Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 506082, 8 pages
doi:10.1155/2012/506082

Research Article

FGA Temperature Control for Incubating Egg

Ismail Yusuf,[1,2] Yusram Yusuf,[3] and Nur Iksan[1]

[1] *Department of Computer, Lamintang Education and Training (LET-Indonesia), Perum Bandaramas Blok D5 1, Kota Batam, Kepri 29464, Indonesia*
[2] *Department Of Information Technology, Sekolah Tinggi Teknik (STT) IBNU SINA BATAM, Jalan, Teuku Umar Lubuk Baja, Kota Batam, Kepri 29432, Indonesia*
[3] *Department Of Mechanical Enginering, Sekolah Menengah Kejuruan (SMK) 5 Makassar, Jalan, Sunu 162, Makassar, Sulawesi Selatan 90214, Indonesia*

Correspondence should be addressed to Ismail Yusuf, ariel_ismail@yahoo.com

Received 11 May 2011; Revised 14 July 2011; Accepted 14 July 2011

Academic Editor: Salvatore Sessa

This paper investigates the use of genetic algorithms (GA) in the design and implementation of fuzzy logic controllers (FLC) for incubating egg. What is the best to determine the membership function is the first question that has been tackled. Thus it is important to select the accurate membership functions, but these methods possess one common weakness where conventional FLC use membership function generated by human operators. The membership function selection process is done with trial and error, and it runs step by step which takes too long in solving the problem. This paper develops a system that may help users to determine the membership function of FLC using the GA optimization for the fastest processing in solving the problems. The data collection is based on the simulation results, and the results refer to the transient response specification which is maximum overshoot. From the results presented, we will get a better and exact result; the value of overshot is decreasing from 1.2800 for FLC without GA to 1.0081 with GA (FGA).

1. Introduction

Clearly automatic control has played an important role in the advance of engineering and science. In addition to its extreme importance in robotic systems, and at time, automatic control has become an important and integral part of modern manufacturing and industrial processes. Automatic control is essential in such industrial operations as controlling pressure, temperature, humidity, viscosity, and flow in the process industries.

While modern control theory [1] has been easy to practice, fuzzy logic controllers (FLC) have been rapidly gaining popularity among practicing engineers. This increase of popularity can be attributed to the fact that fuzzy logic provides a powerful vehicle that allows engineers to incorporate human reasoning in the control algorithm.

In our daily life from the production lines in manufacturing plants, medical equipment, and agriculture to the consumer products such as washing machine and air conditioner, FLC can be applied. As for an example, the controller temperature set for incubating egg. Four factors are of major importance in egg incubation artificially: temperature, humidity, ventilation, and turning. Of these factors, temperature is the most critical. If the temperatures are not accurately controlled, the incubated egg will not be uniform. One of the problems with the incubate-egg systems occurs in the design of the temperature controllers. The temperature and relative humidity of incubator should be set at the specific value as proposed by Du et al. [2]. And it is very easy to overheat the eggs in incubators and difficult to maintain proper humidity.

Preferably, these controllers are designed with a high sensitivity to disturbance signals. However, when a change in a temperature set point occurs, there is a danger in saturating the zone temperature controllers as the magnitude of the temperature set point changes are generally greater than the magnitude of disturbances. Hence, the sensitivity of the controller to disturbance signals must be reduced to prevent

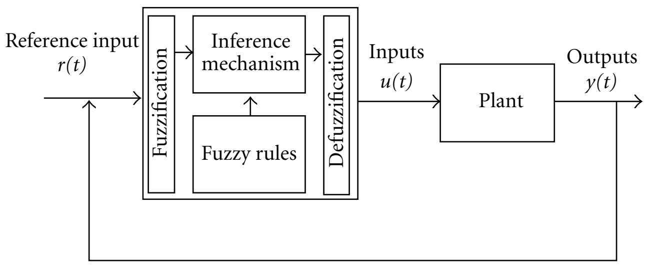


FIGURE 1: Fuzzy controller architecture.

saturation of the controllers to set point changes. Thus it is important to select the accurate membership functions for temperature setting an incubate egg systems.

Taking the above explanation, we propose to use control system based on FLC. The important part in FLC is during the process in selecting the membership function. The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, it represents the degree of an extension of valuation.

Conventional FLC used membership function generated by human operators, who have been manually designing the membership function of FLC. To satisfy such requirements including one common weakness where the membership function selection process is done with trial and error, it runs step by step, which is too long in completing the problem.

A new approach for optimum coding of fuzzy controllers is using GA. GA is used to determine membership function especially designed in situations. We use GA to tune the membership function for terms of each fuzzy variable.

2. Fuzzy Logic Control

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. The fuzzy logic controller block diagram is given in Figure 1, it shows a fuzzy controller integrated in a closed-loop control system. The *plant output* is denoted by *y(t)*, the *plant input* is denoted by *u(t)*, and the *reference input* to the *fuzzy controller* is denoted by *r(t)*.

