Application of Genetic Algorithm to Hybrid Fuzzy Inference Engine

Sae-hie Park*, Sun-tae Chung**, Hong-tae Jeon*

*Dept. of Electronic Eningeering, Chung-Ang Univ.
**Dept. of Electronic Eningeering, Soong-Sil Univ.

Abstract

This paper presents a method on applying Genetric Algorithms(GA), which is a well-known high performance optimizing algorithm, to construct the self-organizing fuzzy logic controller. Fuzzy logic controller considered in this paper utilizes Sugeno's hybrid inference method, which has an advantage of simple defuzzification process in the inference engine. Genetic algorithm is used to find the optimal parameters in the FLC. The proposed approach will be demonstrated using 2 d. o. f robot manipulator to verify its effectiveness.

I. Introduction

Fuzzy logic is well known by its merit of stating the impresice nature of the real world by the fuzzy implication and the compositional rule of inference. Fuzzy logic controller(FLC) is the most outstanding field of the application of fuzzy logic by its simplicity of implementation. The FLC usually consists of fuzzifier, rule base, inference engine and defuzzifier. One of the important points in construction of the FLC is to obtain the proper rule base and the parameters in the fuzzifier and the defuzzifier. For this purpose, several approaches to construct the self-organizing rule base and find the proper parameters have been proposed by fusion with neural networks[1] and genetic algorithm(GA)[2][3].

The FLC may be classified into the direct, indirect and hybrid ones by the inference method. In this paper, the hybrid inference method is adopted. And an approach that can construct the self-organizing rule base and can determine the proper parameters in the process of fuzzifier and defuzzifieris proposed using GA.

This paper is organized as follows. The general concept of GA is explained in section . and the self-organizing hybrid FLC by using GA is described in section. In section . the computer simulation of the proposed approach is shown. Also in section . another modified SGA is used as an alternative generation. Finally in section the conclusions are discussed.

Ⅱ. Genetic Algorithm

Genetic algorithm is a robust search algorithm based on natural selection and evaluation [4][5][6]. It differs from the traditional optimizing algorithms from some aspects: 1) using coding of parameter, 2)using population of search points, 3)not requiring any mathematical information of the function - derivatives, continuity etc.

In GA, all the parameters to be optimized are decoded from the strings in the population and the coding is usually done in the binary numbers. Then three major operators, which are reproduction, crossover and mutation, are applied to find the optimal value. The reproduction operator copies some parts of strings in the population to form a gene pool, with the selection chance according to the fitness value. Each string that constructs the gene pool is called the parent. And crossover operator selects two individual genes with a crossover rate, and exchange the parts of strings to form new strings from the previous search results. In the crossover operator, the cutting point, at which the bits are exchanged, is usually called crossover site and this value is selected by random generation of integer value.

Fig. 1 shows an example where two new strings (A_1', A_2') are generated from (A_1, A_2) through the crossover with crossover site 3.

$$A_1 = 00000000 \rightarrow A_1' = 00011111$$
$$A_2 = 11111111 \rightarrow A_2' = 11100000$$

Fig. 1 Crossover of two strings

Meanwhile the mutation opertator is a bit exchange from 1 to 0 and vise versa. This operator is a second operator and has a lower rate than that of crossover. By mutation operator, the escape from the premature saturation can be done.

The GA consisting of the obove major operators is called Simple Genetic Algorithm(SGA). The SGA can be summerized as follows.

