Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol II,
IMECS 2010, March 17 - 19, 2010, Hong Kong

A Temperature Control for Plastic Extruder Used Fuzzy Genetic Algorithms

Ismail Yusuf *[1], Nur Iksan [2], Nanna Suryana Herman [3]

Abstract—**This paper investigates an application of Genetic Algorithms (GA) in the design and the implementation of Fuzzy Logic Controllers (FLC). This idea used in a real case application called extruder for plastic. The comparison of various parameters shows that GA is helpful in improving the performance of FLC.**
A FLC is fully defined by its membership function. What is the best to determine the membership function is the first question that has be 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 is too long in completing the problem.
This research develops a system that may help users to determine the membership function of FLC using the GA optimization for the fastest processing in completing the problems. The data collection is based on the simulation results and the results refer to the 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 1.0011 for FGA.

Index Terms—**extruder, fuzzy logic, genetic algorithm, membership function, fitness function.**

I. Introduction

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 [9] has been easy to practice, fuzzy Logic Controllers (FLC) has 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 plastic extruders by FLC [7]. When extruding certain materials, the temperatures along the extruder must be accurately controlled in accordance with properties of the particular polymer and of the extruder. If the temperatures are not accurately controlled, the molten polymer will not be uniform and may decompose as a result of excessive temperatures.

One of the problems associated with the prior art extruder control systems occurs in the design of the barrel zone temperature controllers. 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 saturation of the controllers to set point changes. Thus it is important to select the accurate membership functions for temperature setting an extruder control 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 as 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 include 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 using GA. GA is used to determine membership function especially designed in situations. We use GA to tune of the membership function for terms of each fuzzy variable.

II. Fuzzy Logic Controller

Fuzzy logic process (fuzzy inferences) provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to

August 27, 2009.

*[1] Ismail Yusuf. Faculty of Information and Communication Technology, Technical University of Malaysia Malacca (UTeM). 76109 Durian Tunggal, Malaysia. (corresponding author to mail: Perum Bandaramas Blok D5 No.1 - KEPRI. 29464 Batam - Indonesia. By email: ariel_ismail@ yahoo.com).

[2] Nur Iksan. Department of Computer, Control and Electronic. Faculty of Engineering, Hasanuddin University (UNHAS). Perintis Kemerdekaan Rd, KM 10 Tamalanrea, 90245 Makassar. Indonesia.

[3] Prof. Dr. H. Nanna Suryana Herman. Faculty of Information and Communication Technology, Technical University of Malaysia Malacca (UTeM). 76109 Durian Tunggal, Malaysia.

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

control a system [14]. The fuzzy inference block diagram is given in Figure 1.

The fuzzy controller is composed of the following four elements [10]:

1. *Fuzzy rules* (a set of IF-Then rules), which contains a *fuzzy logic* quantification of the expert's linguistic description of how to achieve good control.
2. An *inference mechanism* (also called *inference engine* or *fuzzy inference module*), which emulates the experts decision making in interpreting and applying knowledge about how is the best to control the plant.
3. A *fuzzification interface*, which convert inputs into information that the *inference mechanism* can easily use to activate and apply *rules*.
4. A *defuzzification interface*, which converts the conclusions of the *inference mechanism* into actual inputs for the process.

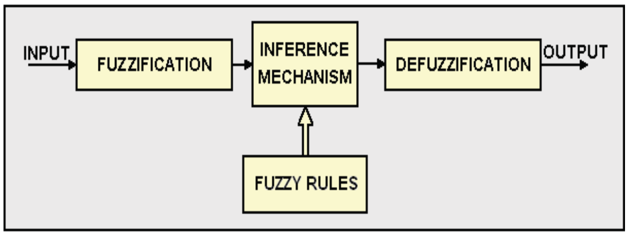


Figure 1 A block diagram of a fuzzy logic system

The *fuzzy logic controller* block diagram is given in Figure 2, where we show a *fuzzy logic system* integrated in a *closed-loop control system*. In a *closed loop control system* the *actuating error signal*, that is the difference between the input signal and the *feedback signal*, is fed into the controller so as to reduce the *error* and bring the output of the system to a desired value [2].

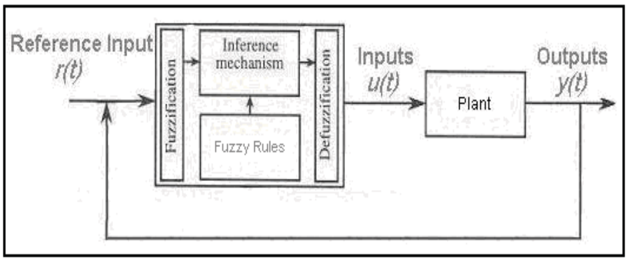


Figure 2 Fuzzy controller architecture

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)*. 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].