Basically, we can view the fuzzy controller as an artificial decision maker that operates in a closed-loop system in real-time. In gathered plant output data *y(t)*, compare it to the reference input *r(t)*, and then decide what the plant input *u(t)* should be to ensure that the performance objectives will be met [3].

Performance of various control system can be analyzed by concentrating time response. The time response of a control system consists of two parts: the transient response and the steady-state response. The transient response of a practical control system often exhibits damped oscillations before reaching steady state. In this research, specifying the transient-response characteristics of a control system to a unit-step input it is common to specify the maximum (percent) overshoot. The maximum overshoot (M_p) is the maximum peak value of the response curve measured from unity.

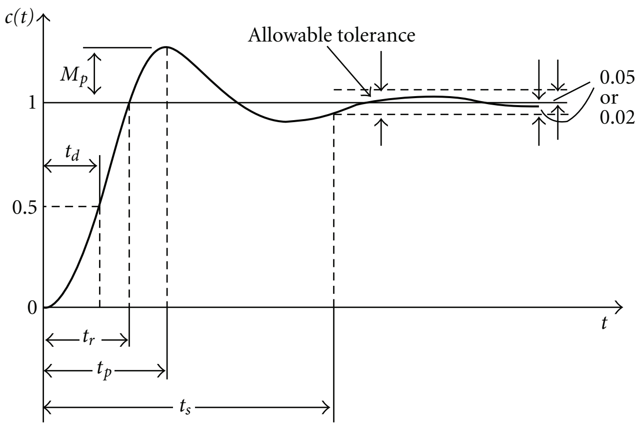


FIGURE 2: Transient and steady-state response.

The amount of the maximum (percent) overshoot directly indicates the relative stability of the system [1, 3, 4]. These specifications are defined in what follows and are shown graphically in Figure 2.

3. Genetic Algorithm

The GA borrow ideas and attempt to simulate Darwin's theory on natural selection and Mandel's work in genetics on inheritance. The usual form of genetic algorithms was described by Goldberg [5]. Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and natural genetics. Genetic algorithms, differing from conventional search techniques, start with an initial set of random solutions called "population." Each individual in the population is called a "chromosome," representing a solution to the problem at hand. For three variable problems hence, chromosomes will arrange three genes.

The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness. To create the next generation, new chromosomes, called offspring, are formed by crossover and mutation operator. A new generation is formed by selecting and rejecting. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimal solution of the problem.

4. Design and Implementation

Basically, genetic algorithms (GA) have had a great measure of success in search and optimization problems. In this research, the GA are used to improve the performance of the fuzzy controller. Considering that the main attribute of the GA is its ability to solve the topological structure of an unknown system, then the problem of determining the fuzzy membership functions can also fall into this category.

For obtaining final (tuned) membership function by using GA, some functional mapping of the system will be given. Parameters of the initial membership function

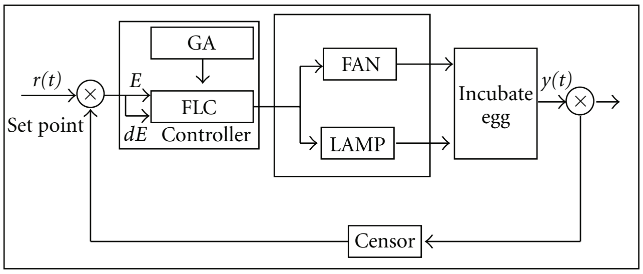


Figure 3: Block diagram of incubate egg by fuzzy genetic algorithms.

are then generated and coded as real numbers that are concatenated to make one long string to represent the whole parameter set of the membership function. A fitness function is then used to evaluate the fitness value of each set of membership function. Then the reproduction, crossover, and mutation operators are applied to obtain the optimal population (membership function), or more precisely, the final tuned value describes the membership function which is proposed.

Having now learnt the complicated procedures of designing FLC, a practical realization of this system is not easy to determine the *membership function* in FLC.

The conceptual idea is to have an automatic and intelligent scheme to tune the *fuzzy membership functions* of the closed-loop control for incubate egg, as indicated in Figure 3.

4.1. Data Structures. The most important data structures in GA are those that represent genes and chromosomes. Most researchers represent a chromosome as a string that is a binary code of a set of genes, but in our case, the real numbers code will be used; *real numbers code* is a more natural representation than *binary code* [6]. In our case, each *gene* corresponds to one *linguistic variable* whose definition is what the GA tries to evolve. In *fuzzy systems*, we represent the value of a *linguistic variable* by *membership function* and for our simulation design; there are three kinds of *linguistic variables*: the variable error signal as input-1 is parameter X, the rate of change in error as input-2 is parameter Y, and then the controller output is parameter Z, as shown in Figure 4.

For every variable, there are five shapes of *membership functions*; three are triangular and two trapezoidal ones. If *membership function* has triangular form, then it can be described by three parameters, a fixed number of *real number* is used to define each of the three parameters (which define completely the specific triangular membership function): if it is trapezoidal, it requires four parameters, as shown in Figure 5.