[Step 1]. Set string size and population size and crossover/mutation rates
[Step 2]. Initialize initial population
[Step 3]. Convert string to parameters
[Step 4]. Evaluate parameter fitness
[Step 5]. If all strings are tested, continue to step 6 and else go to step 3
[Step 6]. Make new population by reprodectuion, crossover and mutation
[Step 7]. Go to step 3 until some condition is satisfied

Ⅲ. Construction of Self-organizing Hybrid FLC using GA

The FLC can be devided into three major classes by its inference method - direct, indirect, and hybrid. A well known direct method is Mamdani's min-max composition. Mean while the

Sugeno's hybrid one has a linear equation in the consequent of the rule. The hybrid type rules with two input variables are represented as follow.

$$\begin{aligned}
&\text{Rule 1: if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } f_1=a_1{*}x_1+b_1{*}x_2+c_1\\
&\text{Rule 2: if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } f_2=a_2{*}x_1+b_2{*}x_2+c_2\\
&\quad\vdots\\
&\text{Rule N: if } x_1 \text{ is } A_n \text{ and } x_2 \text{ is } B_n \text{ then } f_n=a_n{*}x_1+b_n{*}x_2+c_n
\end{aligned} \tag{1}$$

Here x1 and x2 are the input values, and Ai, Bi(i=1.n) are the input lingistic variables. The merit of this infernece method is that the defuzzification process in the inference engine can be accomplished without a particular defuzzifier as follows.

$$y^* = \frac{\sum A_i(x_1)\cdot B_i(x_2)\cdot f_i(x_1, x_2)}{\sum A_i(x_1)\cdot B_i(x_2)} \tag{2}$$

where $A_i(x_1)$, $B_i(x_2)$, $f_i(x_1,x_2)$ are, respectively, the grade of membership value of input x_1 of lingusitic value A_i, input x_2 of B_i and linear function evaluation value.

However, the effectiveness of the above hybrid inference metod depends on the following facts; 1)how to determine the membership functions of linguistic variables, 2)how to choose the parameters (a_i, b_i, c_i, for i=1,2,...,n) of the consequents and 3)how to select the universe of discourse[7].

Tagagi and Sugeno[8] proposed an approach that these parameters could be determined by the given input-output relations and a stable Kalman filter. Also S. Horikawa[1] suggested a Fuzzy Neural Network to find the parameters. In this method, the input and output relations must be known to teach the neural network. But this approach may have some difficiulities in obtaining the exact input-output relations and the optimal values of the parameters.

Thus an approach that can overcome the above difficulties and determine the proper values of a_i, b_i and c_i and automatically adjust the membership function to the control purpose is explained in the followings.

1) Membership function of the antecedent.

Two representative shapes of the membership functions of the fuzzy variables are the triangle and exponential(bell) shapes. In this paper, the membership function of bell shape is chosen as follows

$$\mu(x) = \exp(-(x-c)^2/w^2)) \tag{3}$$

where the value 'c' is the center of the bell on the universe of discourse with $\mu(c) = 1.0$ and the value 'w' adjusts the width of the bell. In Eq. (3), if 'w' is small, the shape will be narrow, and become less fuzzier, and if 'w' is large, the membership function becomes more fuzzier.

The above parameters 'w' and 'c' are coded into the binary numbers to apply the GA.

2) Parameters of the consequent.

The consequent of each rule has three parameters a_i, b_i and c_i. The parameters a_i and b_i are coded into the unsigned binary system and the parameter c_i value, which is an offset, is coded into the signed binary system. By finding the optimized values of parameters by GA, the proper consequent of the rules can be self-organized.

3) Fitness function

The object of the FLC is to control the process with a given command with the minimum error and energy. These profits are used to construct the fitness value of the strings of the FLC. But since GA is a maximizing algorithm, it is necessary to map the minimum payoff into maximum profit. In this paper, the inverse of payoff to construct the fitness function is used as follows.

$$\text{Fitness} = \frac{K_1}{\sum |\text{error}|} + \frac{K_2}{\sum |\text{change of error}|} + \frac{K_3}{\sum |\text{applied energy}|} \tag{4}$$

where K_1, K_2 and K_3 are the normalizing coefficients.

With the above parameters to be optimized and the fitness function, the process of self-organizing of the FLC by using GA can be summerized as follows.