In analyzing and designing control system, we must have a basis of comparison of performance of various control system. 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*. By *transient response*, we mean the one which goes from the initial state to the final state. The *transient response* of a practical control system often exhibits damped oscillations before reaching *steady state*. In specifying the *transient-response characteristics* of a control system to a *unit-step input*, it is common to specify the following delay time, rise time, peak time, maximum overshoot and settling time [2, 3 and 9]. These specifications are defined in what follows and are shown graphically in Figure 3.

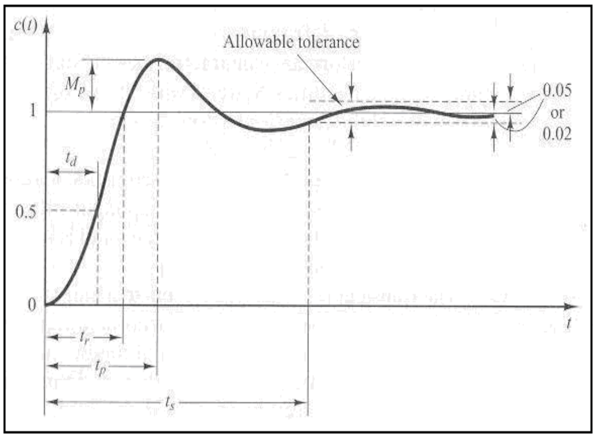


Figure 3 Transient and Steady-state response analyses

III. GENETIC ALGORITHMS

The GA borrow ideas and attempt to simulate Darwin's theory on natural selection and Mandel's work in genetic on inheritance. The usual form of genetic algorithms was described by Goldberg [4]. 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 [11]. To create the next generation, new chromosomes, called offspring, are formed by either:

(1) Merging two chromosomes from current generation using a crossover operator, and
(2) Modifying a chromosome using a mutation operator.

A new generation is formed by:

(1) Selecting, according to the fitness values, some of the parents and offspring, and
(2) Rejecting others so as to keep the population size constant.

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. GA are a class of stochastic search algorithms based on biological evolution.

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. 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 are then generated and coded as real numbers code that are concatenated to make one long string to represent the whole parameter set of 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 dynamic variation of fuzzy input membership functions is the main stumbling block to this design. Manually operating procedures for these variables may not only yield a sub-optimal performance, but can also be dangerous if the complete fuzzy set is augmented wrongly.

Hence, it is the purpose of this section to introduce the GA for the designing of FLC and has the following structure:

Begin
for $t \leftarrow 0$ to pop_size;
$\leftarrow$ [0,…1];
initialize $P(t)$;
repeat;
select a random number k from set x;
calculate corresponding membership function;
evaluate $P(t)$;
while (not termination condition) do
reproduction $P(t)$ to yield $C(t)$;
crossover $P(t)$ to ;
mutation $P(t)$ to yield $C(t)$;
evaluation $C(t)$;
select $P(t+1)$ from $P(t)$ and $C(t)$;
$t \leftarrow t+1$;
end
proposed membership function;
end

The conceptual idea is to have an automatic and intelligent scheme to tune the *fuzzy membership functions* of the *closed loop control* for extruder machine, as indicated in Figure 4. There are three possible zones in a thermoplastic screw. Since terminology is not standardized in the industry, different names may refer to these zones. Different types of polymer will have differing screw designs, some not incorporating all of the possible zones [7, 15]:

1. Feed zone. Also called solids conveying. This zone feeds the resin into the extruder.
2. Melt zone. Also called the transition zone. The resin is melted in this section.
3. Pressurizing zone. Also called metering or melt conveying. This zone gives the plastic uniform pressure and flow characteristics.