4.2. Fitness Function. An individual is evaluated, based on a certain function as the measurement performance. In the evolution of the nature, the highest valuable individual fitness will survive whereas the low valuable individual will die. The fitness calculation is a measure to know what the best particular solution to resolve the problem is.

The fitness function is the basis of the survival of the fittest premise of genetic algorithms. It is responsible for evaluating the parameter sets, and choosing which parameter sets are suitable. Since the fuzzy controller operates in a closed-loop specification, it can be analyzed by the maximum overshoot [1].

The *maximum overshoot* (M_p) is the maximum peak value of the response curve measured from the unity, and the amount of the maximum (percent) overshoot directly indicates the relative stability of the system.

4.3. Genetic Parameters. An individual is evaluated. The decision to make in implementing a genetic algorithm is to set the values for the various parameters, such as population size, probability of crossover rate, and probability of mutation rate. These parameters typically interact with one another nonlinearly, so they cannot be optimized for all situations. There are no conclusive results on what is the best; most people use what has worked well in previously reported cases. In our case, we use population size of 5, 10, and 100 for comparing, while the probability of crossover rate is 0.1, 0.7, and 0.9 where the probability of mutation rate is 0.001, 0.05, and 0.1.

4.4. Termination Conditions. Genetic algorithms will typically run forever, until an appropriate termination condition is reached. For our research, the termination condition was the one that defines the maximum number of generations to be produced. When the generation number is completed by the GA, the new populations generating process is finished, and the best solution is the one among the individuals more adapted to the evaluation function.

5. Result and Analysis

As mentioned before, most important to implementing a genetic algorithm for improving the performance is how to set values of the various parameters, such as population size, the probability of crossover, and mutation rate. These parameters typically interact with each other.

1st experiment: combination of population size: (5, 0.7, 0.001), (10, 0.7, 0.001), and (100, 0.7, 0.001).

2nd experiment: combination of mutation rate: (10, 0.7, 0.001), (10, 0.7, 0.005), and (10, 0.7, 0.05).

3rd experiment: combination of crossover rate: (10, 0.1, 0.001), (10, 0.7, 0.001), and (10, 0.9, 0.001).

4th experiment: combination of crossover rate: (10, 0.1, 0.05), (10, 0.7, 0.05), and (10, 0.9, 0.05).

Referring to researches available beforehand [7, 8], it seems that (10, 0.7, 0.001) is the best combination rate, and therefore that value is chosen for the tests, although further tests show that it does not give even optimum results.

For the first step, we make a comparison of convergence rates for populations of 5, 10, and 100 individuals; the probability of crossover rate is 0.7, and the probability of mutation rate is 0.001. All the data can be shown in the

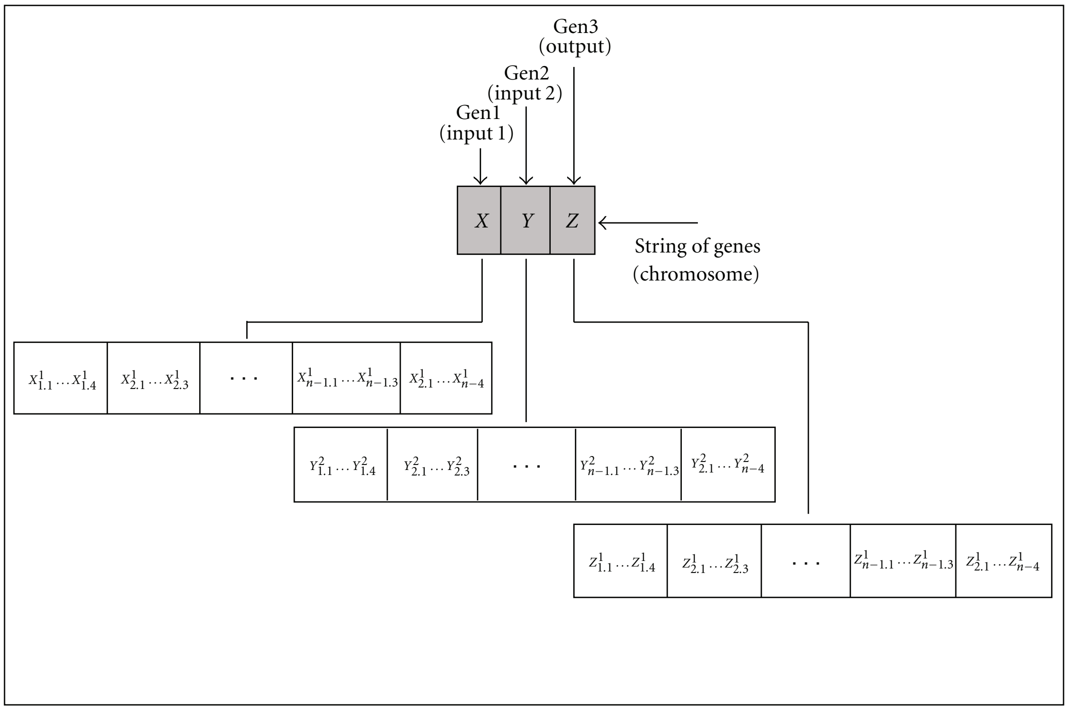
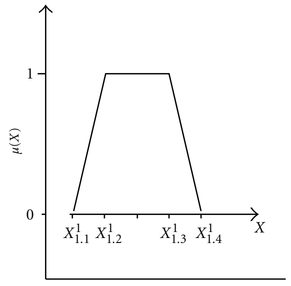
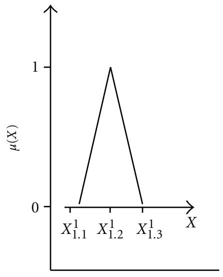