[Step 1]. Decide the length of the string by the number of the parameters of the FLC
[Step 2]. Construct the population by random generation of strings
[Step 3]. Decode the string to constuct the rule base and the parameters of the FLC
[Step 4]. Calculate the fitness by controlling the process with a test command
[Step 5]. Do steps 3 and 4 until the fitness values of all strings are calculated
[Step 6]. Do generation by reproduction, crossover and mutation
[Step 7]. Go to step 3 until some condition is satisfied

Ⅳ. Experimental results by computer simulation

To verify the effectiveness of the proposed approach, computer simulations are done by using a 2 d. o. f robot manipulator which has the mechanical properties as shown in Table 1.

Table 1. The length and mass of a 2 d. o. f robot

	Link 1	Link 2
Length	0. 53m	0. 47m
Mass	4.5Kg	4.0Kg

And the codings of each parameters are accomplished by 8bit signed/unsigned binary number system and crossover and mutation rate are selected to 0.66, 0.033, respectively. Also the population size is set to 80 and each input value has three linguistic variables which are NEgative, ZerO and POsitive.

With the above robot and GA parameters, a rectangular trajectory is given as a test command and two runs are executed. Fig. 2. shows the variation of the fitness value as the generation progress. It can be seen from Fig. 2, that the fitness value goes to a steady state with some oscillations. The steady-state value can be consider as an optimum one. With these optimum parameter, the rectangular trajectory which are accomplished by the robot is shown in