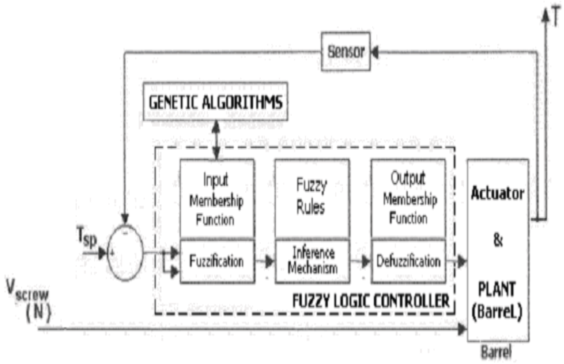


Figure 4 Block diagram Plastic Extruder by Fuzzy genetic Algorithms

Each zone will be equipped with one or more thermocouples for temperature control. Research by Ismail [7] was design a system control based on FLC for controlled the temperature so as the melting point that was wanted in the "pressurizing zone" could maintained. A mathematical model of plastic polymer for single screw extruder is given by [7]:

$$T(s) = \frac{(0.0023 \bullet Q_h) + (240 \bullet T_m) + (0.000343 \bullet T_u)}{6S + 240.000343} \qquad \text{……. (1)}$$

where:

Q_h	=	heat input rate (kcal/sec)
T_m	=	temperature of polymers (110°C)
T_u	=	temperature of outlet air (40°C)
$T(s)$	=	transfer function of temperature *t(s)*

A. 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 gene but in our case, the real numbers code will be used; real numbers code is a more natural representation than binary code [5].

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 and the rate of change in error as input-2 is parameter Y and then the controller output is parameter Z, as shown in Figure 5.

For every variable, there are five shapes of *membership functions*; three are triangular and two trapezoid. If

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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.

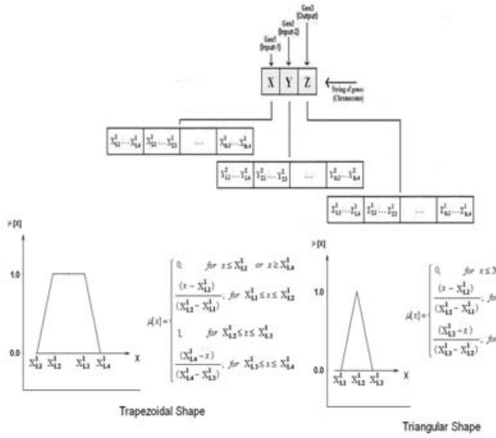


Figure 5 Structure of chromosome

B. Fitness Function

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

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. The most difficult part of the fitness function is to design the function for producing parameters that are reliable and effective. Since the fuzzy controller operates in a closed-loop specification, which can be analyzed by the maximum overshoot [9 and 13]. 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.

The fitness function measures how close each individual in the population meets the given specification at a given instant of time. Thus for a given individual in the population, the maximum overshoot is computed respectively by performing a simulation of the closed loop control system with the candidate controller (FGA) and model of the plant. The main objective of this research is to develop a system which may help users to determine the membership function of FLC for the fastest processing in completing the problems and more accurate in order to find the optimum result.

C. 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 which 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.

D. Termination Condition

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.

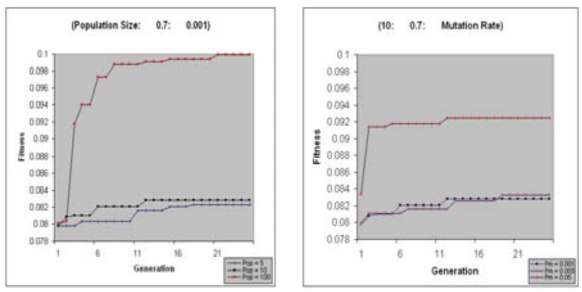
V. Result and Analysis

As mentioned before, the 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.

Experiment 1st: Combination of population size: (5: 0.7: 0.001), (10: 0.7: 0.001) and (100: 0.7: 0.001).
Experiment 2nd: Combination of mutation rate: (10: 0.7: 0.001), (10: 0.7: 0.005) and (10: 0.7: 0.05).
Experiment 3rd: Combination of crossover rate: (10: 0.1: 0.001), (10: 0.7: 0.001) and (10: 0.9: 0.001).
Experiment 4th: 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 [1, 6 and 10], 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 dies 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 show in the following graph in figure 6a.



a) b)

Figure 6 Comparison of convergent:
a) Population size, and b) Mutation rate

These results show that the performance of the GA for first generation with a population size of 5 and 10 are the same i.e. 0.0798. Then this result is increased for next generation till final generation. When the population size is 10, it is a little bit better. The fitness is 0.0823 at the 25th generation for the

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

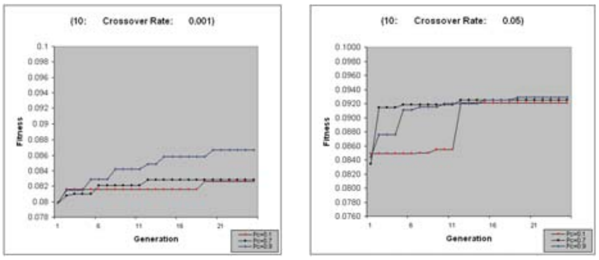
training data when the population size is 5, while it is 0.0828 at the 25th generation when the population size is 10.