Gen3 (output)
Gen2 (input 2)
Gen1 (input 1)
X
Y
Z
String of genes (chromosome)
$X^1_{1.1}\cdots X^1_{1.4}$
$X^1_{2.1}\cdots X^1_{2.3}$
$\cdots$
$X^1_{n-1.1}\cdots X^1_{n-1.3}$
$X^1_{2.1}\cdots X^1_{n-4}$
$Y^2_{1.1}\cdots Y^2_{1.4}$
$Y^2_{2.1}\cdots Y^2_{2.3}$
$\cdots$
$Y^2_{n-1.1}\cdots Y^2_{n-1.3}$
$Y^2_{2.1}\cdots Y^2_{n-4}$
$Z^1_{1.1}\cdots Z^1_{1.4}$
$Z^1_{2.1}\cdots Z^1_{2.3}$
$\cdots$
$Z^1_{n-1.1}\cdots Z^1_{n-1.3}$
$Z^1_{2.1}\cdots Z^1_{n-4}$

Figure 4: Structure of chromosome.



$$\mu(X)=\begin{cases}0; & \text{for } x \le X^1_{1.1} \text{ or } x \ge X^1_{1.4}\\ \dfrac{(x-X^1_{1.1})}{(X^1_{1.2}-X^1_{1.1})}; & \text{for } X^1_{1.1} \le x \le X^1_{1.2}\\ 1; & \text{for } X^1_{1.2} \le x \le X^1_{1.3}\\ \dfrac{(X^1_{1.4}-x)}{(X^1_{1.4}-X^1_{1.3})}; & \text{for } X^1_{1.3} \le x \le X^1_{1.4}\end{cases}$$

Trapezoidal shape



$$\mu(X)=\begin{cases}0; & \text{for } x \le X^1_{1.1} \text{ or } x \ge X^1_{1.3}\\ \dfrac{(x-X^1_{1.1})}{(X^1_{1.2}-X^1_{1.1})}; & \text{for } X^1_{1.1} \le x \le X^1_{1.2}\\ \dfrac{(X^1_{1.3}-x)}{(X^1_{1.3}-X^1_{1.2})}; & \text{for } X^1_{1.3} \le x \le X^1_{1.2}\end{cases}$$

Triangular shape

Figure 5: Mathematical characterization trapezoidal and triangular MF.

following graph in Figure 6(a). These results show that the performance of the GA for first generation with a population size of 5 and 10 are the same, that is, 0.0799. Then this result is increased for next generation till the final generation. When the population size is 5, the fitness is 0.0817 at the 25th generation while the fitness is 0.0855 at the 25th generation when the population size is 10.

For the best result, the fitness is 0.0992 at 25th generation for the training data when the population size is 100, but very significantly, the consumed time is increased from 377 minutes when the population size is 10 to 985 minutes when the population size is 100. These results show that the performance of our GA is very sensitive to the population size.

From Figure 6(b), we can see the effect of a probability mutation rate on the fitness value. If the value of mutation rate is high, the fitness value gets better. The highest fitness value is 0.0932 for a probability mutation rate of 0.05. Secondly, it is 0.0872 (probability mutation rate = 0.005). The lowest value is 0.0855 (probability mutation rate = 0.001). For all these values the probability of crossover rate and the size of population are the same.

Figure 7(a) shows a comparison of convergence rates for three values of crossover rate 0.1, 0.7, and 0.9. In this case, the combination parameter (10, 0.9, 0.001) shows the best performance of GA, while the combination parameter (10, 0.1, 0.001) shows the lowest performance GA.

We can see the effect of a probability crossover rate on the fitness value. If the value of crossover rate is high, the fitness value gets better. The highest fitness value is 0.0880 for a probability crossover rate of 0.9. Secondly, it is 0.0855 (probability crossover rate = 0.7). The lowest value is 0.0825 (probability crossover rate = 0.1). For all these values the probability of mutation rate and the size of population are the same.