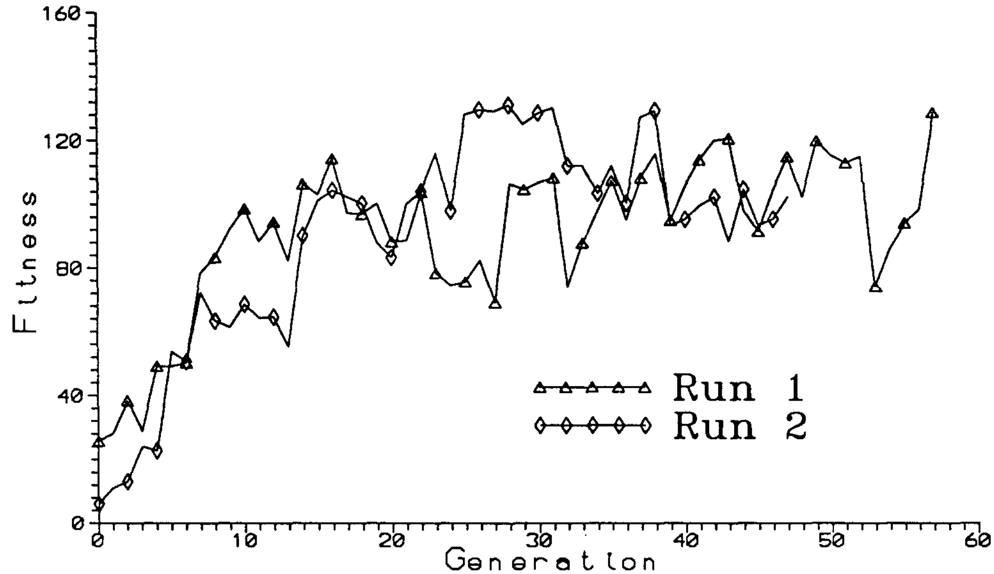


Fig. 2 Variation of fitness value

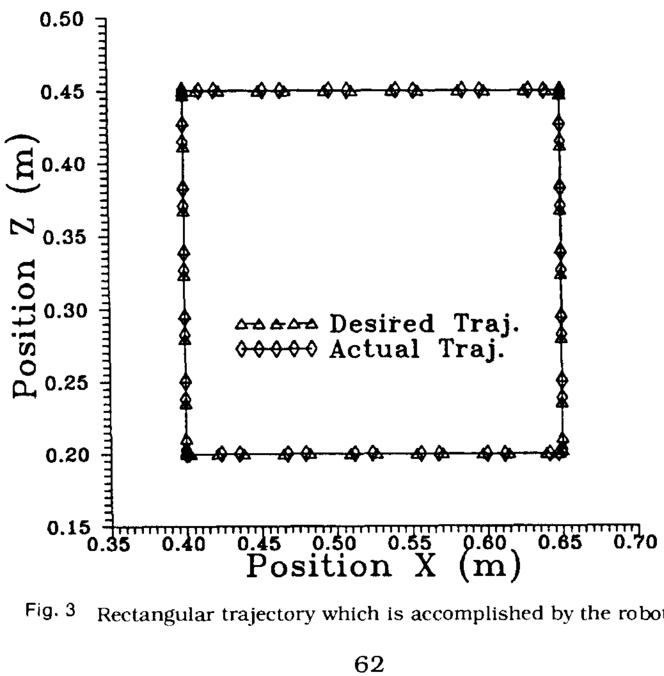


Fig. 3 Rectangular trajectory which is accomplished by the robot

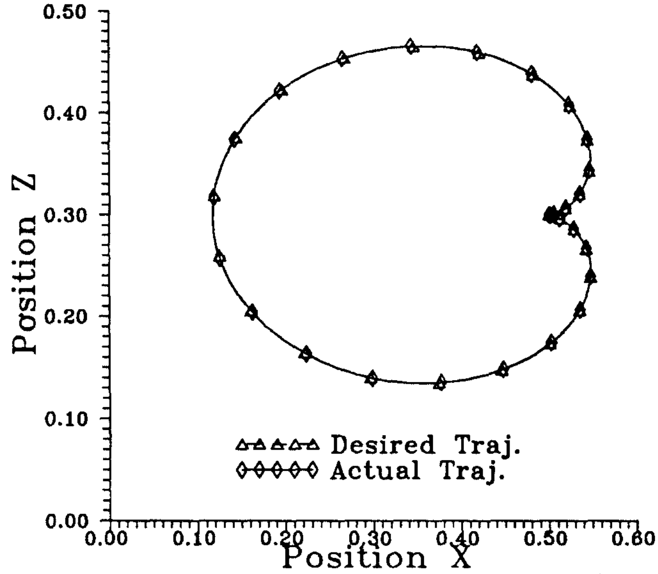


Fig. 4 Heart shape trajectory

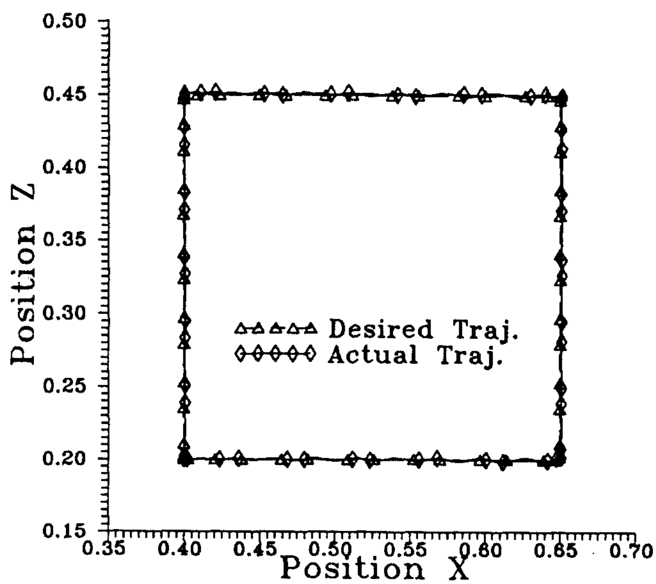


Fig. 5 Rectangular trajectory with variation of payload

Fig. 3. and the heart trajectory in Fig. 4. Also, when increasing the weight of the second link by 250%, the simulation results are shown in Fig. 5. It can be realized that the proposed approach can be also robust against the parameter variation of the robot.