For the best result, the fitness is 0.0999 at 25th generation for the training data when the population size is 100, but very significant, the consumed time is increased from 370 minutes when the population size is 10, to 1205 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 6b, 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 0.0925 for a probability mutation rate of 0.05. Secondly it is 0.0833 (probability mutation rate of 0.005). The lowest value is 0.0828 (probability mutation rate of 0.001). For all these values the probability of crossover rate and the size of population are the same.

Figure 7a 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, which 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 0.0867 for a probability crossover rate of 0.9. Secondly it is 0.0828 (probability crossover rate of 0.7). The lowest value is 0.0826 (probability crossover rate of 0.1). For all these values the probability of mutation rate and the size of population are the same.



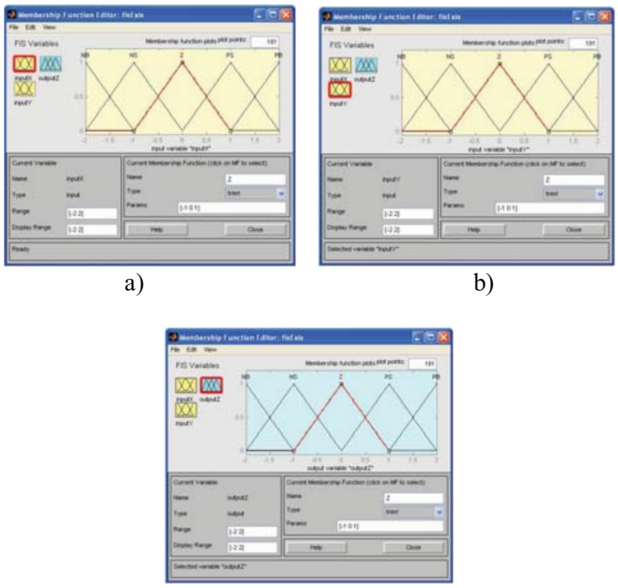
a) b)

Figure 7 Comparison of convergent for crossover rate, while mutation rate set to a) 0.001 and b) to 0.05

The Figure 7b shows a comparison of convergence rates for three values of crossover rate 0.1, 0.7 and 0.9. In this case, the combination parameters (10: 0.9: 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 Figure 7a and Figure 7b shown the fitness value only a little bit different for any value for probability crossover rate, 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 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.

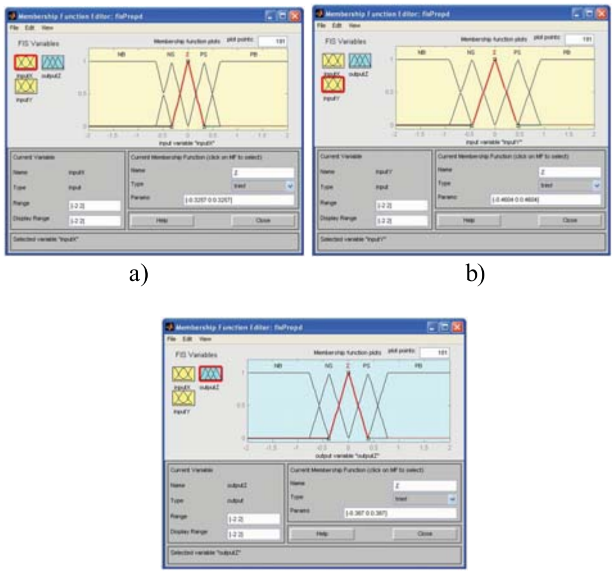
From the results presented in this experiment, the system which we developed is very helpful to determine membership function for the fastest processing in completing the problem. Figure 8a, 8b and 8c shows screen produced from system; the *membership function* exists (without GA) and will be used in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



a) b) c)

Figure 8 Existing membership (Without GA): a) input-1, b) input-2, and c) output

Compare with Figures 9a, 9b and 9c, genetic algorithms were applied into fussy system to determining the membership function with GA (proposed membership function).



a) b) c)

Figure 9 Proposed membership (With GA / FGA): a) input-1, b) input-2, and c) output

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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 both FLC with and without GA. Generally speaking, after the membership function have been tuned with the GA, the improvement in performance of the FLC by using the GA is encouraged.