The Figure 7(b) shows a comparison of convergence rates for three values of crossover rate 0.1, 0.7, and 0.9. In this

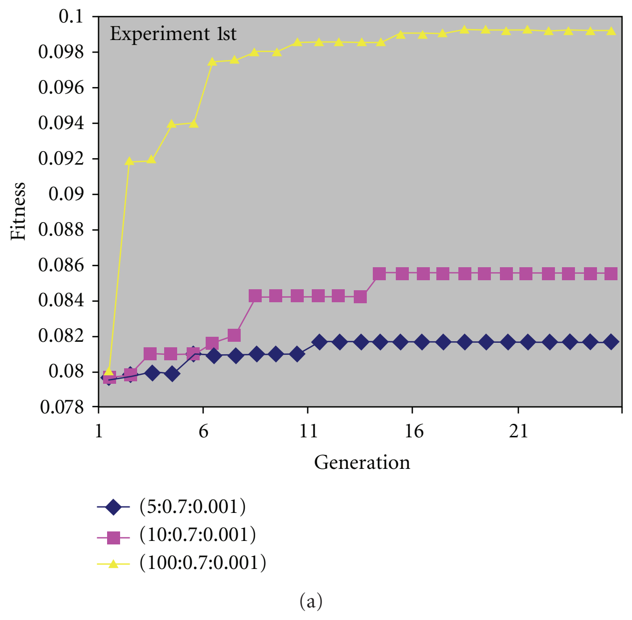


Figure 6: Comparison of convergent for (a) population size and (b) mutation rate.

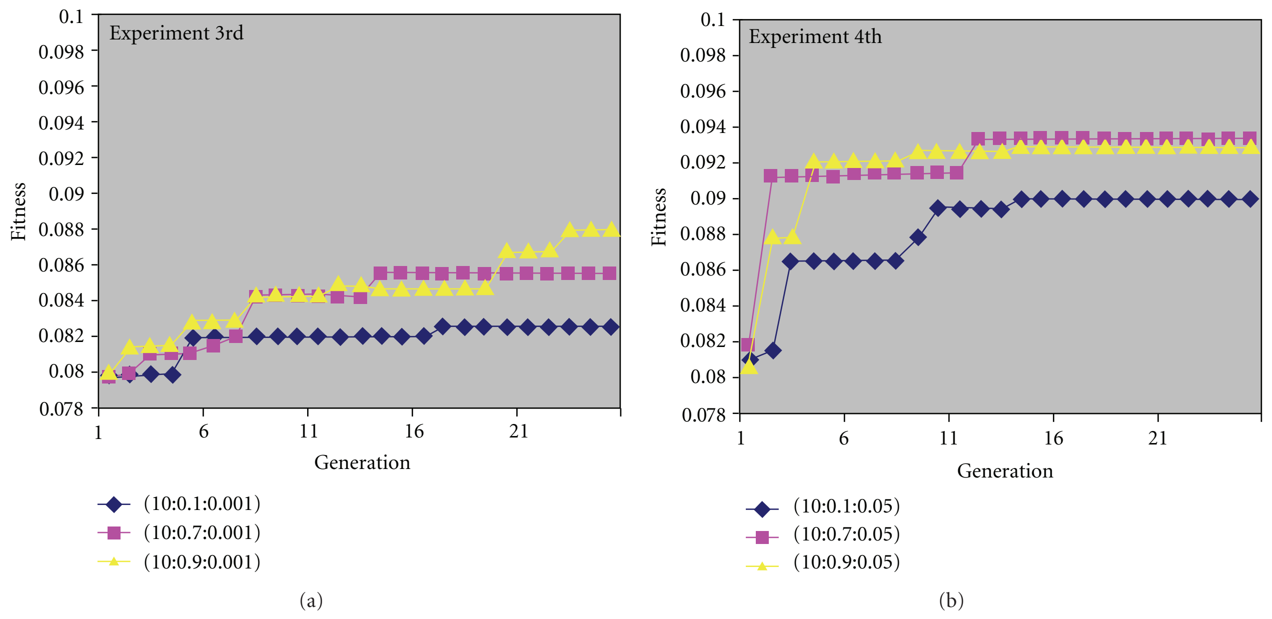
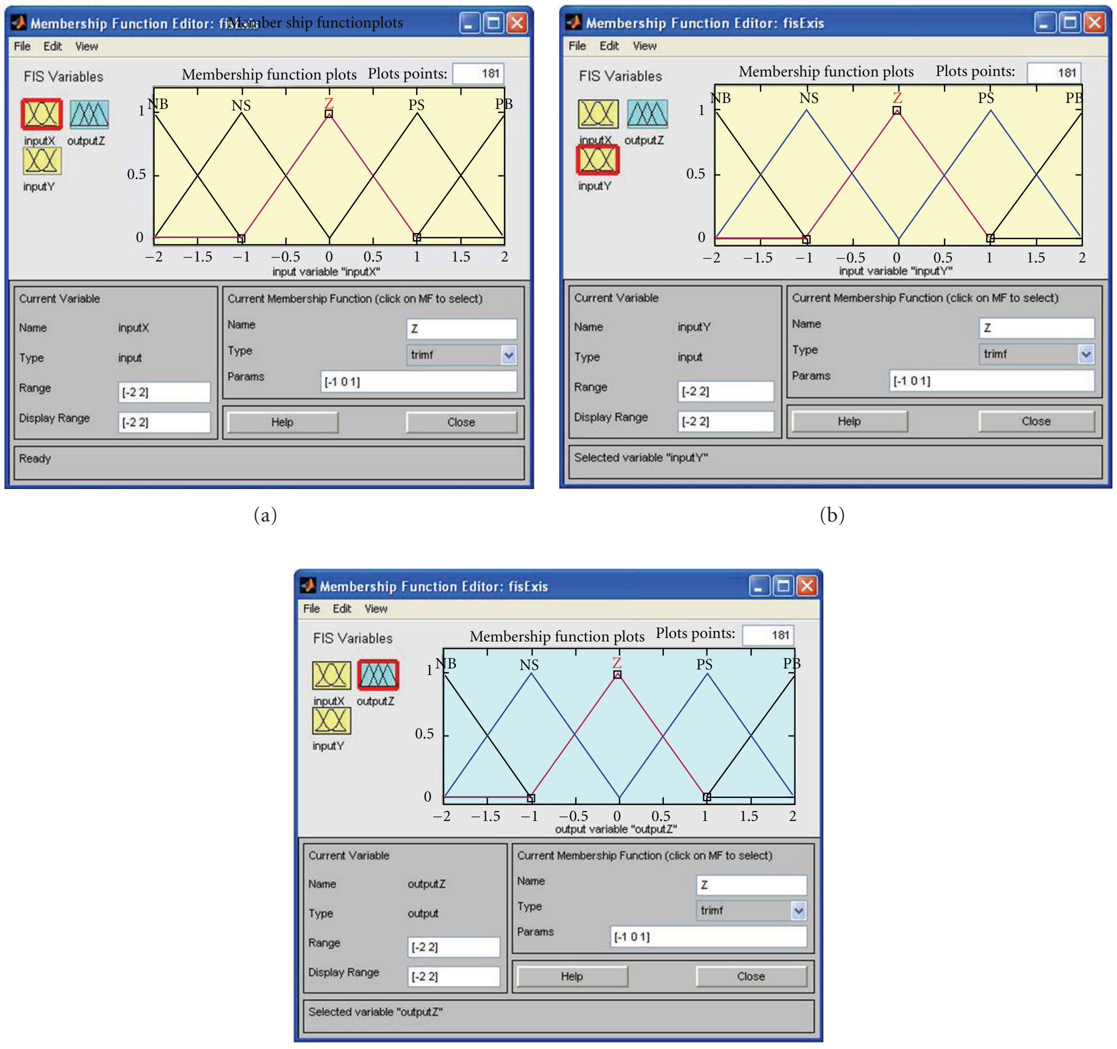


