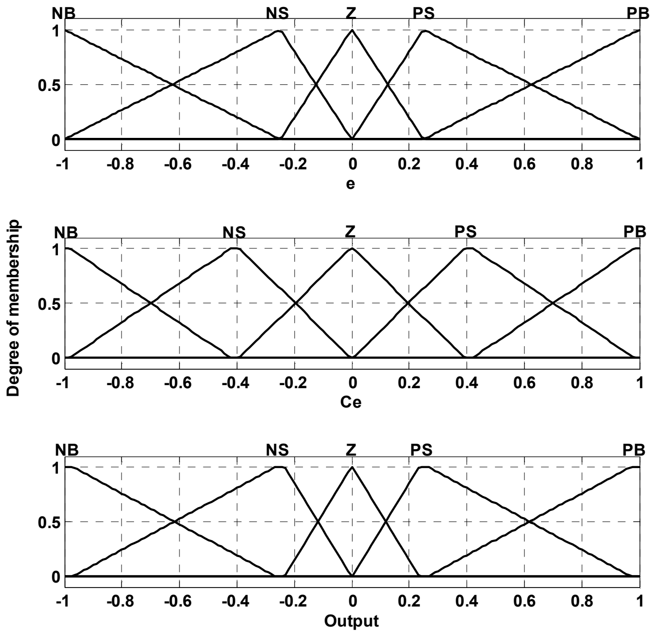


**Figure 12. Optimized Membership function for error, Change in error and output of GAIFLC**

## 9. CONCLUSION

Optimization of a fuzzy logic controller can prove a lengthy process when performed heuristically. In this work it has been shown that the use of genetic algorithms offers a feasible method for the optimization of the knowledge-base of fuzzy logic controllers. The proposed approach is capable of designing a GAIFLC with five membership function and 25 rules also a better control compared to human designed IFLC which have seven membership function and 49 rules. In real time application the GAIFLC tale less computational time compared to human designed IFLC.

## REFERANCE

[1] B.G. Liptak: Instrumentation Engineer’s Handbook: Process Control, Third Edition, CRC Press, 1995.

[2] Zadeh, Lotfi A., 1965, “Fuzzy sets. Information Control”, 8, 338-353.

[3] Lee, C. C. “Fuzzy logic control systems: fuzzy logic controller part-I” IEEE Transactions on Systems, Man and Cybernetics, (1990). 20(2), 404–418.

[4] A.M.F. Fileti, A.J.B. Antunes, F.V. Silva, V. Silveira Jr, J.A.F.R. Pereira Experimental investigations on fuzzy logic for process control”Control Engineering Practice, Volume 15, Issue 9, September 2007, Pages 1149-1160.

[5] Sk.Faruque Ali, Ananth Ramaswamy, “Optimal fuzzy logic control for MDOF structural systems using Evolutionary algorithms” Engineering Applications of Artificial Intelligence 22(2009)407–419

[6] Godwin Anand and Albertraj, Proceedings of International Conference on Intelligent Knowledge Systems (IKS-2004), August 16-20,(2004) paper 36

[7] M. Suresh, Srinivasan, R. Rani Hemamalini “Integrated Fuzzy Logic Based Intelligent Control of Three Tank System” Serbian Journal of Electrical Engineering, Vol. 6, No. 1, May 2009, PP 1-14

[8] S.S. Patil and P. Bhaskar “International Journal of Electronic Engineering Research” Volume 1 Number 1 (2009) pp. 13–25

[9] Dragan.D.K,. Kuzmanovic.S.B,. Emil.L “Design of a PID-like compound fuzzy logic controller”Engineering Applications of Artificial Intelligence, 14 (2001) 785-803

[10] Shi.Yuhui.,. Eberhart.Russell,. Chen.Yaobin “Implementation of Evolutionary Fuzzy Systems” IEEE Transactions on Fuzzy Systems, Vol. 7, No. 2, April 1999

[11] Visioli.A “Tuning of PID controllers with fuzzy logic” IEEE Proc.-Control Theory Appl., Vol.148, No.1, January 2001

[12] Hyun-Joon.Cho,. Kwang-Bo.Cho,. Bo-Hyeun.Wang “Fuzzy-PID hybrid control: Automatic rule generation using genetic algorithms” Fuzzy Sets and Systems 92 (1997) 305-316

[13] Gurocak ,H.B. “A Genetic Algorithm method for tuning Fuzzy Logic Controllers” Fuzzy Sets and Systems 108 (1999) 305-316

[14] Chaiyaratana, N.; Zalzala, A.M.S. “Recent developments in evolutionary and genetic algorithms: theoryand applications”Genetic Algorithms in Engineering Systems: Innovations and Applications, Second International Conference On (Conf. Publ. No. 446) 2-4 Sep 1997 Page(s):270 – 277

[15] Cordon, O.; Alcala, R.; Alcala-Fdez, J.; Rojas, I. “Guest Editorial Genetic Fuzzy Systems: Whatapos;s Next? An Introduction to the Special Section”, IEEE Trans. on Fuzzy Systems Volume 15, Issue 4, Aug. 2007 Page(s):533 – 535

[16] P.C. Chen, C.W. Chen and W.L. Chiang, “GA-based modified adaptive fuzzy sliding mode controller for nonlinear systems”, Expert Systems with Applications 36 (2009) 5872–5879

[17] Brad L. Miller and David E. Goldberg.1995, Genetic Algorithms, Tournament Selection and the Effects of Noise, vol 9, pp 193-212

[18] Arslan, A.,Kaya,M.,2001. Determination of fuzzy logic membership functions using genetic algorithms. Fuzzy Sets and Systems118,297–306.

[19] Zhao, Y.,CollinsJr.,E.G.,2003. Fuzzy parallel parking control of autonomous ground vehicles in tights paces. In: IEEE International Symposium on Intelligent Control, Houston,TX,pp.811–816.

[20] Sk.Faruque Ali, Ananth Ramaswamy, “Optimal fuzzy logic control for MDOF structural systems using Evolutionary algorithms” Engineering Applications of Artificial Intelligence 22(2009)407–419