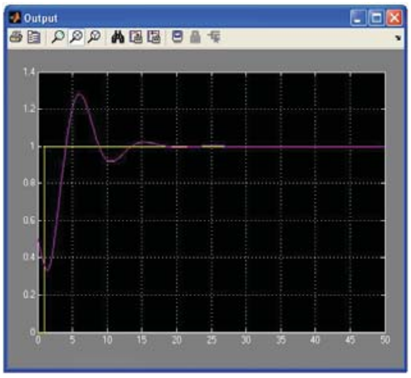


Figure 10 Response system for:
Existing membership function (without GA)

We can compare about *fuzzy logic* with and without genetic algorithms thus visualization.

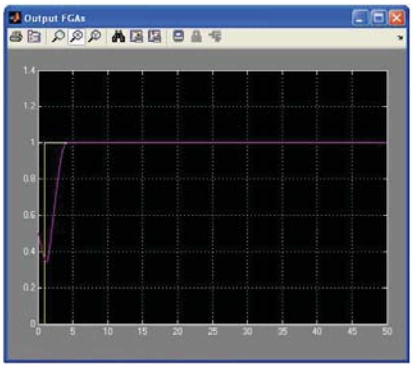


Figure 11 Response system for:
Proposed membership function (with GA / FGA)

From Figures 10 and 11, we can see that, after the execution of program and end of genetic algorithms, the membership function is regulated automatic. We will get a better and exact result; the value of overshot is decreasing from 1.2800 for FLC without GA, to 1.0011 for FGA. It is clear that the GA are very promising in improving the performance of the FLC, to get more accurate in order to find the optimum result.

VI. Conclusion And Future Research

GA has been successfully applied to solve many optimization problems. In this research, genetic algorithms 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.

For this research, the probability of crossover rate is 0.9 and the probability of mutation rate is 0.05, while the population size used is 10. the value of overshot is decreasing from 1.2800 for FLC without GA, to 1.0011 for FGA. We mean the value of fitness function is not so bad and not longer in processing time.

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] Andrew Wirth Dr, Nasir Hayat. *Genetic Algorithms and Machine Scheduling With Class Setups*. University of Melbourne, Australia. 2008
[2] Benoit Boulet. *Fundamentals of Signal and Systems*. Hingham, MA: Da Vinci Engineering Press. 2006
[3] Charles L. Phillips and Royce D. Harbor. *Feedback Control System*. Upper Saddle River, N. J.: Prentice Hall. 1996
[4] David E Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. California: Addison Wesley Publishing . 1989
[5] David Todd. *Multiple Criteria Genetic Algorithms In Engineering Design And Operation*. PhD Thesis. University Of Newcastle; 1997
[6] Eicher West, Schutter E.De, Wilcox. *using Evolutionary Algorithms to Search for Control Parameters in a Nonlinear partial Differential Equation*. University of Minnesota, 55455 USA.
[7] Ismail. *Analysis the Controller Temperature Set for Maintained the Melting Point of Plastic Extruders by FLC*. Thesis. University of Jayabaya Indonesia; 2005
[8] Jinwoo Kim and Bernard P. Zeigler. Hierarchical Distributed Genetic Algorithms: A Fuzzy Logic Controller Design Application. *IEEE*, 1996. pp. 76-84
[9] Katsuhiko Ogata. *Modern Control Engineering*. 3rd ed. Upper Saddle River, N. J.: Prentice Hall. 1997
[10] Kevin M. Passino and Stephen Yurkovich. 1998. *Fuzzy Control*. Menlo Park, California: Addison Wesley Longman.
[11] Mitsuo Gen and Runwei Cheng. 1997. *Genetic Algorithms & Engineering Design*. New York: John Wiley & Sons.
[12] Netadelica. (2007, March 16). *Comparing Different Parameters for Evolving a Bit Counter*. Available: http://www.netadelica.com/ga/
[13] Peter Vas. Artificial Intelligence Based: *Electrical Machines and Drives*. Oxford University Press, USA. 2000
[14] Shehu S. Farinwata, Dimiltar Filev, Reza Langari. F*uzzy Control: Synthesis And Analysis*. *IEEE*, 2001. pp. 95
[15] Wikipedia. (2007, June 5). *Plastic Extucion*. [Online]. Available: http://en.wikipedia.org/wiki/Plastics_extrusion

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)