Figure 7: Comparison of convergent for crossover rate while mutation rates set to (a) 0.001 and (b) 0.05.

case, the combination parameters (10, 0.7, 0.05) show the best performance of GA while the combination parameters (10, 0.1, 0.05) show the lowest performance GA.

The comparison between Figures 7(a) and 7(b) shows the fitness value only a little bit different for probability crossover rates are 0.7 and 0.9, when the probability mutation rate sets to 0.05. This situation shows that the values of probability crossover and mutation rate interact; both of them will affect each other. The determination of the probability crossover and mutation rate is more important. The interaction between crossover rate and mutation rate is significant; both of them will affect each other. The parameters settings vary from problem to problem.

From the results presented in this experiment, the system which we developed is very helpful to determine the membership function for the fastest processing in completing the problem. Figures 8(a), 8(b), and 8(c) show screen produced from system; the membership function exists (without GA) and will be used in initial population for the first chromosome, in the first population and generation (existing membership function, without GA). Compare with Figures 9(a), 9(b), and 9(c), GA were applied into fuzzy



(a)
Plots points: 181
Current Variable
Name inputX
Type input
Range [-2 2]
Display Range [-2 2]
Current Membership Function (click on MF to select)
Name Z
Type trimf
Params [-1 0 1]
Ready

(b)
Plots points: 181
Current Variable
Name inputY
Type input
Range [-2 2]
Display Range [-2 2]
Current Membership Function (click on MF to select)
Name Z
Type trimf
Params [-1 0 1]
Selected variable "inputY"

(c)
Plots points: 181
Current Variable
Name outputZ
Type output
Range [-2 2]
Display Range [-2 2]
Current Membership Function (click on MF to select)
Name Z
Type trimf
Params [-1 0 1]
Selected variable "outputZ"

FIGURE 8: Existing membership (without GA): (a) input-1, (b) input-2, and (c) output.

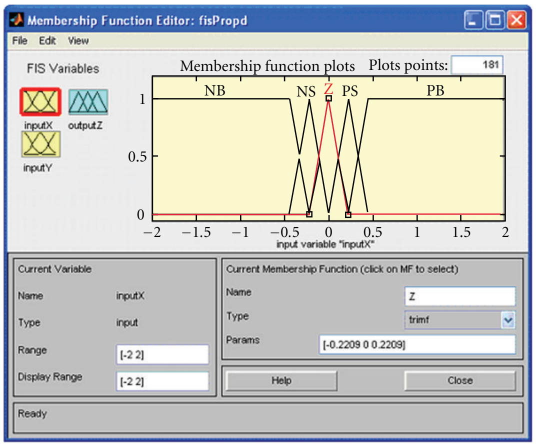
system to determine the membership function with GA/ FGA (proposed membership function).