Ⅴ. Modified SGA

The previous results show that the highest fitness value ocsillates in the steady state. This fact means that the string of the highest fitness value can be lost in the subsequent generations. So we adopt another method called as the micro-GA[9] to preserve the highest fitness string in the population to contribute to sequent generations.

Two strings of the population are selected by selection operator and crossover, and mutaion are accomplished by the rate of 1.0 and 0.033 respectively. Then the string of the least value of fitness is deleted from the population and one of new two strings is replaced in this place. By repeating this procedure, the string of the highest fitness value can be conserved and will contribute to the succeeding evaluations. Fig. 6 show the highest value in each evaluation. The highest value is slightly higher than that of SGA.

The relations between evaluation process and generation number are as follows. One generation has the evaluation process by the amount of strings in the population. For example, the population 80 and the generation number 30 would be accomplished by 2400 evaluation number, or on the contrary 4000 evaluation number is equal to the 50 generations.

Table 2 and 3 show the parameters found by SGA and modified SGA respectively. Also Fig. 7 shows the membership functions of the antecedent lingistic variables abtained from GA.

Table 2. The parameters found by SGA

		FLC of link 1			FLC of link 2		
Err	Cerr	a	b	c	a	b	c
NE	NE	2. 4532250	1. 7501000	0. 3281250	1. 7969750	2. 8282250	0. 6562500
	ZO	3. 2969750	1. 0001000	0. 1093750	2. 6563500	3. 7657250	-1. 9687500
	PO	2. 8126000	1. 9844750	-1. 5625000	1. 6407250	2. 2188500	1. 7187500
ZO	NE	1. 3594750	2. 8126000	1. 5937500	3. 5782250	0. 3907250	-1. 7500000
	ZO	3. 9532250	1. 7969750	0. 1875000	3. 8126000	0. 8751000	-0. 0625000
	PO	1. 7657250	3. 7188500	-1. 0000000	3. 2501000	2. 8126000	-1. 7812500
PO	NE	3. 1407250	2. 9688500	0. 9375000	1. 2657250	1. 4219750	0. 1093750
	ZO	3. 1563500	2. 0938500	0. 4062500	2. 1719750	3. 5782250	-0. 2031250
	PO	0. 2813500	3. 4688500	-0. 0781250	0. 3751000	0. 7657250	1. 1718750

Table 3. The parameters found by modified SGA

Err	Cerr	FLC of link 1 a	FLC of link 1 b	FLC of link 1 c	FLC of link 2 a	FLC of link 2 b	FLC of link 2 c
NE	NE	1. 2657250	3. 4844750	1. 3281250	3. 8282250	2. 5157250	0. 7968750
	ZO	3. 9688500	0. 0782250	-1. 7656250	2. 1719750	0. 7032250	-1. 9531250
	PO	3. 4844750	3. 5469750	-0. 0937500	3. 8282250	0. 3438500	-1. 5000000
ZO	NE	3. 1407250	3. 2657250	1. 5625000	1. 0626000	2. 8126000	-0. 2812500
	ZO	3. 6876000	0. 5626000	0. 7500000	3. 0626000	3. 7813500	0. 0937500
	PO	3. 5001000	0. 9219750	-0. 0156250	0. 1407250	0. 3438500	0. 3125000
PO	NE	0. 1407250	1. 4688500	0. 7187500	1. 4219750	0. 5469750	-1. 0781250
	ZO	2. 7032250	2. 3594750	-0. 4062500	3. 0626000	1. 0469750	0. 8281250
	PO	0. 2032250	0. 2344750	0. 3593750	3. 2969750	1. 0469750	0. 3906250