During the execution of the GA, the fitness of each result is recorded. After evolution is complete, the evolved membership function is tested using the data and the results are compared with both FLC with and without GA. Generally speaking, after the membership function has been tuned with the GA, the improvement in performance of the FLC by using the GA is encouraged. We can compare fuzzy logic with and without genetic algorithms.

It is clear that GA are very promising in improving the performance of the FLC, to get more accurate in order to find the optimum result. From Figures 10 and 11, we can see that, after the execution of program and end of GA, the membership function is regulated automatically. We will get a better and exact result; the value of overshot is decreasing from 1.2800 for FLC without GA, to 1.0081 for FGA.

This is a research using *roulette wheel* or *fitness-based* method selection. But even each chromosome has a chance of selection directly proportional to its fitness but is done in a random manner, and then the effect of this situation causes the possibility of chromosomes with good fitness not joining as well as in the next generation because of not slipping away the selection. Therefore, we need the existence of a method of combination in the level of this selection that can maintain chromosomes with good fitness value so as these chromosomes can be maintained in the following generation.

Our research proposes one method, by recording the value of fitness of each chromosome for every generation. Afterwards we will choose chromosome with the best fitness to enter the further generation directly. It was stated [5, 9, 10] that GA can be represented by a sequence of procedural steps for moving from one population of artificial "chromosomes" to a new population. GA uses "natural" selection and genetic-inspired techniques known as crossover and mutation. Nature has an ability to adapt and learn without being told what to do. In other words, nature finds good chromosomes blindly.



Plots points: 181

Current Variable

Name inputX

Type input

Range [-2 2]

Display Range [-2 2]

Current Membership Function (click on MF to select)

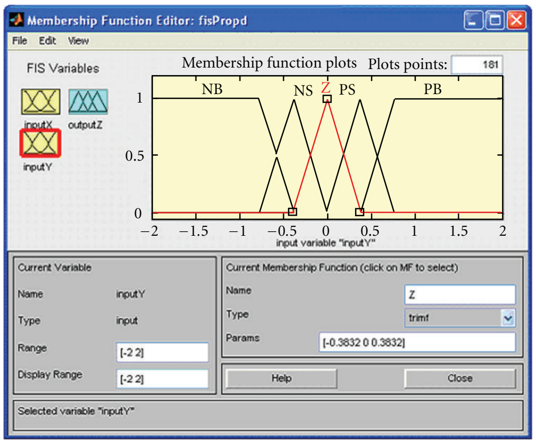
Name Z

Type trimf

Params [-0.2209 0 0.2209]

Ready

(a)



Plots points: 181

Current Variable

Name inputY

Type input

Range [-2 2]

Display Range [-2 2]

Current Membership Function (click on MF to select)

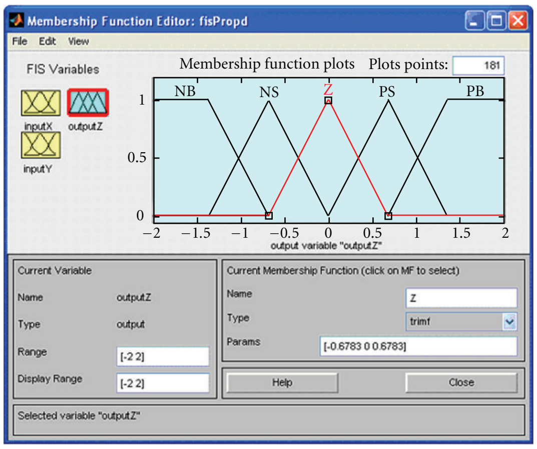
Name Z

Type trimf

Params [-0.3832 0 0.3832]

Selected variable "inputY"

(b)


Membership Function Editor: fisPropd
File Edit View
FIS Variables
inputX
outputZ
inputY
Membership function plots
NB
NS
Z
PS
PB
output variable "outputZ"
Help
Close

Plots points: 181

Current Variable

Name outputZ

Type output

Range [-2 2]

Display Range [-2 2]

Current Membership Function (click on MF to select)

Name Z

Type trimf

Params [-0.6783 0 0.6783]

Selected variable "outputZ"

(c)

Figure 9: Existing membership (with GA/FGA): (a) input-1, (b) input-2, and (c) output.

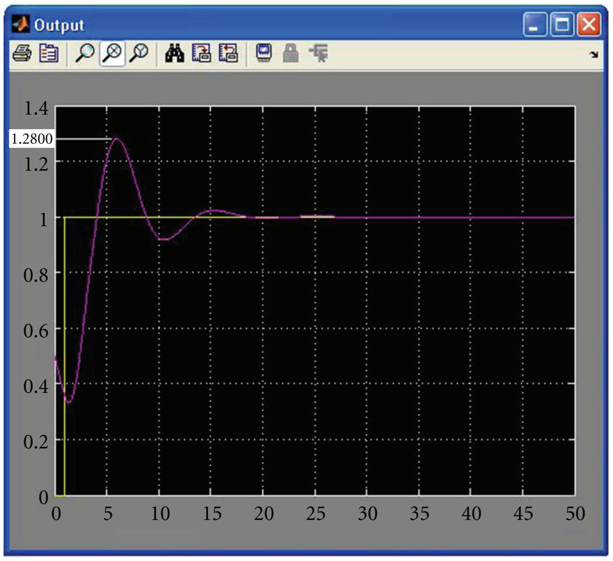


Figure 10: Response system for existing membership function (without GA).

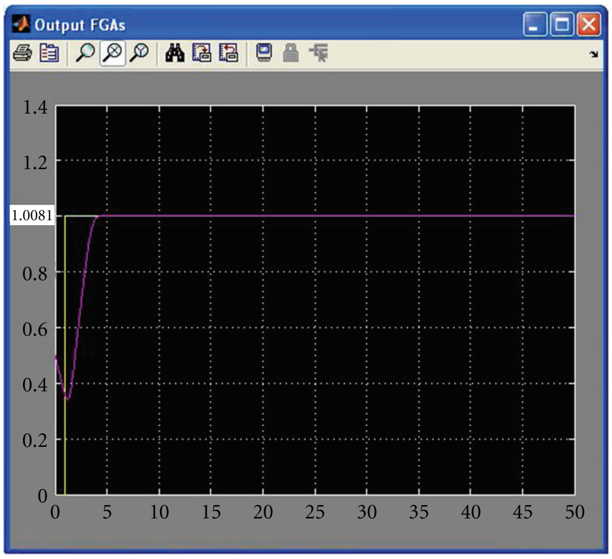


Figure 11: Response system for proposed membership function (using GA/FGA).

From our experiment, we found several matters which become important factors that influence performance of GA. They can be considered as follows.

(1) The selection procedure is used to choose suitable candidates for mating.

(2) The variable α for scaling factor in arithmetic crossover method.

(3) The value of r and b in dynamic or nonuniform mutation method.

6. Conclusion and Future Research

GA has been successfully applied to solve many optimization problems. In this research, GA are implemented to a system (programming language) for determining the *membership function* of FLC. By designing compact data structures for genes and chromosomes and an accurate fitness evaluation function, GA have been implemented which is very effective in finding more accurate membership functions for the fuzzy system. The data structures adopted are compact, and thus very convenient to manipulate by genetic operators.

From our experiment, we found that the *population size* was a significant factor to improve the performance of GA. Generally speaking, the larger *population size* will be better for performance of GA, but longer in processing time. A larger *population size* will be more diverse and thus will contain more *chromosomes*. A reasonable assumption held here is that, when more chromosomes are present, more good *chromosomes* will be present in the population. This is helpful to achieve a better solution.

GA need a longer time if probability of crossover and mutation rate is higher. So the interaction between *crossover rate* and *mutation rate* is significant; both of them will affect each other. The parameters settings vary from problem to problem.

In this research,

(i) combination parameter (10, 0.9, 0.05) will give the best value for solving our problem; the value of fitness is 0.0928 (maximum overshoot = 1.0776) and the processing time is 723 minutes.

(ii) Combination parameter (100, 0.7, 0.001) will give the best fitness value; the value of fitness is 0.0992 (maximum overshoot = 1.0081) and the processing time is 985 minutes.

The performance of GA can be further improved by using different combinations of selection strategies, *crossover* and *mutation methods*, and other genetic parameters such as population size, probability of crossover, and mutation rate.

References

[1] K. Ogata, *Modern Control Engineering*, Dorling Kindersley Pvt, India, 4th edition, 2008.

[2] W.-G. Du, L. Wang, and J.-W. Shen, "Optimal temperatures for egg incubation in two Geoemydid turtles: ocadia sinensis and Mauremys mutica," *Aquaculture*, vol. 305, no. 1–4, pp. 138–142, 2010.

[3] C. L. Phillips and R. D. Harbor, *Feedback Control System*, Prentice Hall, Upper Saddle River, NJ, USA, 4th edition, 2000.

[4] B. Boulet, *Fundamentals of Signal and Systems*, Da Vinci Engineering Press, Hingham, Mass, USA, 2006.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, Calif, USA, 1989.

[6] I. Yusuf, N. Iksan, and N. S. Herman, "A temperature control for plastic extruder used FGA," in *Proceedings of the International Multi-conference of Engineers and Computer Scientists (IMECS '10)*, pp. 1075–1080, Hongkong, March 2010.

[7] Netadelica, "Comparing different parameters for evolving a bit counter," 2010, http://www.netadelica.com/ga/ .

[8] A. J. Scholand, "The GA parameter setting," 2010, http://eislab.gatech.edu/people/scholand/gapara.htm.

[9] D. Todd, *Multiple criteria genetic algorithms in engineering design and operation*, Ph.D. thesis, University Of Newcastle, 1997.

[10] E. West, E. De Schutter, and G. L. Wilcox, "Using evolutionary algorithms to search for control parameters in a nonlinear partial differential equation," University of Minnesota, 55455 USA.