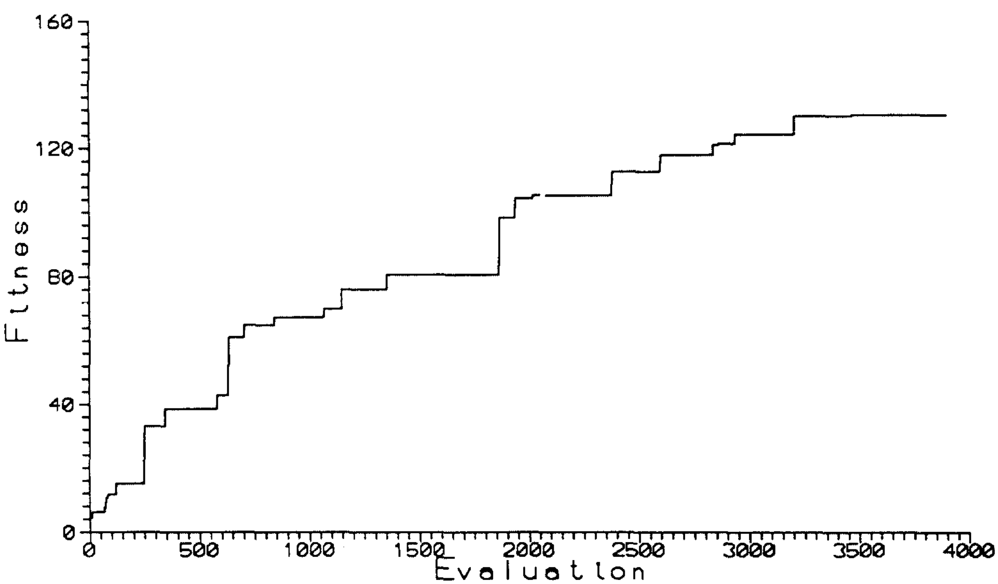
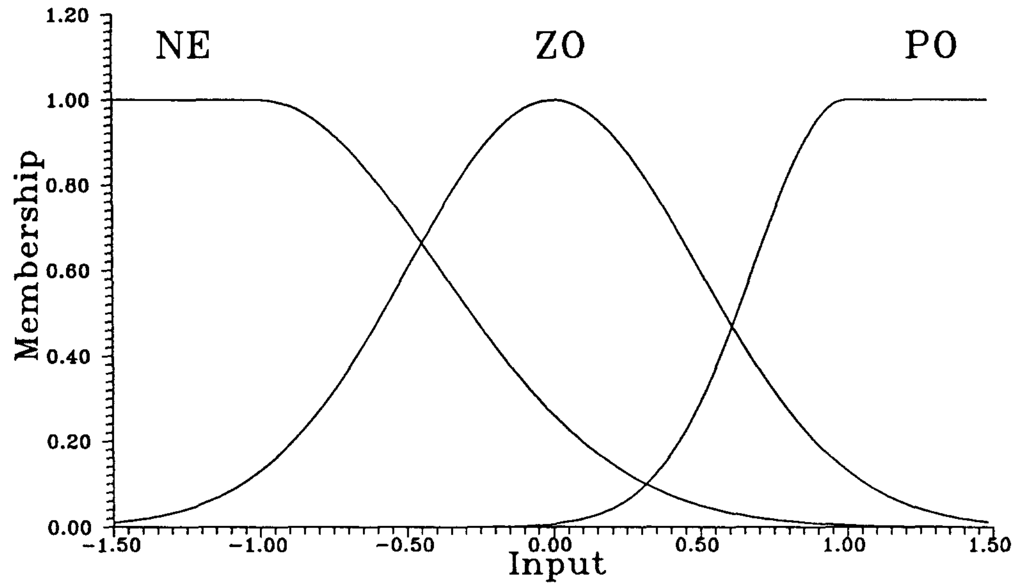
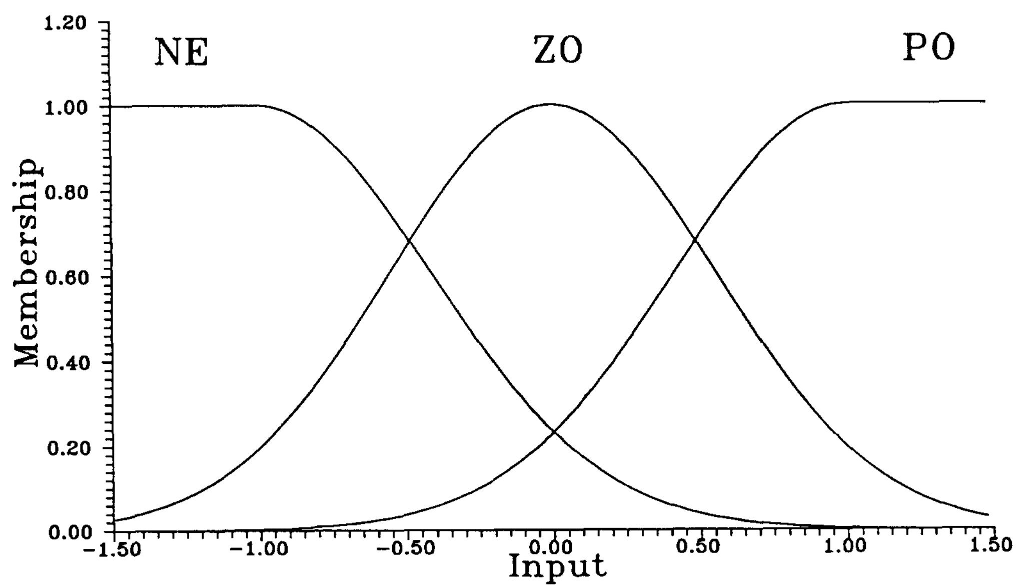


Fig 6. Fitness function of modified SGA



(a)Membership of error



(b)membership of change of error

Fig. 7 Membership function of input lingistic variables.

Ⅶ. Conclusion

This paper proposes a self-organizing FLC, which is based on Sugeno's hybrid inference method, by using genetic algotithm. The proposed approch can overcome the difficulties in determining the membership function and the parameters of the fuzzy rule base. This fact can be verified by computer simulation.

References

1. Shin-ichi Horikawa, Takeshi Furuhashi, Toshiki Uchikawa and Takashi Tagawa, "A Study on Fuzzy Modeling using Fuzzy Neural Networks," Fuzzy Engineering toward Human Friendly Systems, Vol. 1, pp. 562 - 573, 1991
2. Charles L. Karr, "Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm," Proceeding on the fourth international conference on Genetic Algorithm, 1991, pp. 450 - 457
3. Philip Thrift, "Fuzzy Logic Synthesis with Genetic Algorithms," Proceeding on the fourth international conference on Genetic Algorithm, 1991, pp. 509 - 513
4. D. E. Gordberg, Genetic Algorithms in Search Optimization & Machine Learning, Addison Wesley, 1989
5. Yuval Davidor, Genetic Algorithms and Robotics, World Scientific, 1991
6. Lawrence Davis, HandBook of Genetic Algorithm, Van Nostrand Reinhold, 1991
7. M. Braae and D. A. Rutherford, "Selection of parameters for a fuzzy logic controller," Fuzzy Sets and Systems 2(1979), 185-199, North-Holland Pub. Comp.
8. T. Tagagi, M. Sugeno, "Fuzzy Identification of Systems and Its Application to Modeling and Control," IEEE Transaction on Systems, Man, and Cybernetics, Vol SMC-15, NO. 1, January/February, 1985
9. K. S. Krishnakumar. "Micro genetic algorithms for stationary and nonstationary function optimization," Proc. of SPIE Intelligent Control and Apative Systems Conf., 1989
10. Bart Kosko, Neural Networks and Fuzzy Systems,Prentice-Hall, Inc, 1992